# WHY YOU NEED TO WATCH THIS SECTION?

➤ This part of the class is the only theoretical one. However, it is of no less importance than the lab-based sections or the final project.

➤ You must have a firm understanding of DevOps as a set of tools, a culture, and a number of best practices.

➤ DevOps is used by companies and organizations to boost their productivity, and evolve rapidly.

➤ Adopting DevOps will enable you to deliver specific types of products and services at a much higher speed than when using the conventional software development path.

THE NON-DEVOPS WAY

➤ In order to understand the concept of DevOps and why it was introduced at the first place, let's see the traditional way of software development.
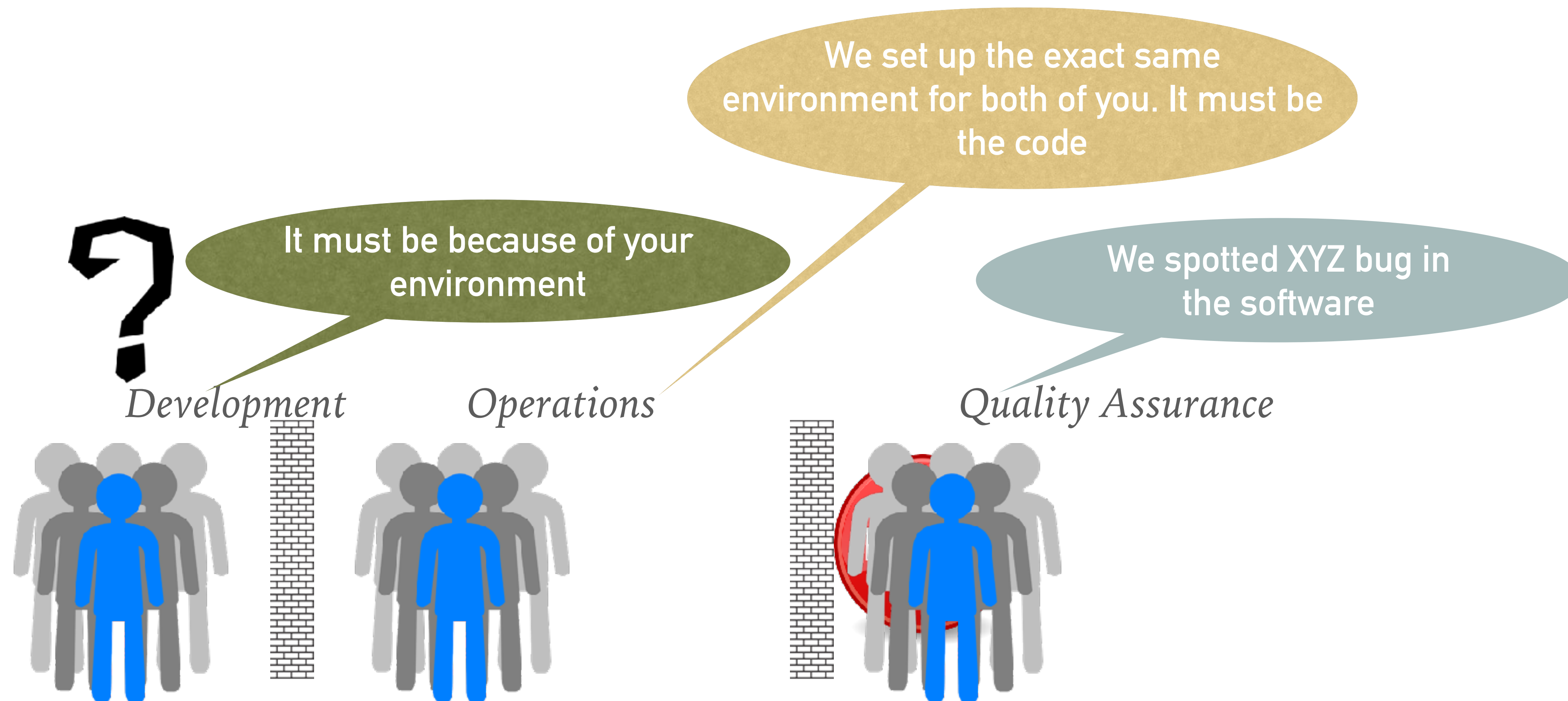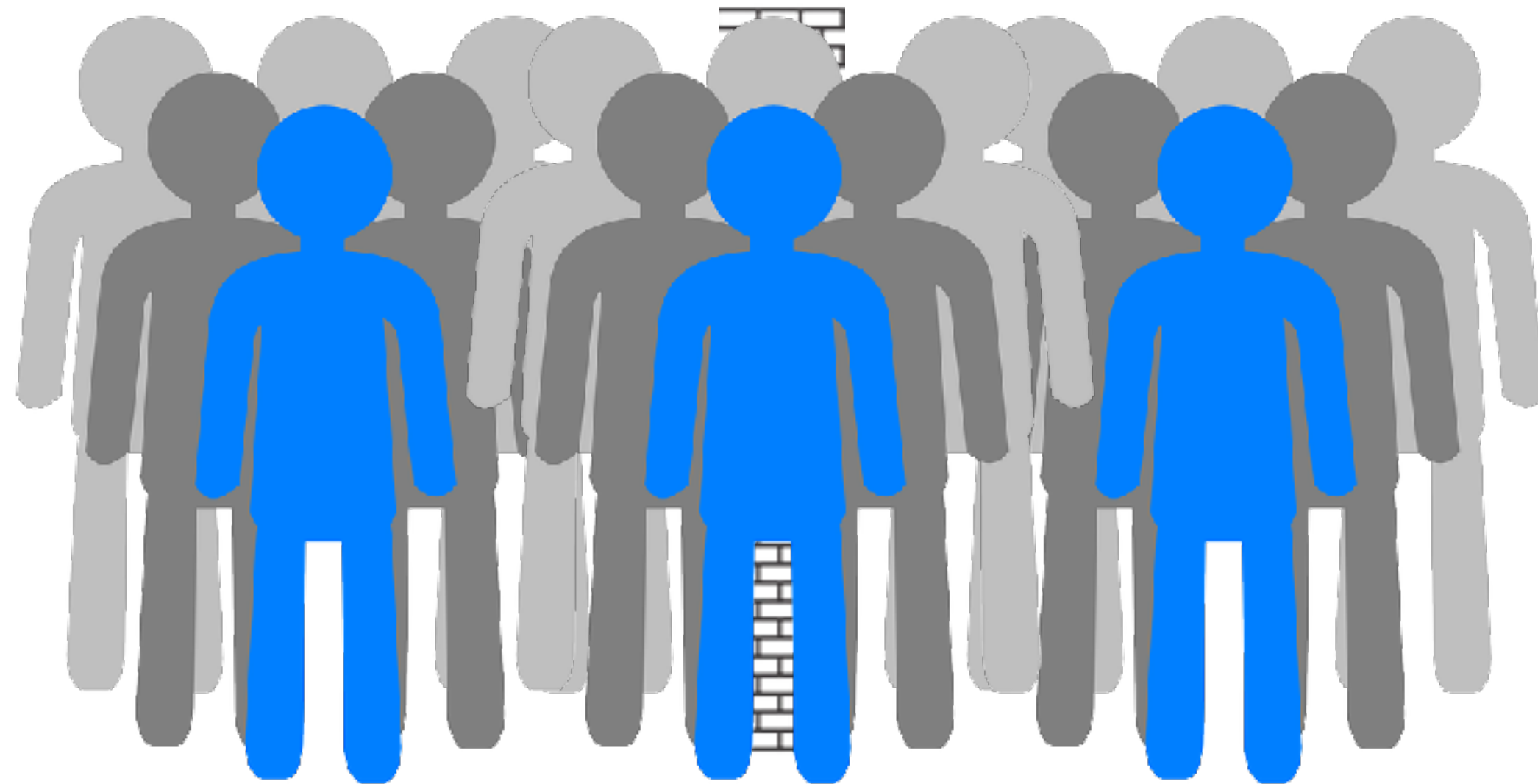
THE DEVOPS WAY

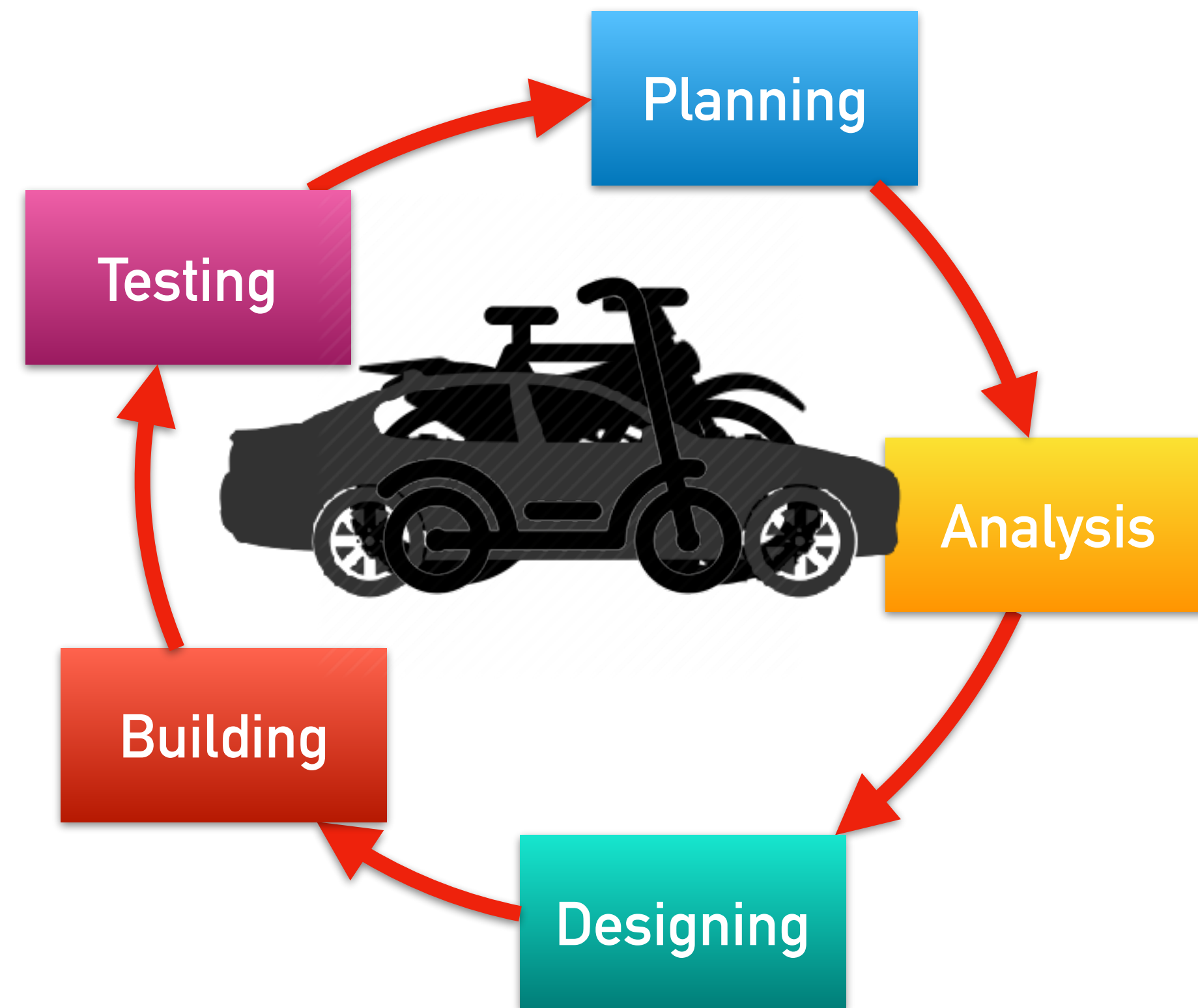*Development*   *DevOps*   *Operations*

WHAT ARE THE BENEFITS?

➤ Adopting a DevOps culture in an organization basically means there will be decrease in the time spent in:

   ➤ Software building and deployment

   ➤ Application maintenance and bug fixing

   ➤ Releasing new features and versions of the application.

➤ Of course this means higher product quality, faster productivity rate, increased customer satisfaction, more revenues and, thus, more profits.

WHAT BROUGHT THE CONCEPT?

➤ The emergence of DevOps is often attributed to the rise of Agile development as a modern way of software production.

➤ To understand Agile development, let's examine the traditional way that was used to develop applications, called the waterfall method.

Requirements	*No software yet*
Design	*No software yet*
Implementation	*No software yet*
Verification	*Software partially available*
Maintenance	*Software available*

Once a phase is finished *It takes months to finish*

Changes

THE AGILE METHODOLOGY
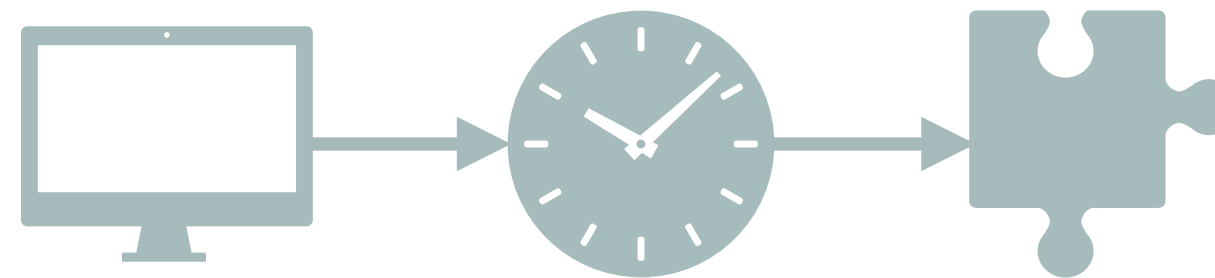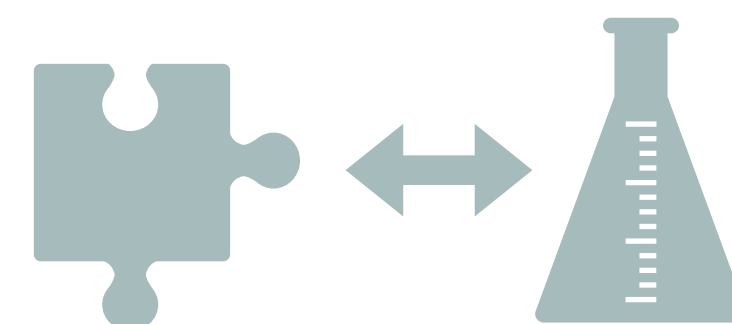
HOW DEVOPS FITS IN?

➤ DevOps helps organizations achieve the benefits of Agile development methodologies by shifting the workflow paradigm towards faster development phases.

➤ The following are the main stages of a project developed with DevOps principles in mind:
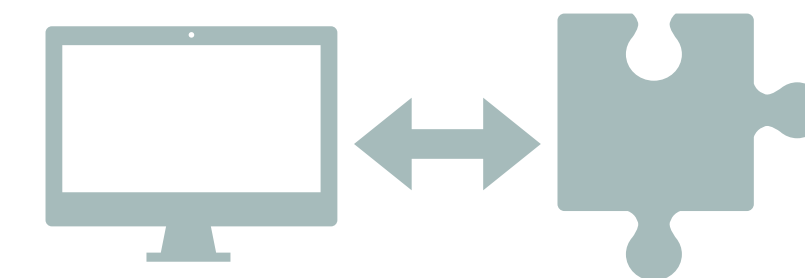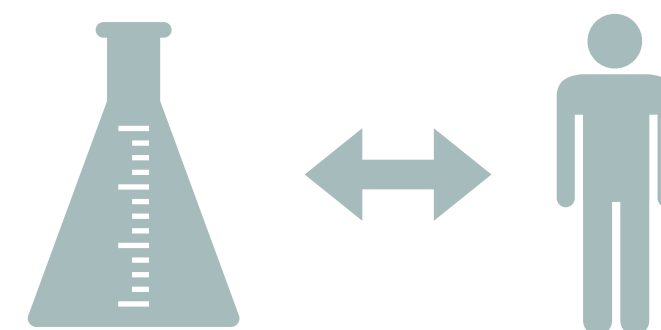
  ➤ Continuous integration

  ➤ Continuous delivery

  ➤ Continuous deployment

HOW TO DO IT?

➤ The strongest aspect of DevOps is the adoption of automation whenever possible. Automation boosts speed and lowers the error margin.

➤ In order to automate a task, you need a tool. DevOps tasks include planning, coding, building, testing, releasing, deploying, operating, and monitoring.

➤ DevOps tools form an ecosystem, where the output of each tool becomes the input of the next one.

➤ The advent of virtualization and - following it - the rise of cloud computing helped ease the process even more. Whole sets of tools, or even a complete infrastructure can be built on the cloud using providers like Amazon AWS, Google Cloud among others.

➤ Virtualization refers to abstracting the hardware resources on the machine so that a single server can run several machines at the same time. This yields massive decrease in the total cost of running a similar infrastructure using traditional physical hardware.

DEVOPS TOOLS

➤ For virtualization: VirtualBox, VMWare, and Vagrant (which builds upon one them)

➤ Containerization: Docker

➤ Continuous integration, continuous delivery, and continuous deployment: Jenkins

➤ Configuration management: Ansible

➤ Version control: Git

➤ The remaining part of this class will be dedicated to briefly introducing each of those tools. At the end, we are going to build a web application that adopts the DevOps culture explained earlier.

➤ The project will use the above tools to demonstrate a real-world working example.