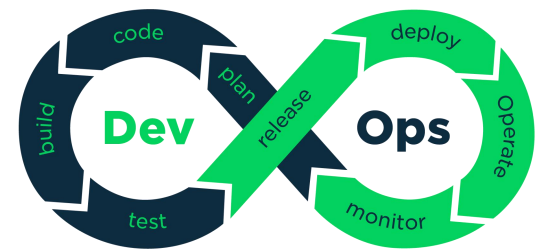


DevOps (Principles & Tools)



Introduction

Name

Total Experience

Background – Development / Infrastructure / Database / Network

Experience on Containers

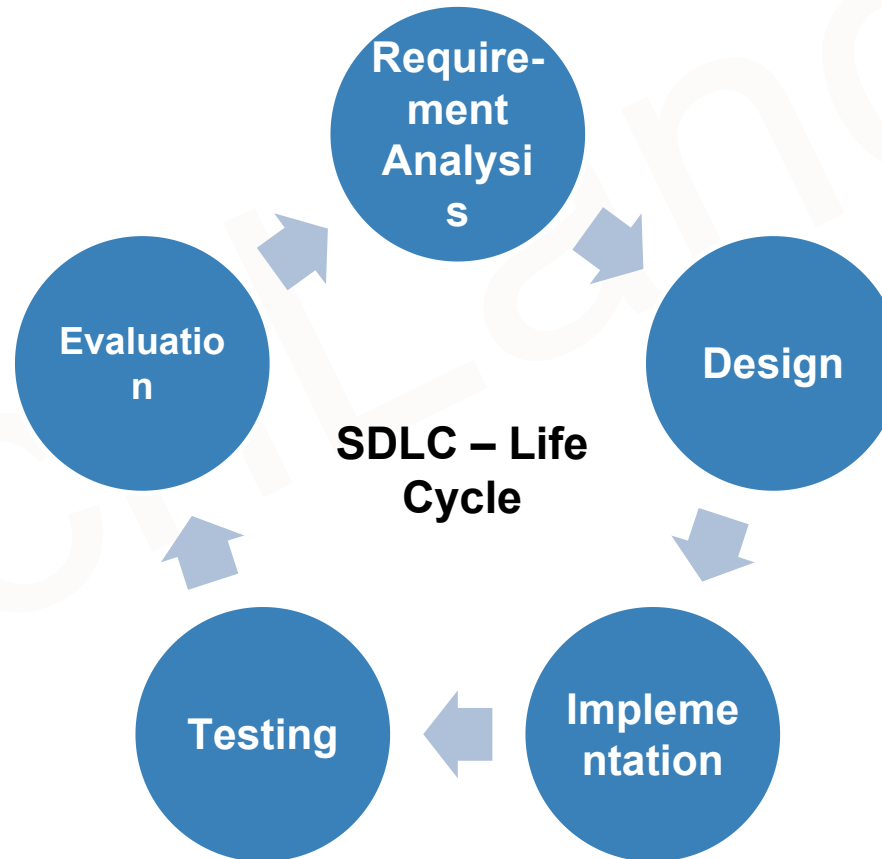
Your expectations from this training

DevOps

What is DevOps?

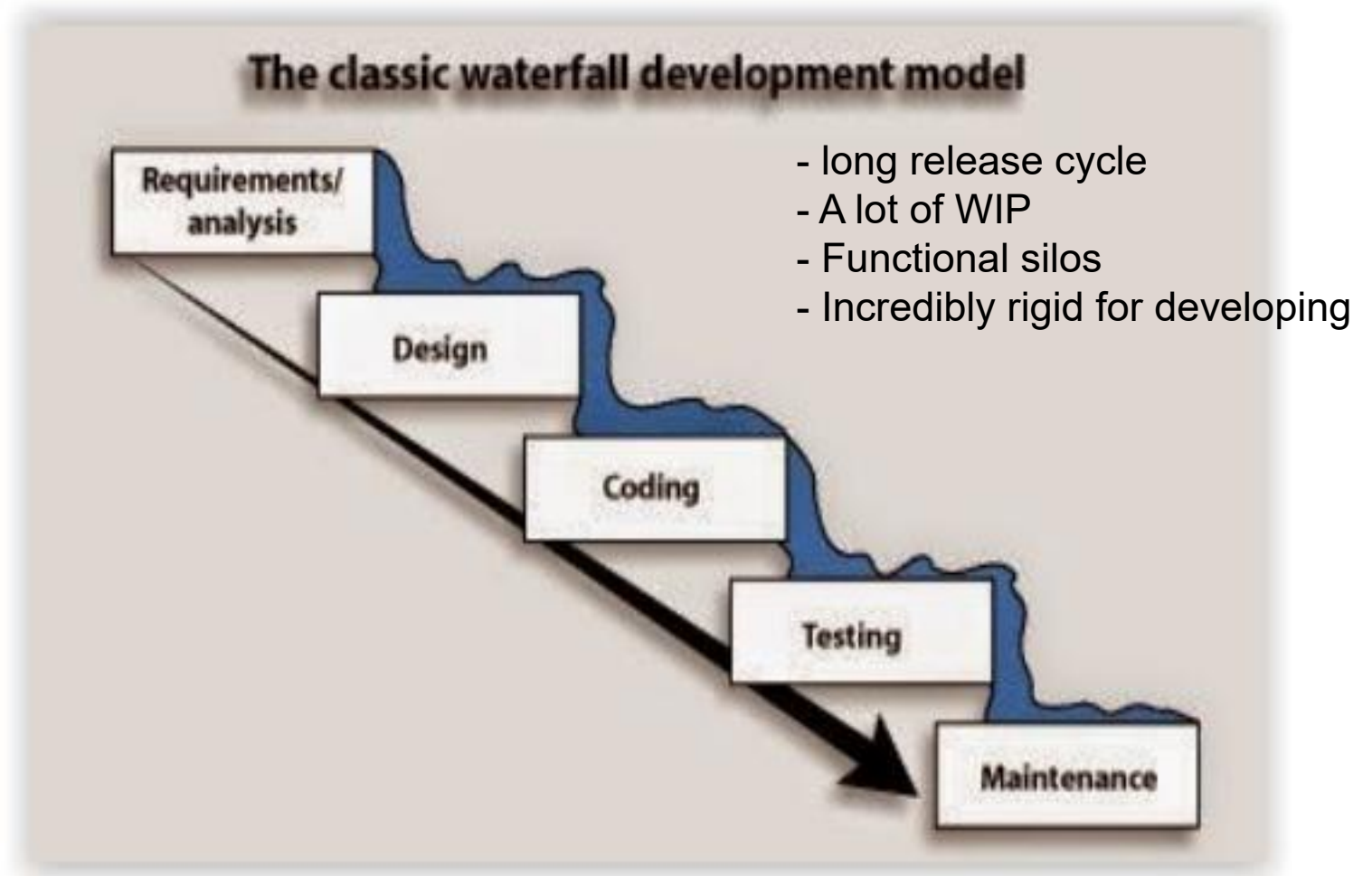
SDLC Model

- A systems development life cycle is composed of **several clearly defined and distinct work phases** which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information systems



Waterfall Model

1. Determine the Requirements
2. Complete the design
3. Do the coding and testing
(unit tests)
4. Perform other tests
(functional tests, non-functional tests, Performance testing, bug fixes etc.)
5. At last deploy and maintain



Agile

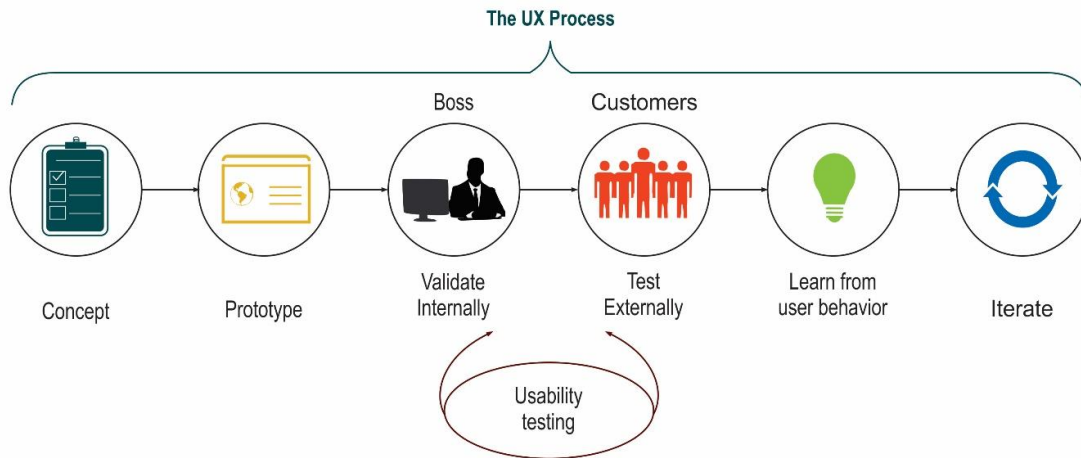
Agile Methodology



- Shorter release cycle
- Small batch sizes (MVP)
- Cross-functional teams
- Incredibly agile

Lean Development

Lean Development (LD)



Not like this...



...instead like this!



- Suddenly ops was the bottleneck (more release less people), again WIP is more!

Challenges

Some of the challenges with the traditional teams of Development and Operations are:



A Typical Case Study

■ Development Team:

- Monday Morning, the writing of code done, unit tests completed, code delivered to the Integration teams to get the code included in CI builds.
- To get the services tested, a ticket is opened for QA teams

■ Build/Release/Testing/Integration Team:

- Tuesday Morning, ticket accepted, a tester put an email to the developer asking deployment instructions. There is not automated deployments, developer updated to the tester, lets come online and we will deploy the services to the QA environment together.
- Call started, developer identified the “test environment” is not compatible.
- Tuesday afternoon, a ticket raised in Ops Team with new specifications.

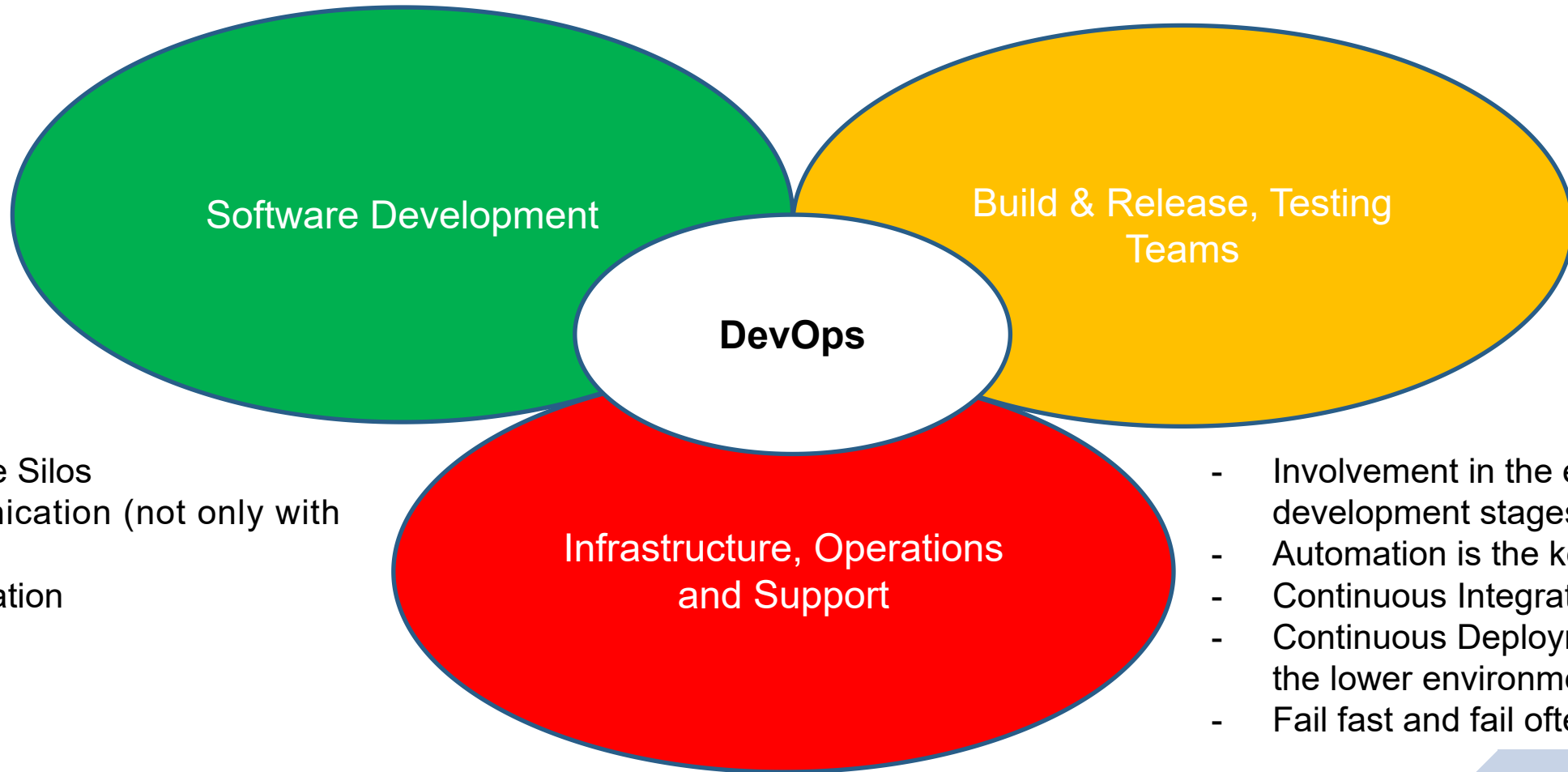
■ Ops Team:

- Wednesday morning, ticket accepted, specifications checked , a new port open request was identified.
- Ticket raised for Security team, ticket accepted, change approved, port opened, email received by the Ops team the work is done.

A Typical Case Study

- **Ops Team:**
 - Identified the provisioning requirements again and started work on building the environment.
- **Build/Release/Testing/Integration Team:**
 - Thursday Morning, updates received – the environment is ready. Developer and Tester again on call to deploy new services. Services deployed; tester is running test scripts. Next phase is to run regression test cases. Again a new ticket is raised for new test data with production teams and day ends.
- **Ops Team:**
 - Its Friday and the work is not on full swing, ticket accepted but not worked as production team has to complete rest of the works. Somehow the test data is gathered by Friday Evening.
- **Build/Release/Testing/Integration Team:**
 - Monday morning, tester gets the data, regression tests run, a defect found, and ticket returned to the development team.

DevOps

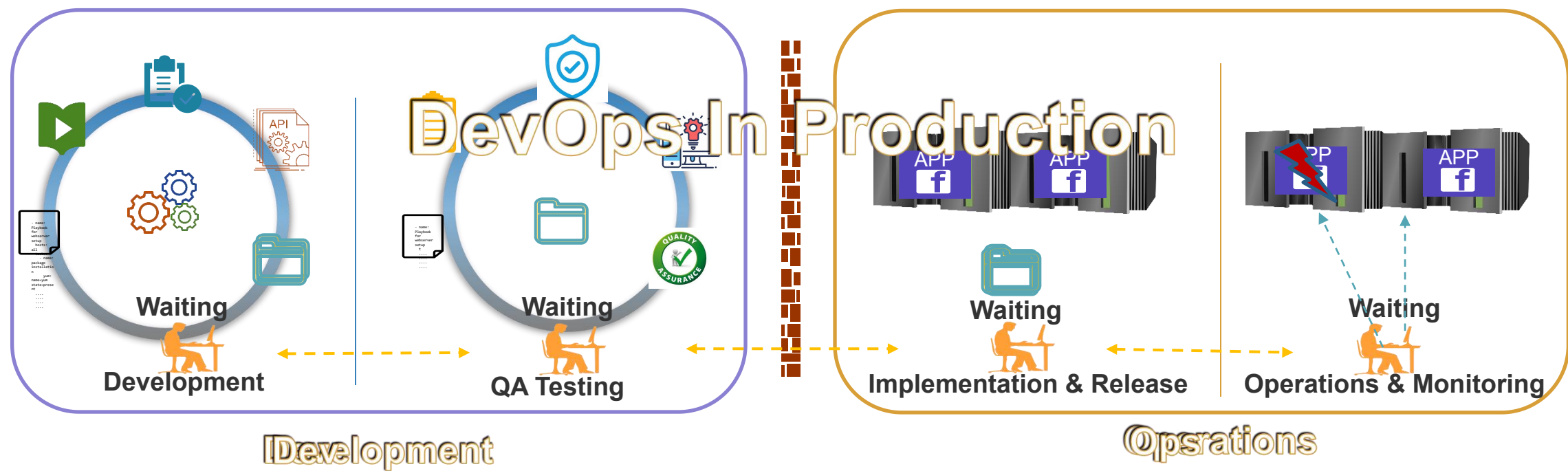


- Break the Silos
- Communication (not only with emails)
- Collaboration
- Trust

- Involvement in the early development stages
- Automation is the key
- Continuous Integration
- Continuous Deployments in the lower environments
- Fail fast and fail often

Continuous Feedback Continuous Improvement Continuous Planning

Continuous Delivery Continuous Deployment Continuous Monitoring



DevOps Toolsets

