Welcome to:

Introduction to DevOps

Unit objectives

After completing this unit, you should be able to:

- Understand the DevOps Process
- Gain knowledge on the difference between DevOps and Agile
- Learn about Kanban and Scrum
- Understand Software release plan

What is DevOps?

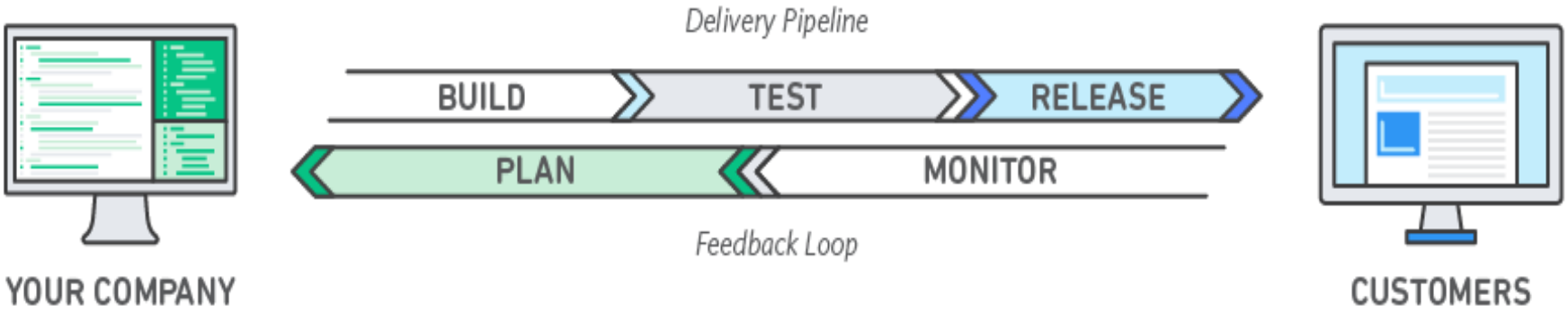

Figure: DevOps overview

Source: https://images.app.goo.gl/BDQSUrU8CDdnVSBv9

How DevOps works

- DevOps is the activity of processes and software engineering collaborating over the whole lifecycle of the project, from the planning and construction phases to the launches and maintenance of manufacturing.

- The team works with each other to accomplish the final aim, ranging from design and construction to automation evaluation and from software deployment to quality production.

- Such two teams are periodically combined into one department where engineers operate during the product lifecycle, learning a broad range of expertise, from development to implementation.

- People with both development and operational skills work together and use different CI-CD (Continuous Integration - Continuous Deployment) and Monitoring tools to respond quickly to customer needs and fix problems and bugs.

- The Quality Assurance (QA) and protection teams will incorporate production and operations into the lifecycle of certain DevOps models. If a DevOps unit works on defense it is often named DevSecOps.

Benefits of DevOps

- Break down the Silos.
- Speed.
- Fast delivery.
- Reliability.
- Scale.
- Team collaboration.
- Security.
- Risk management.

Why DevOps matters

- Technology and the web have changed the environment from retail and finance sectors. Technology no longer just serves one industry; rather it is an important component of any aspect of the company.

- Businesses communicate with their consumers as all forms of tools through web platforms or apps, and portable software.

How to adopt a DevOps model

- DevOps cultural philosophy.

- Shifting to DevOps needs a shift in mindset and community. DevOps is simply around increasing obstacles that are typically Siloed amongst production and operations. Advanced engineering and operational departments might not even be present in certain companies; engineers could do each.

- The two departments work closely with DevOps to maximize efficiency and operating stability for developers. They also aim to collaborate, increase productivity, and enhance the level of consumer services they offer.

- We assume direct control of their programs; also known historically as the desires of the end-user and the responsibilities or names we interpret so they may relate to fulfilling those desires. Such departments will also collaborate with the quality control and health staff.

DevOps practice explained

- Several main strategies allow businesses to evolve more efficiently by automating and streamlining the processes of product engineering and network management. Any of these routines are conducted with the correct method.

- The most common method is to perform non-trivial updates.

- This way businesses will develop with their consumers quicker. These upgrades are usually periodic, instead of sporadic changes under conventional means of updating. Frequent but small upgrades will make any implementation less risky. Since groups can recognize the final extension which leads to errors, they can allow teams to repair bugs more quickly.

- Though releases range in frequency and duration, organizations following the DevOps framework are patched more often than businesses utilizing conventional software development approaches.

DevOps practices

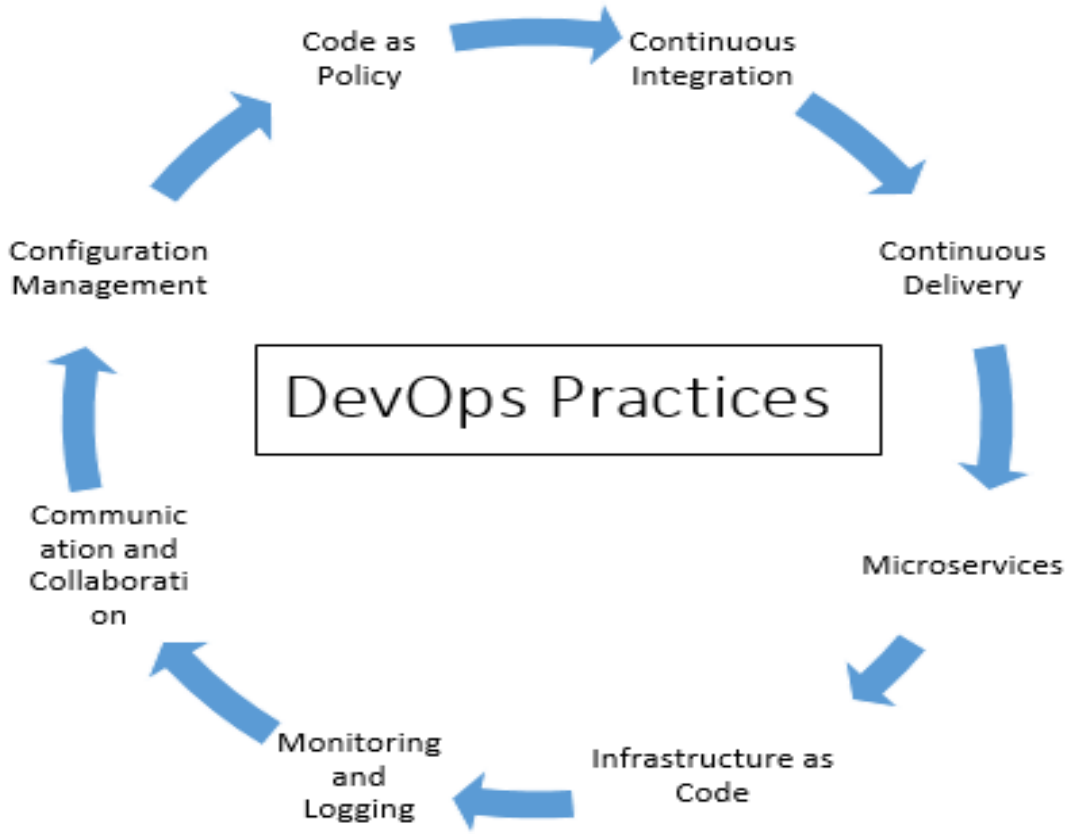

Figure: DevOps practices

DevOps history: How it start?

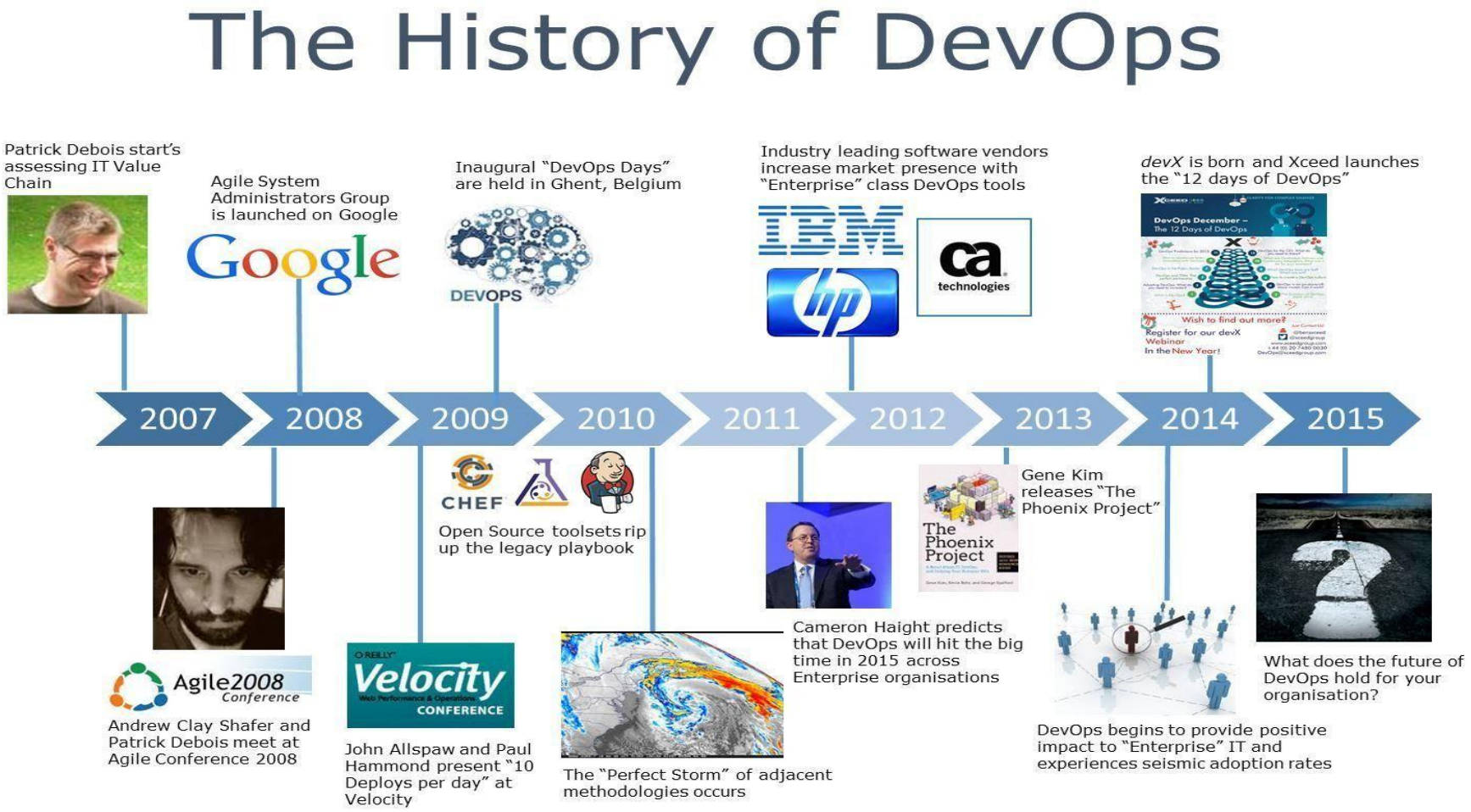

Figure: DevOps History

Source: https://images.app.goo.gl/ynepJfkk2MK2cL5p9

DevOps tools:
Phase 1 – Continues development

- Phase 1: Continuous development.
- Phase 2: Continuous integration.
- Phase 3: Continuous testing.
- Phase 4: Continuous deployment.
- Phase 5: Continuous monitoring.

Phase 2: Continuous integration

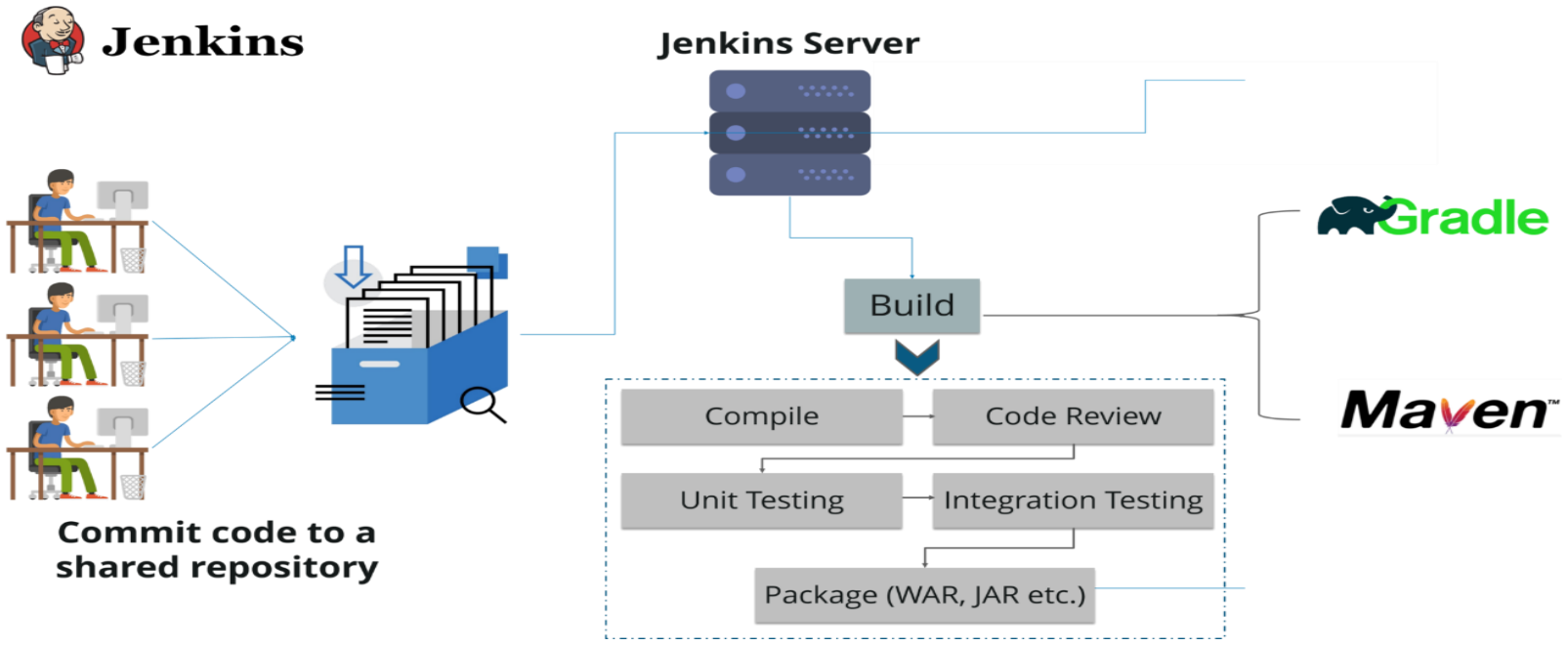

Figure: Jenkins process

Source: https://images.app.goo.gl/xAr4xZdV62NVMkno8

Self evaluation: Exercise 1

- To continue with the training, after learning the various steps involved in DevOps, it is instructed to utilize the concepts to perform the following activity.

- You are instructed to write the following activities using DevOps management tool.

- Exercise 1: Setup Environment for Devops.

Phase 3: Continuous testing

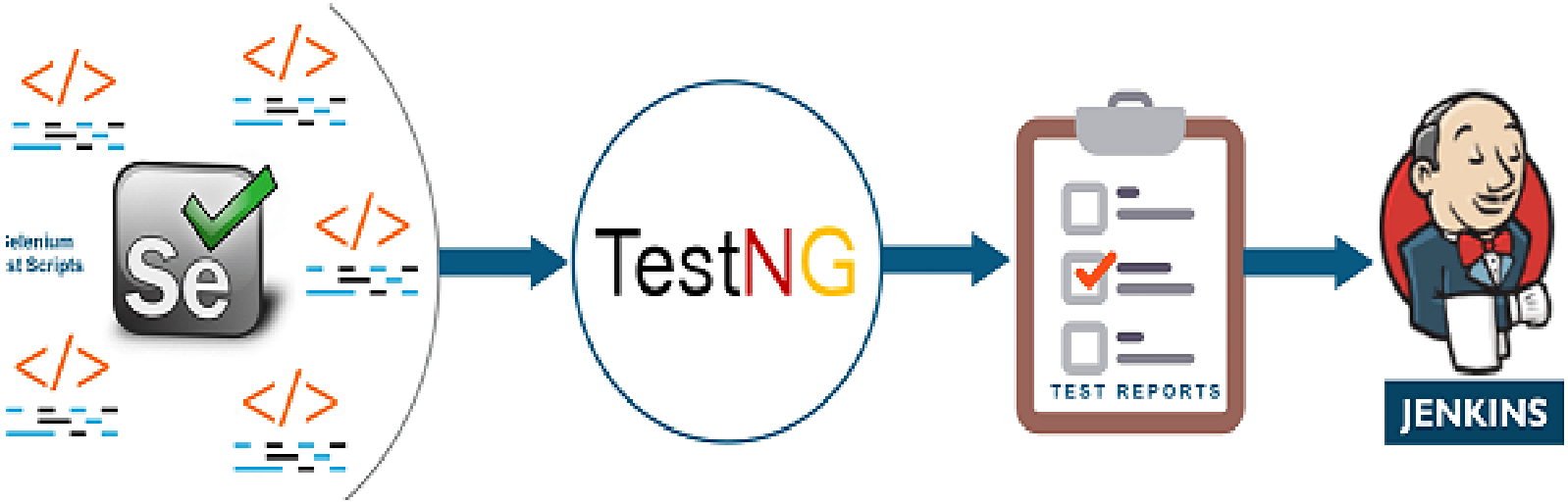

Figure: Selenium and TestNG process

Source: https://images.app.goo.gl/GHARdYqUPeRtAm5M8

Phase 4: Continuous deployment

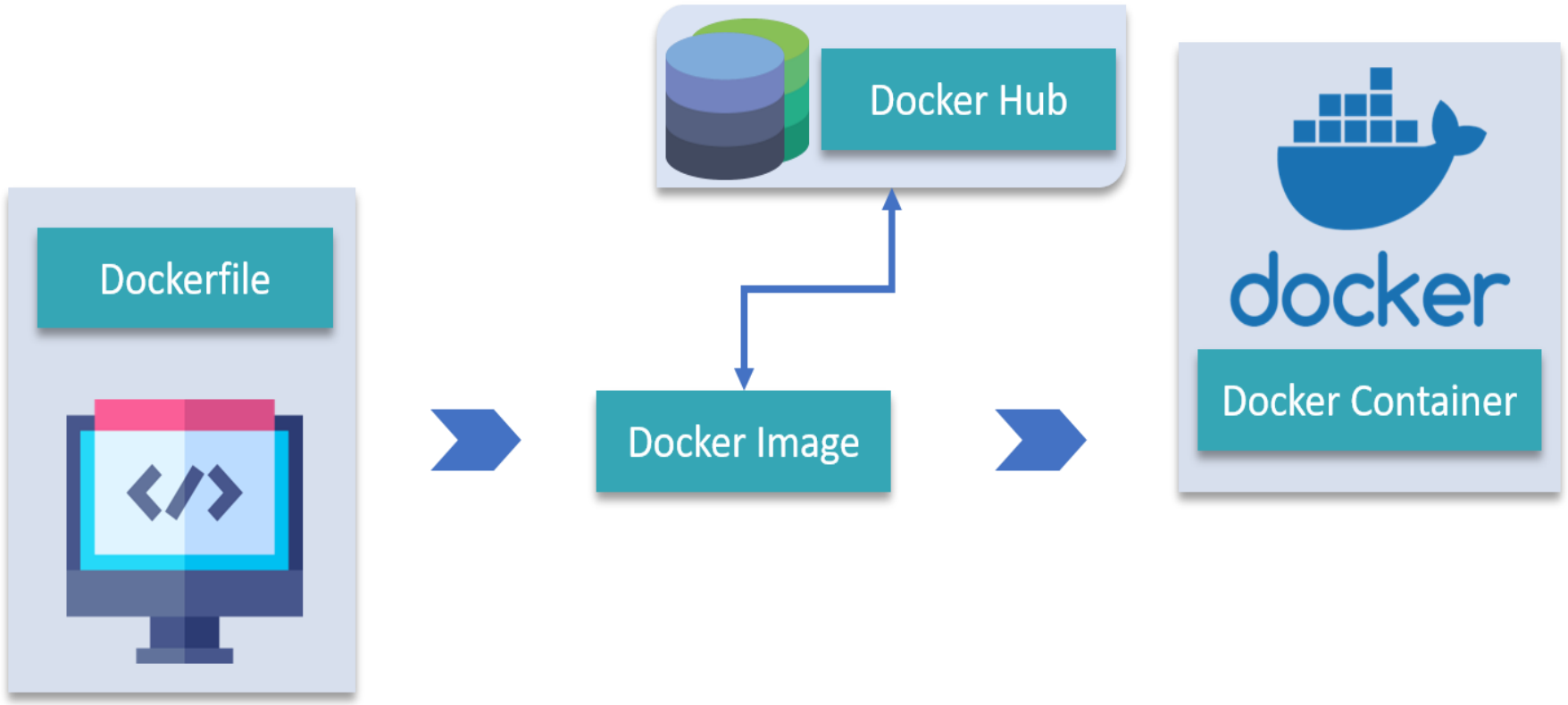

Figure: Docker Process

Source: https://images.app.goo.gl/6ZeVghAFSics11tc6

Self evaluation: Exercise 2

- To continue with the training, after learning the various steps involved in DevOps, it is instructed to utilize the concepts to perform the following activity.

- You are instructed to write the following activities using DevOps management tool.

- Exercise 2: Setup Environment for Docker.

Phase 5: Continuous monitoring

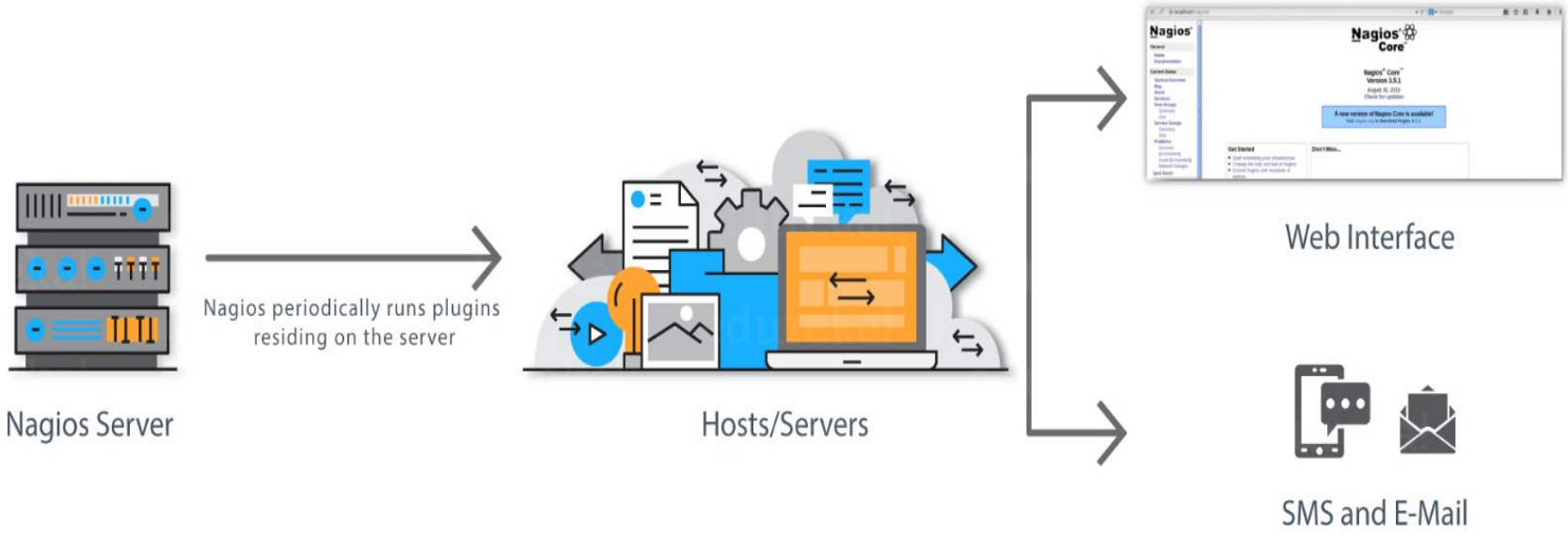

Figure: Nagios Process

Source: https://images.app.goo.gl/6qv8QNE38JErrQSX6

DevOps: Why is it popular?

- 46% more frequent code deployment
- 440x shorter time from commit to deploy
- 96x faster TTR(Time to recover)
- 5x ower change failure rate

Automation is on a rise. Companies now do less manual work and have automated.

- 33% more of their configuration management
- 30% more their deployment
- 27% more of their change approval process
- 27% more of their testing

Figure: DevOps Popularity

Source: https://images.app.goo.gl/EkokhcdjAurGdE4r8

Self evaluation: Exercise 3

- To continue with the training, after learning the various steps involved in DevOps, it is instructed to utilize the concepts to perform the following activity.

- You are instructed to write the following activities using DevOps management tool.

- Exercise 3: Setup Environment for chef fundamentals.

Agile vs DevOps

Features	DevOps	Agile
Agility	Agility in both Development & Operations	Agility in only Development
Processes/ Practices	Involves processes such as CI, CD, CT, etc.	Involves practices such as Agile Scrum, Agile Kanban, etc.
Key Focus Area	Timeliness & quality have equal priority	Timeliness is the main priority
Release Cycles/ Development Sprints	Smaller release cycles with immediate feedback	Smaller release cycles
Source of Feedback	Feedback is from self (Monitoring tools)	Feedback is from customers
Scope of Work	Agility & need for Automation	Agility only

Table: Features of DevOps and Agile

Source: https://images.app.goo.gl/TDiP7e1phxop2BKE6

Introduction to kanban

- Kanban is a visual system for managing work as it moves through a process. Kanban visualizes both the process (the workflow) and the actual work passing through that process.

- The goal of kanban is to identify potential bottlenecks in your process and fix them so work can flow through it cost-effectively at optimal speed or throughput.

Types of kanban board

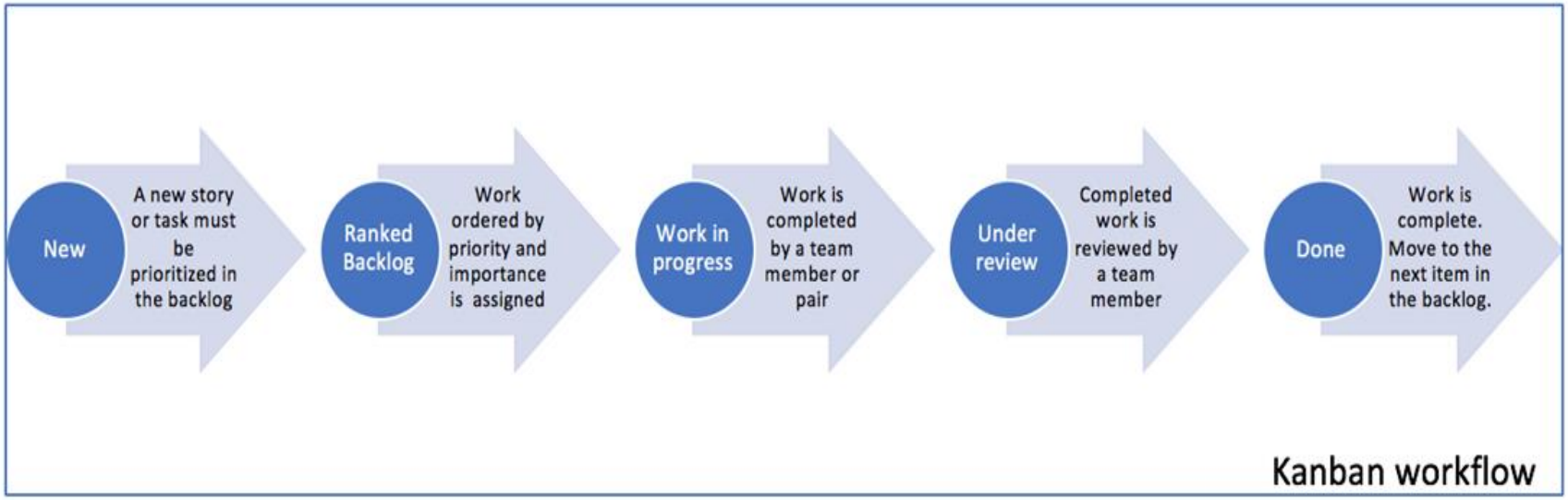

Figure: Kanban workflow

Source: https://www.ibm.com/garage/method/practices/culture/practice-kanban-method

Create a kanban board

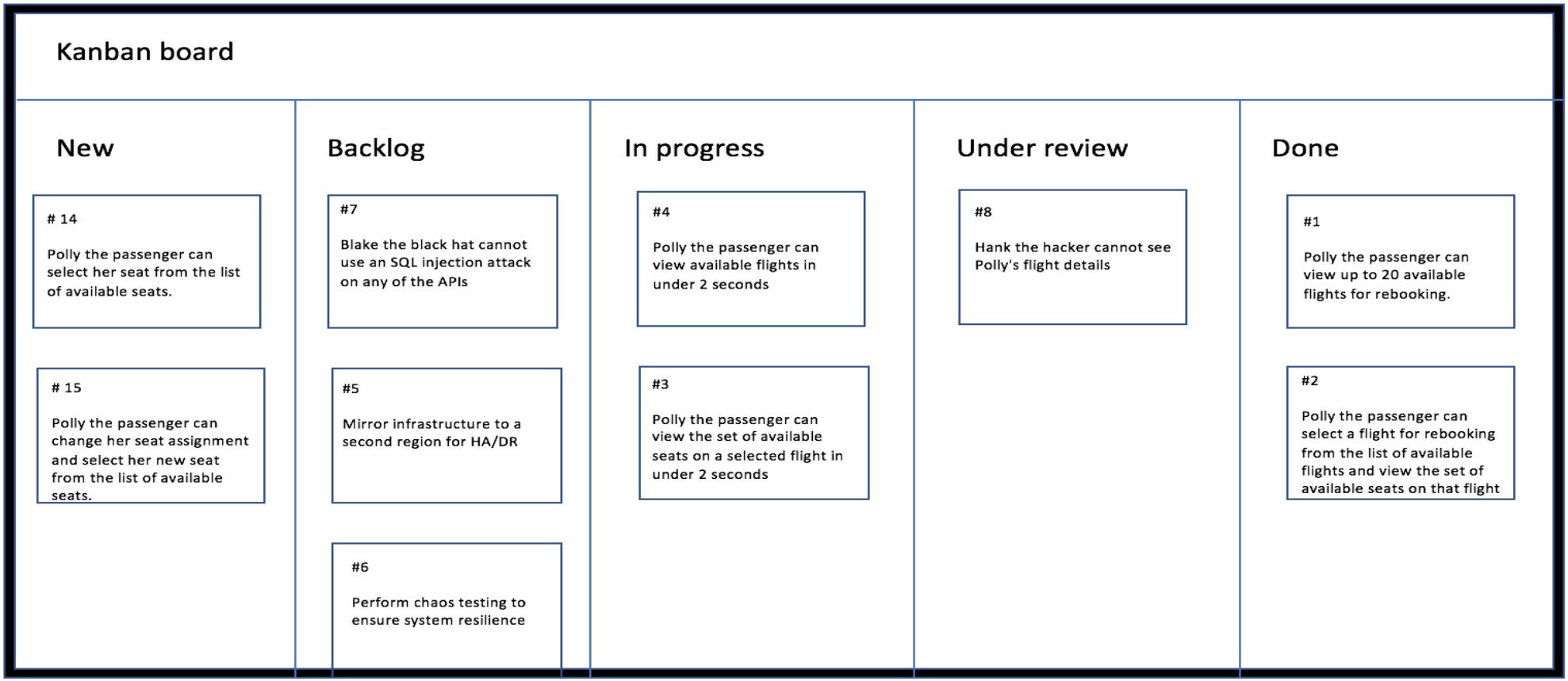

Figure: Kanban board

Source: https://www.ibm.com/garage/method/practices/culture/practice-kanban-method

Kanban with IBM tools

- IBM Engineering Workflow Management.

- IBM® Engineering Workflow Management (formerly IBM Rational Team Concert™) is a team collaboration tool that integrates development tasks, including iteration planning, change management, defect tracking, source control, build automation, and reporting.

- You can use IBM Engineering Workflow Management on-premises or on IBM Cloud. The cloud offering provides these benefits:
  - Eliminates the need to install, configure, and maintain the software.
  - Is secure and highly available.
  - Provides a scalable single-tenant infrastructure.
  - Reduces costs with a pay-as-you-go model. Alternatively, you can bring your own license.
  - Can be customized to meet your needs.

Scrum application delivery pipeline and support team

- The benefits of an automated software delivery pipeline:
  - An automated software delivery pipeline brings great value to teams.
  - By providing automation, a pipeline removes the need for expensive and error-prone manual tasks.
  - New team members can get started and become productive faster because they don't need to learn a complex development and test environment.
  - Teams can detect any code that is not fit for delivery and then reject the code and provide feedback as early as possible.
  - A pipeline provides visibility into and confidence in the code as it progresses through successive stages where the testing becomes more like production.

Self evaluation: Exercise 4

- To continue with the training, after learning the various steps involved in DevOps, it is instructed to utilize the concepts to perform the following activity.

- You are instructed to write the following activities using DevOps management tool.

- Exercise 4: Setup Environment for Puppet Fundamentals.

An orchestration framework for continuous delivery

- A final aspect of building a delivery pipeline is developing the orchestration framework that ties all the tools together. For example, IBM Cloud Continuous Delivery is a framework that includes toolchains, which are sets of integrated tools, such as Delivery Pipeline and GitHub.

- If you're considering using an open-source tool or framework, use one that is supported by a vibrant and active community with many users. Such tools often support several environments, integrate with other tools, have a wide selection of plug-ins or extensions, and have many online sources for help or bug fixes. In addition, a larger pool of people will know how to use the tool. You might combine several open-source tools, such as Gradle and Jenkins to form the basis of your orchestration framework.

Software release plan

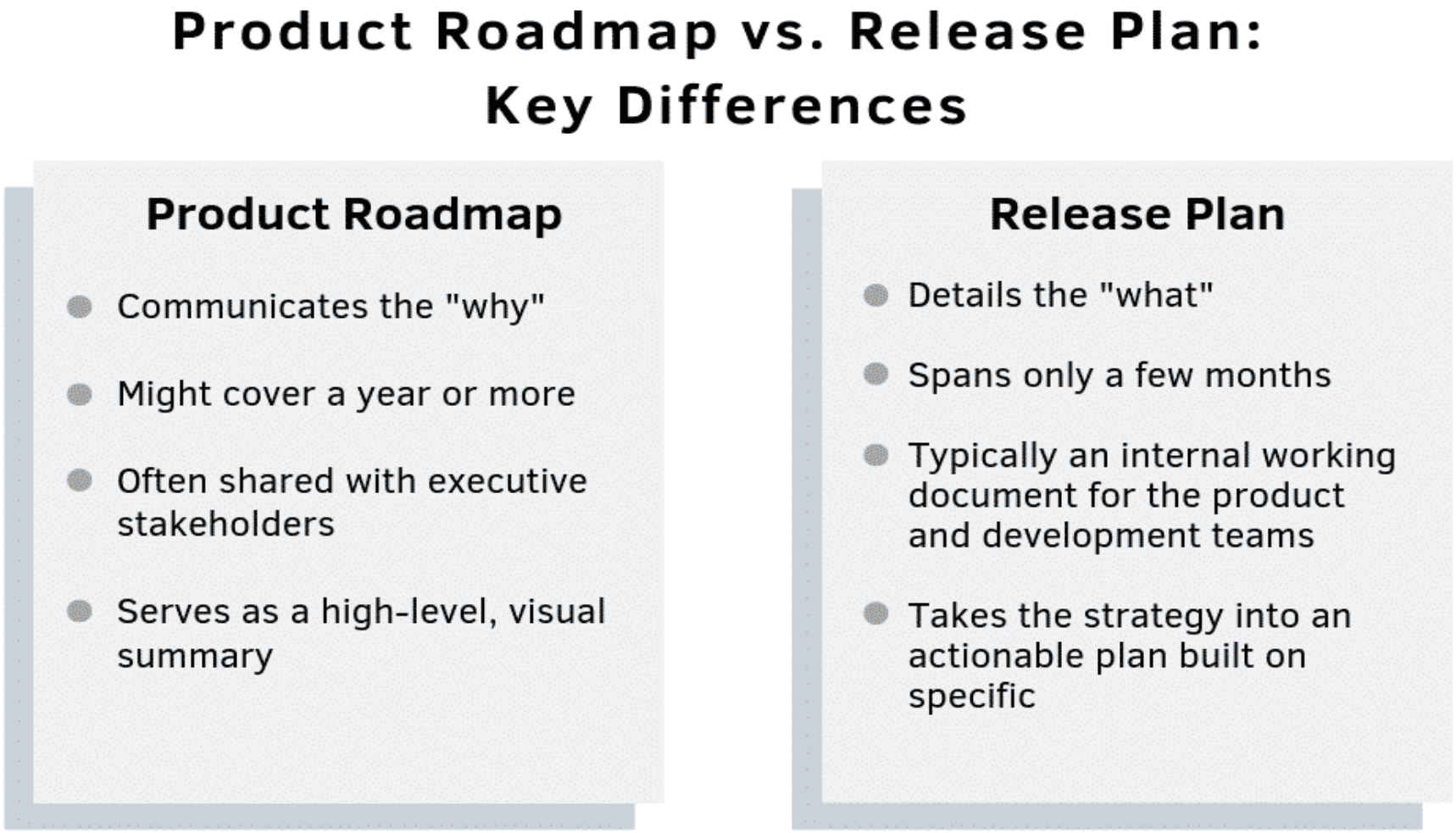

Figure: Product roadmap and release planning

Source: https://images.app.goo.gl/TGWeukSarCJssoDv7

Feedback and learning from feedback and improving the delivery

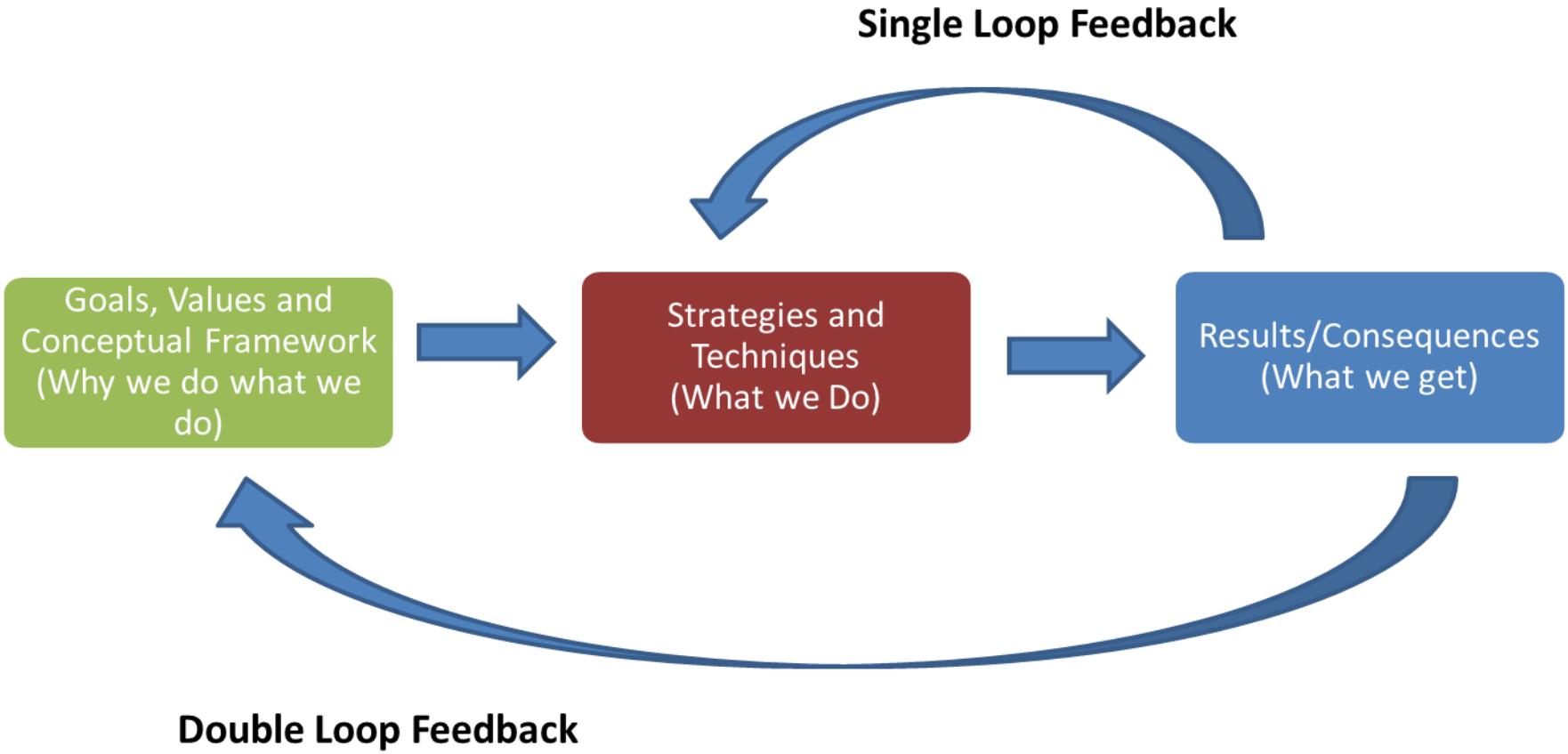

Figure: Feedback system in agile

Source: https://images.app.goo.gl/EW55B8qF8Hitvcfp6

DevOps toolchain

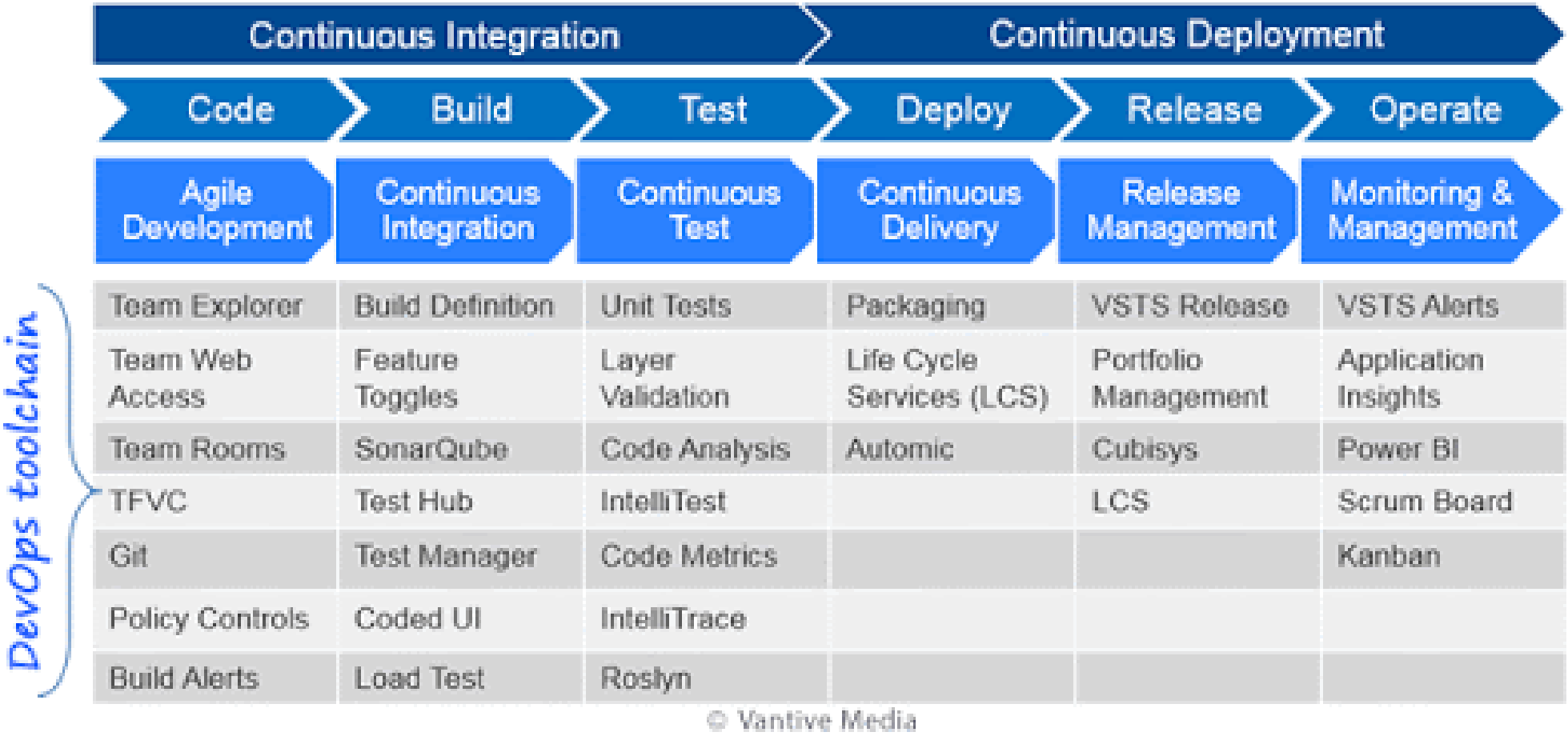

Figure: DevOps Tool Chain

Source: https://images.app.goo.gl/yB5rRnvpYU19qGyE8

DevSecOps

Figure: DevSecOps

Source: https://images.app.goo.gl/2RHxzB481cigEjya8

DevOps vs SRE

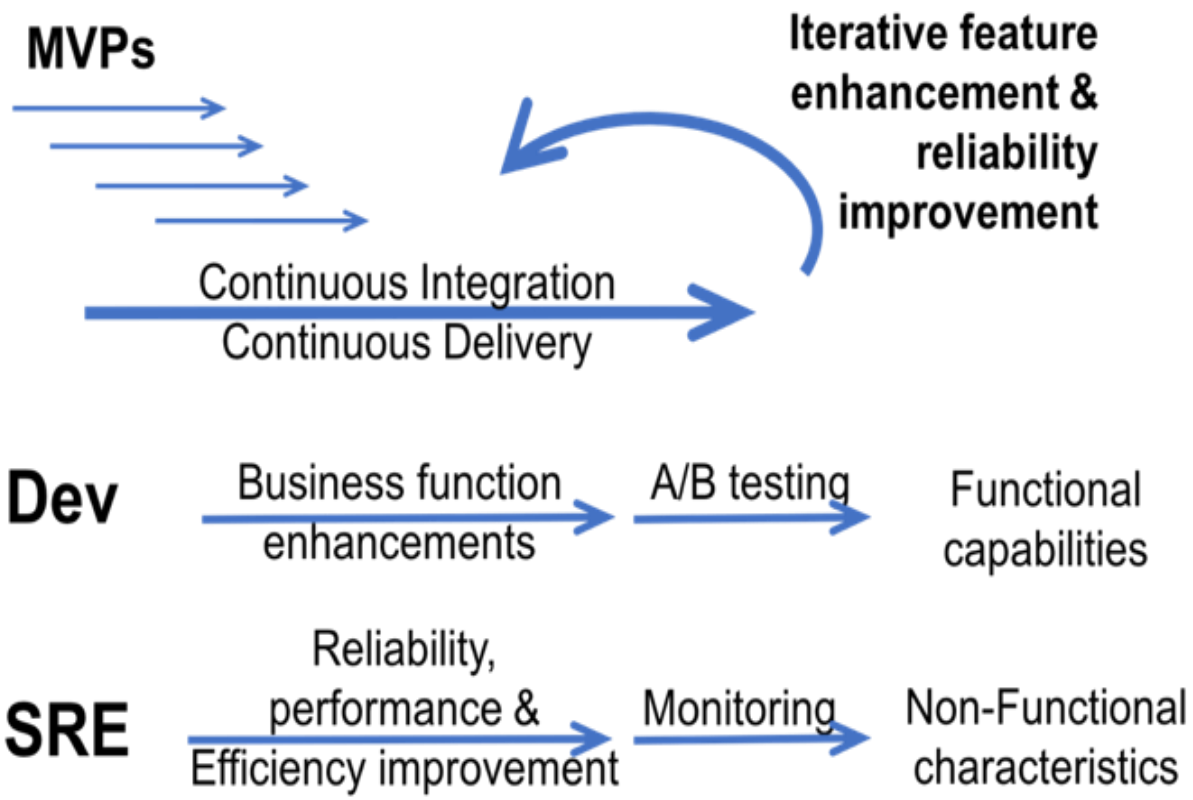

Figure: SRE View

Source: https://images.app.goo.gl/VieeNMJxA91uj56n7

Self evaluation: Exercise 5

- To continue with the training, after learning the various steps involved in DevOps, it is instructed to utilize the concepts to perform the following activity.

- You are instructed to write the following activities using DevOps Management tool.

- Exercise 5: Setting up Learning Environment for Jetkins.

Checkpoint (1 of 2)

Multiple choice questions:

1. What is the full form of DevOps?
   a) Development and Operations
   b) Digital and Operations
   c) Drive and Operations
   d) None of the above

2. Which of the followings are the popular tools for DevOps?
   a) Jenkins & Monit
   b) Nagios & ELK
   c) Jenkins & Ansible
   d) All the above

3. What are the key components of DevOps?
   a) Continuous Integration
   b) Continuous Testing
   c) Continuous Delivery & Monitoring
   d) All the above

Checkpoint solutions (1 of 2)

Multiple choice questions:

1. What is the full form of DevOps?
   - **a) Development and Operations**
   - b) Digital and Operations
   - c) Drive and Operations
   - d) None of the above

2. Which of the followings are the popular tools for DevOps?
   - a) Jenkins & Monit
   - b) Nagios & ELK
   - c) Jenkins & Ansible
   - **d) All the above**

3. What are the key components of DevOps?
   - a) Continuous Integration
   - b) Continuous Testing
   - c) Continuous Delivery & Monitoring
   - **d) All the above**

Checkpoint (2 of 2)

Fill in the blanks:

1. _____ is most important DevOps KPI?
2. _____ is the Popular Scripting Language Of DevOps?
3. DevOps is an extension of _____.
4. _____ team gets priority in DevOps.

True or False:

1. Is ansible an open-source tool. True/False
2. Maven is a source code management tool. True/False
3. Body element is not a part of the data structure of a Git repository. True/False

Checkpoint solutions (2 of 2)

Fill in the blanks:

1. **Deployment frequency** is most important DevOps KPI?
2. **Python** is the Popular Scripting Language of DevOps?
3. DevOps is an extension of **Agile.**
4. **Operational and Development** team gets priority in DevOps.

True or False:

1. Is Ansible an open-source tool. **True**
2. Maven is a source code management tool. **False**
3. Body element is not a part of the data structure of a Git repository. **False**

Question bank

Two mark questions:

1. What do you know about DevOps?
2. How is DevOps different from agile methodology?
3. Which are some of the most popular DevOps tools?
4. What are the different phases in DevOps?

Four mark questions:

1. Mention some of the core benefits of DevOps.
2. How will you approach a project that needs to implement DevOps?
3. What is the difference between continuous delivery and continuous deployment?
4. What is the role of configuration management in DevOps?

Eight mark questions:

1. How does continuous monitoring help you maintain the entire architecture of the system?
2. What is the role of AWS in DevOps?

Unit summary

Having completed this unit, you should be able to:

- Understand the DevOps Process
- Gain knowledge on the difference between DevOps and Agile
- Learn about Kanban and Scrum
- Understand Software release plan