Software Engineering

3
Software Processes (3)
DevOps

School of Computer Science
Prof. Euijong Lee

Chapter2-3. DevOps

Topics covered

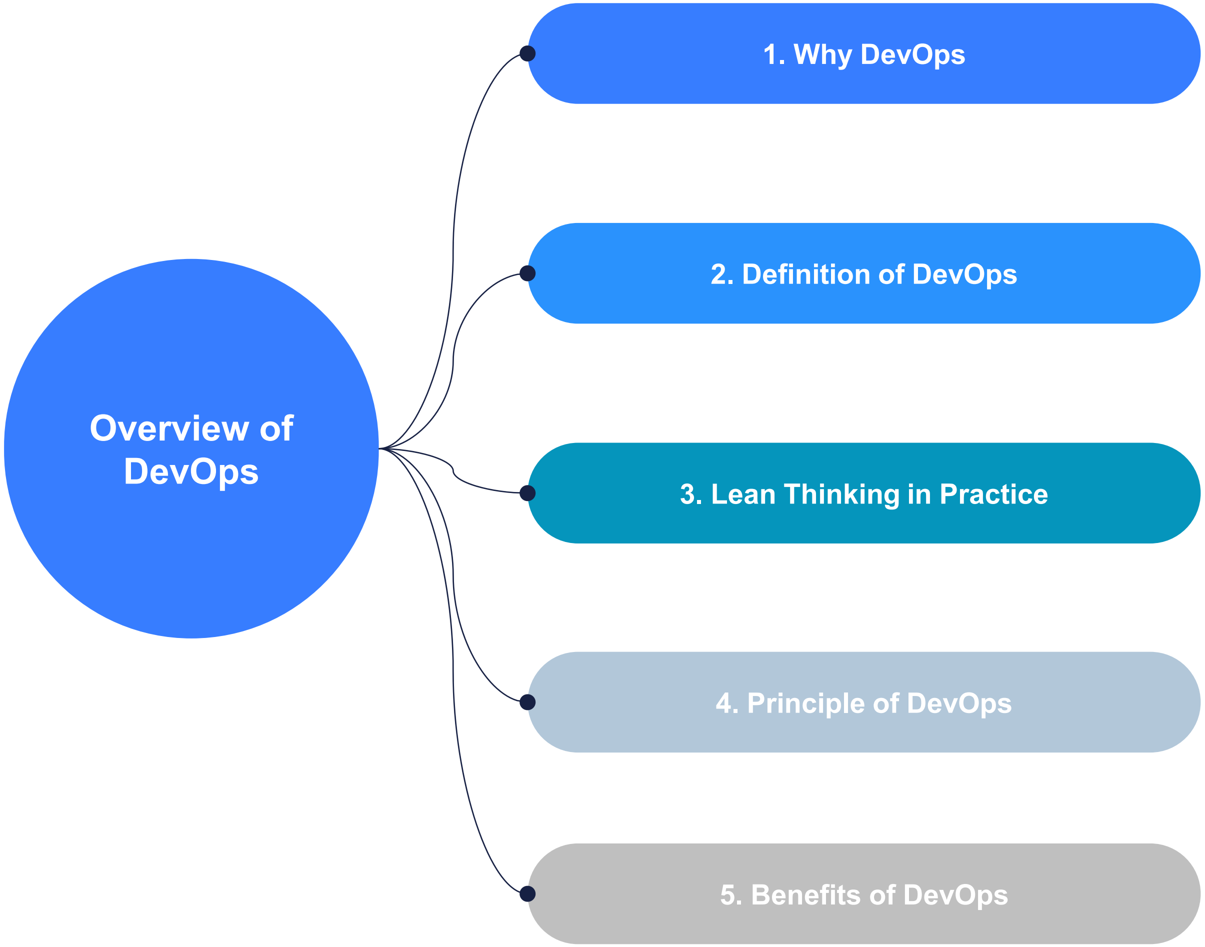
01 ▌ Overview of DevOps

02 ▌ DevOps Processes

03 ▌ DevOps in Practices

Chapter2-3. DevOps

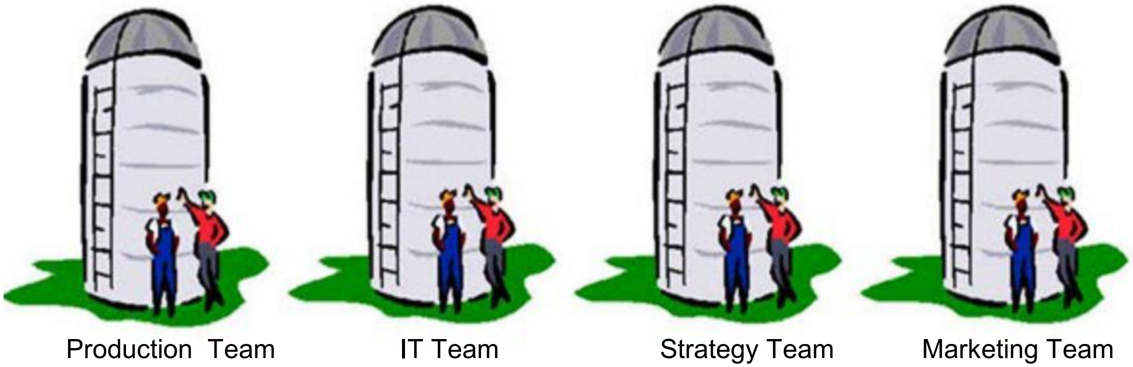
3-1: Overview of DevOps



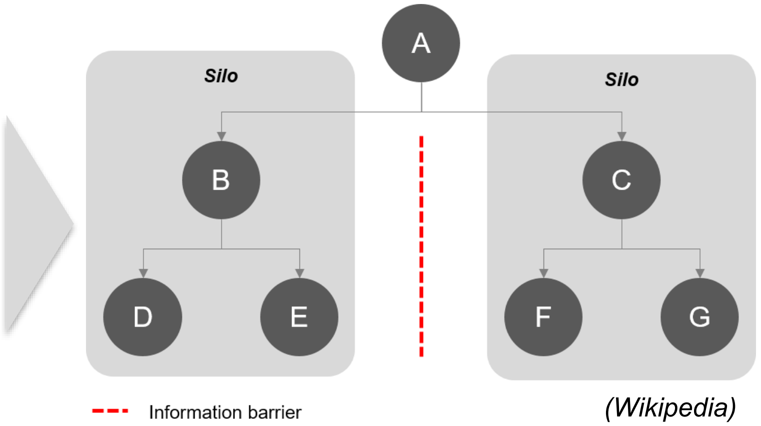
Overview of DevOps

Why DevOps?

- **Silos based development process**
  - After each stage of software development is completed, proceed to the next
  - Absence of interaction between each step



(syfte blog,2016)

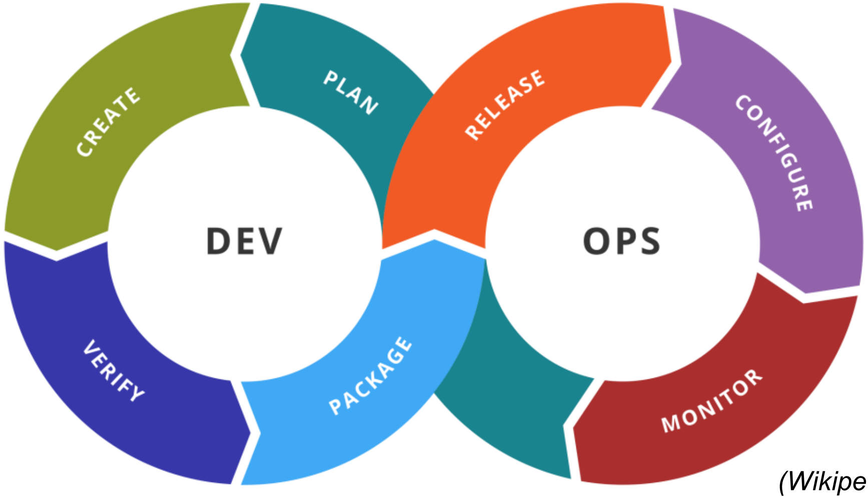


(Wikipedia)

Overview of DevOps

Why DevOps?

- **DevOps?**
  - **DevOps** is a software development methodology using iterative mechanism
  - Interconnect Development(Dev), Operations(Ops), and user feedbacks
  - Based on Agile and Lean methods
  - To solve insufficient communication between development team, development delay, and other problems in projects during operation



(Wikipedia)

Overview of DevOps

Definition of DevOps

Wikipedia

- **DevOps** is a methodology in the **software development** and IT industry
- Used as a set of practices and tools, DevOps **integrates and automates** the work of **software development (Dev)** and IT **operations (Ops)** as a means for **improving and shortening** the systems development life cycle
- **Lean** and **Agile** based software development methodology

IBM

- **DevOps** **speeds delivery of higher quality software** by combining and automating the work of **software development and IT operations teams**

Overview of DevOps

Definition of DevOps

O’Reilly

- **DevOps** is a **cultural movement that changes** how individuals think about their work, values the diversity of work done, supports intentional processes that **accelerate the rate** by which businesses realize value, and measures the effect of social and technical change.

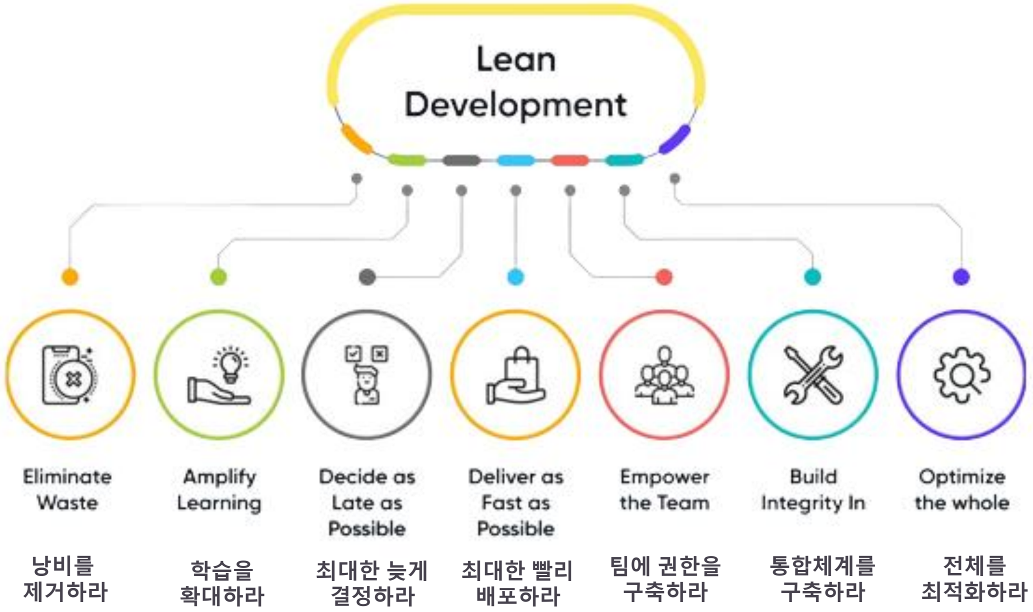
IEEE std.

- **Set of principles and practices** which enable **better communication and collaboration** between relevant stakeholders for the purpose of specifying, developing, and operating software and systems products and services, and continuous improvements in all aspects of the life cycle.

Overview of DevOps

Lean Thinking

- **Lean Software Development (LSD)**
  - **“Think Big, Act Small, Fail Fast, Learn Rapidly”**
  - Lean development is a methodology that was first applied to manufacturing, but is now being applied to management, UX, startups, and to software development
  - Lean software development is the process of optimizing how value is created by reducing waste



(https://www.netsolutions.com/insights/7-principles-of-lean-software-development/)

Overview of DevOps

Principles of DevOps

Collaboration (협업)

- The key premise behind DevOps is collaboration
- Development and operations teams coalesce into a functional team that communicates, shares feedback, and collaborates throughout the entire development and deployment cycle
- Often, development and operations teams merge into a single team that works across the entire application lifecycle.

Automation (자동화)

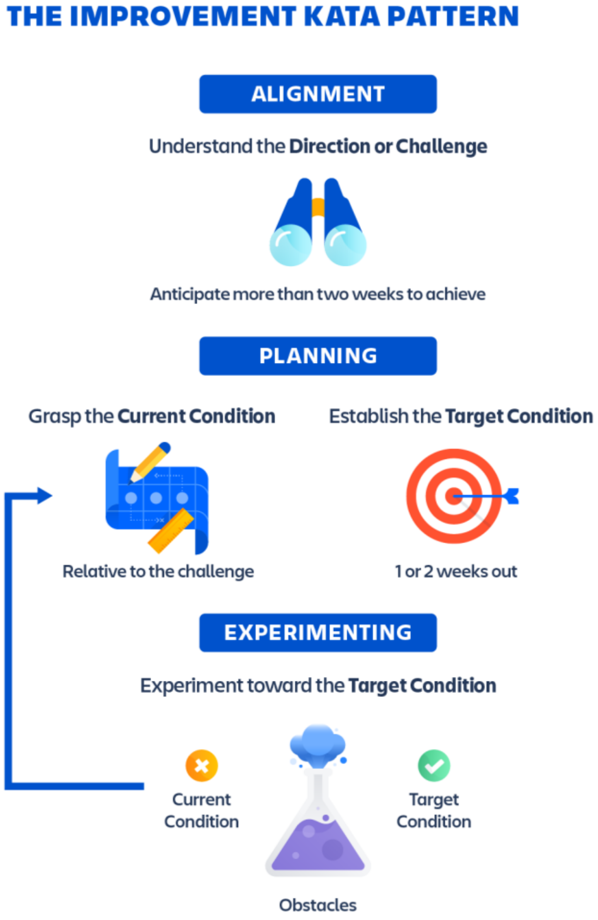
- Automation gives developers more time to write code and develop new features
- Automation is a key element of a CI/CD (Continuous Integration / Continuous Delivery) pipeline and helps to reduce human errors and increase team productivity
- Teams achieve continuous improvement with short iteration times, which allows them to quickly respond to customer feedback.

https://www.atlassian.com/devops/what-is-devops

Principles of DevOps

Continuous Improvement (지속적 개선)
Continuous integration and delivery (지속적 통합/배포)

- Continuous improvement was established as a staple of agile practices, as well as lean manufacturing and Improvement Kata.
  - Kata? *Improvement Kata is a method where team leaders and members continually practice a kata routine that develops and channels their abilities to solve problems.*
- It's the practice of focusing on experimentation, minimizing waste, and optimizing for speed, cost, and ease of delivery.
- Continuous improvement is also tied to continuous delivery, allowing DevOps teams to continuously push updates that improve the efficiency of software systems.
- The constant pipeline of new releases means teams consistently push code changes that eliminate waste, improve development efficiency, and bring more customer value.


THE IMPROVEMENT KATA PATTERN
ALIGNMENT
Understand the Direction or Challenge
Anticipate more than two weeks to achieve
PLANNING
Grasp the Current Condition
Establish the Target Condition
Relative to the challenge
1 or 2 weeks out
EXPERIMENTING
Experiment toward the Target Condition
Current Condition
Target Condition
Obstacles

https://www.atlassian.com/devops/what-is-devops

Principles of DevOps

Customer-centric action (고객-중심 작업)

- DevOps teams use short feedback loops with customers and end users to develop products and services centered around user needs.
- DevOps practices enable rapid collection and response to user feedback through use of real-time live monitoring and rapid deployment.
- Teams get immediate visibility into how live users interact with a software system and use that insight to develop further improvements.

Create with end in mind (끝을 염두에 두고 만들기)

- This principle involves understanding the needs of customers and creating products or services that solve real problems.
- Teams shouldn’t ‘build in a bubble’, or create software based on assumptions about how consumers will use the software.
- Rather, DevOps teams should have a holistic understanding of the product, from creation to implementation.

https://www.atlassian.com/devops/what-is-devops

Overview of DevOps

Benefits of DevOps

Collaboration and trust (협업 및 신뢰)

- Building a culture of shared responsibility, transparency, and faster feedback is the foundation of every high-performing DevOps team.
- In fact, collaboration and problem-solving ranked as the most important elements of a successful DevOps culture, according to our 2020 DevOps Trends survey.

Release faster and work smarter (더 빠른 릴리스, 더 스마트한 업무 진행)

- Speed is everything. Teams that practice DevOps release deliverables more frequently, with higher quality and stability.
- In fact, the DORA "2019 State of DevOps" report found that elite teams deploy 208 times more frequently and 106 times faster than low-performing teams. (https://cloud.google.com/blog/products/devops-sre/the-2019-accelerate-state-of-devops-elite-performance-productivity-and-scaling?hl=en)

Overview of DevOps

Benefits of DevOps

Accelerate time-to-resolution (문제 해결 시간 단축)

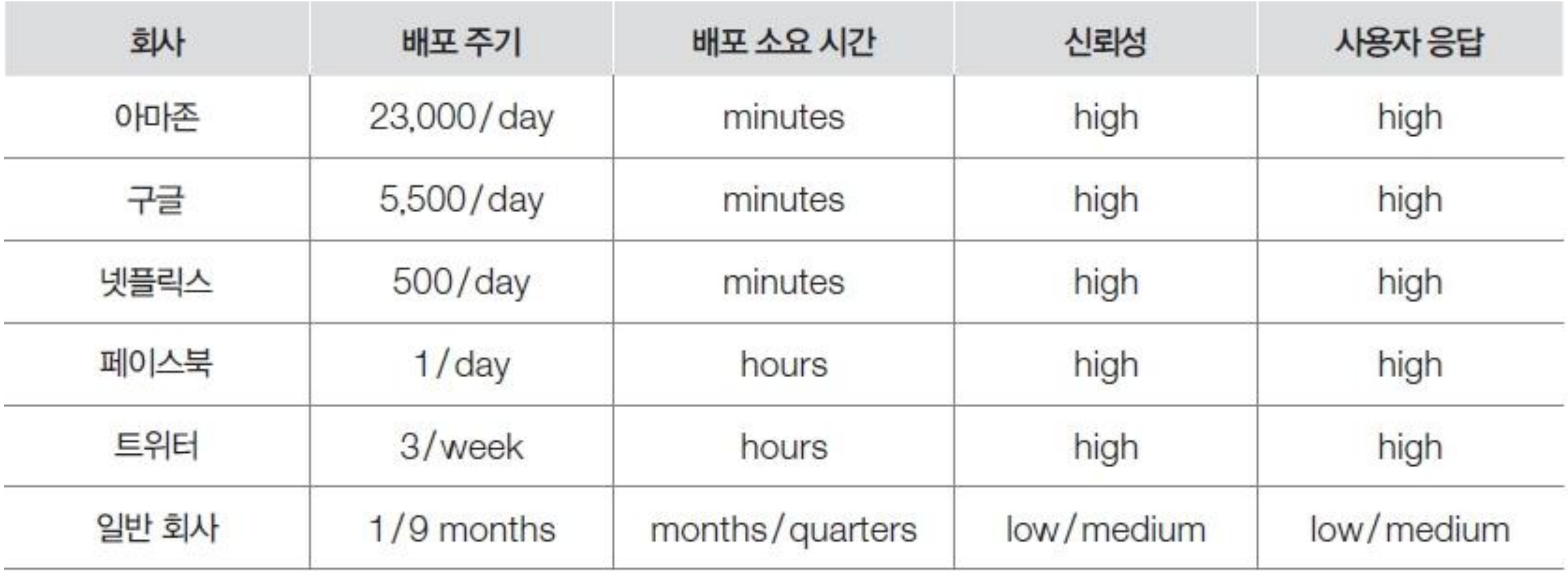
- The team with the fastest feedback loop is the team that thrives.
- Full transparency and seamless communication enable DevOps teams to minimize downtime and resolve issues faster.
  - If critical issues aren't resolved quickly, customer satisfaction tanks.

Better manage unplanned work (계획되지 않은 업무를 쉽게 관리)

- Teams who fully embrace DevOps practices work smarter and faster, and deliver better quality to their customers.
- The increased use of automation and cross-functional collaboration reduces complexity and errors, which in turn improves the Mean Time to Recovery (MTTR) when incidents and outages occur.

DevOps Application Trend

- **DevOps Application Trend**
    - IT organizations that must provide customer service 24/7 have applied DevOps for software development and operation strategies
(e.g., Amazon, Google, and Netflix)



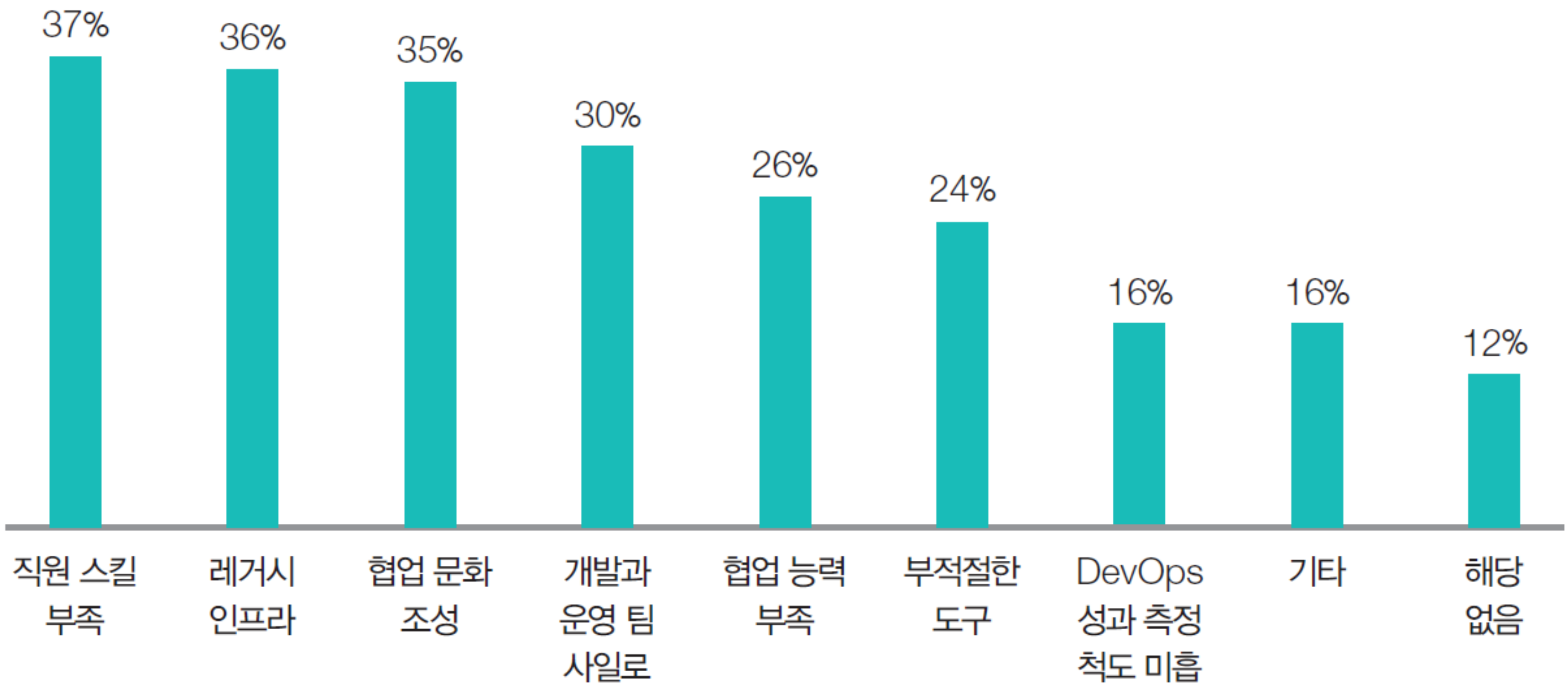
회사	배포 주기	배포 소요 시간	신뢰성	사용자 응답
아마존	23,000/day	minutes	high	high
구글	5,500/day	minutes	high	high
넷플릭스	500/day	minutes	high	high
페이스북	1/day	hours	high	high
트위터	3/week	hours	high	high
일반 회사	1/9 months	months/quarters	low/medium	low/medium

(“소프트웨어 공학 이론과 실제”, 홍장의, 한빛아카데미)

Overview of DevOps

DevOps Application Trend

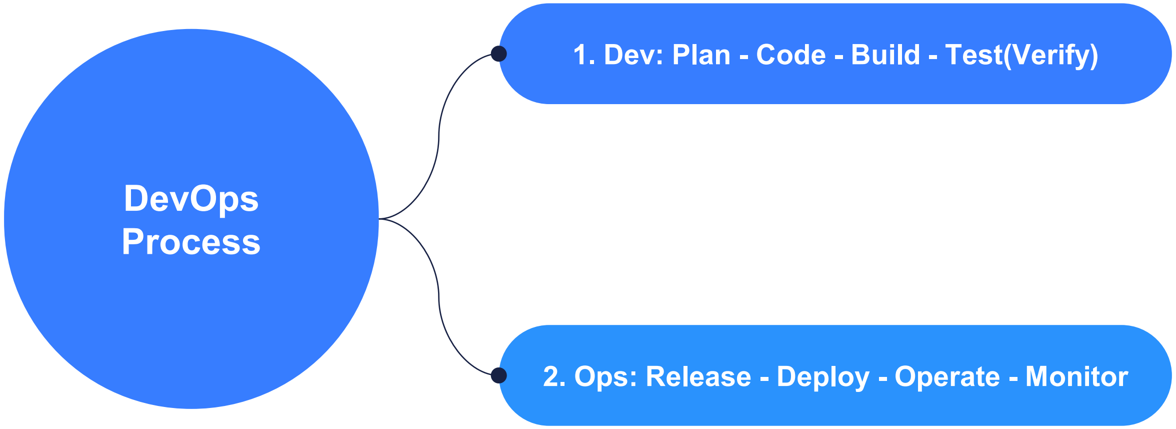
- **Obstructive factors to apply DevOps**



("소프트웨어 공학 이론과 실제", 홍장의, 한빛아카데미)

Chapter2-3. DevOps

3-2: DevOps Process

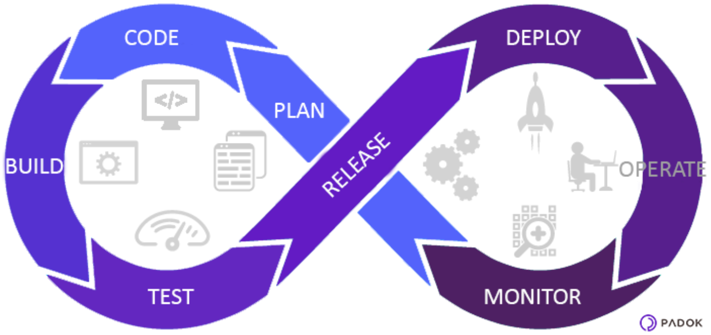


DevOps Process

1. Dev: Plan - Code - Build - Test(Verify)

2. Ops: Release - Deploy - Operate - Monitor

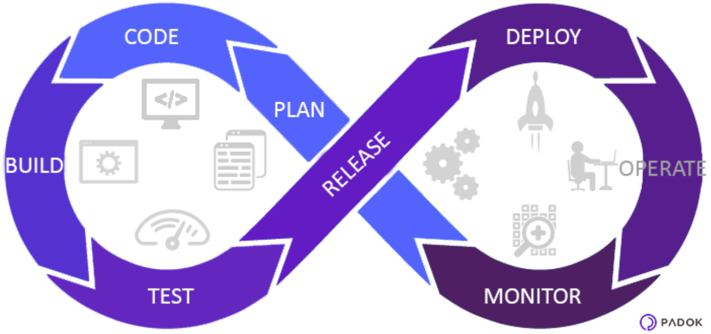
DevOps Process



- **Plan**
  - This is the part of the project where you are **organizing the tasks, and schedules** and setting up your project management tools.
  - The idea is to plan tasks using the **user story** process from the agile methodology.
  - A perfect user story as a what (who, where, trigger), a why, and acceptance criteria.
- **Code**
  - Doing **code development and code review.**
  - In DevOps practice, it is important to share a code tool between Ops and developers teams like Github or Gitlab.

(https://www.padok.fr/en/blog/devops-process)

DevOps Process

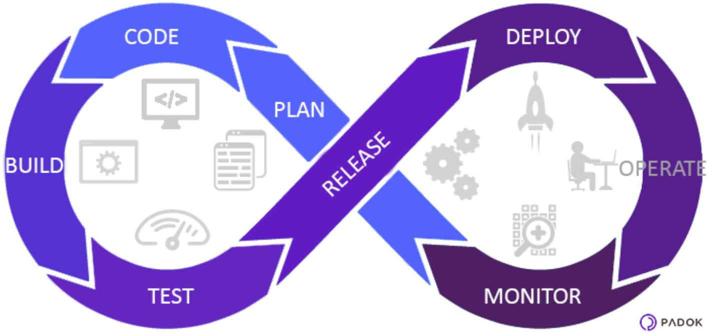


- **Build**
  - The goal here is to **build the source code** into one desired format, compiling, testing, and deploying in a particular place of the infrastructure.
  - Once this step sets up the **continuous integration (CI) and delivery (CD) tools** can check and verify the source code from Source Code Management and build it.

- **Test (Verify)**
  - The continuous testing process **reduces risks.**
  - **Automatic tests** ensure that no bugs will be implemented in production.
  - You have to implement testing tools in your workflow to ensure the best development quality for your software.

(https://www.padok.fr/en/blog/devops-process)

DevOps Process

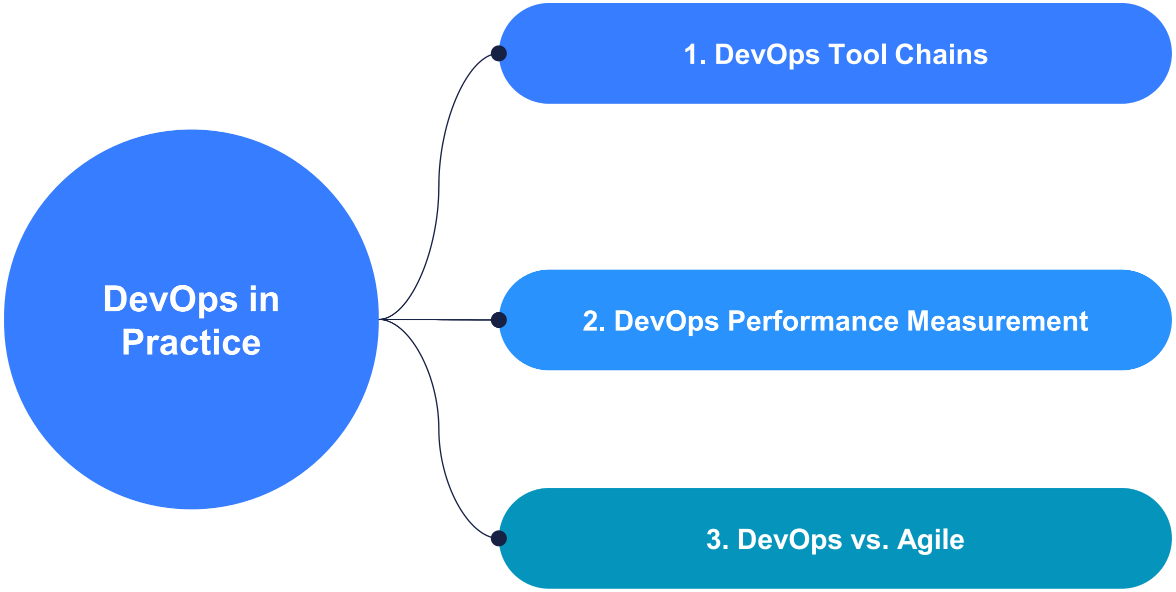


- **Release**
  - The code has passed the testing (continuous integration) process and is **ready to be deployed.**
- **Deploy**
  - The operational team is **deploying the new feature in production**.
  - But as automation is one of the DevOps principles, it is possible to set up continuous deployment.
- **Operate / configure infrastructure**
  - The Ops builder **maintain a scalable infrastructure**, infrastructure as code, and check security issues and log management.
- **Monitor**
  - Monitoring is an important step as it allows you to fix incidents faster and create a better experience for your end-user.

(https://www.padok.fr/en/blog/devops-process)

Chapter2-3. DevOps

3-3: DevOps in Practice



DevOps in Practice

1. DevOps Tool Chains

2. DevOps Performance Measurement

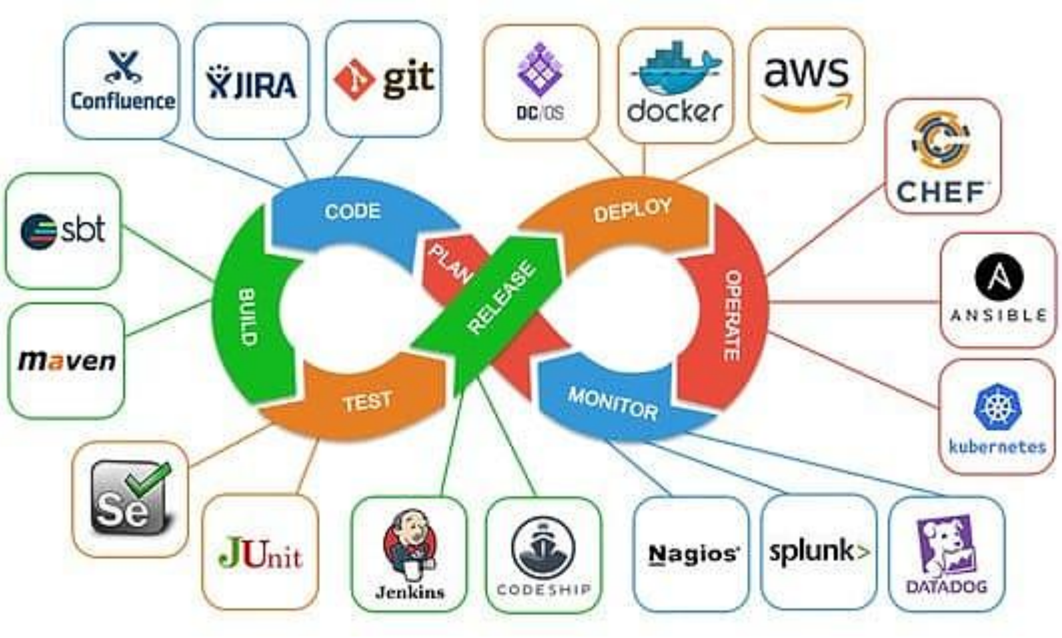
3. DevOps vs. Agile

DevOps in Practice

DevOps Tool Chains

▪ DevOps Tool chains?

- A DevOps toolchain is **a collection of tools**, often from a variety of vendors, that operate as an integrated unit to design, build, test, manage, measure, and operate software and systems.
- It enables development and operations teams to collaborate across the entire product lifecycle and tackles key DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration.

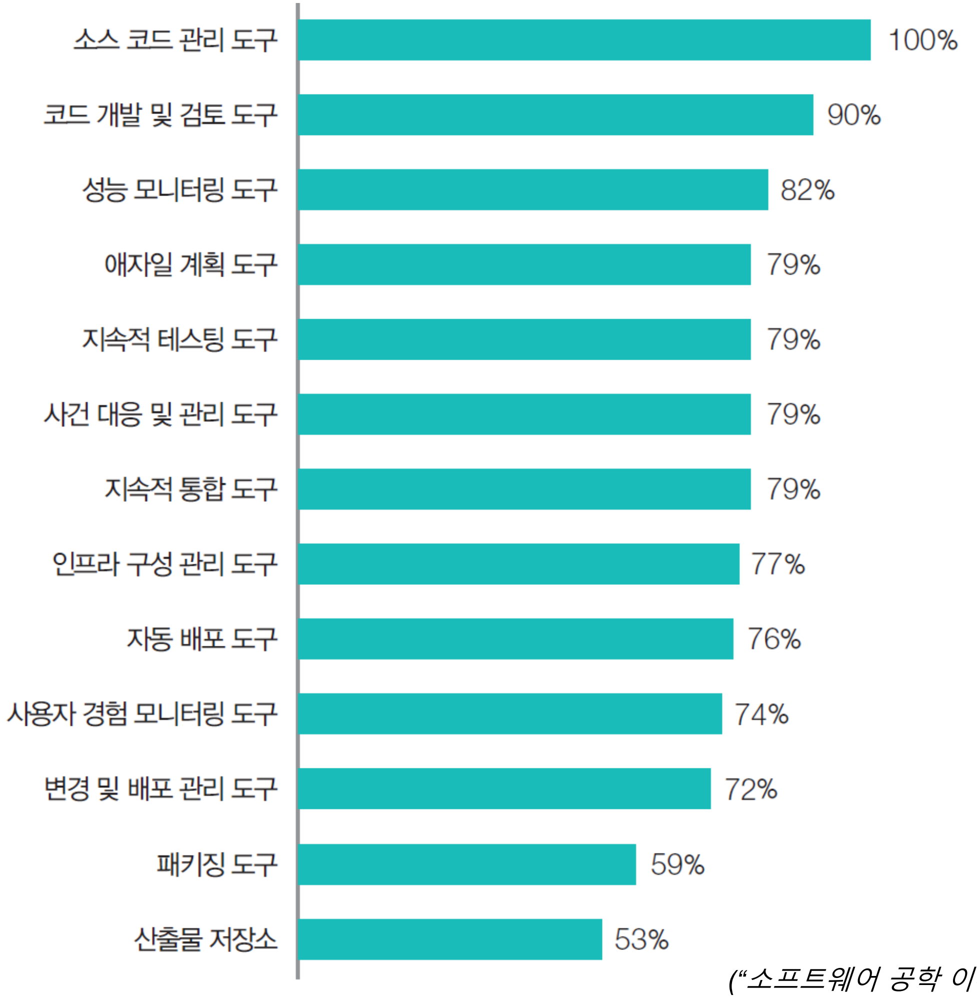


(https://www.padok.fr/en/blog/devops-process)

DevOps in Practice

DevOps Tool Chains

▪ DevOps Tool chains?



("소프트웨어 공학 이론과 실제", 홍장의, 한빛아카데미)

DevOps in Practice

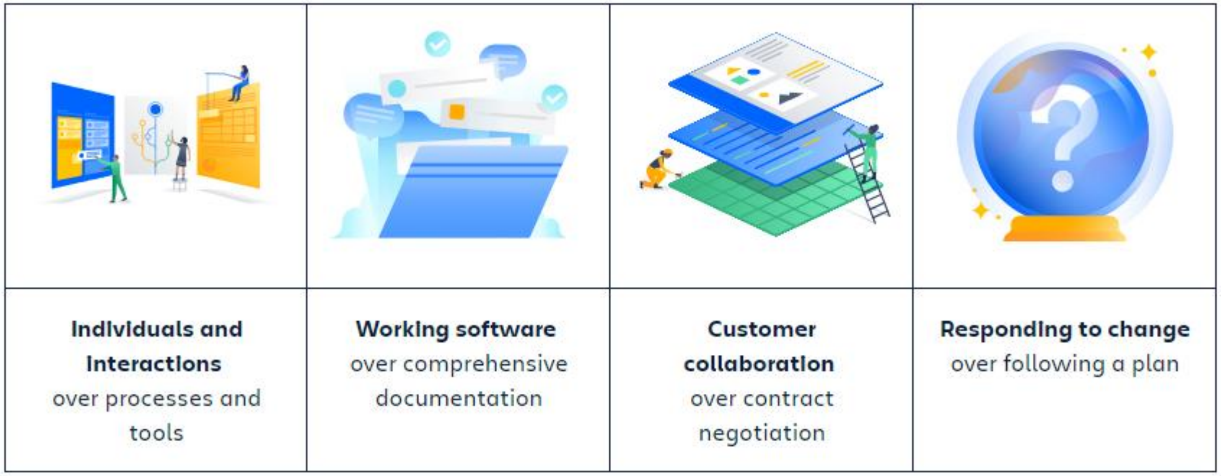
DevOps Performance Measurement

- **Which factors can be used as measurement of DevOps performance**
  - Deployment frequency (설치주기)
  - Deployment size (설치규모)
  - Deployment time (설치시간)
  - Lead time (소요시간)
  - Customer Ticket (고객불만사항)
  - Automated test-pass ratio (자동 테스트 성공률)
  - Defect escape rate (결함 제거율)
  - Availability (가용성)
  - Service level agreement (서비스 수준 협약)
  - Application usage and traffic (서비스 활용 및 트래픽)
  - Application performance (서비스 성능)
  - Mean time to detection (결함발견 간 평균시간)
  - Mean time to recovery (결함 복구 간 평균시간)

DevOps vs. Agile

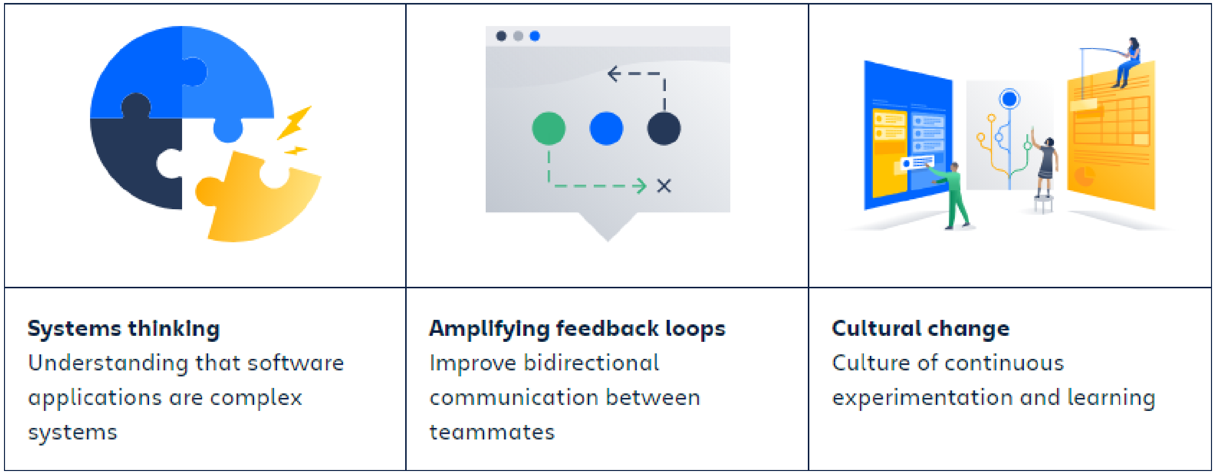
- **Characteristics of Agile and DevOps**

Agile



Individuals and interactions over processes and tools	**Working software** over comprehensive documentation	**Customer collaboration** over contract negotiation	**Responding to change** over following a plan

DevOps



Systems thinking Understanding that software applications are complex systems	**Amplifying feedback loops** Improve bidirectional communication between teammates	**Cultural change** Culture of continuous experimentation and learning

https://www.atlassian.com/devops/what-is-devops

DevOps in Practice

DevOps vs. Agile

- **What is difference between DevOps and Agile?**

Agile	DevOps
Agile emphasizes collaboration between developers and product management	DevOps includes the operations team
Agile centers the flow of software from ideation to code completion	DevOps extends the focus to delivery and maintenance
Agile emphasizes iterative development and small batches	DevOps focuses more on test and delivery automation
Agile adds structure to planned work for developers	DevOps incorporates unplanned work common to operations teams

DevOps in Practice

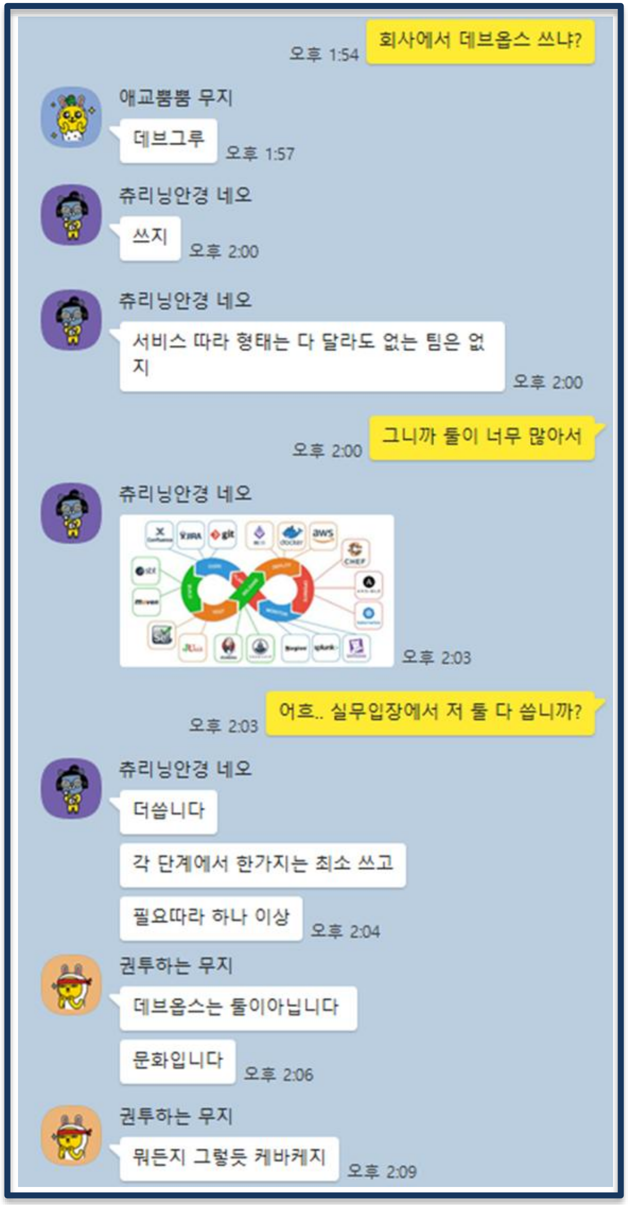
DevOps vs. Agile

- **What is difference between DevOps and Agile?**
  - The Agile Manifesto explicitly prioritizes individuals and interactions, working software, customer collaboration, and responding to change.
  - These are clearly the same priorities of DevOps but **extended beyond the development process and into the management of systems and running applications.**
  - In addition, the **Twelve Principles of Agile Software includes** references to **DevOps principles.**

- **Goals of Agile and DevOps**
  - **Ultimately the goals** of agile and DevOps are the **same**.
  - To improve the speed and quality of software development, and it makes very little sense to talk about one without the other.

DevOps in Practice

Does DevOps have been applied in the real world?



회사에서 데브옵스 쓰냐? 오후 1:54

애교뿜뿜 무지
데브그루 오후 1:57

츄리닝안경 네오
쓰지 오후 2:00

츄리닝안경 네오
서비스 따라 형태는 다 달라도 없는 팀은 없지 오후 2:00

그니까 툴이 너무 많아서 오후 2:00

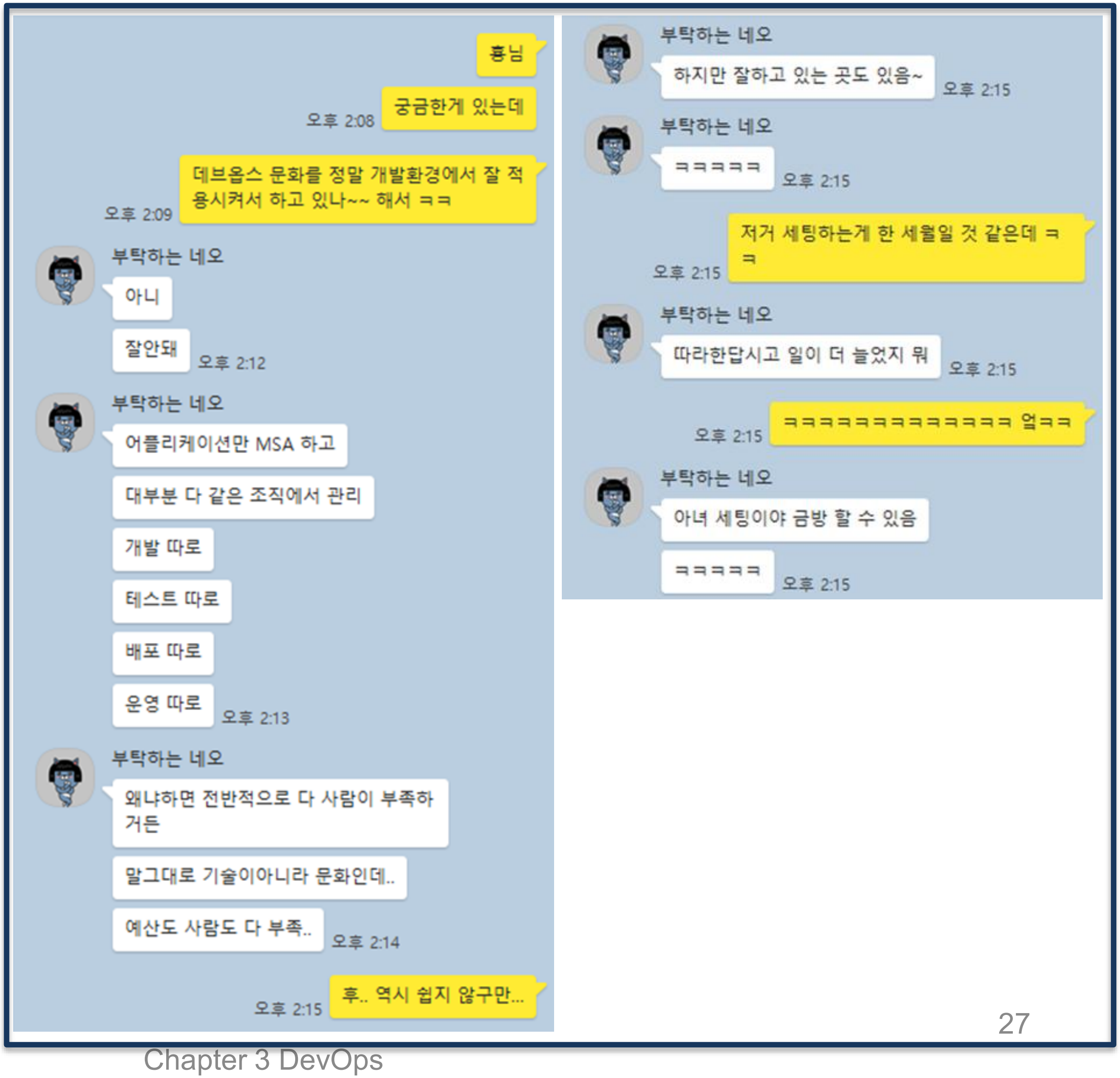
츄리닝안경 네오
오후 2:03

어흐.. 실무입장에서 저 툴 다 씁니까? 오후 2:03

츄리닝안경 네오
더씁니다
각 단계에서 한가지는 최소 쓰고
필요따라 하나 이상 오후 2:04

권투하는 무지
데브옵스는 툴이아닙니다
문화입니다 오후 2:06

권투하는 무지
뭐든지 그렇듯 케바케지 오후 2:09



흉님
궁금한게 있는데 오후 2:08

데브옵스 문화를 정말 개발환경에서 잘 적용시켜서 하고 있나~~ 해서 ㅋㅋ 오후 2:09

부탁하는 네오
아니
잘안돼 오후 2:12

부탁하는 네오
어플리케이션만 MSA 하고
대부분 다 같은 조직에서 관리
개발 따로
테스트 따로
배포 따로
운영 따로 오후 2:13

부탁하는 네오
왜냐하면 전반적으로 다 사람이 부족하거든
말그대로 기술이아니라 문화인데..
예산도 사람도 다 부족.. 오후 2:14

후.. 역시 쉽지 않구만... 오후 2:15

부탁하는 네오
하지만 잘하고 있는 곳도 있음~ 오후 2:15

부탁하는 네오
ㅋㅋㅋㅋㅋ 오후 2:15

저거 세팅하는게 한 세월일 것 같은데 ㅋㅋ 오후 2:15

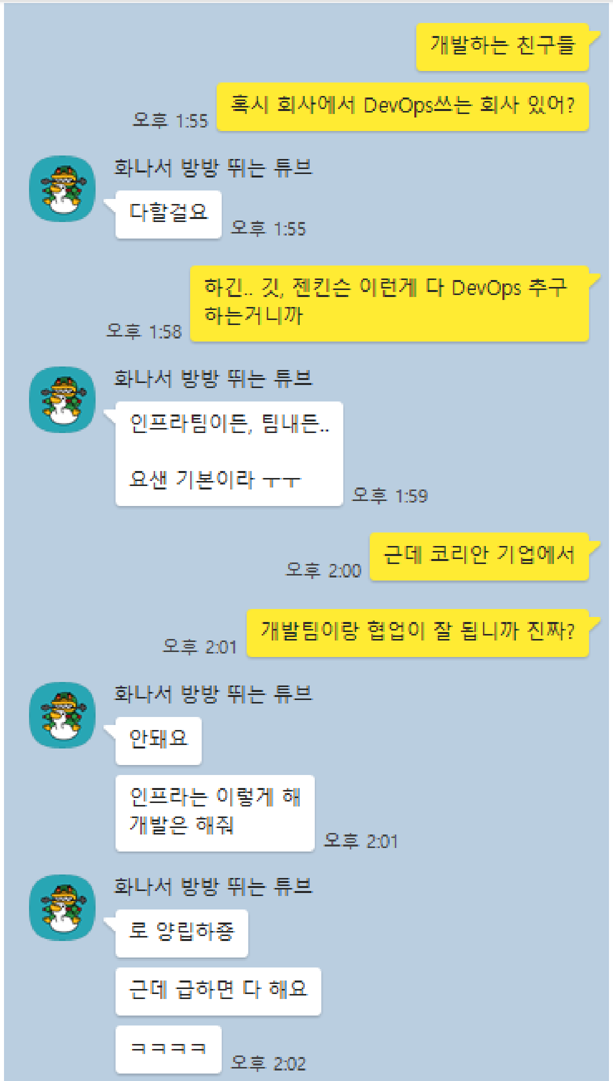
부탁하는 네오
따라한답시고 일이 더 늘었지 뭐 오후 2:15

ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ 엌ㅋㅋ 오후 2:15

부탁하는 네오
아녀 세팅이야 금방 할 수 있음
ㅋㅋㅋㅋㅋ 오후 2:15

DevOps in Practice

Does DevOps have been applied in the real world?



개발하는 친구들

혹시 회사에서 DevOps쓰는 회사 있어? 오후 1:55

화나서 방방 뛰는 튜브

다할걸요 오후 1:55

하긴.. 깃, 젠킨슨 이런게 다 DevOps 추구하는거니까 오후 1:58

화나서 방방 뛰는 튜브

인프라팀이든, 팀내든..

요샌 기본이라 ㅜㅜ 오후 1:59

근데 코리안 기업에서 오후 2:00

개발팀이랑 협업이 잘 됩니까 진짜? 오후 2:01

화나서 방방 뛰는 튜브

안돼요

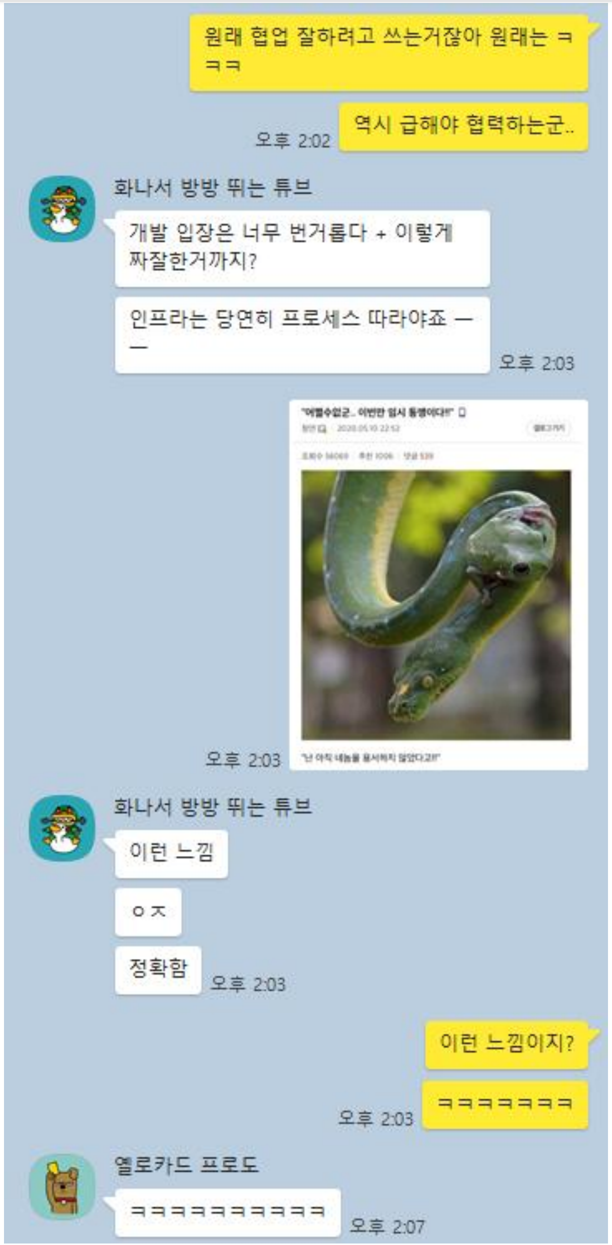
인프라는 이렇게 해 개발은 해줘 오후 2:01

화나서 방방 뛰는 튜브

로 양립하죵

근데 급하면 다 해요

ㅋㅋㅋㅋ 오후 2:02



원래 협업 잘하려고 쓰는거잖아 원래는 ㅋㅋㅋ

역시 급해야 협력하는군.. 오후 2:02

화나서 방방 뛰는 튜브

개발 입장은 너무 번거롭다 + 이렇게 짜잘한거까지?

인프라는 당연히 프로세스 따라야죠 ㅡㅡ 오후 2:03

"어쩔수없군.. 이번만 임시 동맹이다!!"

"난 아직 네놈을 용서하지 않았다고!!" 오후 2:03

화나서 방방 뛰는 튜브

이런 느낌

ㅇㅈ

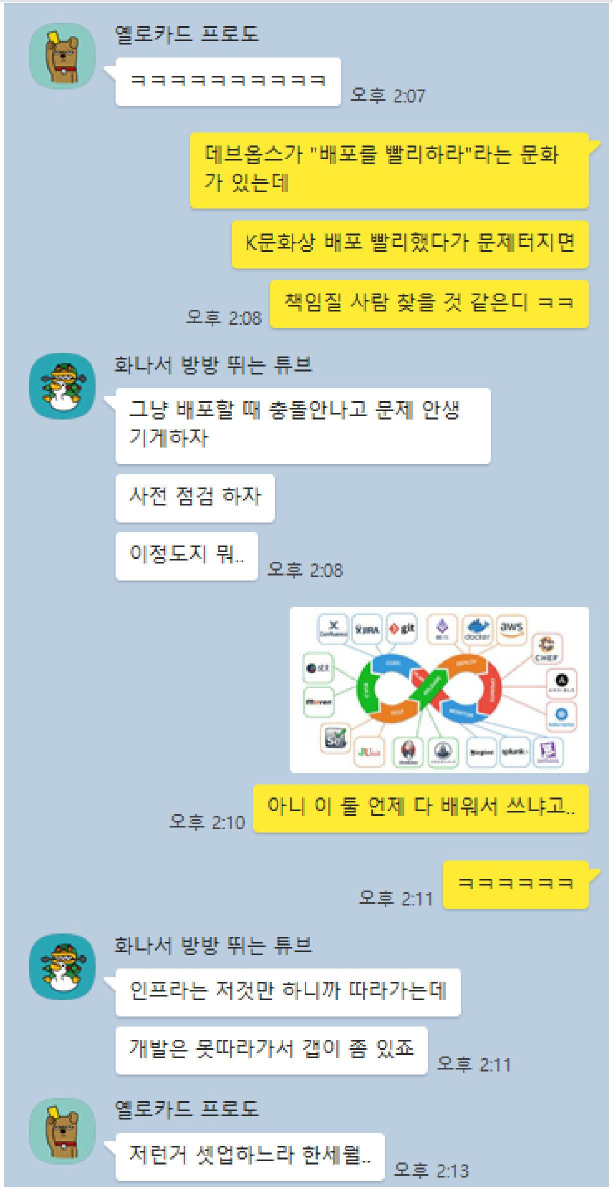
정확함 오후 2:03

이런 느낌이지?

ㅋㅋㅋㅋㅋㅋㅋ 오후 2:03

옐로카드 프로도

ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ 오후 2:07



옐로카드 프로도

ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ 오후 2:07

데브옵스가 "배포를 빨리하라"라는 문화가 있는데

K문화상 배포 빨리했다가 문제터지면

책임질 사람 찾을 것 같은디 ㅋㅋ 오후 2:08

화나서 방방 뛰는 튜브

그냥 배포할 때 충돌안나고 문제 안생기게하자

사전 점검 하자

이정도지 뭐.. 오후 2:08

아니 이 툴 언제 다 배워서 쓰냐고.. 오후 2:10

ㅋㅋㅋㅋㅋㅋ 오후 2:11

화나서 방방 뛰는 튜브

인프라는 저것만 하니까 따라가는데

개발은 못따라가서 갭이 좀 있죠 오후 2:11

옐로카드 프로도

저런거 셋업하느라 한세월.. 오후 2:13

Chapter 3. DevOps

Key Points

✧ DevOps is a software development methodology using iterative mechanism Interconnect Development(Dev), Operations(Ops), and user feedbacks.

✧ DevOps Process consists Plan, Code, Build, Test(Verify), Release, Deploy, Operate (configure infrastructure), and Monitor.

✧ A DevOps toolchain is a collection of tools, often from a variety of vendors, that operate DevOps process

✧ Ultimately the goals of agile and DevOps are the same that is enhancing the speed and quality of software development, and it makes very little sense to talk about one without the other.