

# DevOps Basic

What is DevOps

Why DevOps

Waterfall Model Phase

Advantages and Disadvantages

What is Agile

Agile Process

Advantages of Agile Model

Disadvantages of Agile Model

DevOps Life cycle

continuous development

continuous integration

continuous testing

continuous deployment

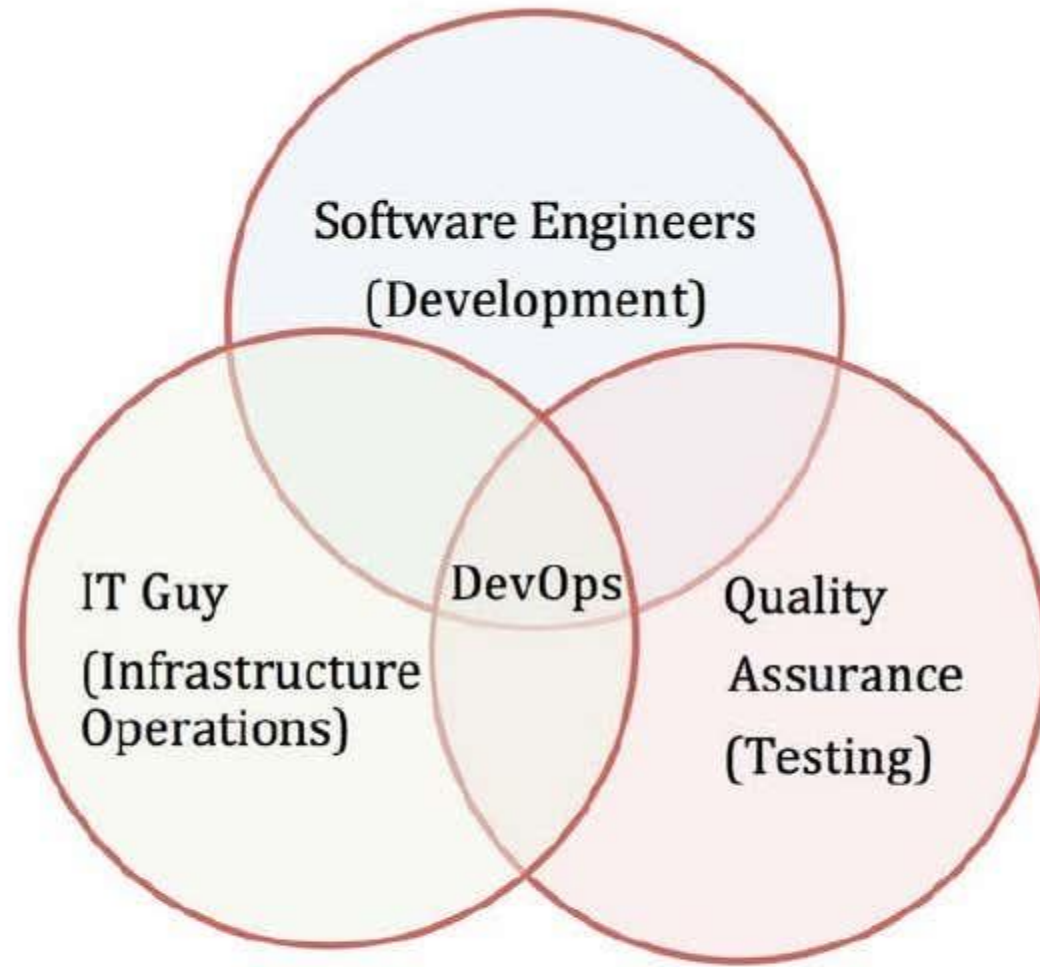
continuous monitoring

DevOps tools

# What is DevOps

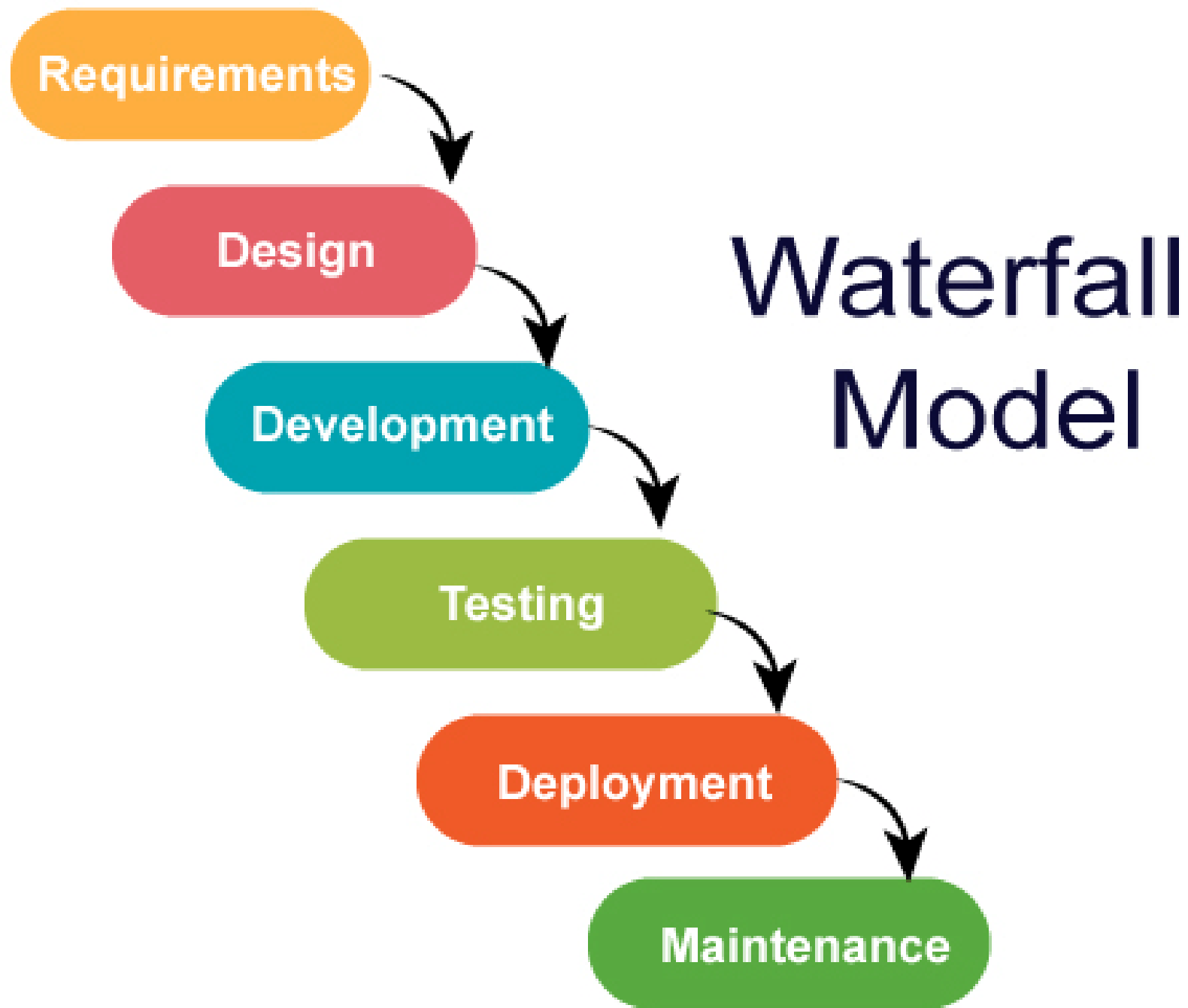
- The word **DevOps** is a combination of two words Development and Operations.
- The development team is responsible for developing, designing, and building the application.
- The operation team deals with the deployment and testing of the application.
- If there are problems with the application, the operation team also provides feedback to the development team.





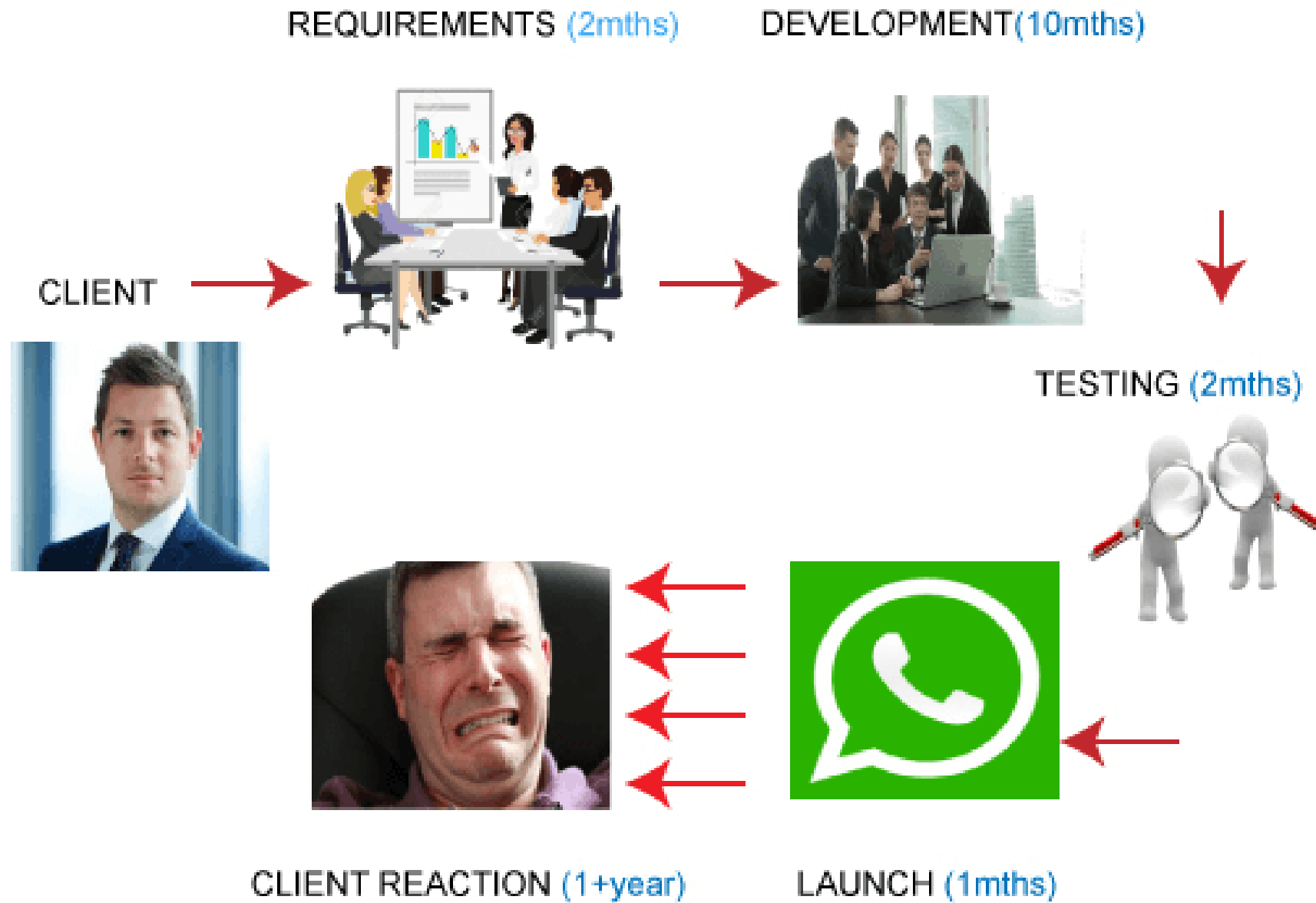
# Why DevOps?

- The problems faced in traditional models like in the **waterfall model** there is a problem of a one-way stream of work.
- If there is any mistake, the whole process repeats and there is no interaction with customers.
- This is solved in agile by splitting the whole development plan into several iterations for a better level of production efficiency.
- The agile model also includes customer interaction with the company to rectify the mistakes.
- But there is another problem faced in Agile too.



# Waterfall Model Phase

- phase 1 – Complete Requirement is gathered and SRS is developed
- phase 2 – This System is Planned and Designed using the SRS
- phase 3 – Implementation of the System takes place
- phase 4 – System is tested and its quality is assured
- phase 5 – System is deployed to the end users
- phase 6 – Regular Maintenance of the system is done



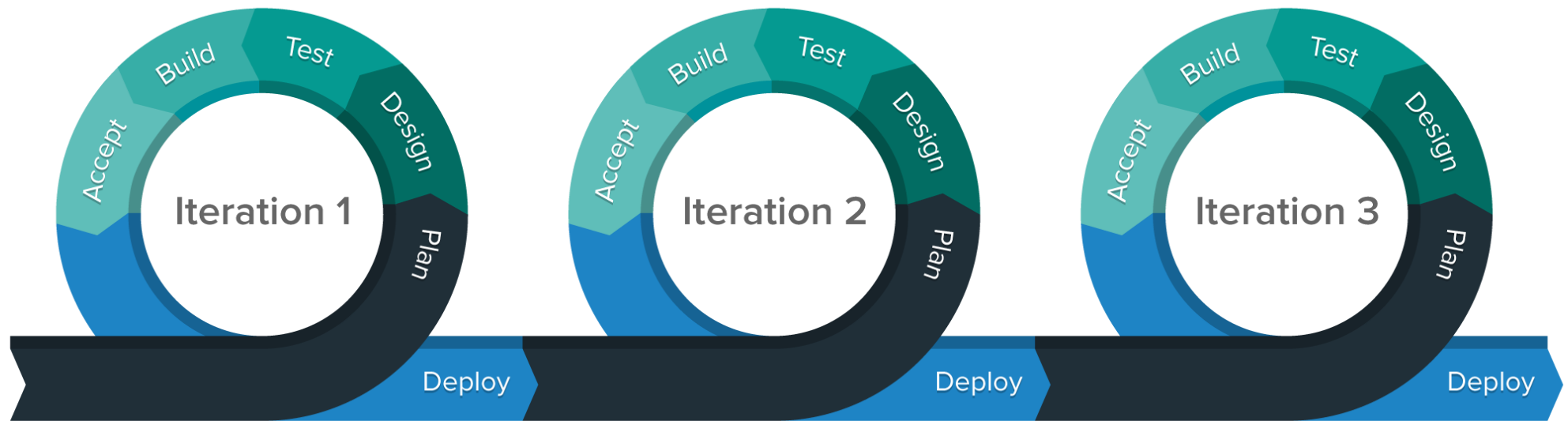


# Advantages and Disadvantages

- Advantages of the Waterfall model
  - Simple and easy to understand and use
  - Specific deliverable and review process
  - Phases do not overlap
  - Good for small projects if all requirements are clearly defined
  - Allows for departmentalization & managerial control
- Disadvantages of the Waterfall model
  - Time to market is high
  - Unexpected results
  - Lack of visibility of the current progress
  - Not suitable when the requirements keep changing
  - Difficult to make changes to the product when it is in the testing phase
  - The end product is available only at the end of the cycle
  - Not suitable for large and complex projects

# What is Agile?

- Agile is a philosophy, i.e., a set of values and principles to make a decision for developing software.
- Agile is based on the iterative-incremental model.
- In an incremental model, we create the system in increments, where each increment is developed and tested individually.



# Agile Process

- In Agile, a company releases the application with some high priority features in the first iteration.
- After its release, the end-users or the customers give you feedback about the performance of the application.
- Then you make the necessary changes into the application along with some new features and the application is again released which is the second iteration.
- You repeat this entire procedure until you achieve the desired software quality.

# Advantages of Agile Model

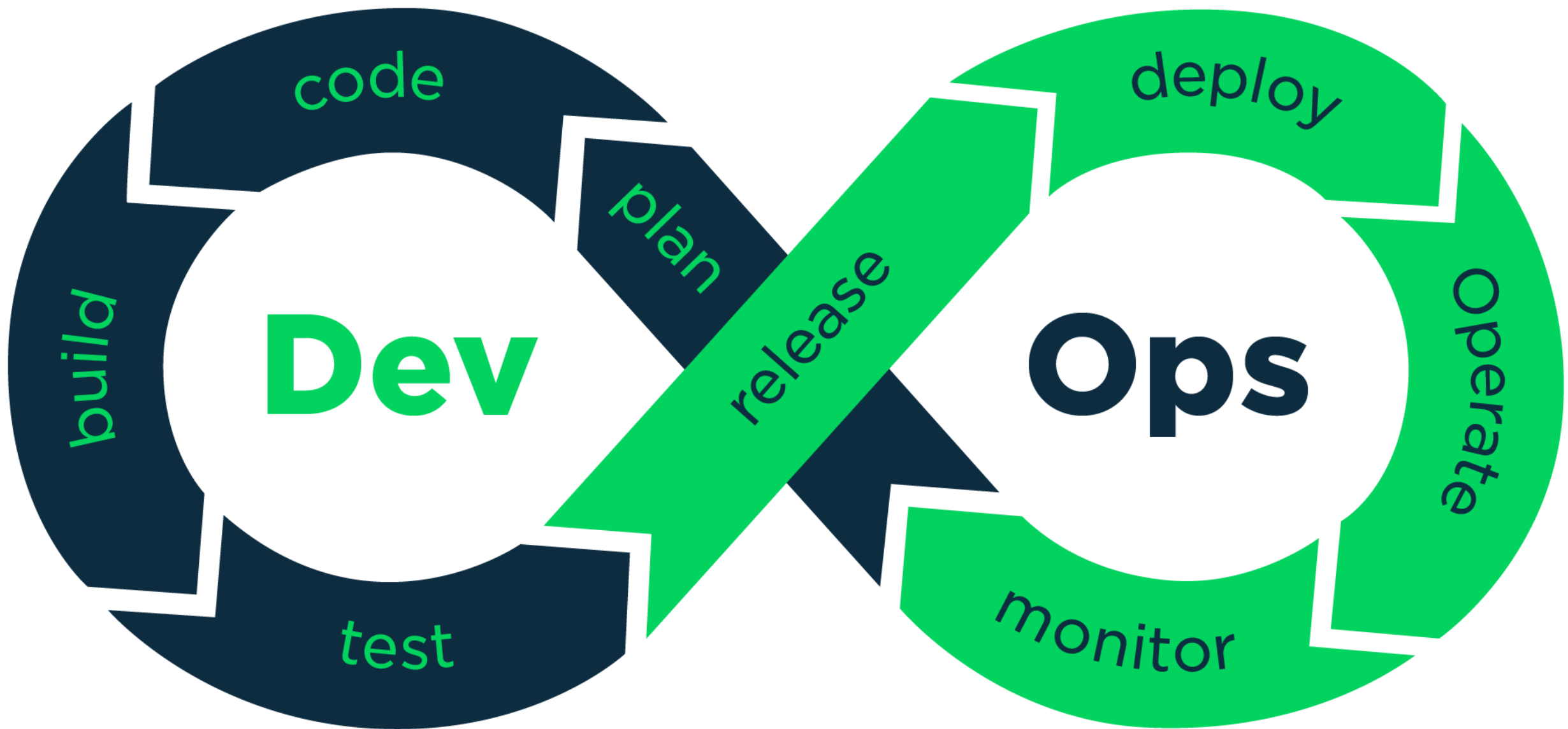
- It adaptively responds to requirement changes
- Fixing errors early in the development process and makes this process more cost-effective
- Improves the quality of the product and makes it highly error-free
- Allows for direct communication between people involved in software project
- Highly suitable for large & long-term projects
- Minimum resource requirements & very easy to manage

# Disadvantages of Agile Model

- Highly dependent on clear customer requirements
- Difficult to predict time and effort for larger projects
- Not suitable for complex projects
- Lacks documentation efficiency
- Increased maintainability risks

# DevOps Life cycle

- continuous development
- continuous integration
- continuous testing
- continuous deployment
- continuous monitoring

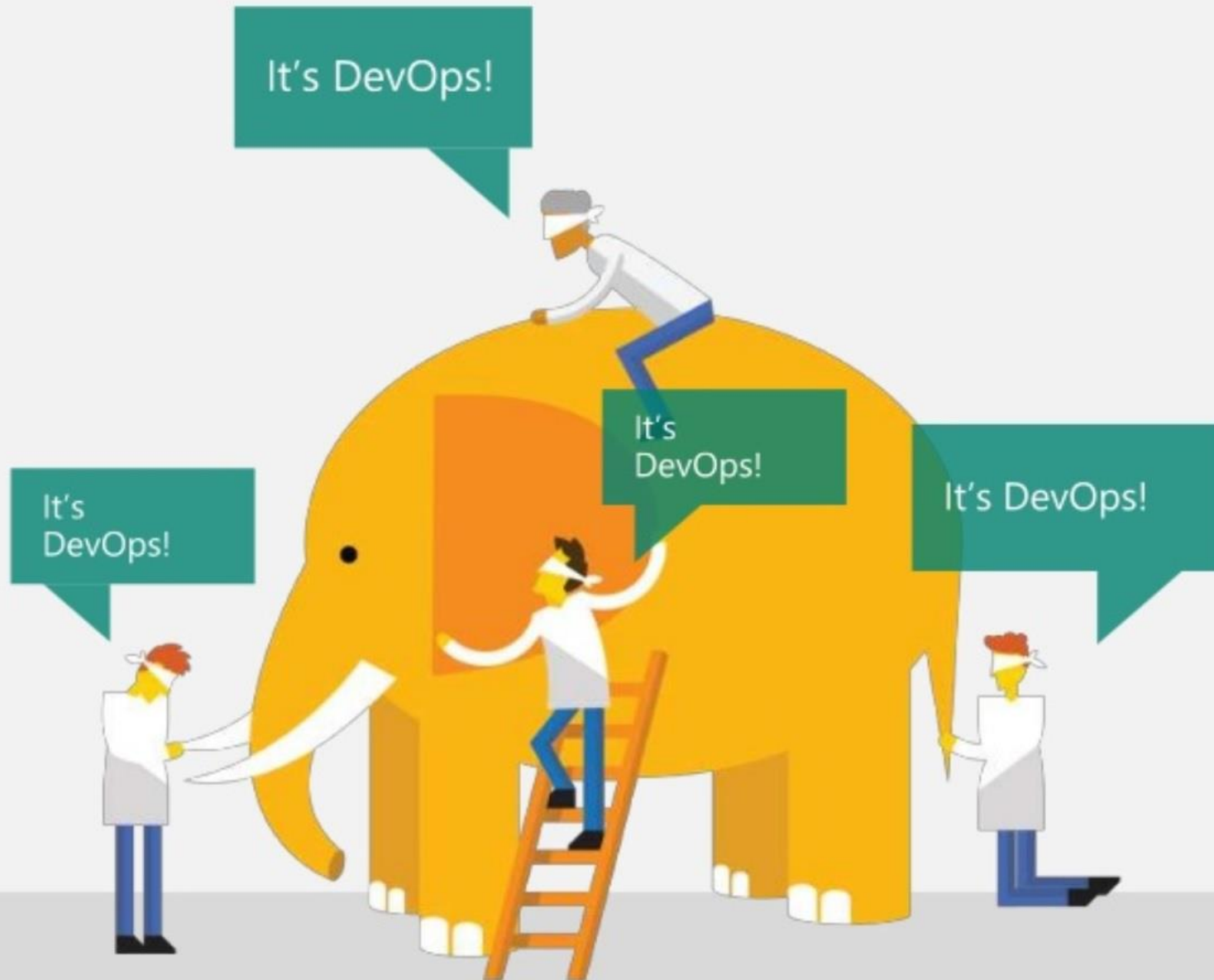




"DevOps is  
development  
and operations  
**collaboration**"

"DevOps  
is using  
**automation**"

"DevOps  
is **small**  
deployments"



"DevOps is  
treating your  
**infrastructure**  
as code"

"DevOps  
is feature  
**switches**"

"**Kanban**  
for Ops?"

# Continuous Development

- 'Planning' and 'Coding' Stage.
- The code can be in any language, but you maintain it by using Version Control tools.
- This process of maintaining the code is known as Source Code Management.
- After the code is developed, then you move to the Continuous Integration phase.
- **Git, SVN, CVS**

# Continuous Integration

- This stage is the core of the entire DevOps life cycle.
- Require to commit changes to the source code more frequently.
- This may be either on a daily or weekly basis.
- You then build every commit and this allows early detection of problems if they are present.
- Building code not only involves compilation but it also includes code review, unit testing, integration testing, and packaging.
- **Jenkins, GitLab**

# Continuous Testing

- Test the developed software continuously for bugs using automation testing tools.
- These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there are no flaws in the functionality.
- In this phase, you can use Docker Containers for simulating the test environment.
- **Jenkins, Selenium TestNG, JUnit**

# Continuous Deployment

- Configuration Management – Chef, Puppet, Ansible
- Containerization – Docker, Vagrant

# Continuous Monitoring

- Tools Used: Splunk, ELK Stack, Nagios, New Relic

## Collaborate

### Application Lifecycle Mgmt.



### Communication & ChatOps



### Knowledge Sharing



## Build

### SCM/VCS



### CI



### Build



### Database Management



## Test

### Testing



## Deploy

### Deployment



### Config Mgmt. / Provisioning



### Artefact Management



## Run

### Cloud / IaaS / PaaS



### Orchestration & Scheduling



### BI / Monitoring / Logging

