



SOFTWARE ENGINEERING

Automated Software Development Process

Asst. Prof. Dr. Rathachai Chawuthai

Department of Computer Engineering

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Agenda

- DevOps
- CI/CD
- CI/CD Tools

DevOps



Why DevOps Matters

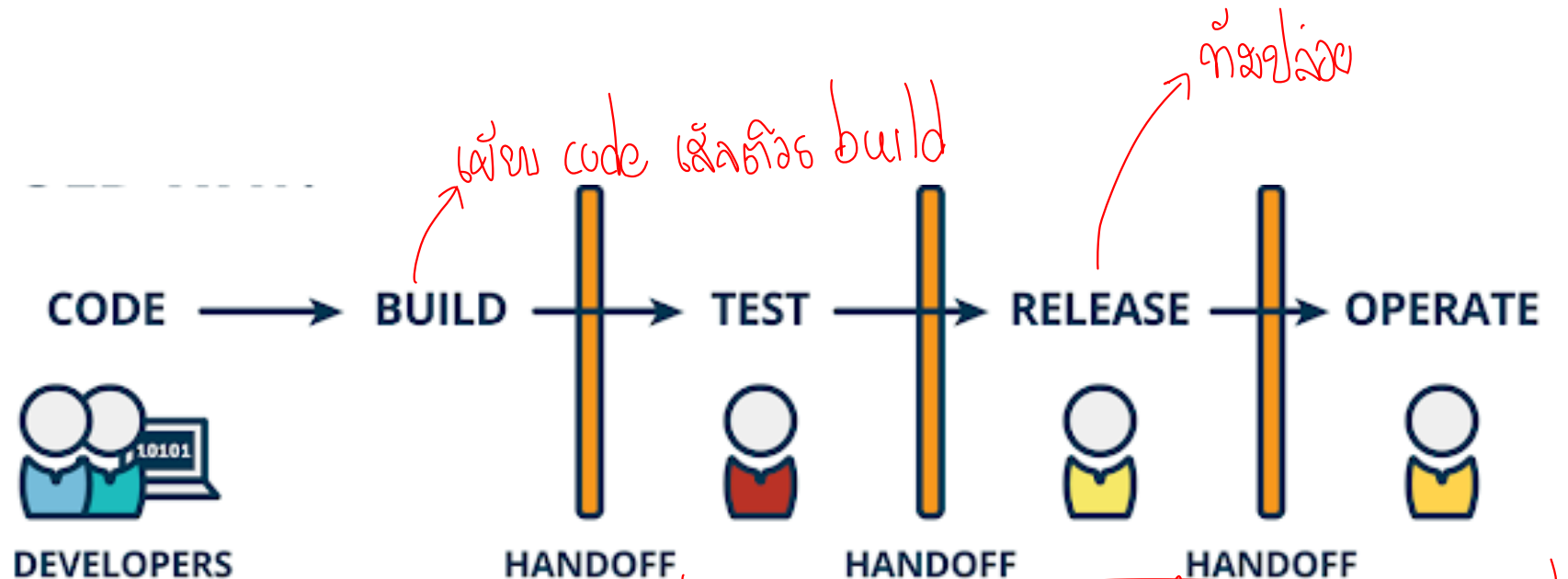
software ការងារအားလုံးလုပ်ပေး

- Software and the Internet have transformed the world and its industries, from shopping to entertainment to banking.
- Software no longer merely supports a business; rather it becomes an integral component of every part .
- Companies interact with their customers through software delivered as online services of a business. or applications and on all sorts of devices.
- They also use software to increase operational efficiencies by transforming every part of the value chain, such as logistics, communications, and operations.
- In a similar way that physical goods companies transformed how they design, build, and deliver products using industrial automation throughout the 20th century, companies in today's world must transform how they build and deliver software.

time to business → တီထွင်ဖန်တီးမှု feature အသစ်
ချိန်အချိန်အတိုအတွက်

Old Way

เขียนโปรแกรมทำ software



งานพัฒนา software
ใช้ตามขั้นตอน

process หนึ่งใช้เวลา

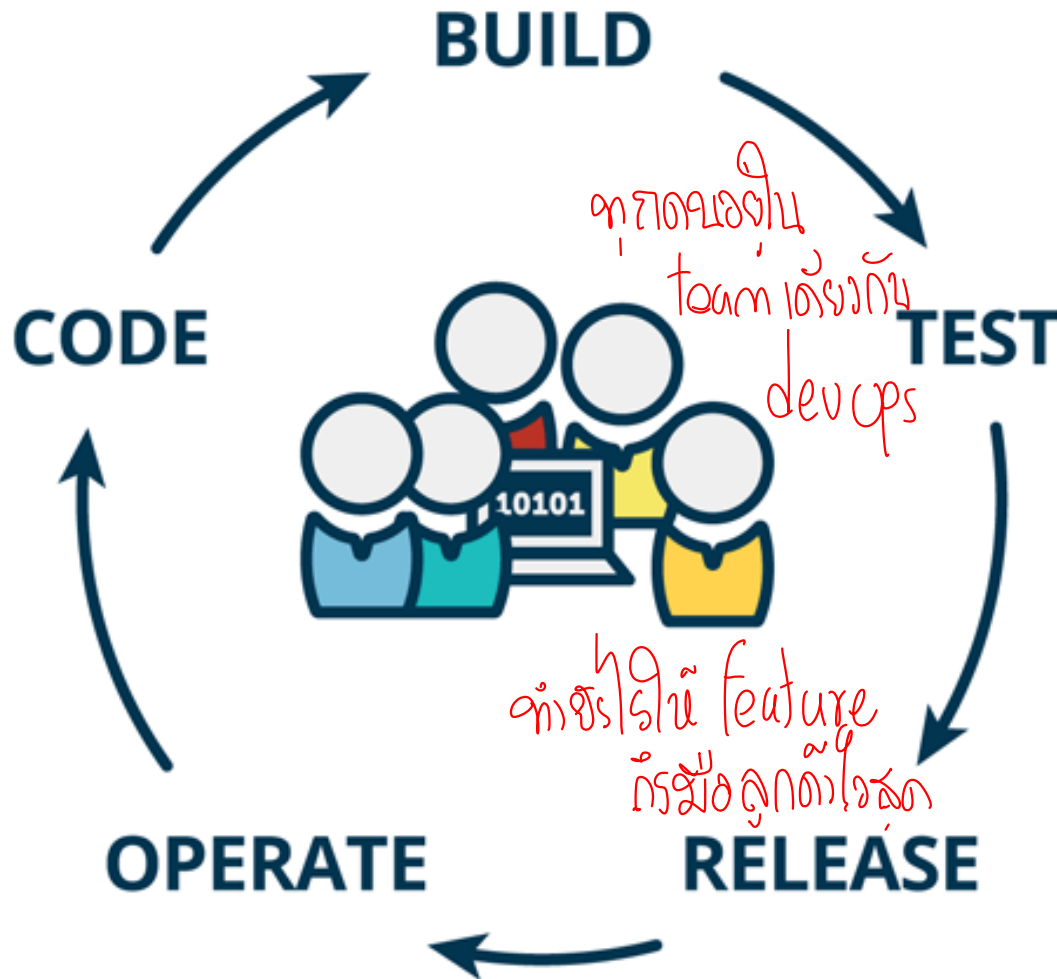
ต้องรอจนเป็นทอดๆ ไป

สามารถ automate
แทนได้

งานส่วนนี้ไม่ซ้ำกัน

New Way

សាលាសិក្សា



DevOps Model Defined

- DevOps is the combination of

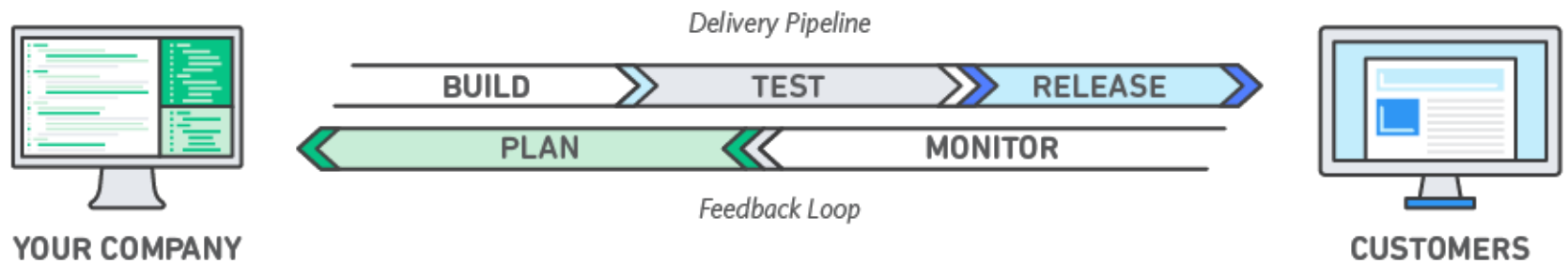
- cultural philosophies, → culture တိတ်ပေလိမ့်
- practices, and → ကလေးပုံအိမ်က
- tools

↗ code မှန်ကန်ပြီး အလုပ်ကိစ္စလုပ်ကိုင်

that increases an organization's **ability to deliver** applications and services **at high velocity**: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.

- This speed enables organizations to better serve their customers and compete more effectively in the market.

Flow



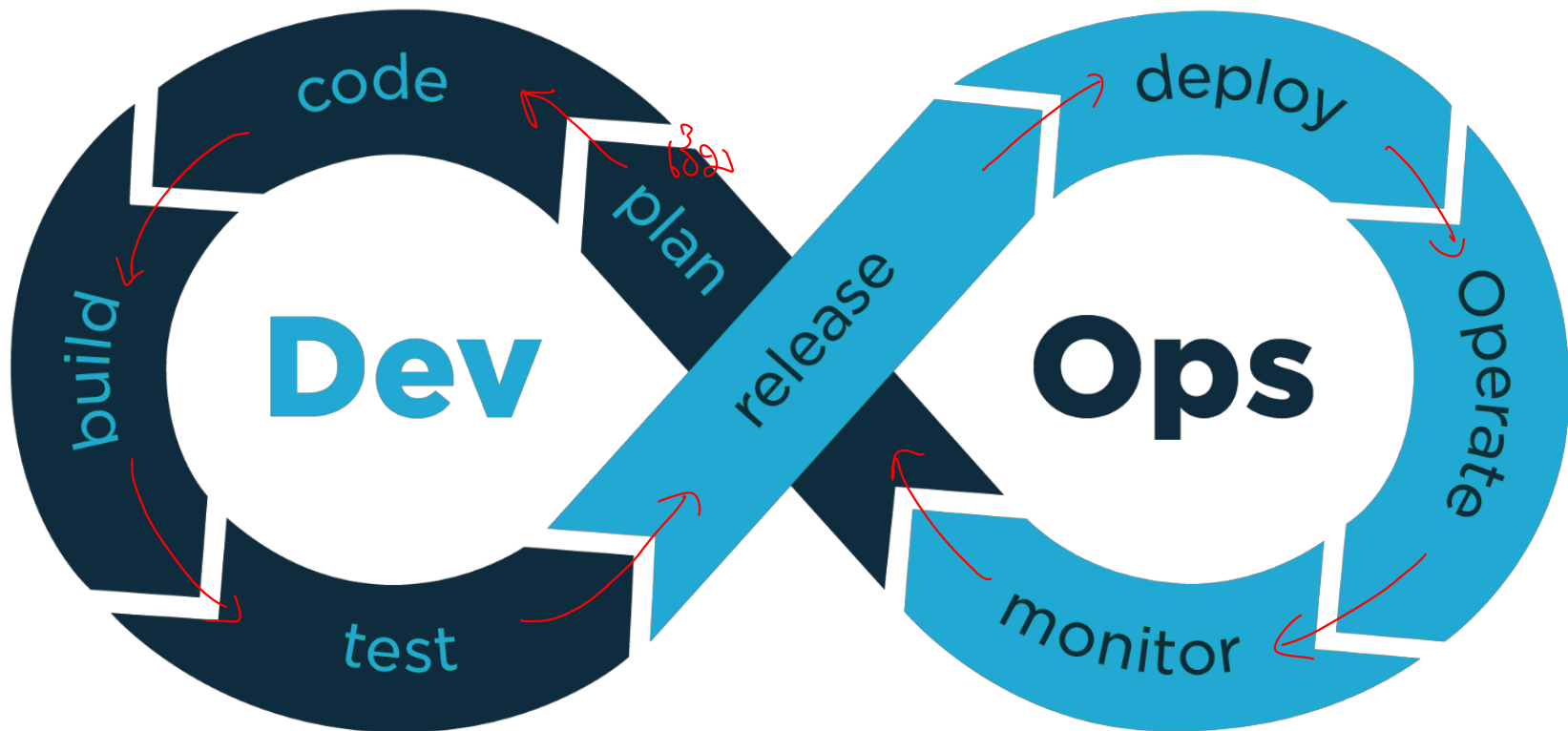
How DevOps Works → ក្នុង dev & operation មាន flow តែមួយ

- ① ការងារ
- ② ឧបករណ៍
- ③ ការងារ

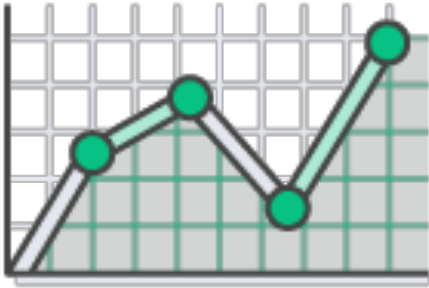
- Under a **DevOps model**, development and operations teams are no longer “siloeed.” Sometimes, **these two teams (dev & ops) are merged into a single team** where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function.
- In some DevOps models, quality assurance and security teams may also become more tightly **integrated with development and operations and throughout the application lifecycle**. When security is the focus of everyone on a DevOps team, this is sometimes referred to as DevSecOps.
- These teams use practices to automate processes that historically have been manual and slow. **They use a technology stack and tooling which help them operate and evolve applications quickly and reliably**. These tools also help engineers independently accomplish tasks (for example, deploying code or provisioning infrastructure) that normally would have required help from other teams, and this further increases a team’s velocity.

DevOps

สิ่งนี้คือ DevOps



Benefit of DevOps



Speed ការធ្វើការងារឱ្យលឿន (លឿនជាងការងារដទៃ)

- Move at high velocity so you can innovate for customers **faster**, **adapt to changing markets better**, and grow more efficient at **driving business results**.
- The DevOps model enables your developers and operations teams to achieve these results. For example, microservices and continuous delivery let teams take ownership of services and then release updates to them quicker.

Benefit of DevOps



Rapid Delivery

ส่งมอบได้ไวขึ้น

- Increase the frequency and pace of releases so you can innovate and improve your product faster.
- The quicker you can release new features and fix bugs, the faster you can respond to your customers' needs and build competitive advantage.
- Continuous integration and continuous delivery are practices that automate the software release process, from build to deploy.

Benefit of DevOps



Reliability → ធានាគុណភាព ផលិតផល → (test)

- Ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining a positive experience for end users.
- Use practices like continuous integration and continuous delivery to test that each change is functional and safe.
- Monitoring and logging practices help you stay informed of performance in real-time.

Benefit of DevOps



Scale → ควบคุมและจัดการ infra ให้สามารถรองรับปริมาณ user ที่เพิ่มขึ้น

- Operate and manage your infrastructure and development processes at scale.
- Automation and consistency help you manage complex or changing systems efficiently and with reduced risk.
- For example, infrastructure as code helps you manage your development, testing, and production environments in a repeatable and more efficient manner.

Benefit of DevOps



Improved Collaboration

ကျောင်းအတွင်း Flow
ပိုမိုလျင်မြန်
အောင်မြင်

- Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability.
- Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows.
- This reduces inefficiencies and saves time (e.g. reduced handover periods between developers and operations, writing code that takes into account the environment in which it is run).

DevOps Cultural Philosophy

culture បំប្លែង

- Transitioning to DevOps requires a change in culture and mindset. At its simplest, DevOps is about removing the barriers between two traditionally siloed teams, development and operations. In some organizations, there may not even be separate development and operations teams; engineers may do both.
- With DevOps, the two teams work together to optimize both the productivity of developers and the reliability of operations. They strive to communicate frequently, increase efficiencies, and improve the quality of services they provide to customers.
- They take full ownership for their services, often beyond where their stated roles or titles have traditionally been scoped by thinking about the end customer's needs and how they can contribute to solving those needs.
- Quality assurance and security teams may also become tightly integrated with these teams. Organizations using a DevOps model, regardless of their organizational structure, have teams that view the entire development and infrastructure lifecycle as part of their responsibilities.

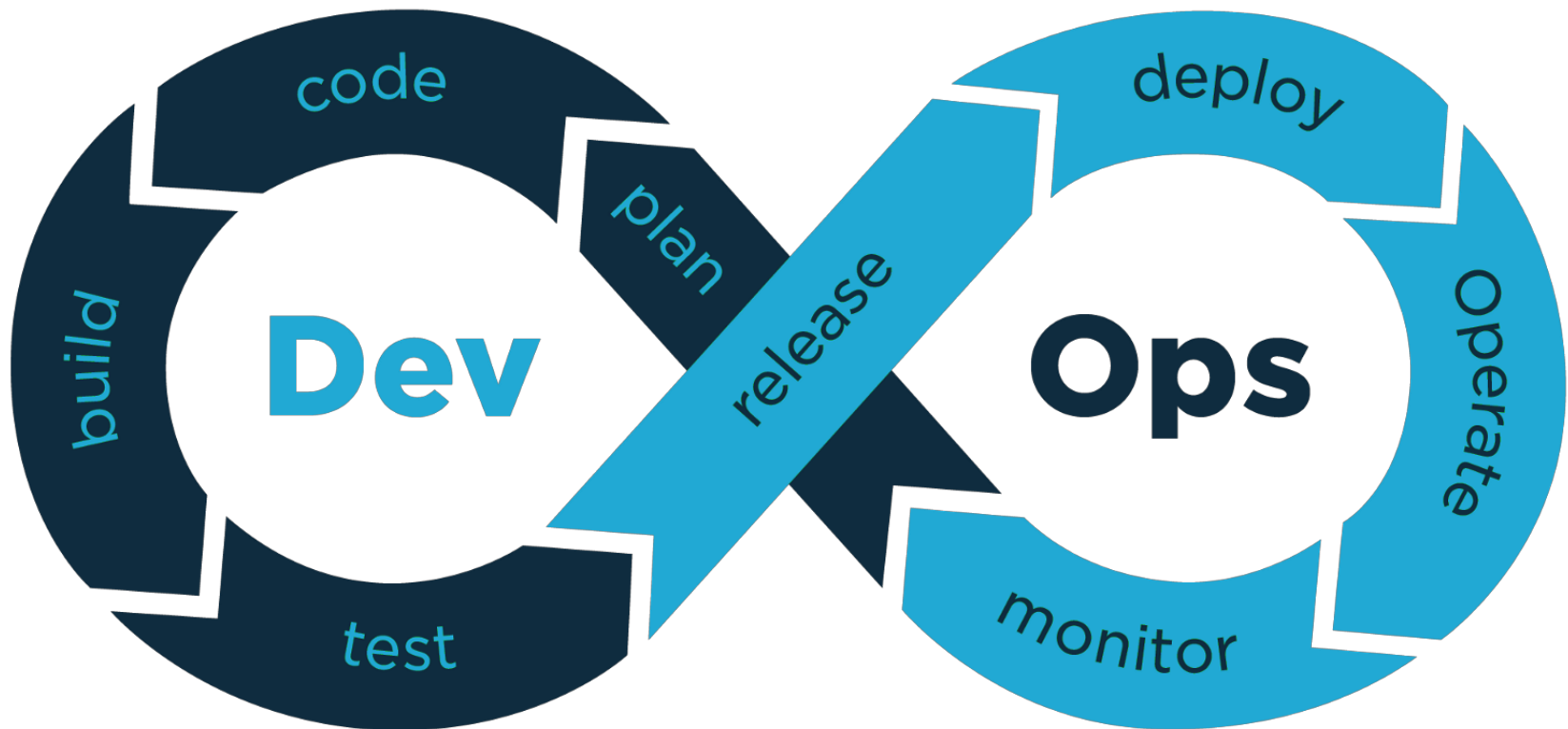
การบูรณาการ dev ops ทั่วทั้ง

CI/CD
↓ ↓ delivery
continuous integration

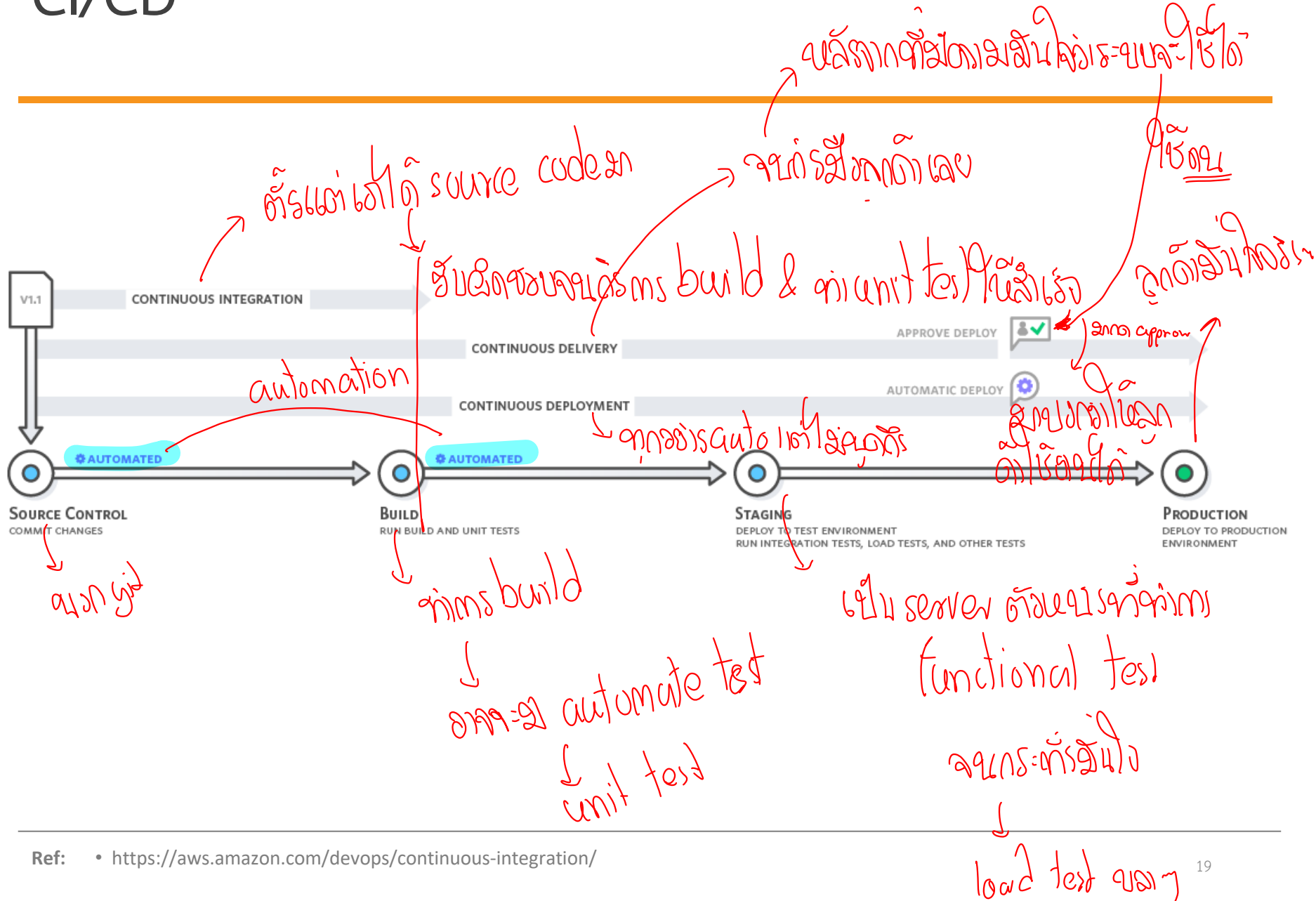


DevOps

ជំរក់ចរន្តការងាររវាង Dev និង Ops



CI/CD



CI : Continuous integration

but(automate)

test(automate)

- Continuous integration is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.
- Continuous integration most often refers to the **build or integration stage of the software release process and entails both an automation component** (e.g. a CI or build service) and a cultural component (e.g. learning to integrate frequently).
- The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

CI: How បំពេញការងារ version control → ឯ CI ទើបនឹងដឹង

- With continuous integration, developers frequently commit to a shared repository using a **version control** system such as Git. Prior to each commit, developers may choose to run local unit tests on their code as an extra verification layer before integrating.
- A continuous integration service automatically builds and runs unit tests on the new code changes to immediately surface any errors.
- Continuous integration refers to the build and unit testing stages of the software release process. **Every revision that is committed triggers an automated build and test.**
↓ ធ្វើការ commit ហើយ code ៗ build → test ប្រសិនបើ
- With continuous delivery, code changes are automatically built, tested, and **prepared for a release to production**. Continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

CD: Continuous Delivery

အသေးစား production

- Continuous delivery is a software development practice where code changes are **automatically prepared for a release to production**. A pillar of modern application development, continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When properly implemented, developers will always have a deployment-ready build artifact that has passed through a standardized test process.
- Continuous delivery lets developers automate testing beyond just unit tests so they can verify application updates across multiple dimensions before deploying to customers. **These tests may include UI testing, load testing, integration testing, API reliability testing**, etc. This helps developers more thoroughly validate updates and pre-emptively discover issues. With the cloud, it is easy and cost-effective to automate the creation and replication of multiple environments for testing, which was previously difficult to do on-premises.

အသေးစား tester ခံ → tester အသေးစား ပြုပြင်

↓
အသေးစား test (auto)

CD: Continuous Delivery

- Continuous delivery automates the entire software release process. Every revision that is committed triggers an automated flow that builds, tests, and then stages the update. The final decision to deploy to a live production environment is triggered by the developer.

ကျွန်ုပ်တို့က automation test

↓
မေးခွန်း (၁) မေးခွန်းများအတွက်

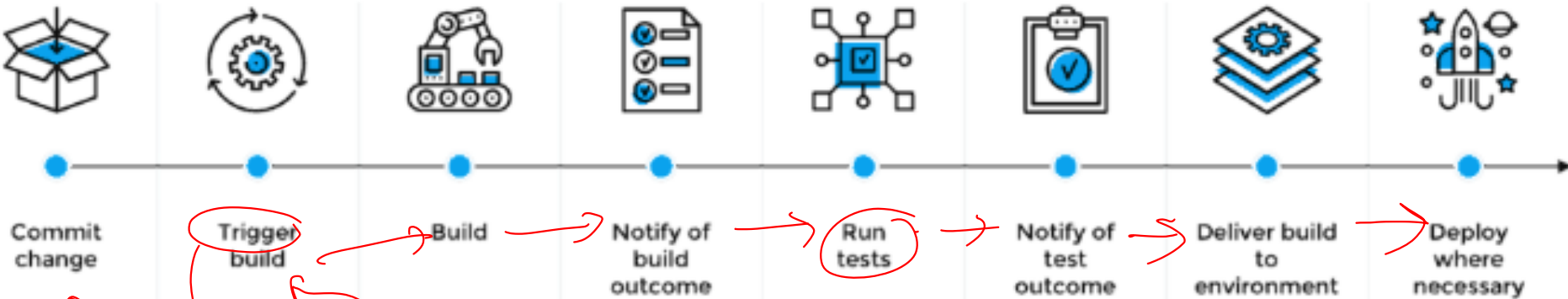
CI/CD Pipeline

coding → software
test script

build ops ทำอะไรหุ

↓
จะ setup pipeline ให้ complete

↓
ให้ automation team



new SC

จน

↓
ถ้า error งานที่ done ก็จะ

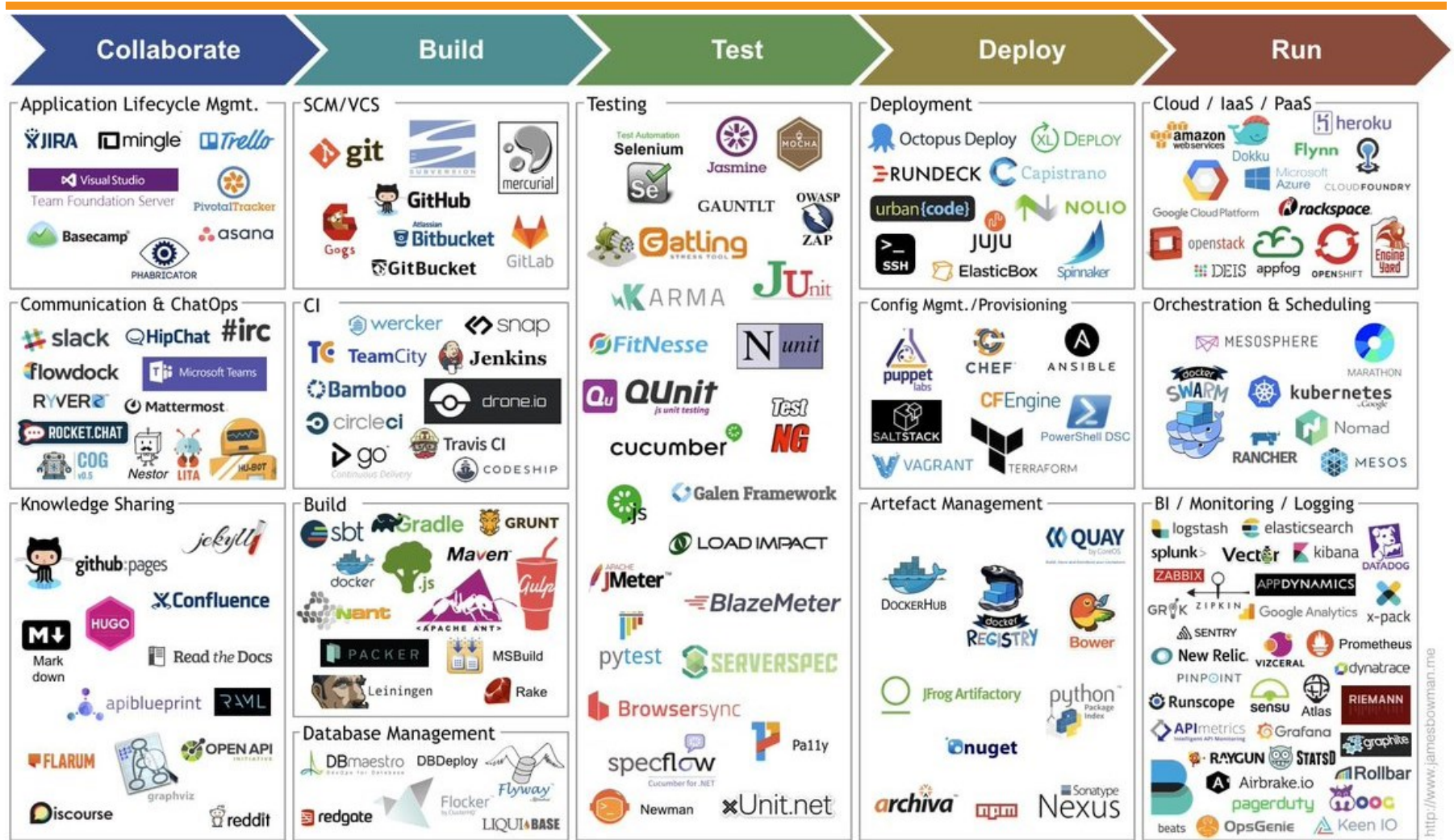
↓
ถ้า success

↓
ให้ automation flow ไป

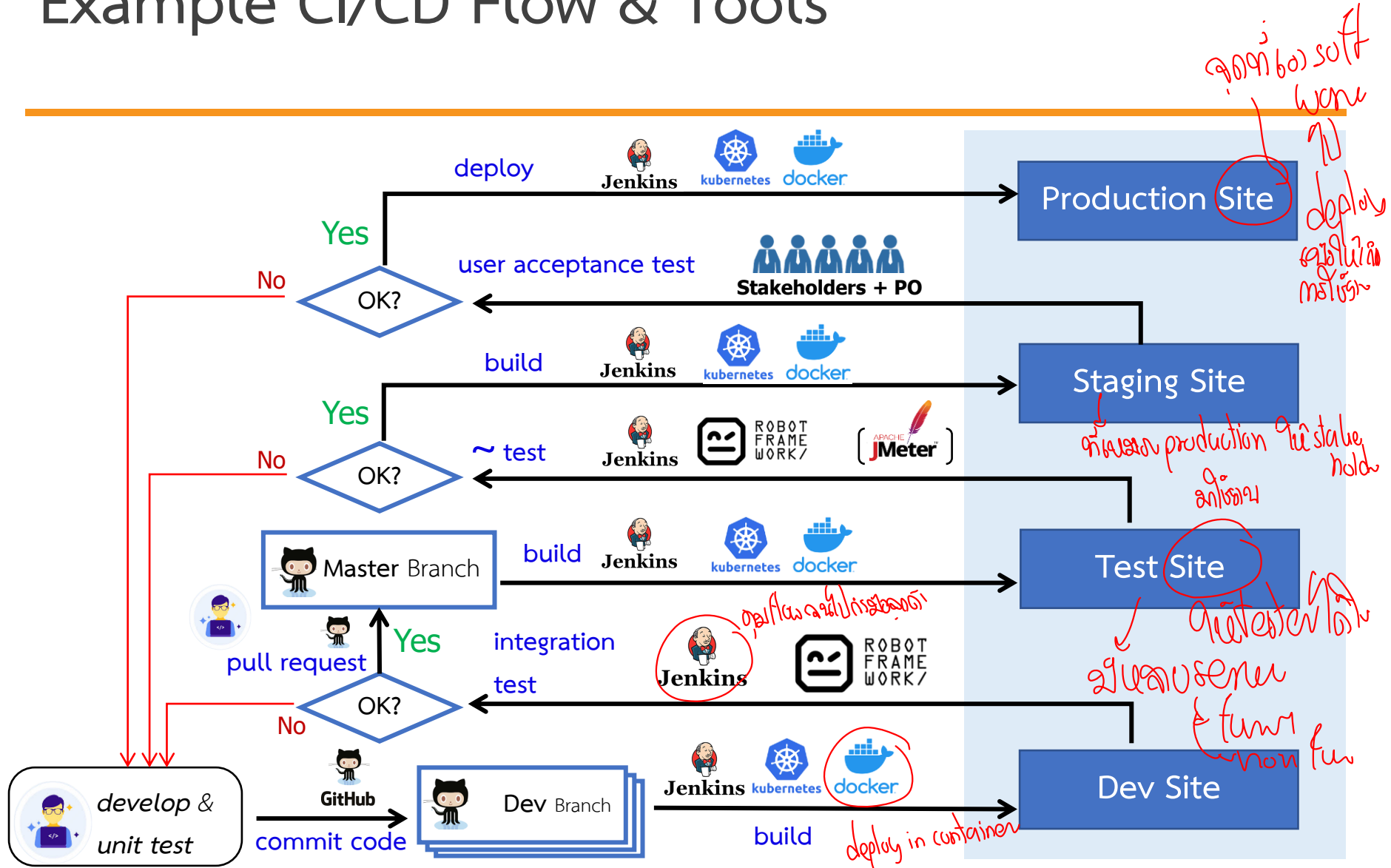
CI/CD Tools



CI/CD Tools

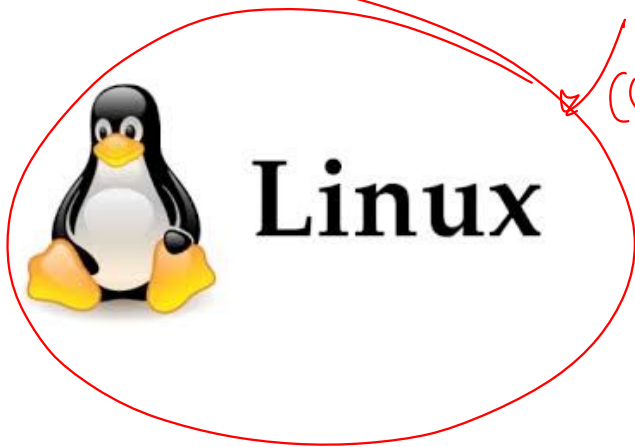


Example CI/CD Flow & Tools



Example CI/CD Skills

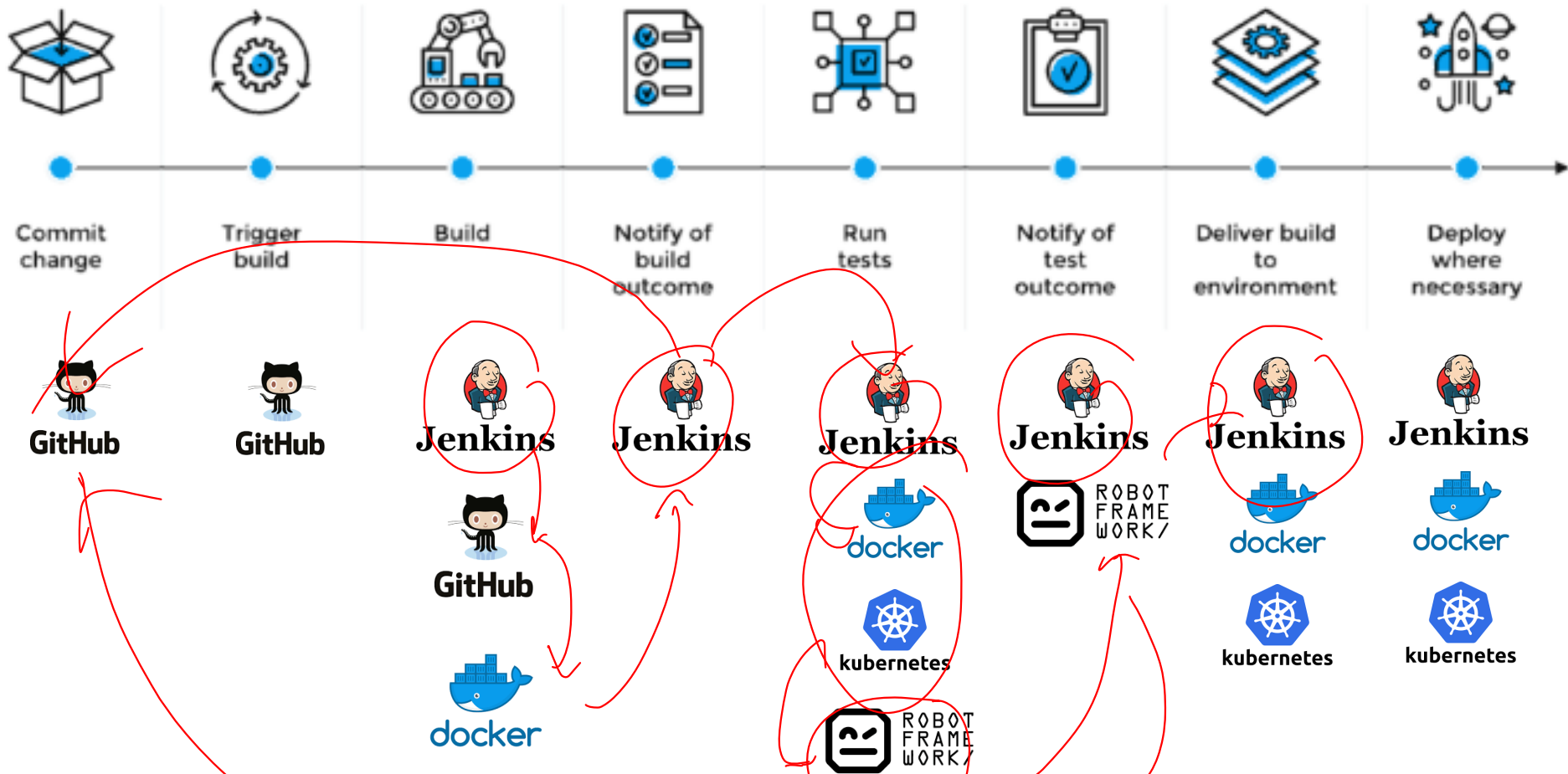
CI, network
config



Summary



Key Points



“

To successfully implement continuous delivery,
you need to change the culture of
how an entire organization views
software development efforts.

”

Tommy Tynjä