Orion Business Innovation

Introduction to DevOps and CI/CD

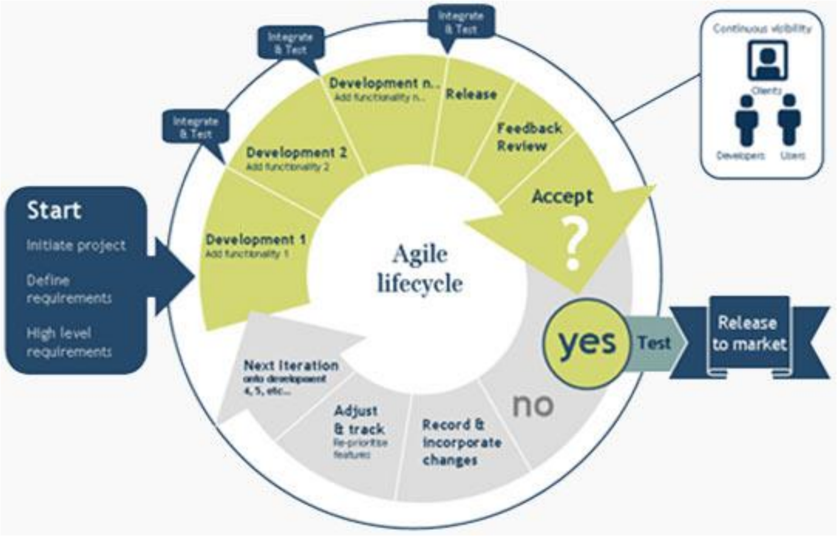
Kunal Kumar Shah

Aug 17, 2017

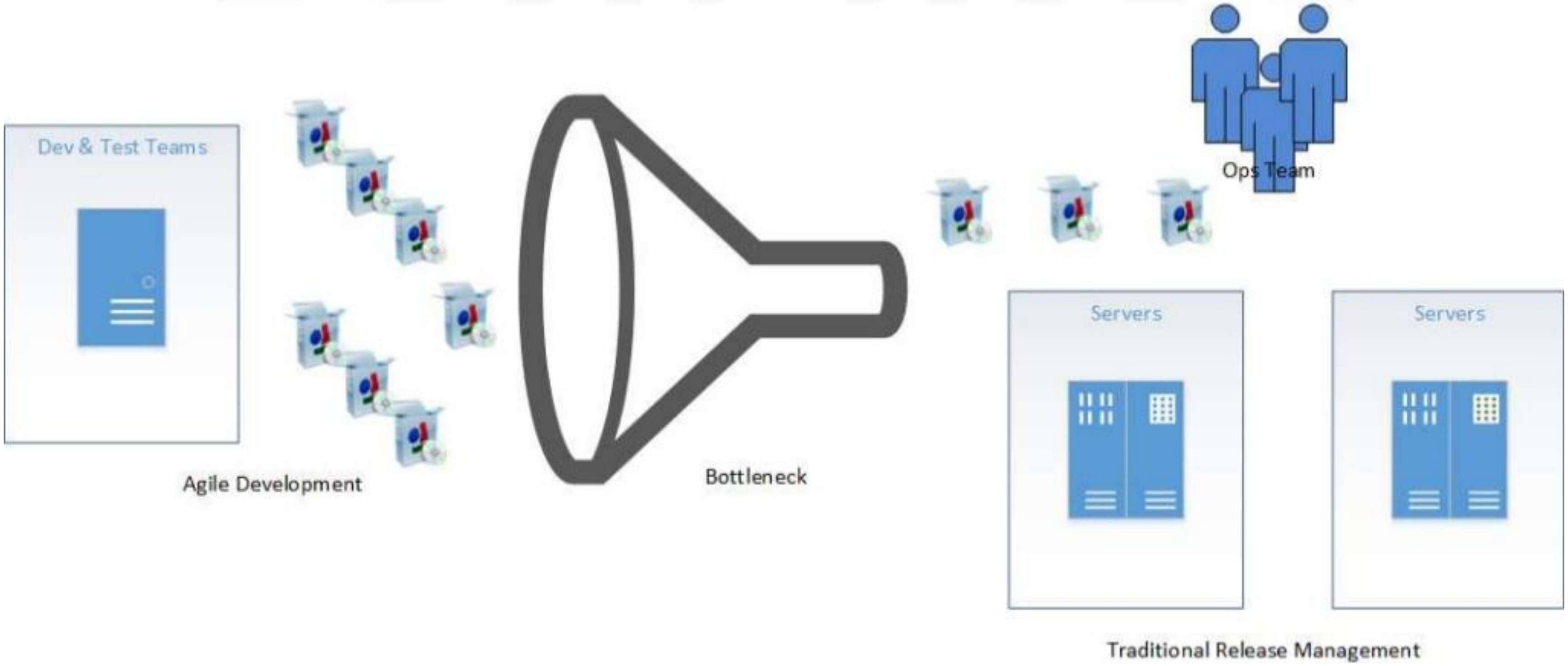
Move your business **forward.**

Agile Approach

- The Agile approach helps in introducing functionality with each release and provides visibility of work to the customer

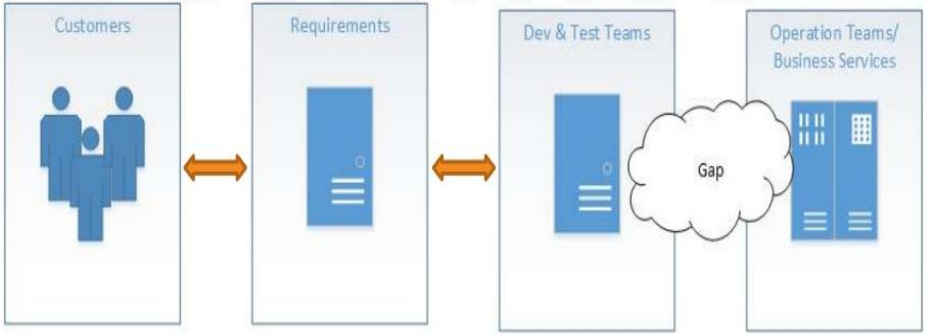


Agile Delivery Approach

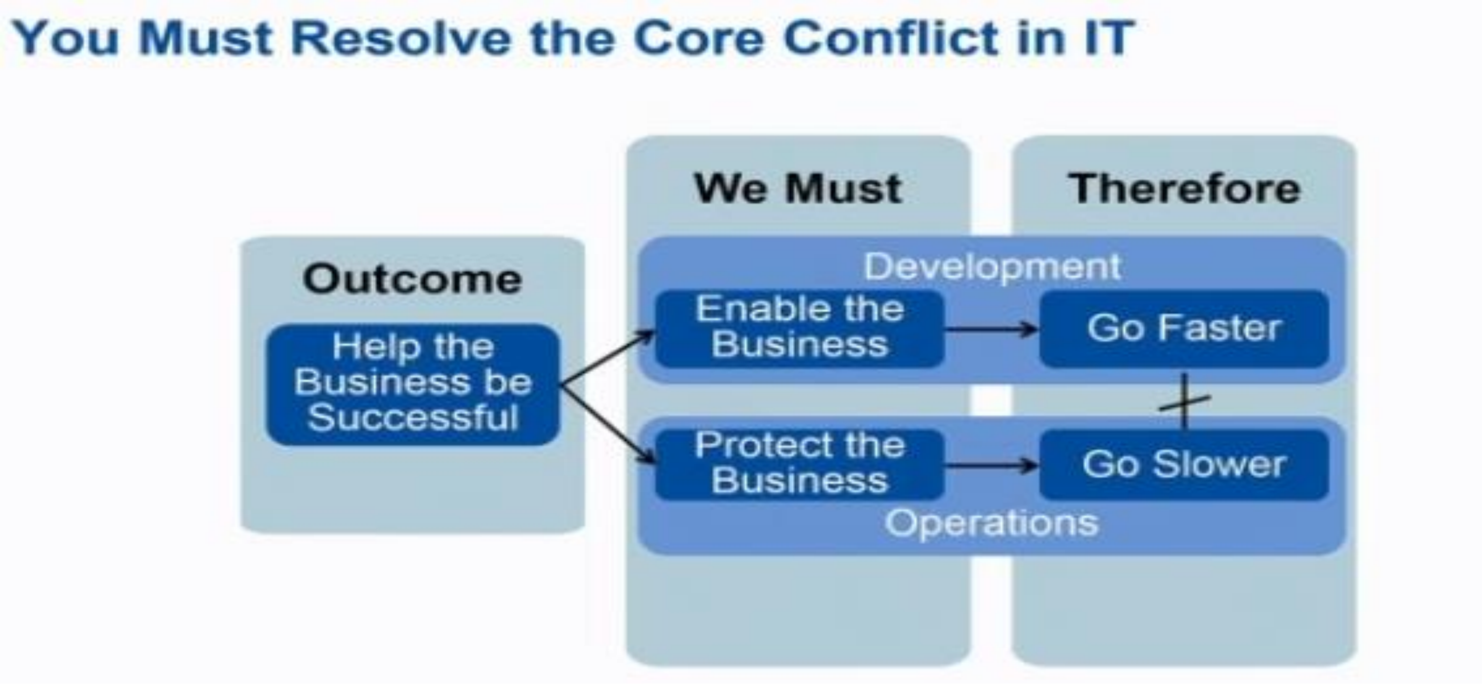


What is missing in Agile Delivery Approach?

- The IT Operations (Ops) team is segregated from the Development team, which results in the following
  - Ops team unaware of the Hardware/network/security requirements of the application being deployed
  - Ops team not able to successfully put the configuration settings in place for each release
  - Incomplete Release Notes leading to delay in deployment
  - Penetration Test failure due to incorrect implementation of the security policies directed by the customer
  - Ops Team failure to provide L2 support due to Lack of product knowledge



Core Conflict in IT



You Must Resolve the Core Conflict in IT

Outcome

Help the Business be Successful

We Must

Therefore

Development

Enable the Business

Go Faster

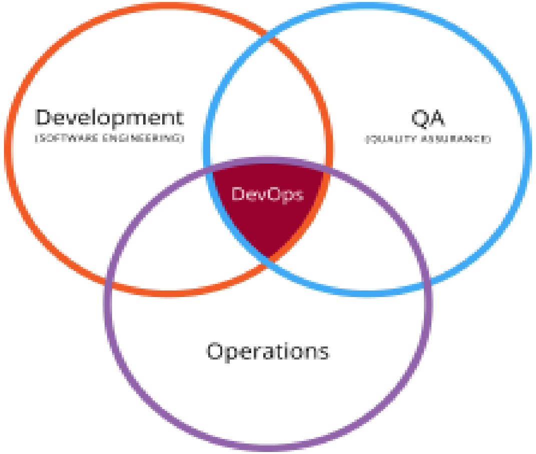
Protect the Business

Go Slower

Operations

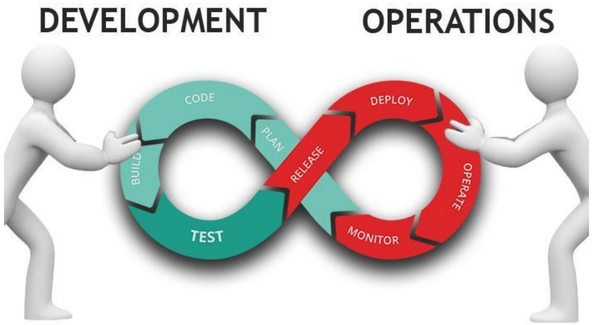
What is DevOps

- **DevOps (development and operations)** is an enterprise software development and delivery process used to mean a type of agile relationship between Development and IT Operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between the two business units.
- DevOps is mainly focused on increasing deployment cycles as well as reducing outages that are change related.
- DevOps Toolchain assists in highly automated approaches to deploy solutions.

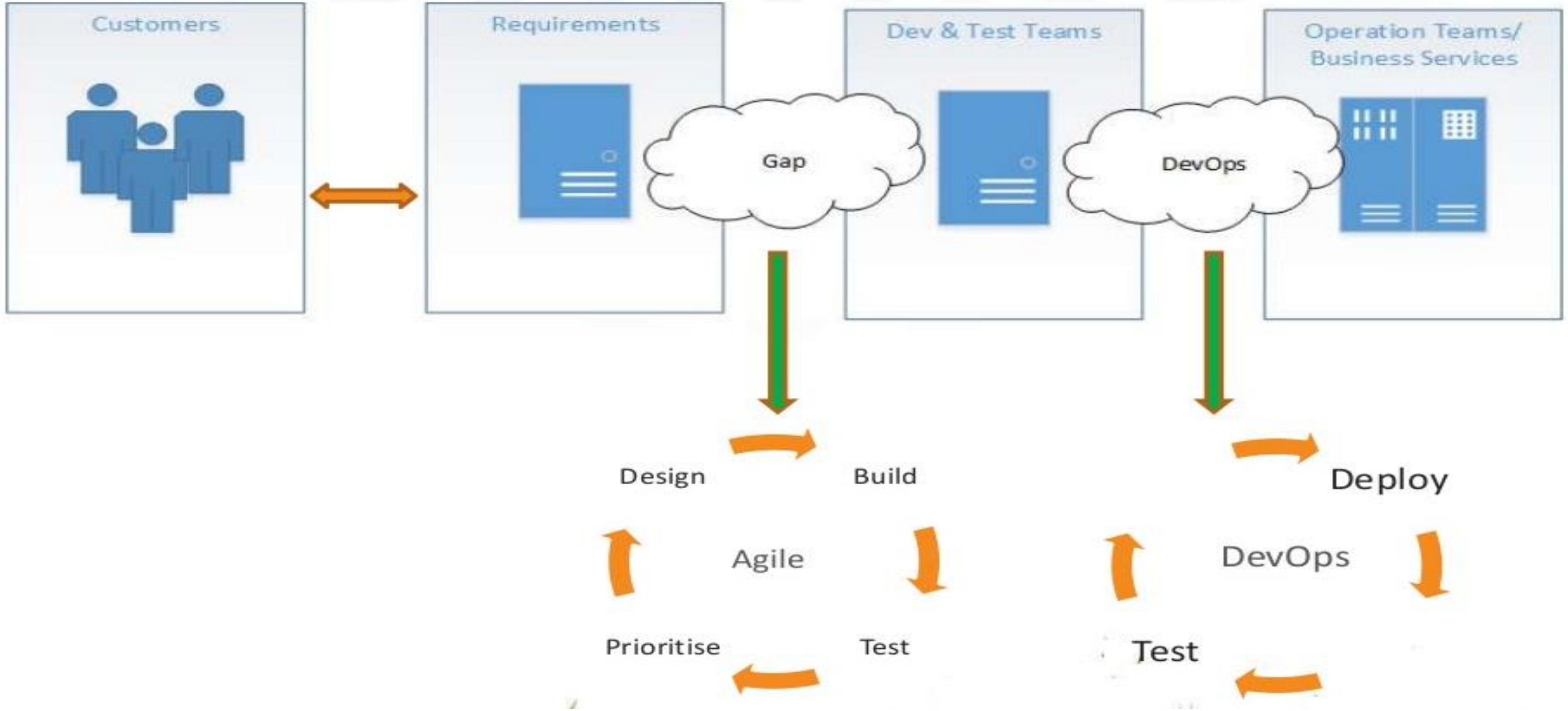


DevOps ToolChain

- As DevOps is a truly cross-functional mode of working, there is no single "DevOps tool": it is rather a set (or "toolchain") of multiple tools.
  - Code — code development and review, source code management tools, code merging
  - Build — **Continuous Integration tools**, build status
  - Test — continuous testing tools that provide feedback on business risks
  - Package — artifact repository, application pre-deployment staging
  - Release — change management, release approvals, **Continuous Deployment**
  - Configure — infrastructure configuration and management, **Infrastructure as Code** tools
  - Monitor — **Application Performance Monitoring**



Agile and DevOps



Steps to move towards DevOps Approach

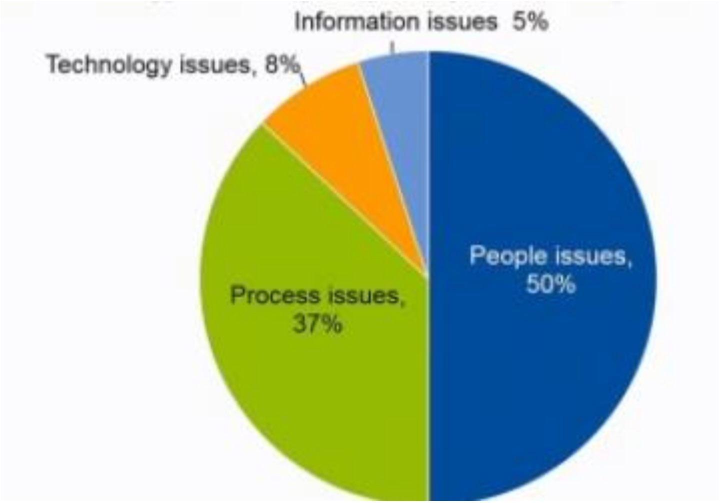
- Organize team, where Dev, Test, Rollout (operations) teams are dis-integrated and combined as 1 or 2 teams having all the above competencies in each team.
- Improve Test team knowledge on vulnerability (application and network), so that they can perform Penetration Testing under the guidance of Security Team.
- Involve Operations team for adding capabilities of virtualization, cloud computing, configuration management, monitoring and containerization.
- Automate routine tasks like, Build, Unit Tests, Production rollout/upgrades and data maintenance/cleanup.

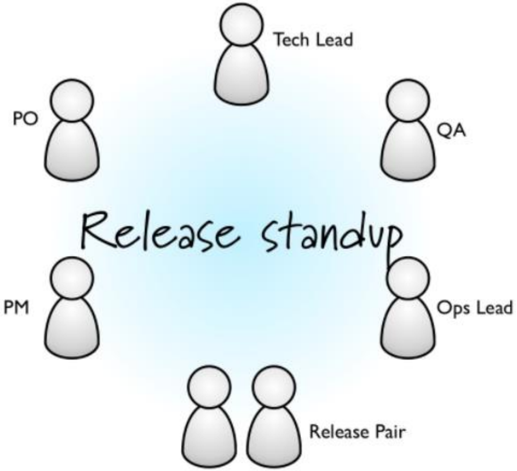
Continuous Integration & Continuous Deployment

- Continuous Integration (CI) is the practice of merging all developer working copies to a shared mainline several times a day.
  - The main aim of CI is to prevent integration problems, referred to as "integration hell"
  - CI is effective if used in combination with automated unit tests written through the practices of test-driven development
  - Predominately, Build Servers are used to automatically run unit tests and report to developers
  - Additionally, Performance Tests can be performed and extraction of Source Code documentation can be done
- Continuous Deployment (CD) is the process of deployment using automation and subsequent run of unit tests.
- Key Benefits of CI/CD
  - Integration bugs are detected. This saves both time and money over the lifespan of a project.
  - Avoids last-minute chaos at release dates, when everyone tries to check in their slightly incompatible versions
- Popular CI/CD Tools : Jenkins, TeamCity, Bamboo, Travis

Can ToolChain and Automation make DevOps success?

- DevOps requires Communication and Integration along with Tools and automation to succeed
- What issues Orgs face in implementing DevOps?
- Release Standup team helps reduce any delivery issues





Goal of DevOps and Why It is Important

- DevOps is important, because of the conflicting nature of departmental roles:
  - Operations — seeks organizational stability
  - Developers — seek change
  - Testers — seek risk reduction
- DevOps focuses on the organization change to support great collaboration between the many functions involved with the following goals
  - Improved deployment frequency
  - Lower failure rate of new releases
  - Faster mean time to recovery (in the event of a new release crashing or otherwise disabling the current system)
- DevOps aims to maximize the predictability, efficiency, security, and maintainability of operational processes. Very often, automation supports this objective.

Thank You.

https://www.linkedin.com/in/kunalkumarshah/

For more Information, please visit: http://www.orioninc.com

Disclaimer: This document is for informational purposes only and is subject to change without notice. This document and its content herein are believed to be accurate as of its date of publication. However, Orion Systems Integrators, LLC (herein referred as Orion) makes no guarantee, representations or warranties with regard to the enclosed information and specifically disclaims the implied warranties of fitness for a particular purpose and merchantability. As each user of Orion services is likely to be unique in their requirements in the use of such software solutions and their business processes, users of this document are always advised to discuss the content of this document with their Orion Representatives.

orion™
Business Innovation