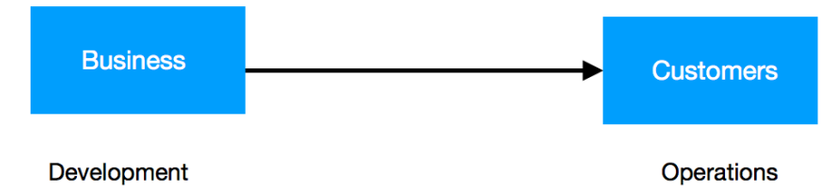
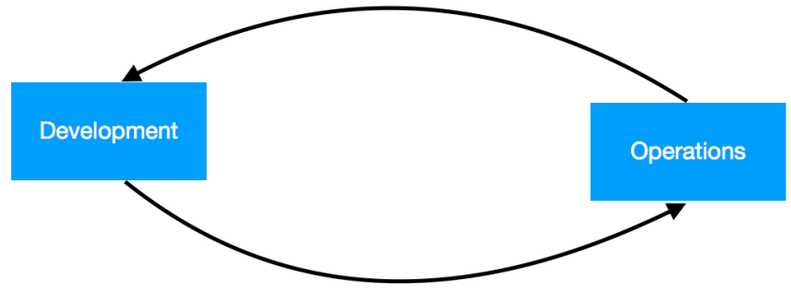
What is DevOps ?

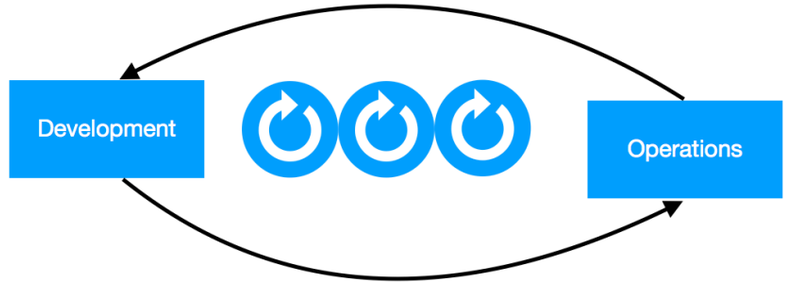
* DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity:
* Evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.
* This speed enables organizations to better serve their customers and compete more effectively in the market.

3 - way thinking

* **Systems Thinking**
  * Small intervals of work
  * Constantly optimize for global goals
* **Feedback Loops**
  * Prevent problems from happening again
  * Faster Detection & Recovery
* **Continuous Experimentation**
  * Repetition to find a better way
  * Innovation and risk taking







Key Technology Practices (As Of Today)

Key Technical Practices

* Open Source Libraries
* Agile Methodology
* Cloud Infrastructure
* Minimal Viable Product (MVP) and Lean Startups
* Monitoring & Observability
* Continuous Testing
* Change Management
* Security Policies
* Influence culture through organizational work

Why Do DevOps ?

* **IT**
  * Importance, autonomy and resources
  * Remove any one of the above and things cannot be done
* **Move Dev + Ops together**
  * Know how much work needs to be done
  * Know your resources
  * Minimize Reliance on constraints, as constraints multiply delays
* **Categorize Tasks**
  * Business Tasks
  * IT Tasks
  * Maintenance Tasks
  * Unplanned Work (The Killer!)
* **Feedback Helps Planning**
  * The more feedback the better, especially, earlier in the process
  * Less frequently, we will repeat the same mistake, but will make newer mistakes and we will solve those
* **Higher Software Delivery Performance == Powerful Business Outcomes** (Software Delivery & Operational Performance)

Performance Measurement

Src: DevOps Research & Assessment LLC

Strong Performers will have upto 3.5 times more availability and still able to optimize for throughput and stability

DevOps Practices

* **Continuous Integration**
  * Developers regularly merge their code changes into a central repository, after which automated builds and tests are run
  * The key goals of continuous integration are to find and address bugs quicker, improve software quality
* **Continuous Delivery**
  * Code changes are automatically built, tested, and prepared for a release to production
  * deployment-ready build artifact that has passed through a standardized test process
* **Infrastructure As A Code**
  * Infrastructure is provisioned and managed using code and software development techniques
  * **Configuration Management**
    * Developers and system administrators use code to automate operating system and host configuration, operational tasks, and more
  * **Policy As Code**
    * Organizations can monitor and enforce compliance dynamically and at scale. Infrastructure that is described by code can thus be tracked, validated, and reconfigured in an automated way
* **Communication And Collaboration**
  * The use of DevOps tooling and automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations

Where DevOps Came From ?

* Originally coined in 2008 by Patrick Debois and Andrew Shafer
* Entered common usage in 2009 in Velocity Conference Community
* Applying Lean principles to IT Value Stream, accelerating flow of work through product management, development, test and operations
* DevOps has benefitted the most from the work of Agile community
* DevOps also benefits from an astounding convergence of philosophical management movements, such as Lean Startup, Innovation Culture, Toyota Kata, Rugged Computing, and the Velocity community. All of these mutually reinforce each other, creating the conditions of a powerful coalition of forces that can accelerate DevOps adoption.

DevOps Myths

* **DevOps Replaces Agile**
  * Code is only “done” when it has been fully tested and is operating in production as designed
  * Agile is not a prerequisite for DevOps
* **DevOps Replaces ITIL/ITSM**
  * DevOps is compatible
  * ITIL/ITSM describe many of the capabilities needed in order for Operations to support a DevOps-style work stream.
* **DevOps Means NoOps**
  * DevOps will often put more responsibility on Development to do code deployments and maintain service levels
  * Many Operations tasks are automated
* **DevOps Is Only For Open Source Systems**
  * DevOps in universal, independent of underlying technologies
  * It starts with support for version control, automation, configuration management
* **DevOps Is For Startups**
  * Applicable and relevant to any organization
  * Planned work through development while maintaining quality, reliability and security for the customer