Ostad

DevOps in Software Development Lifecycle

Traditional SDLC vs. Agile vs. DevOps. The role of CI and CD. Collaboration and Communication.

Understanding the SDLC

Software Development Lifecycle definition: planning, developing, testing, deploying, and maintaining software.

Common models: Waterfall, Agile, DevOps

1. **Planning**
2. **Developing**
3. **Testing**
4. **Deployment**
5. **Maintenance**

Traditional SDLC: Waterfall

Sequential, step-by-step approach with high documentation and low flexibility.

Challenges: Slow delivery, late testing, poor collaboration

Sequential Steps

Each phase must be completed before moving on.

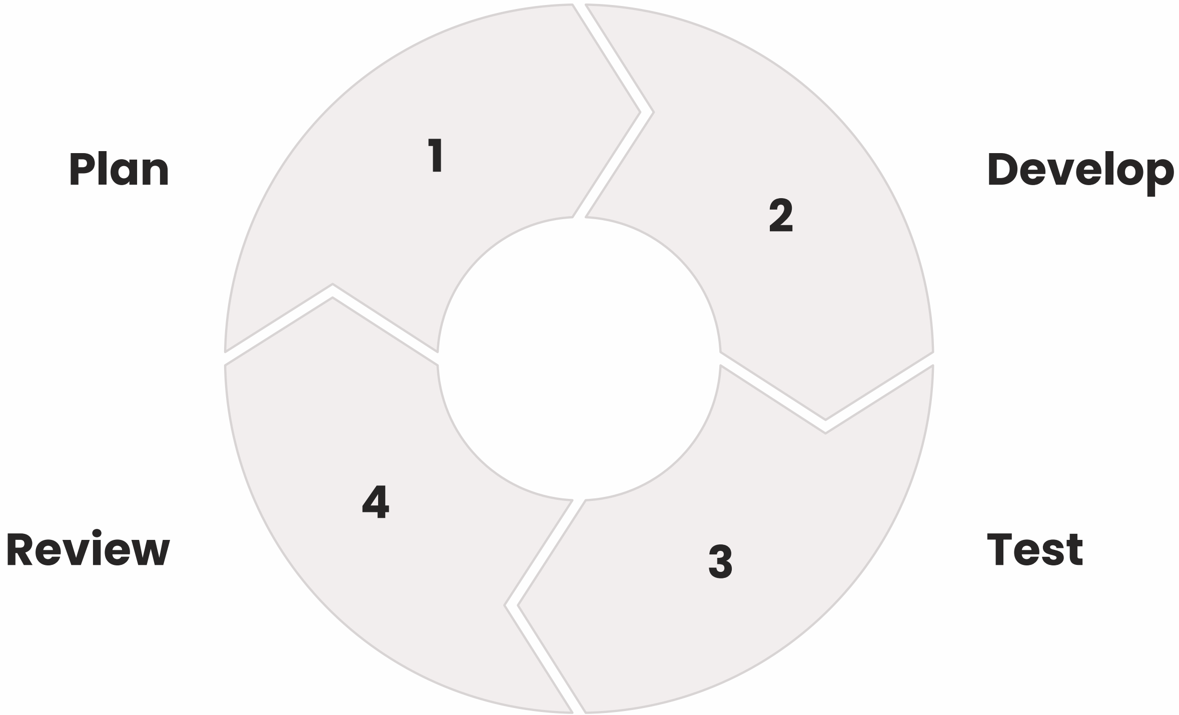
Documentation Heavy

Requires extensive documentation at each stage.

Agile SDLC: Iterative Progress

Iterative & incremental development with frequent releases & feedback. Cross-functional collaboration.

Faster delivery and early problem detection.



DevOps in SDLC

Combines Agile development with IT operations.

Features: Continuous development & delivery. Automation-driven testing & deployment.

1 **Continuous**

Development and delivery

2 **Automation**

Testing and deployment

3 **Collaboration**

Between all teams

SDLC Models Compared

A side-by-side comparison.

Feature	Traditional SDLC	Agile	DevOps
Approach	Sequential	Iterative	Continuous
Speed	Slow	Faster	Fastest
Collaboration	Low	Moderate	High
Automation	Minimal	Some	Extensive
Feedback	Late	Frequent	Continuous

CI: Continuous Integration

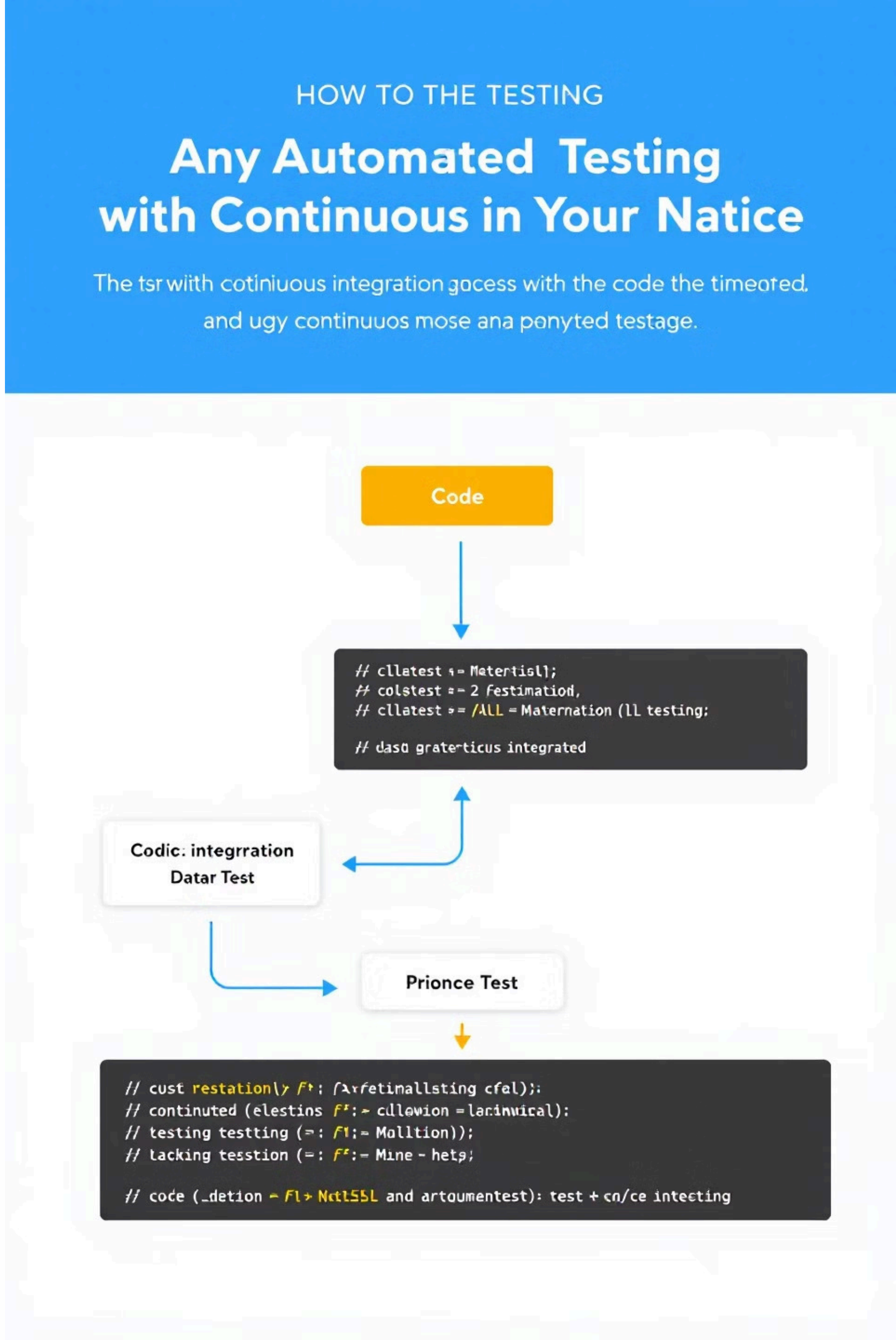
Merging & testing code frequently to detect issues early.

Reduces integration issues and speeds up development.

Early Bug Detection

Reduce Integration Issues

Speeds Development



HOW TO THE TESTING

Any Automated Testing with Continuous in Your Natice

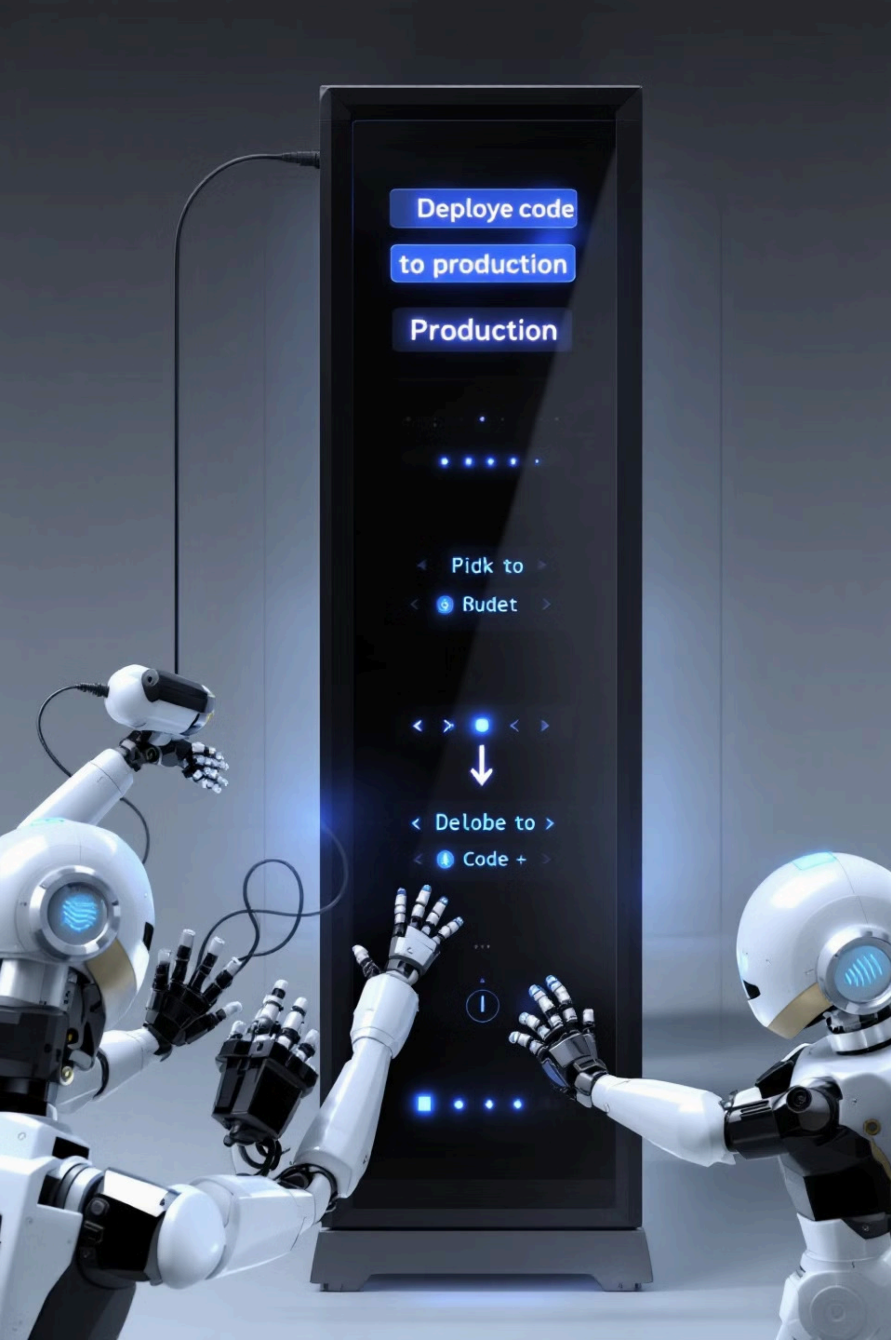
The tsr with cotiniuous integration gacess with the code the timeoted. and ugy continuuos mose ana ponyted testage.

```
// cllatest ፥= Metertisl];
// colstest ፧= 2 festimatiod,
// cllatest ፧= /ALL = Maternation (lL testing;

// dasg grate-ticus integrated
```

```
// cust restation\ሃ f፥: (Axfetimallsting cfel);
// continuted (elestins f፥:- cdllowion =larinwical):
// testing testting (=: f1:= Molltion));
// lacking tesstion (=: f፥:- Mine - hetg;

// code (_detion - Fl→ NctL55L and artoumentest): test + cn/ce intecting
```



CD: Continuous Deployment/Delivery

Automatically deploying tested code to production.

Ensures smooth updates without downtime and reduces errors.

1 **Code Commit**

2 **Automated Testing**

3 **Staging Deployment**

4 **Production Deployment**

Collaboration in DevOps

Reduces friction, increases software quality, and enables faster issue resolution.

Use communication tools and encourage a shared responsibility mindset.

Reduce Friction

Between Dev & Ops

Increase Quality

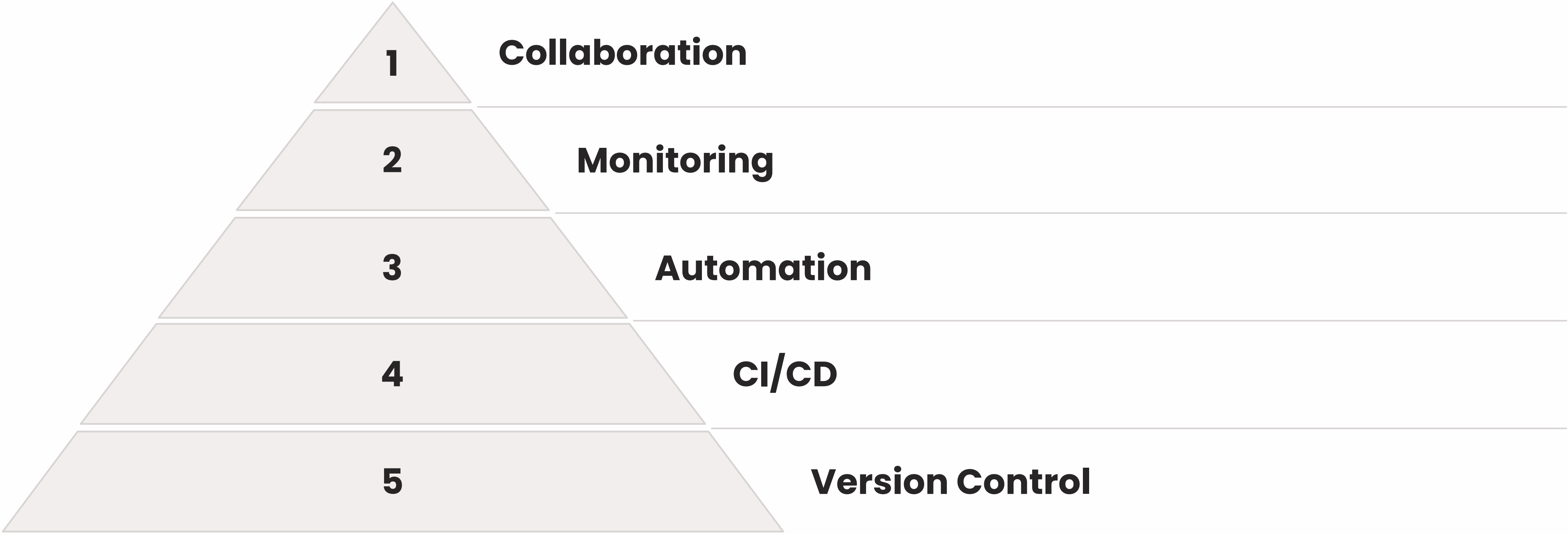
Faster issue resolution

Faster

Issue resolution

DevOps Toolchain

Tools for version control, CI/CD, infrastructure automation, monitoring & logging, and collaboration.



1 Collaboration

2 Monitoring

3 Automation

4 CI/CD

5 Version Control

Key Takeaways

DevOps improves SDLC, CI/CD ensures automation, and collaboration is key.

1 **DevOps**

Improves SDLC

2 **CI/CD**

Ensures automation

3 **Collaboration**

Key to DevOps success