LogiGear

LOGIGEAR CERTIFIED TEST PROFESSIONAL I

BASIC SOFTWARE TESTING SKILLS

Agenda

Chapter 1: Software Development Life Cycles and Testing

Chapter 2: Software Testing Overview

Chapter 3: Test Requirements

Chapter 4: Test Design Techniques

Chapter 5: Working with a Test Case Management System

Chapter 6: Software Error

Chapter 7: Working with a Bug Management System

Chapter 8: Reporting Progress: Status Report

Chapter 9: Introduction to Mobile Application Testing

LogiGear

CHAPTER 1

Software Development Lifecycles (SDLC) and Testing

CHAPTER 1

- SDLC and Testing
- Waterfall Model
- Spiral Model
- V-Model
- Concurrent Model
- Agile Model
- Other SDLC Models
- Testing Phases and Milestones
- DevOps and CI/CD

OBJECTIVES

- An introduction to Common Software Development Life Cycle (SDLC) practices and how software testing fits in those contexts.
- Understanding testing phases, milestones, checkpoints in a SDLC and their purposes
- Learn about how each SDLC choice affects software testing and its implementation
- Learn about DevOps and CI/CD

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

SDLC and Testing

How testing is directly affected by the SDLC choice:

- Documentation availability to test against
- Time to test
- Time to automate or produce effective automation
- Knowledge and understanding of the application as we plan our tests
- Amount of regression testing

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

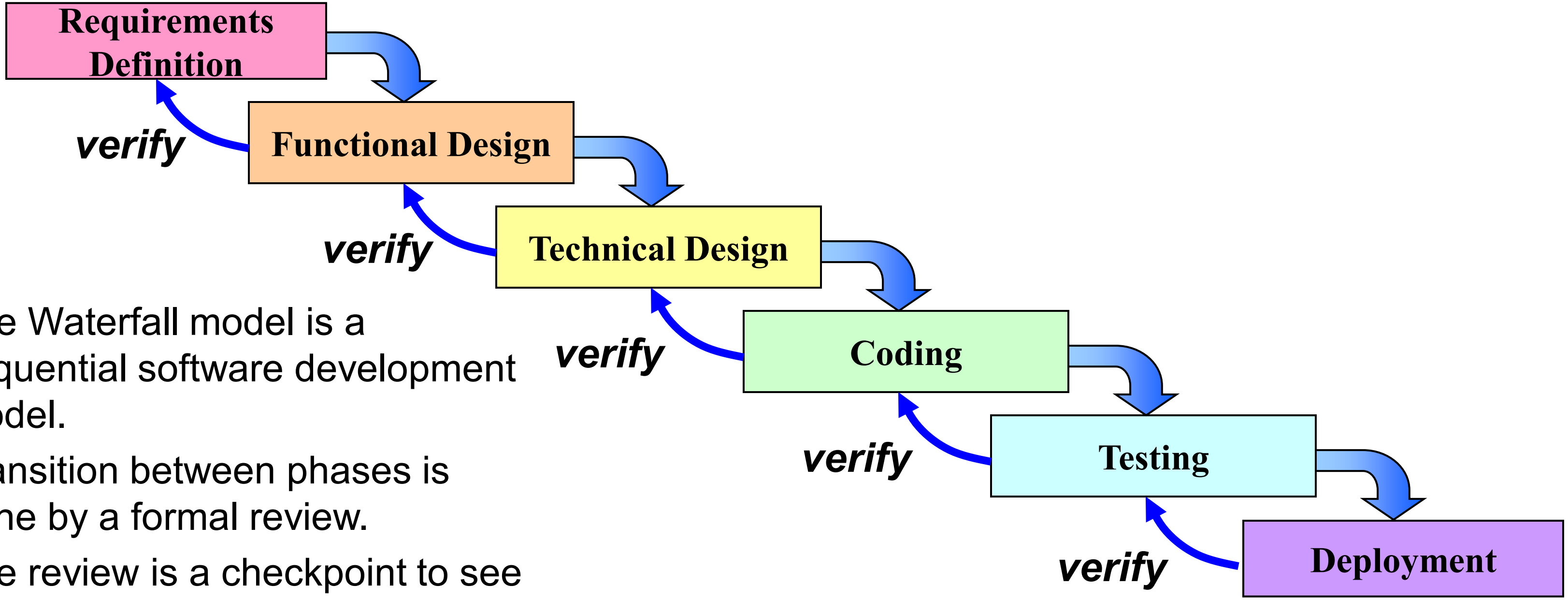
1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

Waterfall Model



- The Waterfall model is a sequential software development model.
- Transition between phases is done by a formal review.
- The review is a checkpoint to see that you are on the right track.

Testing in Waterfall Model

- Testing is not inherent to every phase of the Waterfall model.
- Constant testing from the design, implementation and verification phases is required to validate the phases preceding them.
- Testing phase: Only after coding phase, software testing begins. Different testing methods are available to detect the bugs that were committed during the previous phases. A number of testing tools and methods are already available for testing purposes.

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

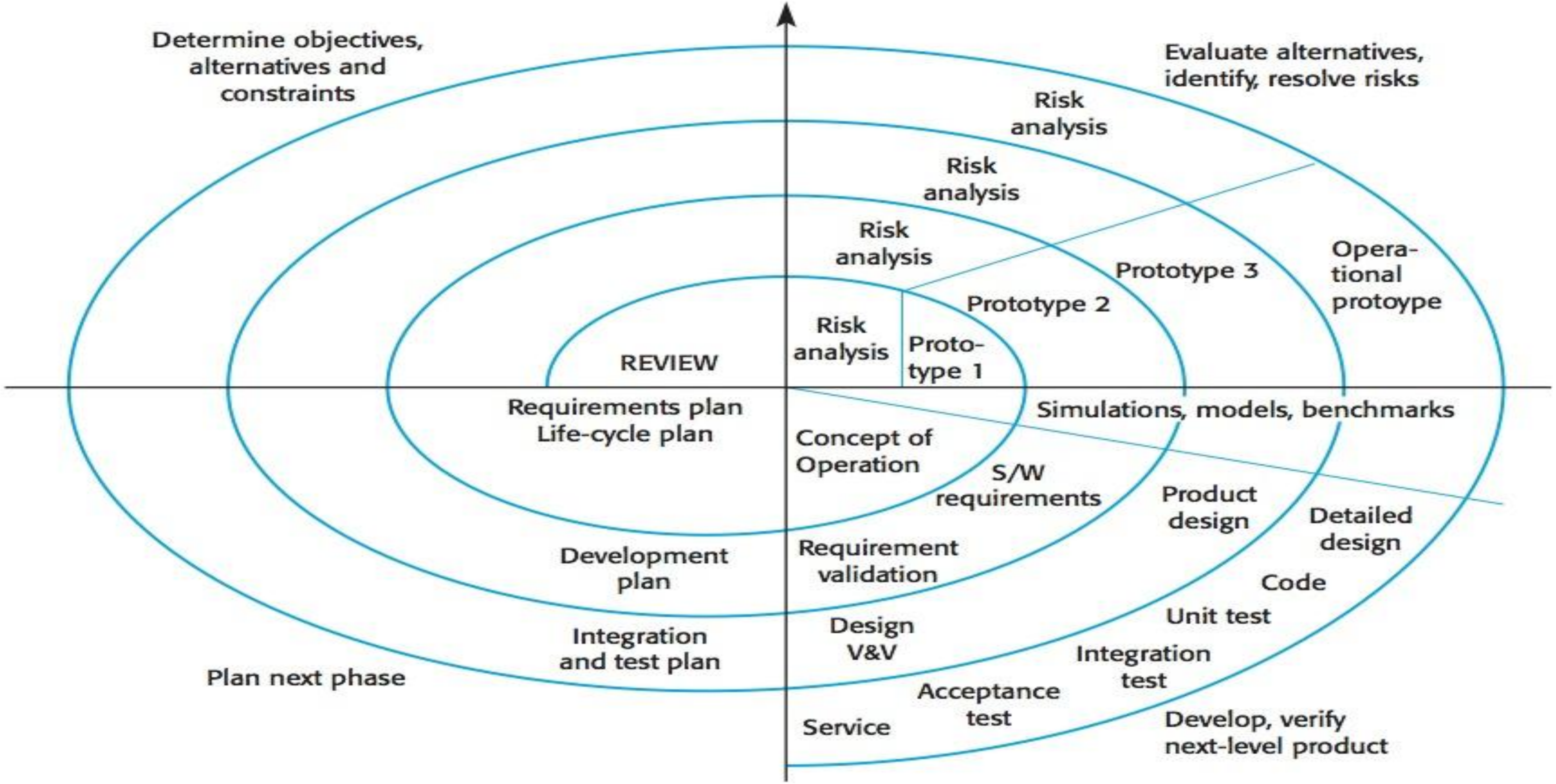
1.6- Agile

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

Spiral Model



Testing in Spiral Model

- Planned and structured releases
- Usually has documentation to test against
- Each spiral iteration can be thought of as a “mini-waterfall”; there are defined testing phases
- Previous releases must be regression tested

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

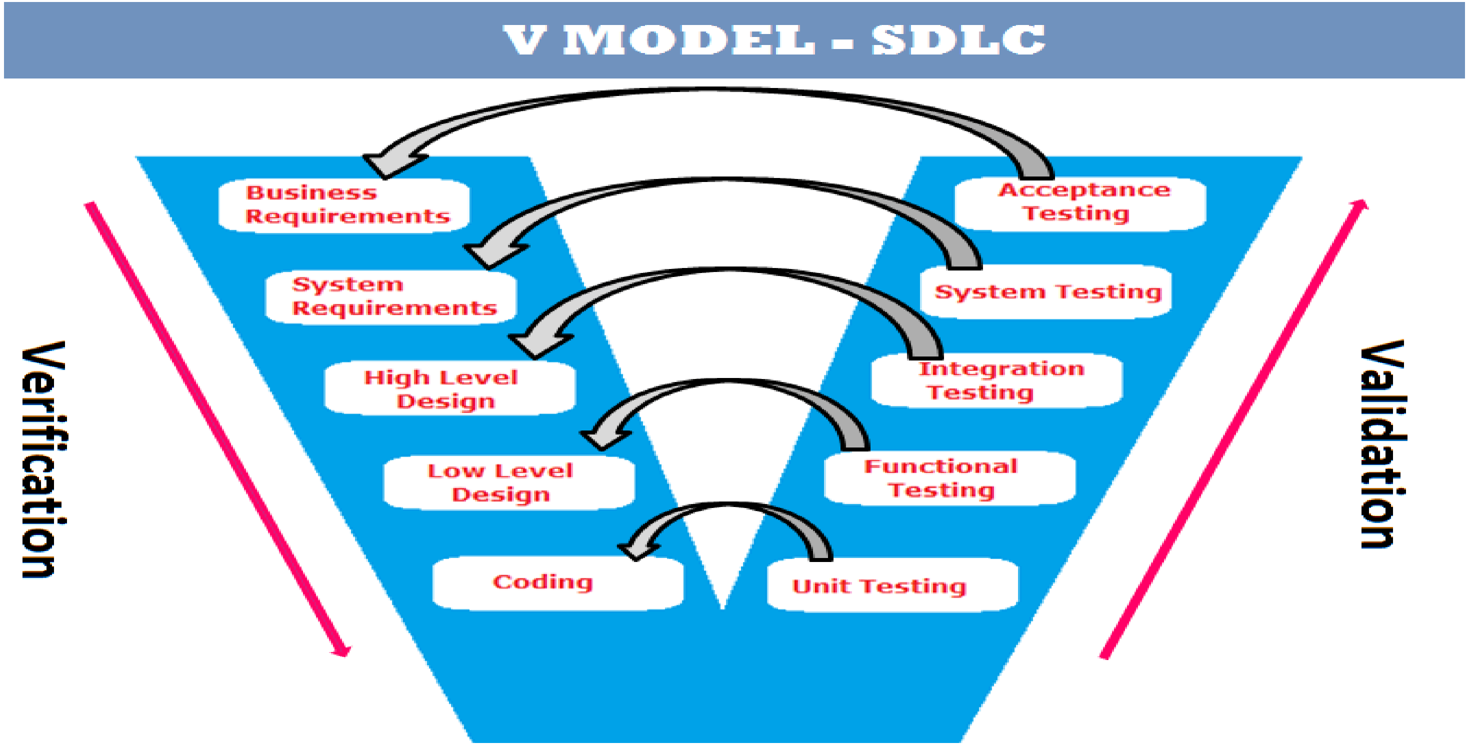
1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

V-Model



© www.SoftwareTestingMaterial.com

Testing in V-Model

- Validation: The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. (PMBOK Guide)
- Verification: The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. (PMBOK Guide)

Do right things >> Validation

Do things right >> Verification

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

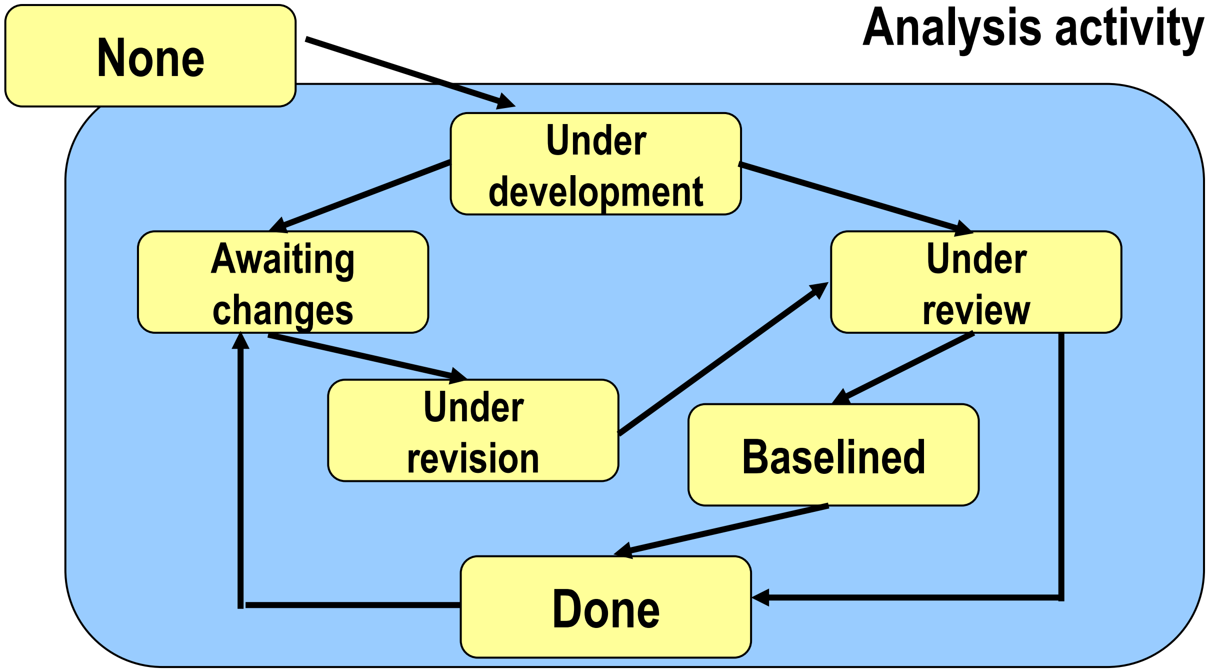
1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

Concurrent Model



All activities concurrent but reside in different states

Testing in Concurrent Model

- Planning is concurrent to design; design is concurrent to development—everything is happening at the same time.
- The whole project is not well planned or well structured.
- Planning, design and development are most dynamic. Product is in constant change. Very difficult to test; impossible to effectively plan testing project.
- Often no documentation to test against. Testing is ad hoc.
- Coverage usually cannot be measured. Structured regression testing is impossible.
- Bugs will be missed because of so much change and so little planning.
- Risk analysis and reporting is crucial.

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

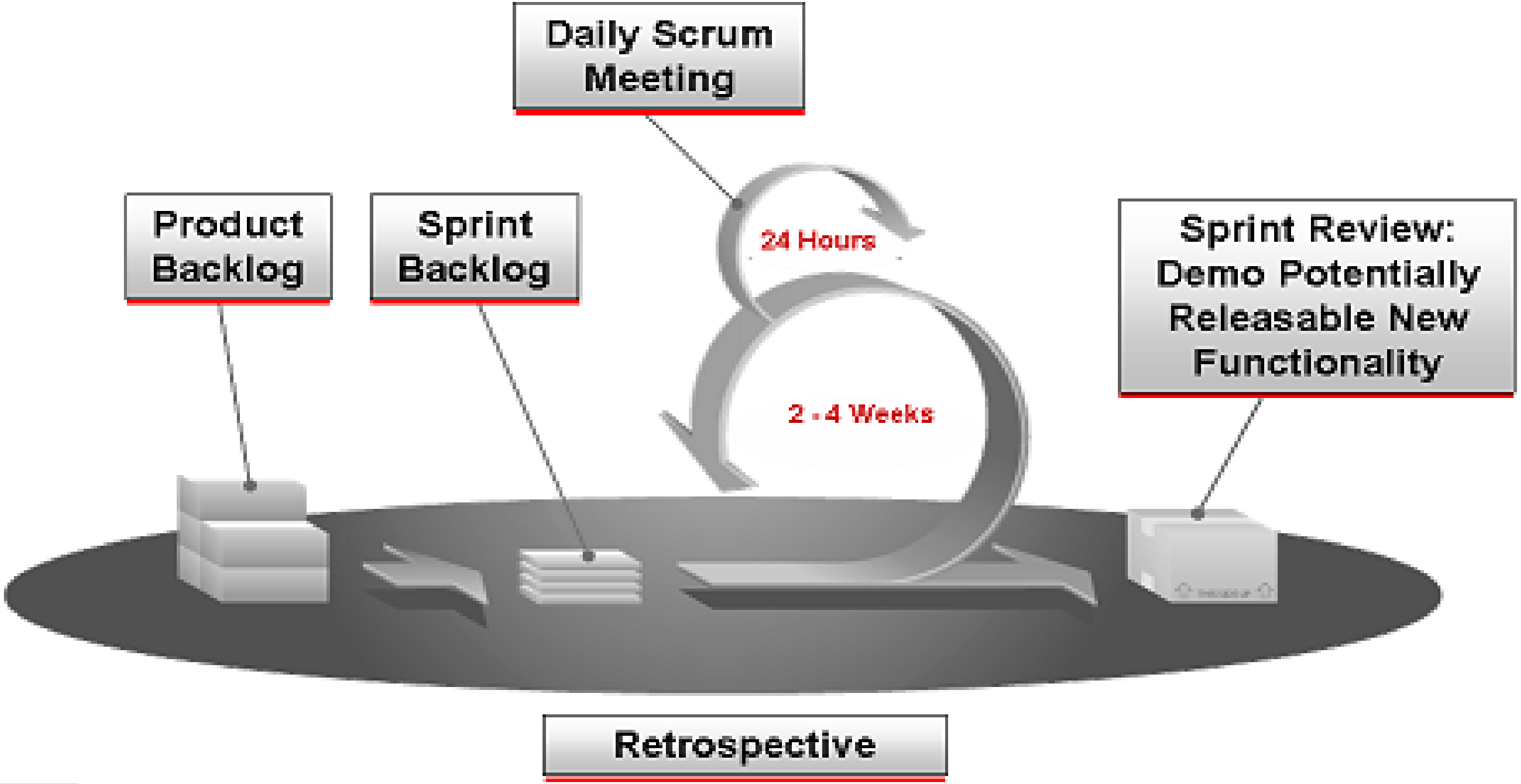
Agile Model

Key principles of Agile:

- Top priority is customer satisfaction
- Welcome changing requirements, even late in the project
- Deliver working software frequently
- Business and development people should work together daily
- Communicate face-to-face, whenever possible
- Working software is the primary measure of progress
- Teams constantly reflect on how to improve their processes and methods

(From http://www.agilemanifesto.org/principles.html)

Agile Model


Daily Scrum Meeting
Product Backlog
Sprint Backlog
24 Hours
2 - 4 Weeks
Sprint Review: Demo Potentially Releasable New Functionality
Retrospective

Agile Model

What is User Story?

- User Story is a story about how the system is supposed to solve a problem or support a business process
- A user story is a very high-level definition of a requirement, containing just enough information collected from client so that the developers can implement it.
- User stories are simple enough that people can understand easily
- Each User Story is written on a User Card (by collecting from client directly), stored in full documents or an ALM tool such as Rally, TFS or IBM Rational.

Agile Model

Acceptance Criteria

- An acceptance criterion is a set of accepted conditions or business rules which the functionality or feature should satisfy and meet, in order to be accepted by the Product Owner/Stakeholders.
- Acceptance criteria define the boundaries of a user story, and are used to confirm when a story is completed and working as intended.

Testing in Agile Model

- Delivery cycles are short. Development is very focused. Diagrams, use cases, user stories, index cards, or discussions of functionality serve as “documentation” to test against.
- Agile projects usually have more unit testing by developers.
- Dynamic nature of development needs structured regression testing.
- Testing is often exploratory, but focused.
- Dynamic style or projects include much change but the change is discussed and side-effects noted.

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

Other SDLC Models

- Rapid Application Development (RAD) Model
- TDD – Test Driven Development, an XP practice
- eXtreme Programming (XP)
- Kanban
- Lean Software Development (LSD)
- …the list goes on

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

1.6- Agile Model

1.7- Other SDLC Models

1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

Phases & Milestones

- Development Phase: A time block within SDLC. There are a number of activities that have to be done in this time block.
- Examples of some Test Phases: Unit Test, Integration Test, System Test, User Acceptance Test, Release Test and Maintenance Test.
- Development Milestone: A significant event in the software developing process in which, the software moves from one phase to another. The milestone let us know where the application is in SDLC
- Criteria: Built or created from the standard of the project.

Testing Phases and Milestones

Test Phase	Pre – Alpha	Alpha	Beta	GM/Release
	Unit	Integration	System	User Acceptance
Performed by	Developers	Developers/(Technical) Testers	Testers	Users
Definition	Test one unit of code at a time	Iteratively more complex system components integrated. Checking the units of code work together incrementally	Big bang. The entire product as it is intended to be used	Requirements-based testing by the project sponsor, testing the production system
Outcome	Fewer Bugs	Bug Finding	Bug Finding/QC	QC
Goals of Testing	To validate each module in isolation	To find bugs, to learn about the product, to validate the integration requirements	To find bugs, to validate the system requirements, data flow, scenarios, load	To validate user requirements

SDLC and Testing

1.1- SDLC and Testing

1.2- Waterfall Model

1.3- Spiral Model

1.4- V-Model

1.5- Concurrent Model

1.6- Agile Model

1.7- Other SDLC Models

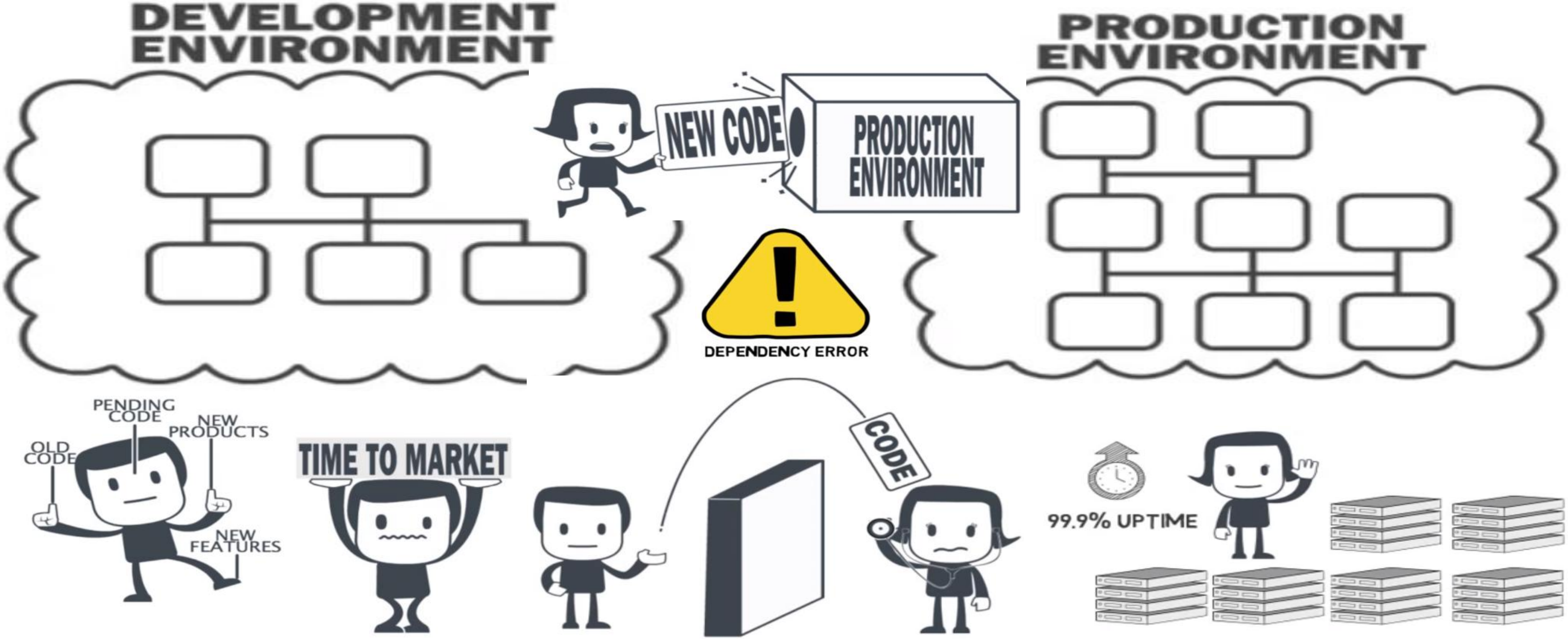
1.8- Testing Phases and Milestones

1.9- DevOps and CI/CD

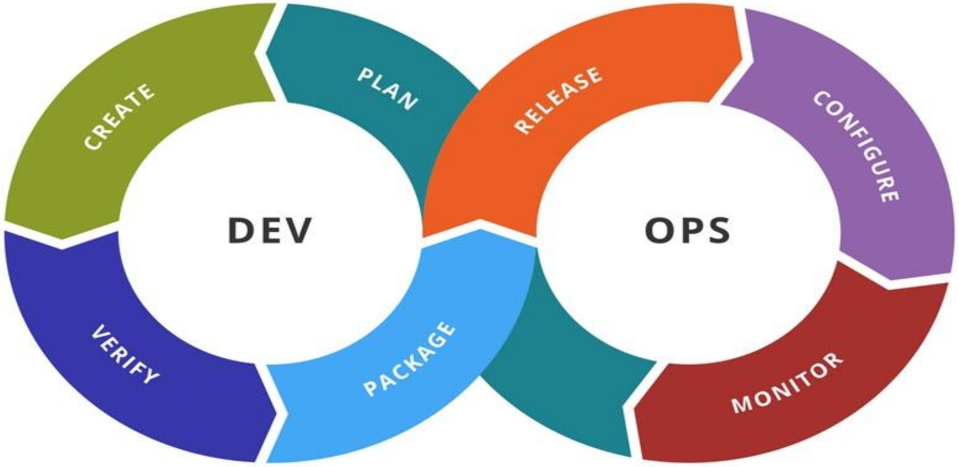
DevOps Overview

- **DevOps** = “software **DEV**elopment” and “**OP**eration**S**”
- DevOps is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management.
- DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.

Life Before DevOps



DevOps Goals



- Improved deployment frequency;
- Faster time to market;
- Lower failure rate of new releases;
- Shortened lead time between fixes;
- Faster mean time to recovery (in the event of a new release crashing or otherwise disabling the current system).

Continuous Integration

Continuous Integration is a key aspect of rapid development:

- Build often
- Automated builds
- Automated unit tests re-run
- Automated build acceptance (smoke) testing
- Automated functional/regression testing

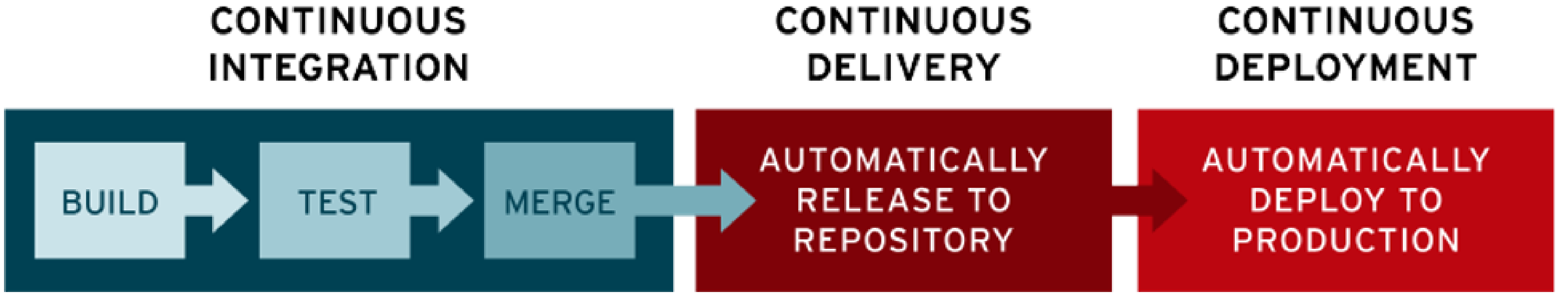
CI / CD

CI / CD / CD

- **Continuous Integration**: Developers practicing continuous integration merge their changes back to the main branch as often as possible.
- **Continuous Delivery**: Continuous Delivery is an extension of Continuous Integration to make sure that the new changes can be deployed to testing / staging environment to go through automation test quickly in a sustainable way.
- **Continuous Deployment**: Continuous Deployment goes one step further than Continuous Delivery. Every change that passes all stages of the production pipeline is released to customers. There's no human intervention.

CI/CD

Continuous Integration – Continuous Delivery – Continuous Deployment



SUMMARY

- Depending on the type of application to be built and or customer's requirement, we choose a suitable SDLC model.
- There are several phases in the SDLC. In each phase, there are milestones and criteria for each milestone. The criteria, milestones and phases of the project help the product to be developed correctly, timely and effectively.
- Depending on the SDLC, the testing and its implementation is affected, usually in time, knowledge about the application, documentation and amount of regression test done.
- DevOps is a software engineering culture and practice, aiming at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.

Thank You!