DevOps

Week 01

Murtaza Munawar Fazal

Class Rules

1. Attendance will be taken at the start of the class. Anyone coming after the attendance would be marked as Late. No leniency in attendance.
2. Quizzes will be unannounced with no retakes.
3. After every quiz/assignment, marks will be uploaded on google classroom. Make sure you verify all your marks.
4. Keep your cell phones on silent.
5. Copied Assignments will be graded 0.

Consultation Hours

- Available only on class day(s)
  - Friday : 03:00 pm – 06:00 pm
  - Saturday : 03:00 pm – 06:00 pm

- For any queries after the consultation hours, please don't hesitate to send me an email : murtaza.fazal.v@nu.edu.pk

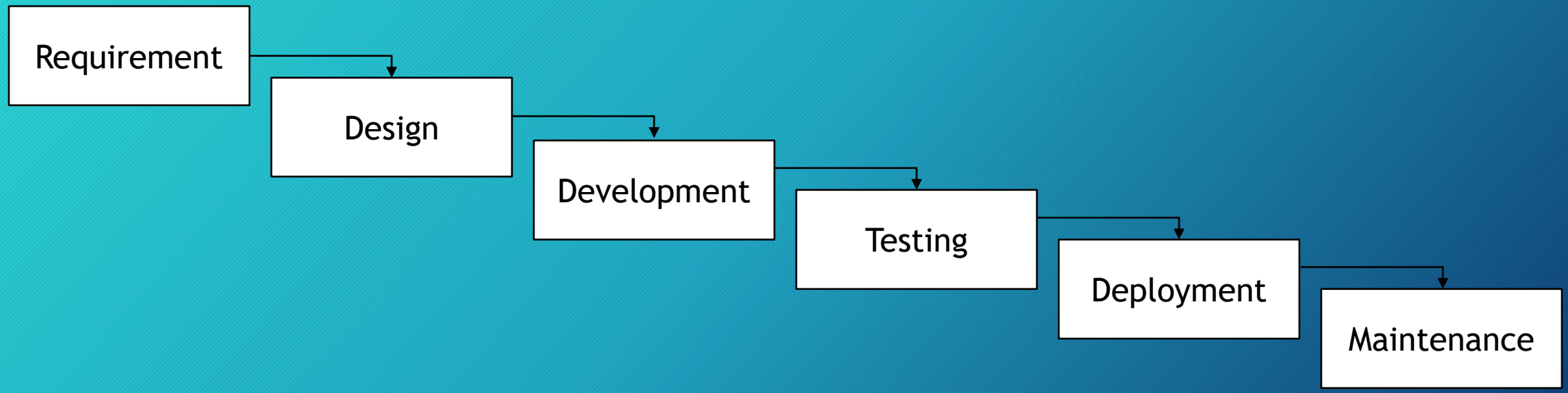
Tools Required

- Visual Studio (Code)
- Github Account
- Azure DevOps Account (nu)
- Microsoft 365 Account (nu)
- Azure (Cloud) Account (nu)

Differentiate

- Software Developer v/s Software Tester
- Software Developer vs Software Engineer

Software Development Life Cycle (SDLC)



Waterfall

Software Development Life Cycle (SDLC)

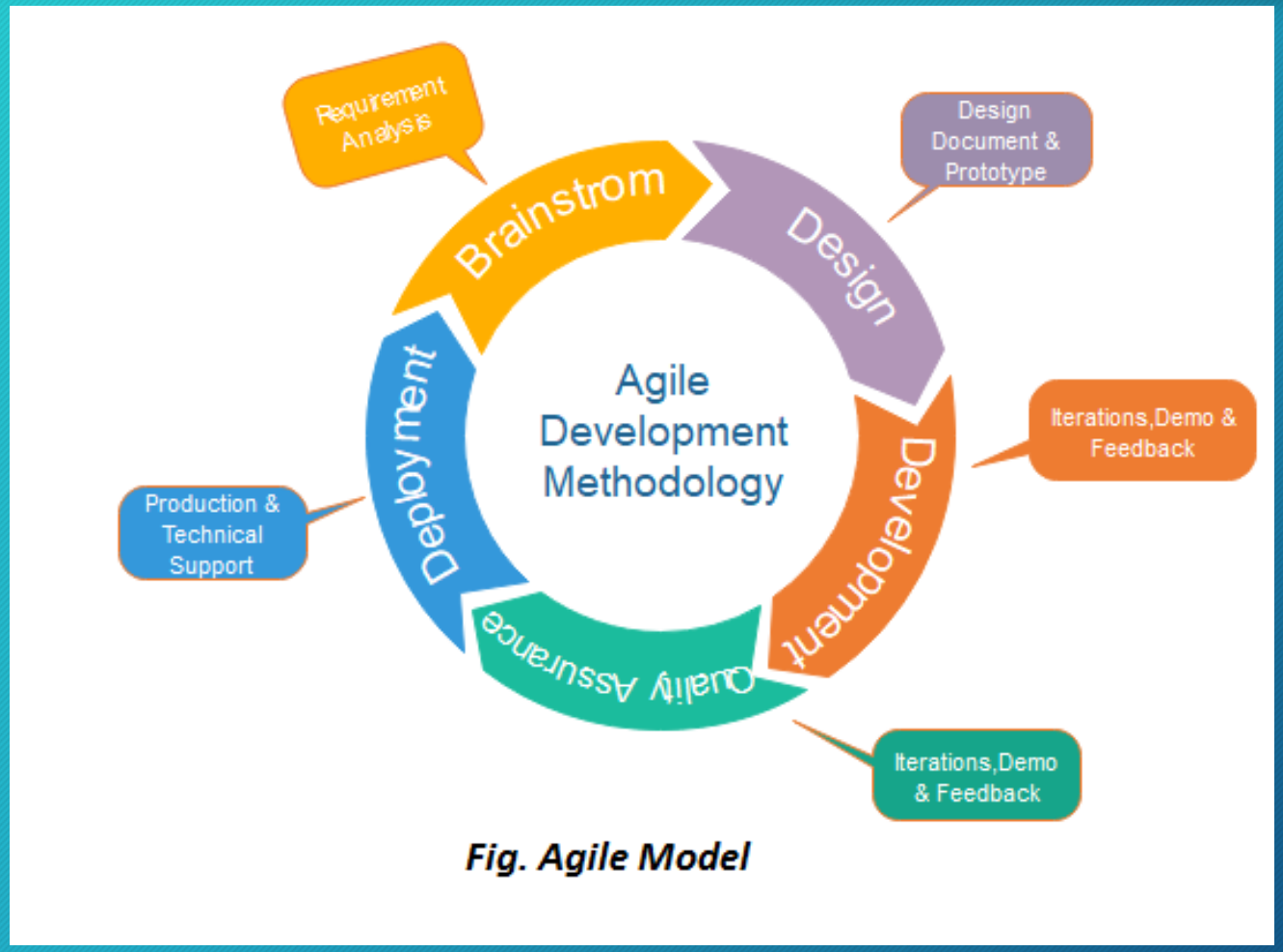


Fig. Agile Model

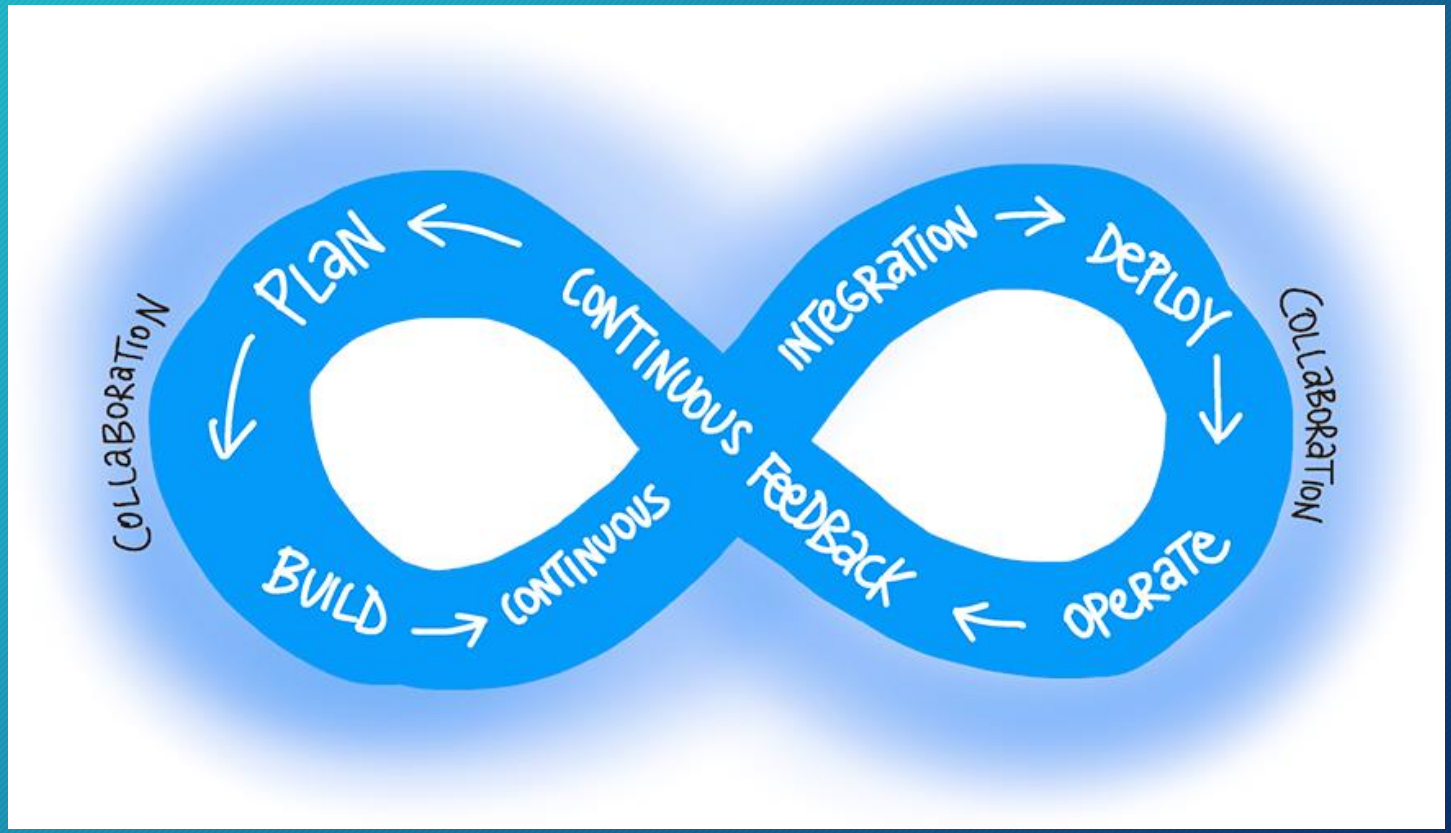
Agile

- Agile methodology constantly emphasizes adaptive planning and early delivery with continual improvement.
- Agile software development methods are based on releases and iterations:
  - One release might consist of several iterations.
  - Each iteration is like a small independent project.
  - After being estimated and prioritization:
    - Features, bug fixes, enhancements, and refactoring width are assigned to a release.
    - And then assigned again to a specific iteration within the release, generally on a priority basis.
  - At the end of each iteration, there should be tested working code.
  - In each iteration, the team must focus on the outcomes of the previous iteration and learn from them.
- Having teams focused on shorter-term outcomes is that teams are also less likely to waste time over-engineering features. Or allowing unnecessary scope creep to occur.

What is DevOps

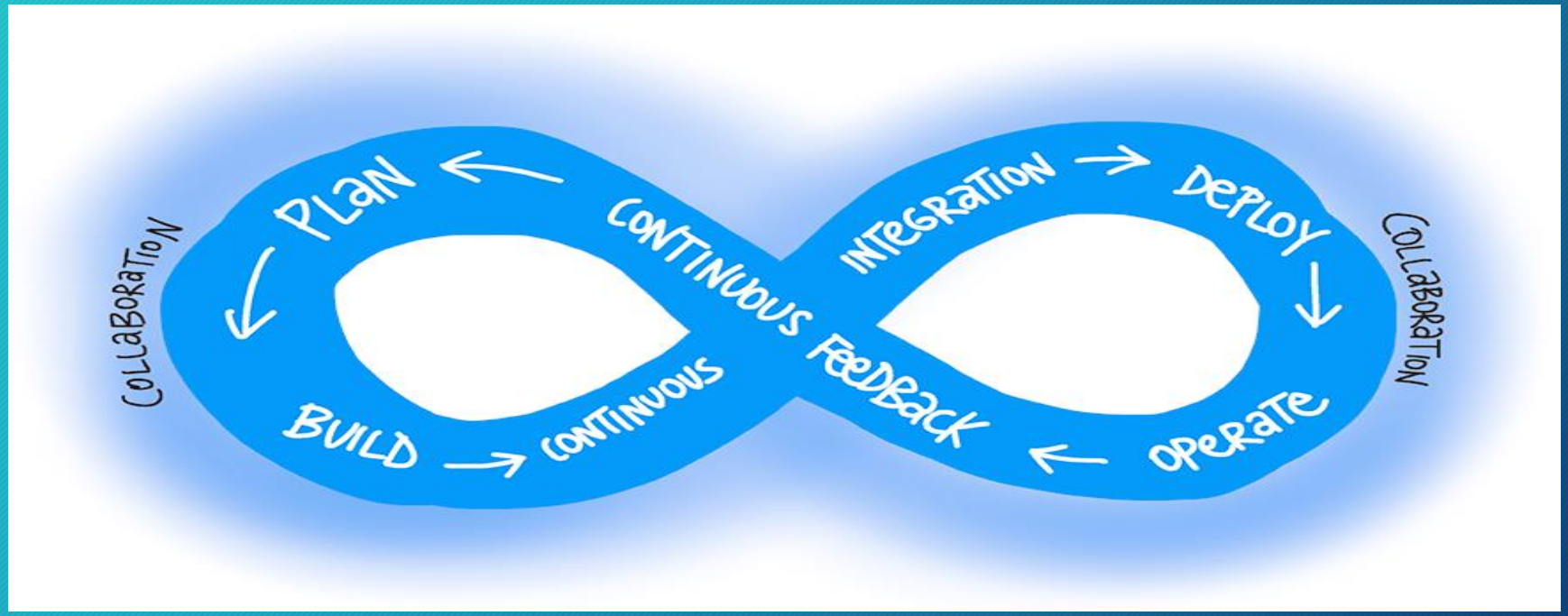
- *"DevOps is the union of people, process, and products to enable continuous delivery of value to end users."*

– Donovan Brown



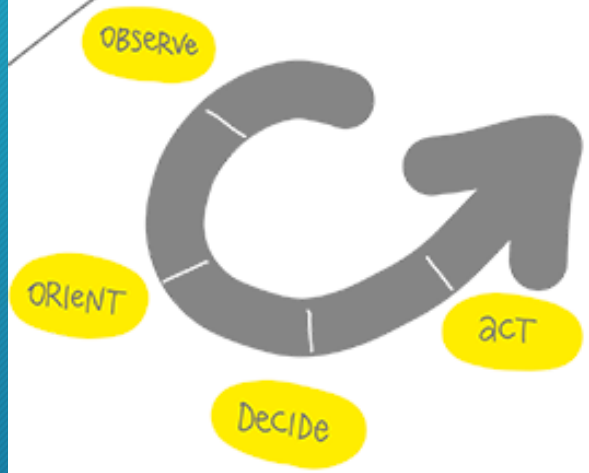
What is DevOps

- The contraction of "Dev" and "Ops" refers to replacing siloed Development and Operations. The idea is to create multidisciplinary teams that now work together with shared and efficient practices and tools. Essential DevOps practices include agile planning, continuous integration, continuous delivery, and monitoring of applications. DevOps is a constant journey.



Understand the Cycle

- You start with observing business, market, needs, current user behavior, and available telemetry data. Then you orient with the enumeration of options for what you can deliver, perhaps with experiments. Next, you decide what to pursue, and you act by delivering working software to real users.

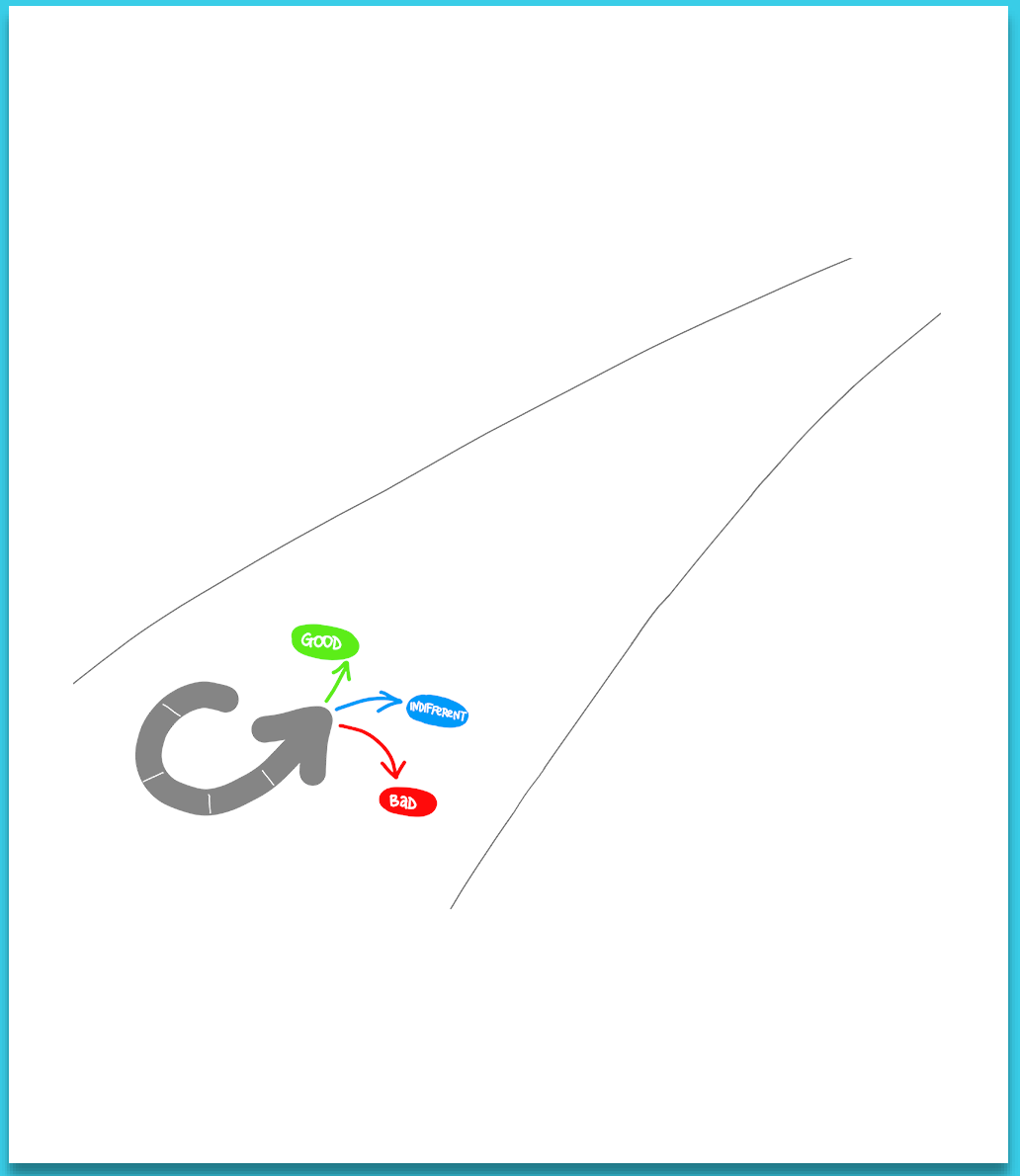


Understand the Cycle

- Many experience reports tell us that roughly one-third of the deployments will have negative business results. Approximately one-third will have positive results, and one-third will make no difference. Fail fast on effects that do not advance the business and double down on outcomes that support the business.

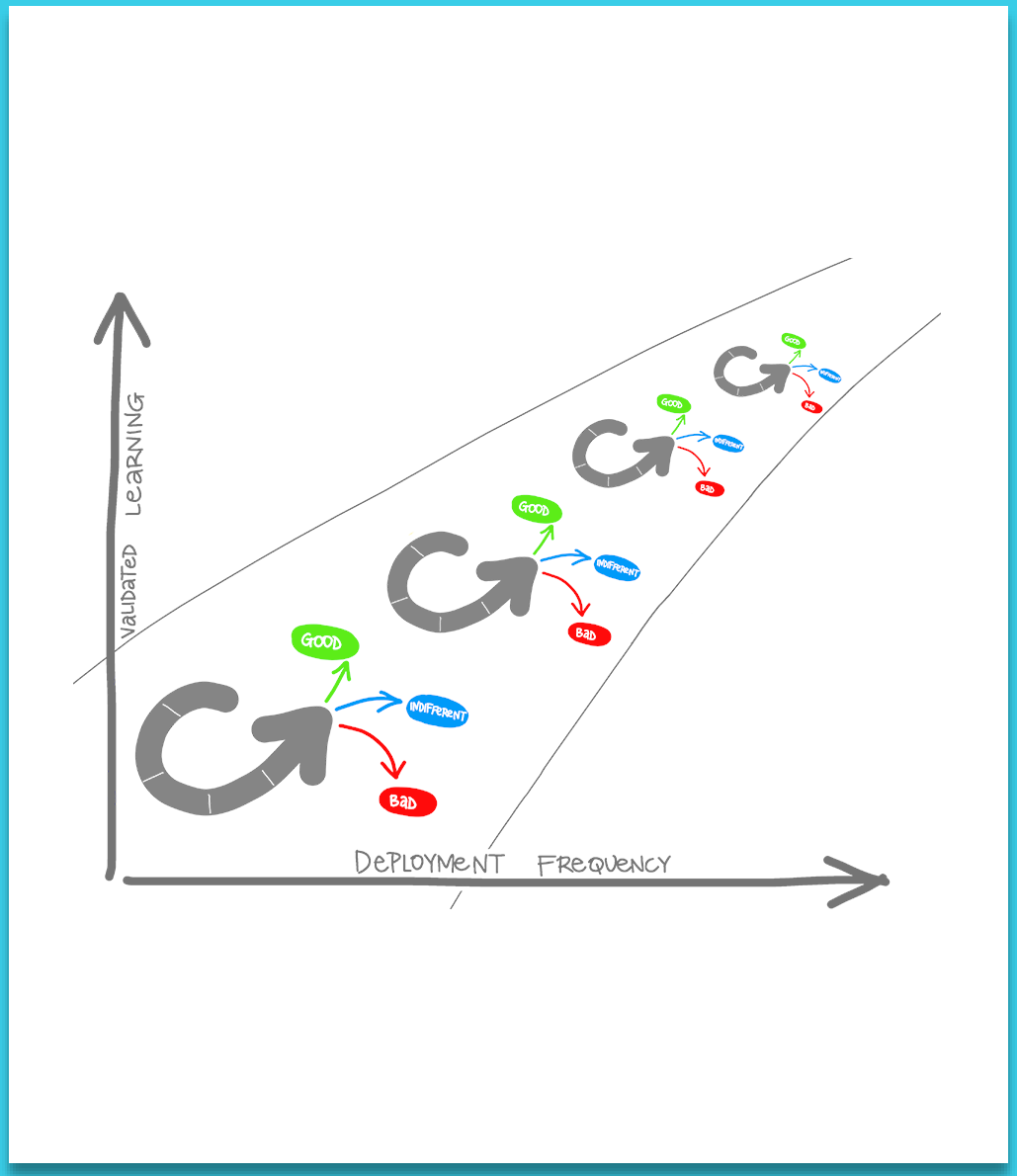
Validate Learning

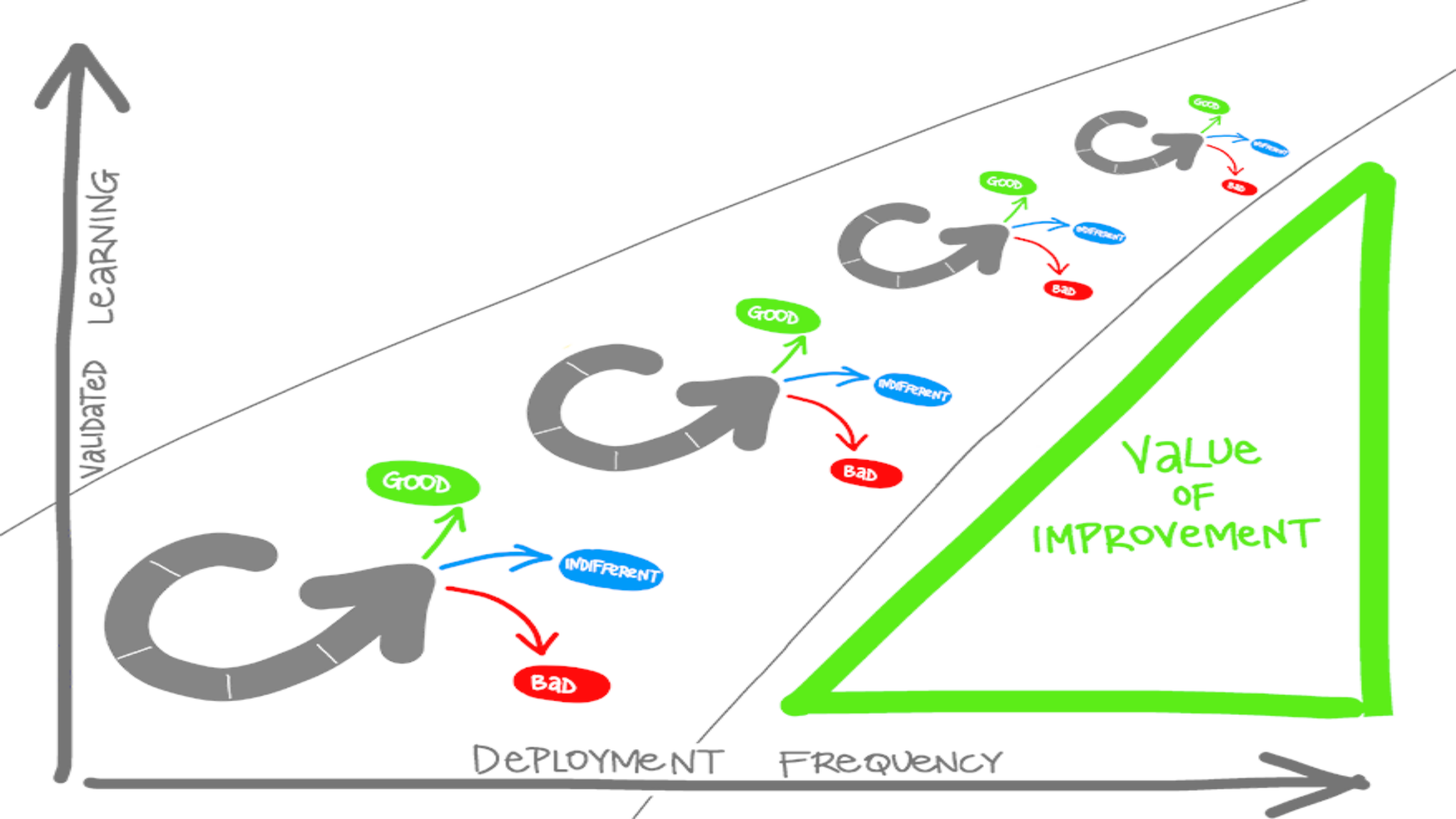
- How quickly you can fail fast or double down is determined by your cycle time. Also, in how long that loop takes, or in lean terms. Your cycle time determines how quickly you can gather feedback to determine what happens in the next loop. The feedback that you collect with each cycle should be factual, actionable data.



DevOps Practices

- You shorten your cycle time by working in smaller batches.
- Using more automation.
- Hardening your release pipeline.
- Improving your telemetry.
- Deploying more frequently.




VALIDATED LEARNING
GOOD
INDIFFERENT
BAD
GOOD
INDIFFERENT
BAD
GOOD
INDIFFERENT
BAD
GOOD
INDIFFERENT
BAD
VALUE OF IMPROVEMENT
DEPLOYMENT FREQUENCY

Continuous Integration

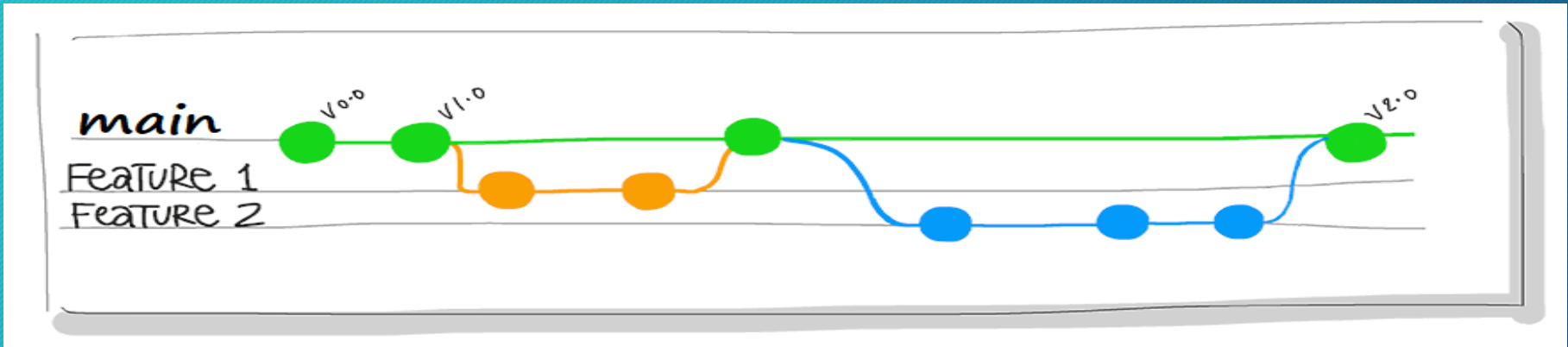
- How long does it take to deploy a change of one line of code or configuration?
  - Continuous Integration drives the ongoing merging and testing of code, leading to an early finding of defects. Other benefits include less time wasted fighting merge issues and rapid feedback for development teams.

Continuous Delivery

- Continuous Delivery of software solutions to production and testing environments helps organizations quickly fix bugs and respond to ever-changing business requirements.

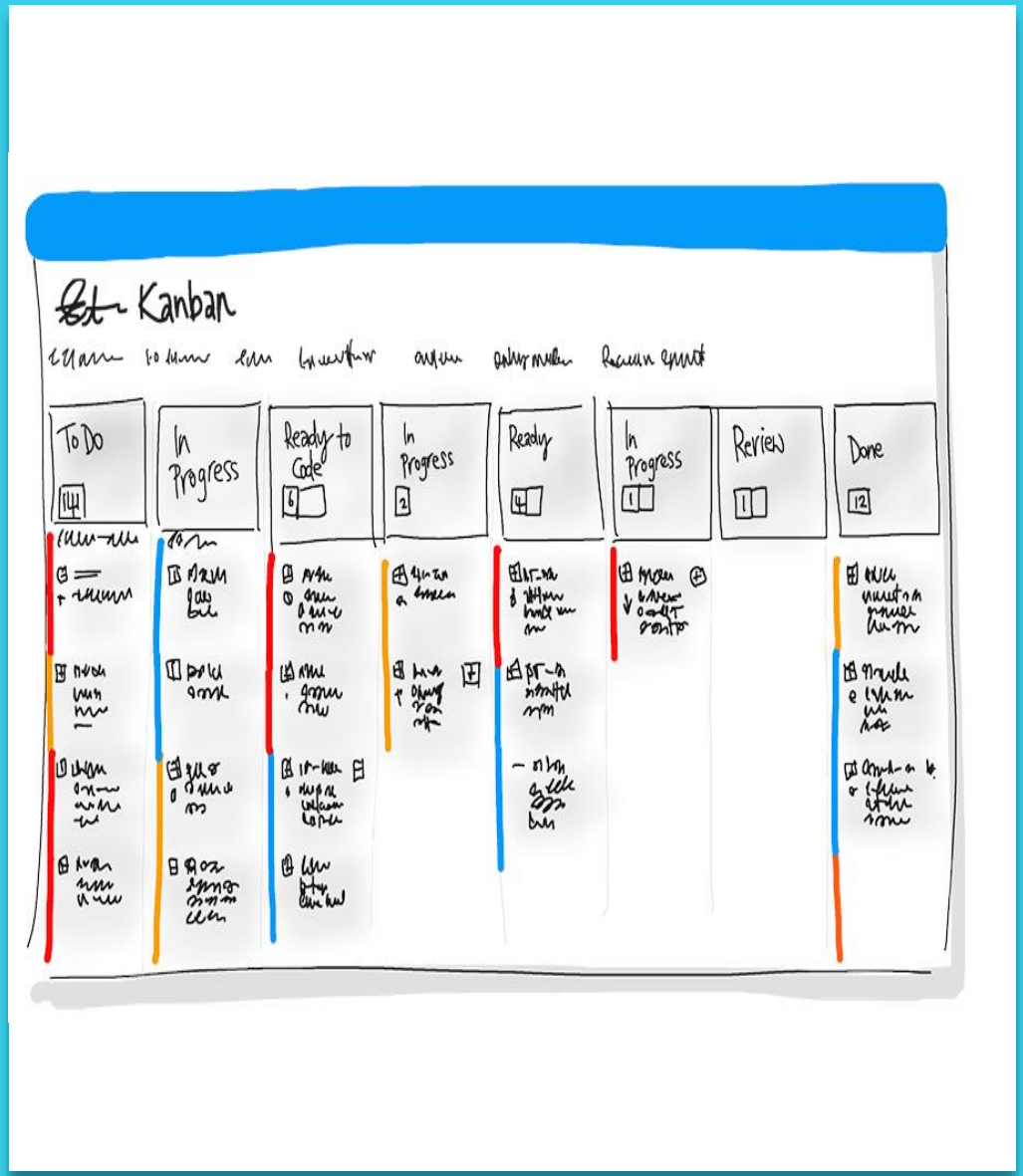
Version Control

- Version Control, usually with a Git-based Repository, enables teams worldwide to communicate effectively during daily development activities. Also, integrate with software development tools for monitoring activities such as deployments.



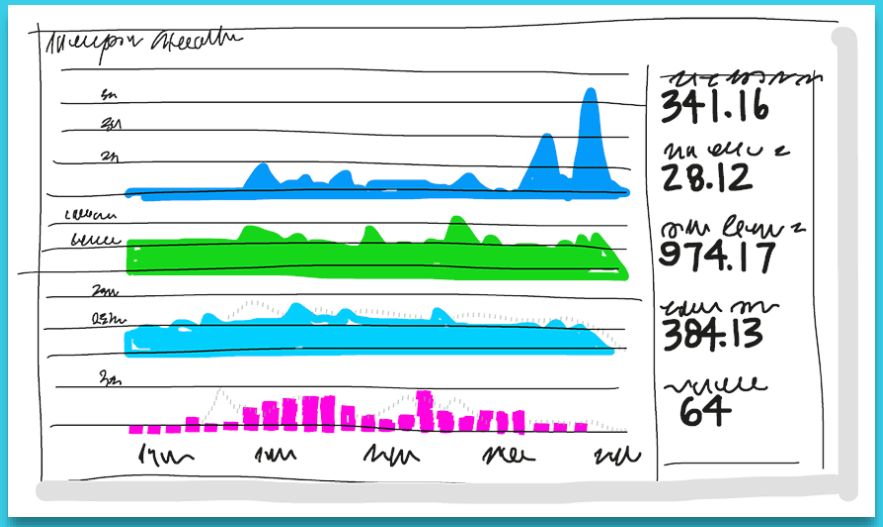
Agile Planning

- Use Agile planning and lean project management techniques to:
  - Plan and isolate work into sprints.
  - Manage team capacity and help teams quickly adapt to changing business needs.
  - A DevOps Definition of Done is working software collecting telemetry against the intended business goals.



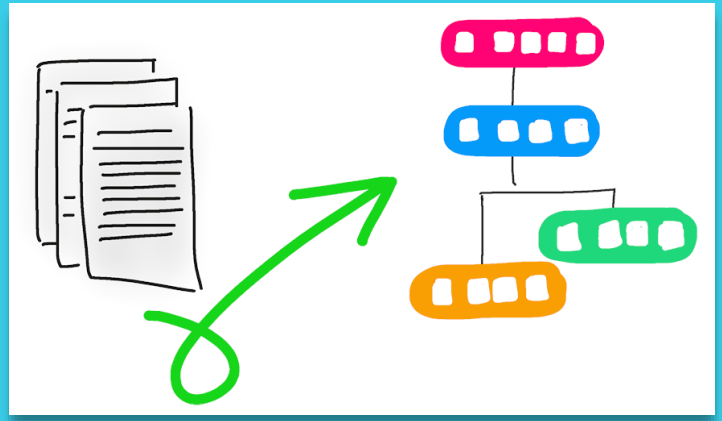
Monitoring and Logging

- Monitoring and Logging of running applications. Including production environments for application health and customer usage. It helps organizations create a hypothesis and quickly validate or disprove strategies. Rich data is captured and stored in various logging formats.



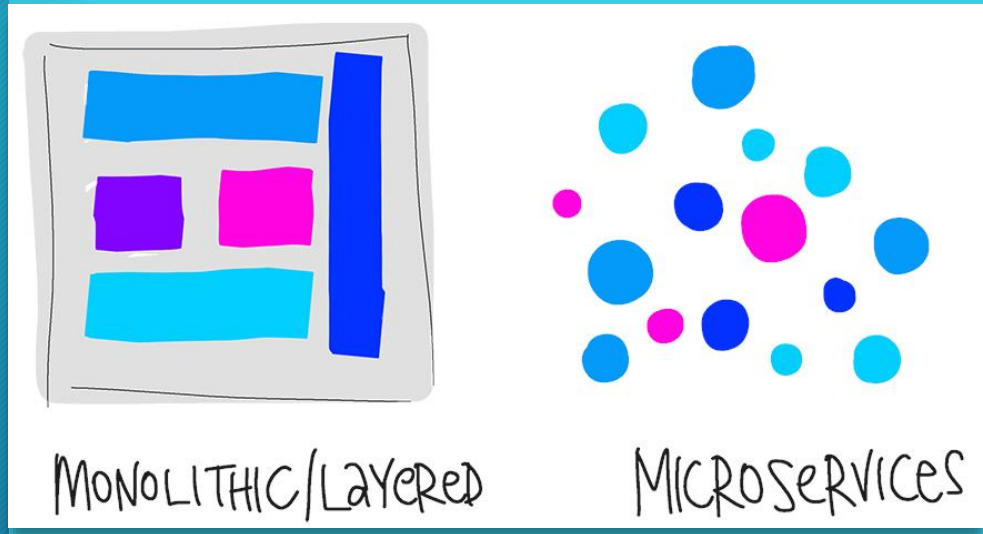
Infrastructure as Code (IaC)

- Enables the automation and validation of the creation and teardown of environments to help deliver secure and stable application hosting platforms.



Microservices architecture / Containers

- Use Microservices architecture to isolate business use cases into small reusable services that communicate via interface contracts. This architecture enables scalability and efficiency.
- Containers are the next evolution in virtualization. They're much more lightweight than virtual machines, allow much faster hydration, and easily configure files.



Apply DevOps to which Project ?

- The terms greenfield and brownfield have their origins in residential and industrial building projects.
- A greenfield project is one done on a green field, undeveloped land.
- A brownfield project is done on the used ground for other purposes. Because of the land use that has once occurred, there could be challenges reusing the land. Like existing buildings, some would be obvious but less obvious, like polluted soil.

Apply DevOps to which Project ?

- Greenfield Project:
  - A greenfield project will always appear to be a more accessible starting point. A blank slate offers the chance to implement everything the way that you want.
  - You might also have a better chance of avoiding existing business processes that do not align with your project plans.
  - Suppose current IT policies do not allow the use of cloud-based infrastructure. In that case, the project might be qualified for entirely new applications designed for that environment from scratch.
  - For example, you can sidestep internal political issues that are well entrenched.

Apply DevOps to which Project ?

- Brownfield Project:
  - Usually, brownfield projects come with:
    - The baggage of existing codebases.
    - Existing teams.
    - A significant amount of technical debt.

Apply DevOps to which Project ?

- Greenfield DevOps project would be easier to manage and to achieve success?
- A common misconception is that DevOps is only for greenfield projects and suits startups best. However, DevOps can also succeed with brownfield projects.
- The beauty of these projects is that there's often a large gap between customer expectations and delivery.
- The teams involved may well realize that the status quo needs to change. They've lived the challenges and the limitations associated with what they're currently doing.
- The system is often crucial for organizations. It might also be easier to gain more robust management buy-in for these projects because of the potential benefits delivered.
- Management might also have a stronger sense of urgency to point brownfield projects in an appropriate direction when compared to greenfield projects that do not currently exist.

Types of System

- Systems of record
  - Systems that provide the truth about data elements are often-called systems of record. These systems have historically evolved slowly and carefully. For example, it is crucial that a banking system accurately reflects your bank balance. Systems of record emphasize accuracy and security.
- Systems of engagement
  - Many organizations have other systems that are more exploratory. These often use experimentation to solve new problems. Systems of engagement are modified regularly. Usually, it is a priority to make quick changes over ensuring that the changes are correct.

Types of Users

- Canary
  - **Canary** users voluntarily test bleeding edge features as soon as they're available.
- Early Adopters
  - **Early adopters** who voluntarily preview releases, considered more refined than the code that exposes canary users.
- Users
  - **Users** who consume the products after passing through canary and early adopters.