COMP 4350 Software Engineering II

Lecture 3 Study Notes

Dr. Shaowei Wang

Version Control Essentials

- **Version Control**: Manages changes to source code over time.
- **Code**: The written instructions developers create to develop software.
- **Dependency**: Code or software that a project requires to execute properly.
- **Configuration**: The arrangement of each of the functional and physical characteristics of equipment, or a group of items arranged for a particular purpose.
- **Environment**: The computing setup where software runs; it can be local, staging, or production.
- **Code Style**: Also known as coding standards; rules and guidelines used to set the style of the written code.
- **Convention**: Commonly agreed-upon guidelines for writing software, including version control best practices.

DevOps Overview

What is DevOps?

- **DevOps**: A set of practices that combines software development (Dev) and IT operations (Ops).
- **Goal**: To shorten the systems development life cycle and provide continuous delivery with high software quality.

Why DevOps?

- Removal of the gap between operation and development teams.
- Faster delivery of features and quicker resolution of problems.
- Improved collaboration and communication both internally and externally.
- Spend more effort on developing new features (e.g., IBM's increase from 58% to 80% in innovation focus).

DevOps Tools

- **CI/CD Tools**: Jenkins, Puppet, Docker.
- **Planning Tools**: Jira, Azure DevOps, Asana.
- **Source Code Management**: Git.
- **Code Review Tools**: Gerrit, Review Board.
- **Build Tools**: Make, Ant, Maven, Gradle.
- **Monitoring Tools**: Nagios, Splunk, New Relic.

DevOps Stages

1. **Plan**

2. **Code**
3. **Build**
4. **Test**
5. **Release**
6. **Deploy**
7. **Operate**
8. **Monitor**

DevOps Practices

- **Continuous Integration (CI)**: Frequent merging of code changes into a main branch.
- **Continuous Delivery (CD)**: Automated delivery of code changes to an environment after the build stage.
- **Continuous Deployment**: Similar to CD, but automatically deploys code changes to production.
- **Automate Everything**: Using "infrastructure as Code" for automated creation of environments.
- **Monitoring**: Real-time tracking of application or infrastructure issues.
- **Blue/Green Deployment**: Reduces downtime risk by having duplicate environments.

Cloud Services

- **Cloud Providers**: AWS, Google Cloud, Microsoft Azure.
- **Service Models**:
  - IaaS (Infrastructure as a Service)
  - PaaS (Platform as a Service)
  - SaaS (Software as a Service)
- **Benefits**:
  - Scalability, cost-effectiveness, immediate availability.
  - Improved performance and security.

Agile vs Waterfall

Process 1: Waterfall

- Sequential development process, phased approach from requirements definition to maintenance.

Limitations of Waterfall

- Rigid structure, inflexibility regarding changes.

Agile Development

- Iterative development process, emphasizing customer feedback and continuous improvement.

Practical Exercises

- Set up CI with TravisCI using a forked GitHub repository.
- Understand how to configure `.travis.yml` for Java-based projects.
- Use TravisCI to automatically build and test the project.
- Explore the differences between Maven and Ant build systems.

References

- *Effective DevOps*, Jennifer Davis & Katherine Daniels, Chapters 2, 4, & 6.
- Online resources for understanding IaaS vs PaaS vs SaaS.
- Documentation on continuous integration, delivery, and deployment.

DevOps Consequences

- Small, autonomous teams for efficient development.
- Adoption of microservices architecture.
- Reduced overhead through reliance on service interfaces for team interactions.

Class Notes

Slide: 1

Dr. Shaowei Wang
Comp 4350 Software Engineering II Lecture 3

Slide: 2

Title: TA available time

Slide: 3

Title: Review

- Version control
- Code Dependency
- Configuration
- Environment
- Code style
- Convention

Slide: 4

Title: DevOps overview

Slide: 5

Title: Agenda

- Why DevOps?
- What is DevOps?
- DevOps Stage
- DevOps tools

Slide: 6

Title: Software Development

- Design and specification
- Testing
- Coding
- Building

Slide: 7

Title: Software Operation

- Deployment
- Operation

Slide: 8

Title: Software Operations

- Definition: The people and management processes associated with IT service management to deliver the right set of services at the right quality and at competitive costs for customers
- Duties:
  - Manage hardware
  - Deploy into production environment
  - Monitor the execution of production (e.g., error, performance)

Slide: 9

Operators

Slide: 10

Title: There is a gap between software developers and operators

- Developers
- Operators
- Does my system perform well in the field?
- What does this error message mean? It doesn’t work in the production.
- How do I resolve it?

Slide: 11

Title: What problem is DevOps trying to solve?

- Poor communication between Dev and Ops
- Developers have more insights on operations
- Operators have more insights on the developers
- Slow product delivery
- Leads to slower release schedules

Slide: 12

Title: DevOps

- Remove the gap between Dev team and Ops team
- Developers
- Operators

Slide: 13

What is DevOps?

- Goal of DevOps:
  - Remove the gap between operation and development
  - Reduce the time between committing a change to a system and the change being deployed into production environment, while ensuring high quality.

Slide: 14

Title: What does high quality mean?

- High quality code
- Well tested
- High quality builds and delivery mechanism
- Automation & more testing
- No interruption of the existing system

Slide: 15

Title: Why companies care?

- IBM: Spending more efforts on developing new features
- Paddy Power (Ireland): Decreased cycle time from conception to production

Slide: 16

Nowadays, a profound shift is taking place and make DevOps happen

Slide: 17

Title: DevOps stages

Slide: 18

Title: Plan

- Requirements and feedback gathering
- Product roadmap creation
- Tools: Jira, Azure DevOps or Asana

Slide: 19

Title: Coding

- Standard toolkit with plugins for consistent code-styling and security

Slide: 20

Title: Build

- Automated building and code review

Slide: 21

- Build systems describe how sources are translated into deliverables

Slide: 22

- Build systems describe how sources are translated into deliverables

Slide: 23

Build systems describe how sources are translated into deliverables

Slide: 24

Title: Build tools

- Make
- Ant
- Maven
- Gradle

Slide: 25

Title: Code review

Slide: 26

Title: Testing

- Manual test
- Automated tests
- Different levels of functional testing

Slide: 27

Title: How deep the test should be?

Slide: 28

Title: Where is the first place DevOps kicks in?

Slide: 29

Title: Continuous integration

Slide: 30

https://itnext.io/git-concepts-for-newcomers-part-2-git-repository-working-tree-and-staging-area-a2e720bf3528

Slide: 31

Title: Continuous integration

Slide: 32

Title: Release

Slide: 33

Title: Deploy

Slide: 34

Title: Cloud service

- Organizations moving deployment to cloud service
- Cloud service providers

Slide: 35

Slide: 36

Title: Benefit of cloud service

Slide: 37

Title: Benefit of cloud service

- Scalability
- Cost-effectiveness

Slide: 38

Title: Benefit of cloud service

- Performance
- Security

Slide: 39

https://www.t4.ai/industry/cloud-computing-market-share

Slide: 40

Title: Continuous Delivery

- Automation to deploy code changes to environments

Slide: 41

Title: Continuous Deployment

- All changes passing verification steps are released to production

Slide: 42

Slide: 43

Title: CD using docker

Slide: 44

Title: Continuous Delivery Pipeline

Slide: 45

Title: Continuous delivery pipeline

Slide: 46

Title: Development strategy - blue/green deployment

- A deployment strategy to reduce downtime and risk

Slide: 47

Slide: 48

Slide: 49

Title: Operate

- Environment automatically scales with load
- Collecting and triaging feedback

Slide: 50

Title: Operate

Slide: 51

Title: Monitor

- Process to monitor and identify compliance issues and security risks

Slide: 52

Title: What type of data we can monitor?

Slide: 53

Title: Monitoring data type

Slide: 54

Slide: 55

Slide: 56

Title: DevOps Practices

- Automate Everything
- Develop infrastructure code with the same practices as application code

Slide: 57

Slide: 58

Title: Reference

- Effective DevOps, Jennifer Davis & Katherine Daniels
- Other online resources

Slide: 59

Title: DevOps Consequences

- Keep teams relatively small
- Amazon's “two pizza rule”

Slide: 60

Title: DevOps Consequences

- Team size drives overall architecture
- Small teams develop small services

Slide: 61

Title: Try it out using TravisCI

Slide: 62

Title: We need automation and tool supports!

Slide: 63

Title: Try it out using TravisCI

Slide: 64

Title: Process 1: Waterfall

- Steps in the waterfall model

Slide: 65

Title: Limitations of waterfall

Slide: 66

Title: Agile development

- Focus on customer feedback

Slide: 67

Slide: 68

https://itnext.io/git-concepts-for-newcomers-part-2-git-repository-working-tree-and-staging-area-a2e720bf3528