Collaboration for Development

5CS024

Learning Objectives

Learning Objectives

- Coding Standards
- Style Guides
- Code Reviews
- Pair Programming
- Collaborative Coding Tools
    - Playgrounds
    - Collaborative Text Editors
- Wikis
- DevOps

1. Coding Standards

1. Coding Standards

- Set of guidelines for a specific programming language that recommend programming style, practices, and methods for each aspect of a program written in that language.
- Also sometimes referred to as ***'Coding Conventions'***, ***'Programming Style'***
- The dream is that any developer familiar with the guidelines can work on any code that followed them.
- Standards range from a simple series of statements to large documented volume of guide.
- Can be - *General*, *Language-specific* or *Project-specific*

1. Coding Standards

- **What does it contain?**
  - Program Design
  - Naming Conventions
  - Formatting Conventions
  - Comments
  - Code Layout
  - File Organization and Modules
  - Miscellaneous rules for specific Language
- **WHY?**
  - Greater consistency between developers
  - Easier to develop and maintain
  - People can quickly get into the problem without worrying about the language
  - Fewer mistakes creep in
  - Maintainability, Readability and Reusability is maximized

1. Coding Standards

Examples:

PEP-8 – This document gives coding conventions for the Python code comprising the standard library in the main Python distribution

PSR-1 - The standard comprises what should be considered the standard coding elements that are required to ensure a high level of technical interoperability between shared PHP code.

AirBnb Javascript Style Guide - AirBnb's documentation of Javascript Best practices that has been accepted and well-regarded throughout the industry

1. Coding Standards

- **Prime Directive**

  "Document every time you violate a standard."

  No standard is perfect for every application, but failure to comply with your standards requires a comment

- **Ambler's Law of Standards**
  - Project Standards > No Standards
  - Organizational Standards > Project Standards
  - Industry Standards > Organizational standards

Review coding standards periodically and Automate checking your code with linters and formatters.

2. Style Guides

2. Style Guides

A style guide is a set of standards for the writing and design of documents

- Focused on content, and directed mostly at content managers and copy writers.
- Improves communication and branding through consistency
- Enforces best practices

2. Style Guides

Mostly concerned with:

- Spelling and grammar
- Punctuation
- Acronyms
- Contractions and abbreviations

Examples:

- Google Developer Documentation Style Guide
- Mozilla Writing Style Guide

3. Code Review

2.Style Guides

3.Code Review

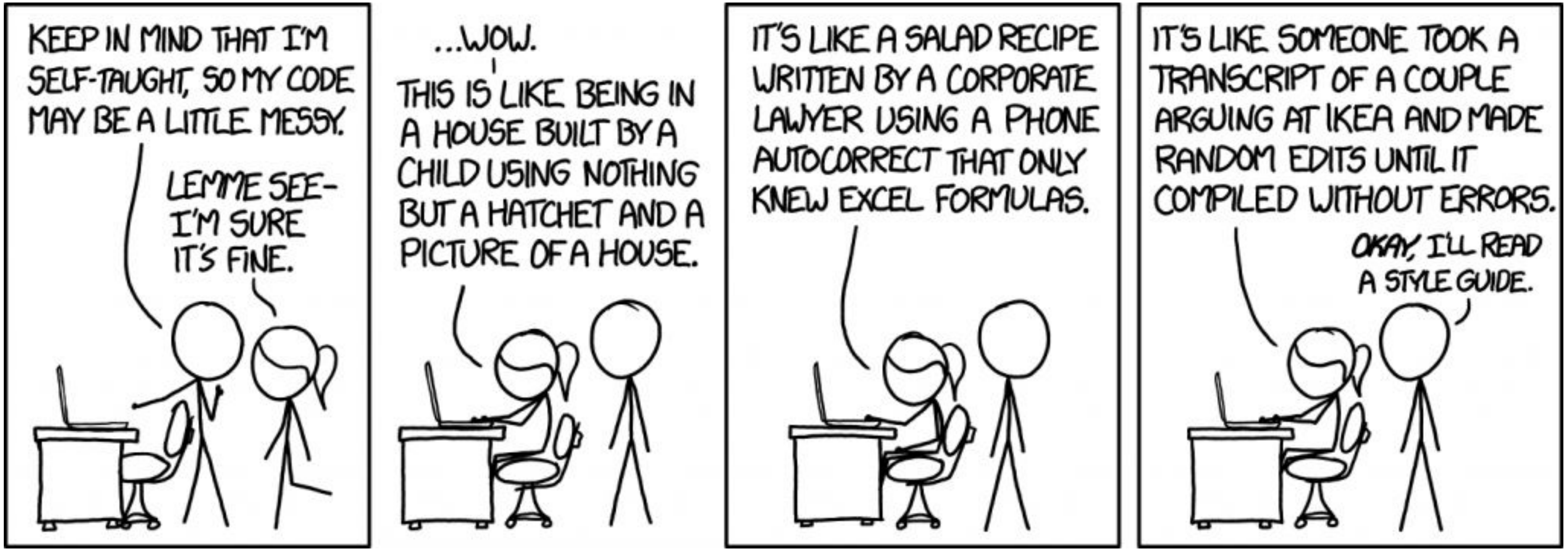
4.Pair Programming

3. Code Review

“Agile teams are self-organizing, with skill sets that span across the team. This is accomplished, in part, with code review. Code review helps developers learn the code base, as well as help them learn new technologies and techniques that grow their skill sets.”

3. Code Review

Code Review is the act of consciously and systematically working with one's fellow programmers to check each other's code for mistakes, and has been repeatedly shown to accelerate and streamline the process of software development like few other practices can.


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE- I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

3. Code Review

When a developer is finished working on an issue, another developer looks over the code and considers questions like:

- Are there any obvious logic errors in the code?
- Looking at the requirements, are all cases fully implemented?
- Are the new automated tests sufficient for the new code? Do existing automated tests need to be rewritten to account for changes in the code?
- Does the new code conform to existing style guidelines?

One example of Code Review Process: https://thoughtbot.com/playbook/developing/code-reviews

3. Code Review

Benefits:

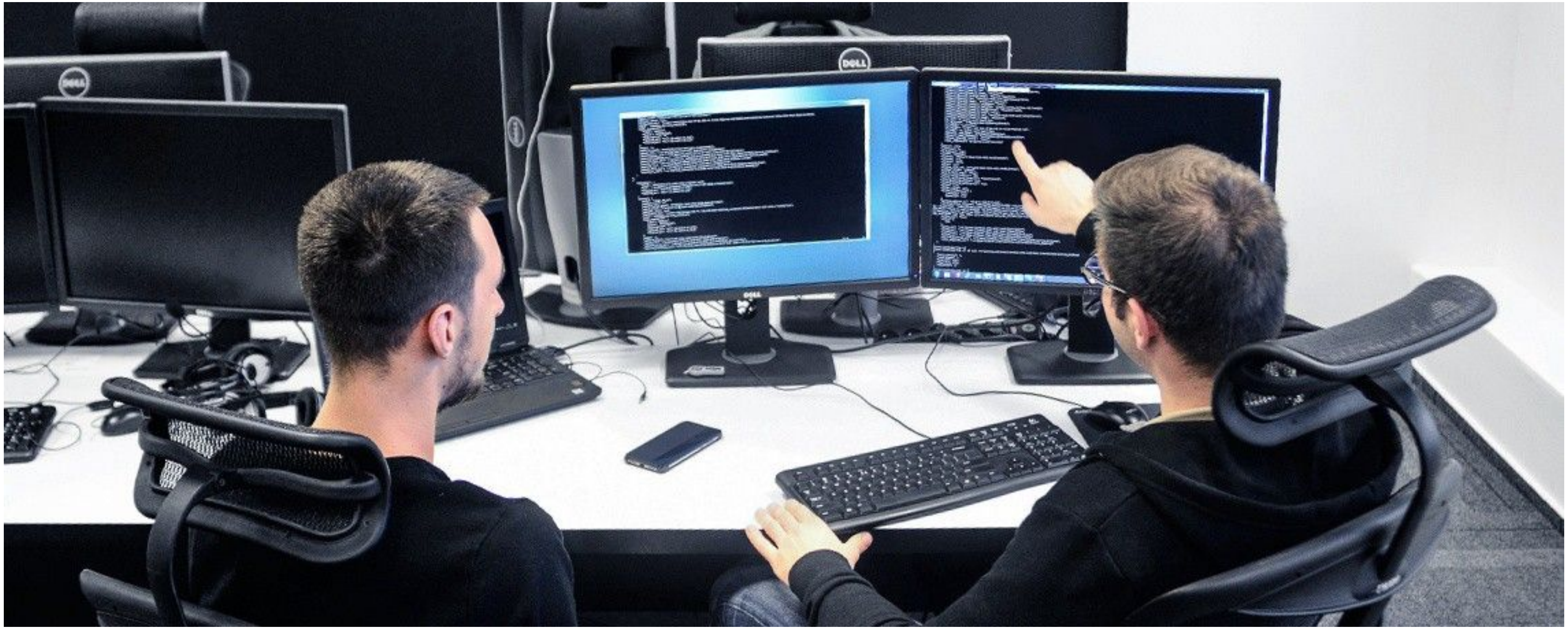
- Consistent design and implementation
- Minimizing your mistakes and their impact
- Ensuring project quality and meeting requirements
- Improving code performance
- Sharing new techniques

3. Code Review

Common Approaches:

- The Email Thread
- Pair Programming
- Over-the-Shoulder
- Tool-Assisted

4. Pair Programming



Pair Programming, also known as **‘Pairing’**

4. Pair Programming

Pair programming consists of two programmers sharing a single workstation (one screen, keyboard and mouse among the pair).

The programmer at the keyboard is usually called the "driver", the other, also actively involved in the programming task but focusing more on overall direction is the "navigator"; it is expected that the programmers swap roles every few minutes or so.

4. Pair Programming

Benefits:

- **Increased discipline:** Pairing partners are more likely to "do the right thing" and are less likely to take long breaks.
- **Better code:** Pairing partners are less likely to go down Internet and Rabbit-holes and tend to come up with higher quality designs.
- **Resilient flow:** Pairing flow happens more quickly: one programmer asks the other, "What were we working on?" Pairing flow is also more resilient to interruptions: one programmer deals with the interruption while the other keeps working.
- **Improved morale:** Pair programming, done well, is much more enjoyable than programming alone, done well.

4. Pair Programming

Benefits:

- **Collective code ownership:** When everyone on a project is Pair Programming, and pairs rotate frequently, everybody gains a working knowledge of the entire codebase.
- **Mentoring:** Pair programming is a painless way of spreading that knowledge.
- **Team cohesion:** People get to know each other more quickly when pair programming. Pair programming may encourage team jelling.
- **Fewer interruptions:** People are more reluctant to interrupt a pair than they are to interrupt someone working alone.

4. Pair Programming

What to be cautious about?

- **Higher Costs:** Studies have shown a 15% - 100% increase in baseline development costs
- **Mixed Results:** Studies on pair programming have not returned conclusive evidence of measurable benefits.
- **Team Fit:** The high-intensity communication of pair programming is not a good fit for every personality.
- **“Watch the Master”:** It can also lead to the more experienced developer dominating the process. The novice developer then becomes disengaged, “watching the master” work.

5. Collaborative Coding Tools

5. Collaborative Coding Tools

Playgrounds

"Code playgrounds offer a quick and dirty way to experiment with client-side code and share with others."

Examples: Codepen, JSFiddle

Collaborative Text Editors

Code editors that let you collaborate with other programmers, help you to edit your code in real time and keep you constantly updated with changes in projects that you may otherwise overlook

Examples: Collabedit, Firepad

6. Wikis

6. Wikis

“A wiki is a website that is collaboratively created by multiple users. It can also be thought of as a collaborative content management system (CMS) for collecting and organizing media that is created and revised by its users.”

It is based on the idea that within any organization, a great deal of knowledge exists among the members. Sharing this knowledge and information can raise the organization’s intelligence level, be it a corporation, association or educational institution.

6. Wikis

Uses:

- Knowledge Management and Information Distribution
- Collaboration
- Training
- Developing Community

5.Collaborative Coding Tools

6.Wikis

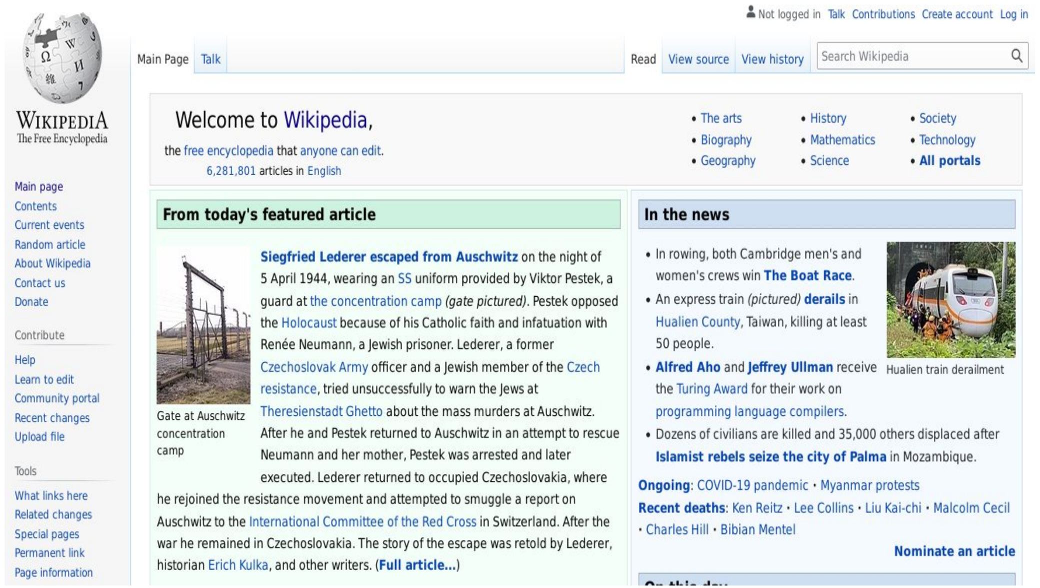
7.DevOps

6. Wikis

Examples:

[Mediawiki](#)

[Confluence](#)



Not logged in Talk Contributions Create account Log in

Main Page Talk Read View source View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
6,281,801 articles in English

- The arts
- Biography
- Geography
- History
- Mathematics
- Science
- Society
- Technology
- **All portals**

From today's featured article

Gate at Auschwitz concentration camp

Siegfried Lederer escaped from Auschwitz on the night of 5 April 1944, wearing an SS uniform provided by Viktor Pestek, a guard at the concentration camp *(gate pictured)*. Pestek opposed the Holocaust because of his Catholic faith and infatuation with Renée Neumann, a Jewish prisoner. Lederer, a former Czechoslovak Army officer and a Jewish member of the Czech resistance, tried unsuccessfully to warn the Jews at Theresienstadt Ghetto about the mass murders at Auschwitz. After he and Pestek returned to Auschwitz in an attempt to rescue Neumann and her mother, Pestek was arrested and later executed. Lederer returned to occupied Czechoslovakia, where he rejoined the resistance movement and attempted to smuggle a report on Auschwitz to the International Committee of the Red Cross in Switzerland. After the war he remained in Czechoslovakia. The story of the escape was retold by Lederer, historian Erich Kulka, and other writers. (**Full article...**)

In the news

Hualien train derailment

- In rowing, both Cambridge men's and women's crews win **The Boat Race**.
- An express train *(pictured)* **derails** in Hualien County, Taiwan, killing at least 50 people.
- **Alfred Aho** and **Jeffrey Ullman** receive the Turing Award for their work on programming language compilers.
- Dozens of civilians are killed and 35,000 others displaced after **Islamist rebels seize the city of Palma** in Mozambique.

Ongoing: COVID-19 pandemic • Myanmar protests
Recent deaths: Ken Reitz • Lee Collins • Liu Kai-chi • Malcolm Cecil • Charles Hill • Bibian Mentel

Nominate an article

7. DevOps

7.1 Context

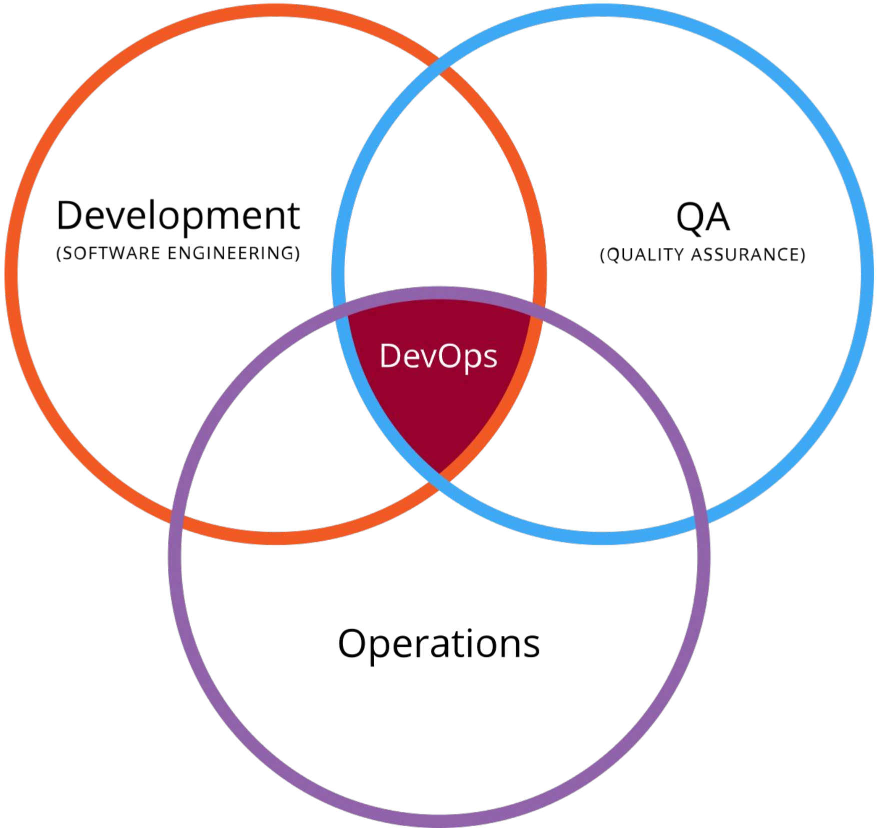
Developers

- Always looks to implement new functionalities and bring changes
- New updates have to be integrated frequently

Operations / Sys Admins

- Change is the enemy of Operations
- It is not reliable and lead to instability

WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net



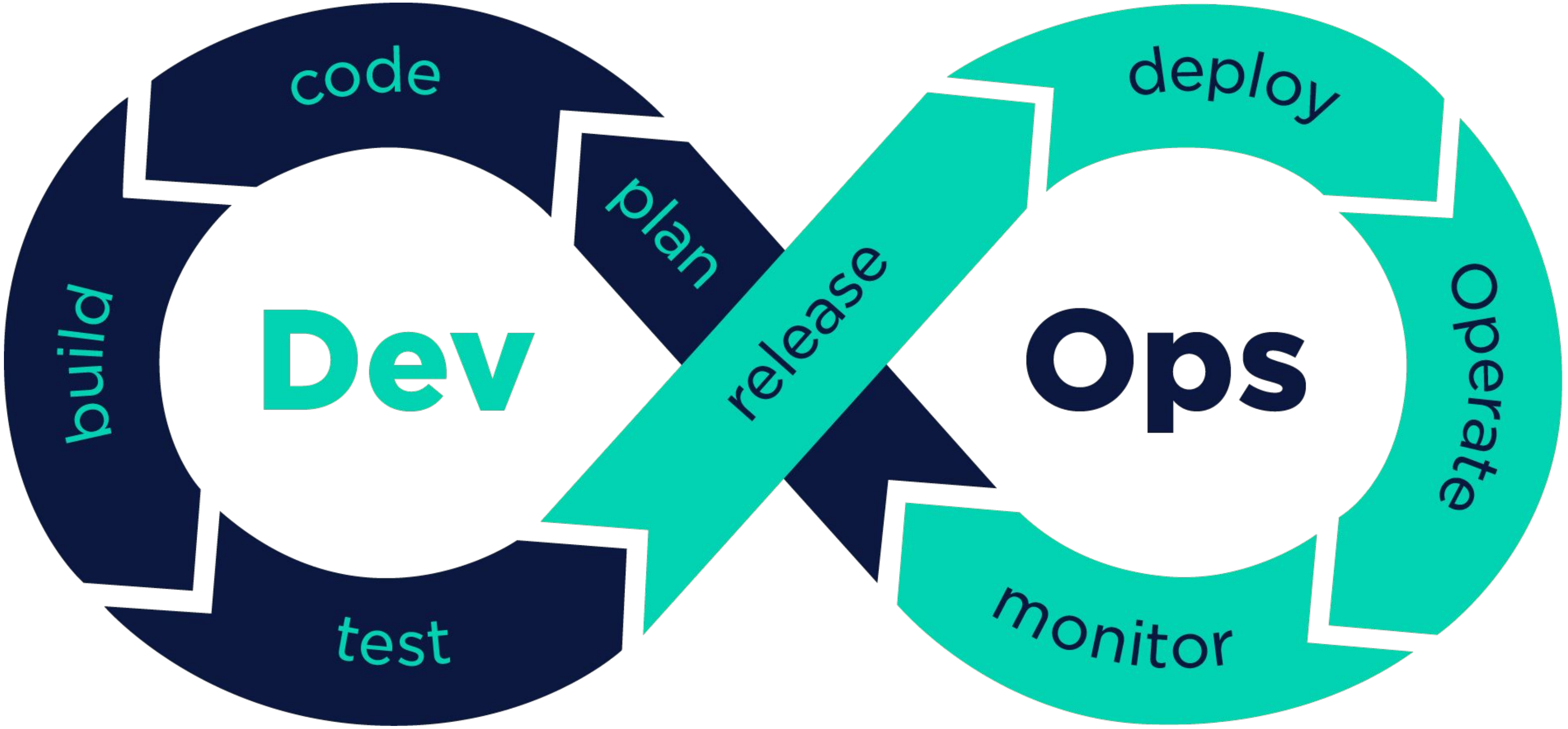
Intersection of Development, Operations and Quality Assurance

7.3 DevOps

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). Collaborative Mindset of Devs and Ops

It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.



Stages in DevOps

7.3 DevOps

Continuous Integration

Continuous integration is a software development practice in which developers merge code changes frequently into the main code branch.

Continuous integration employs automated testing, which runs every time new code is committed so the code in the main branch is always stable.

Continuous Deployment

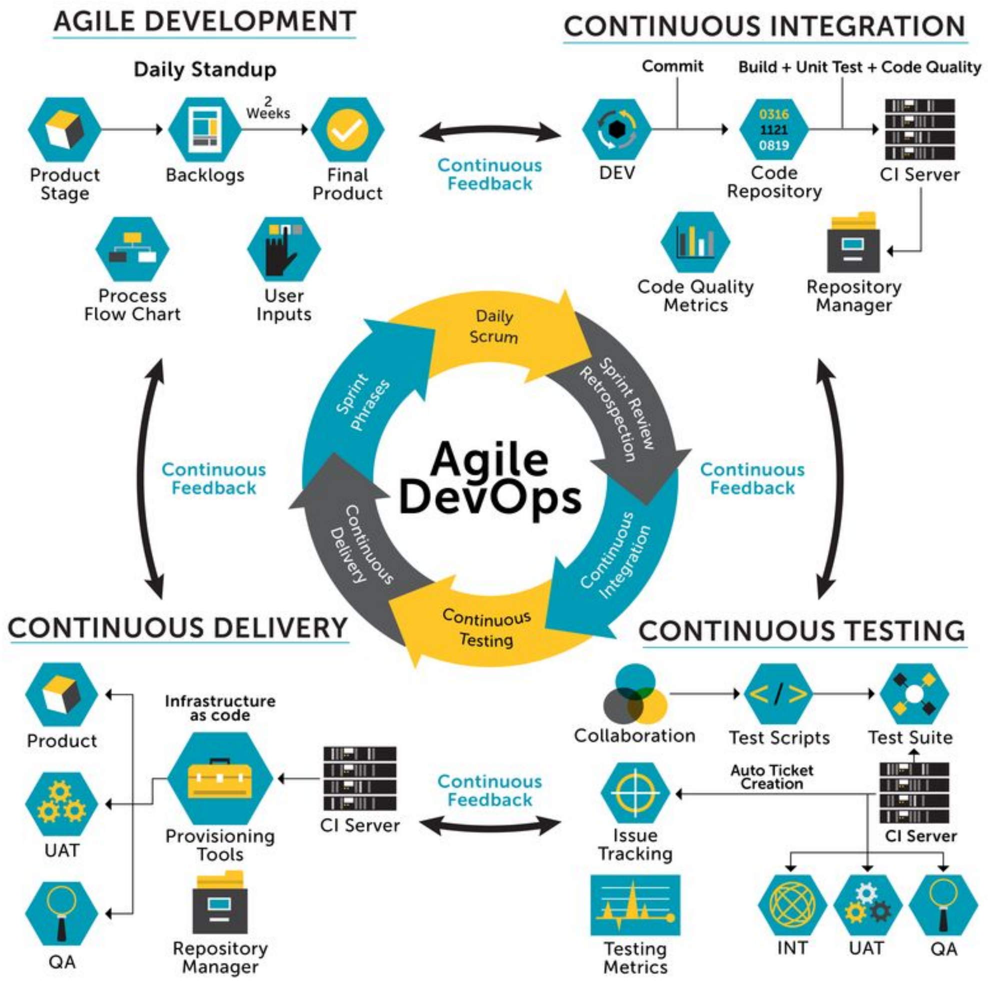
Continuous delivery is the frequent, automated deployment of new application versions into a production environment.

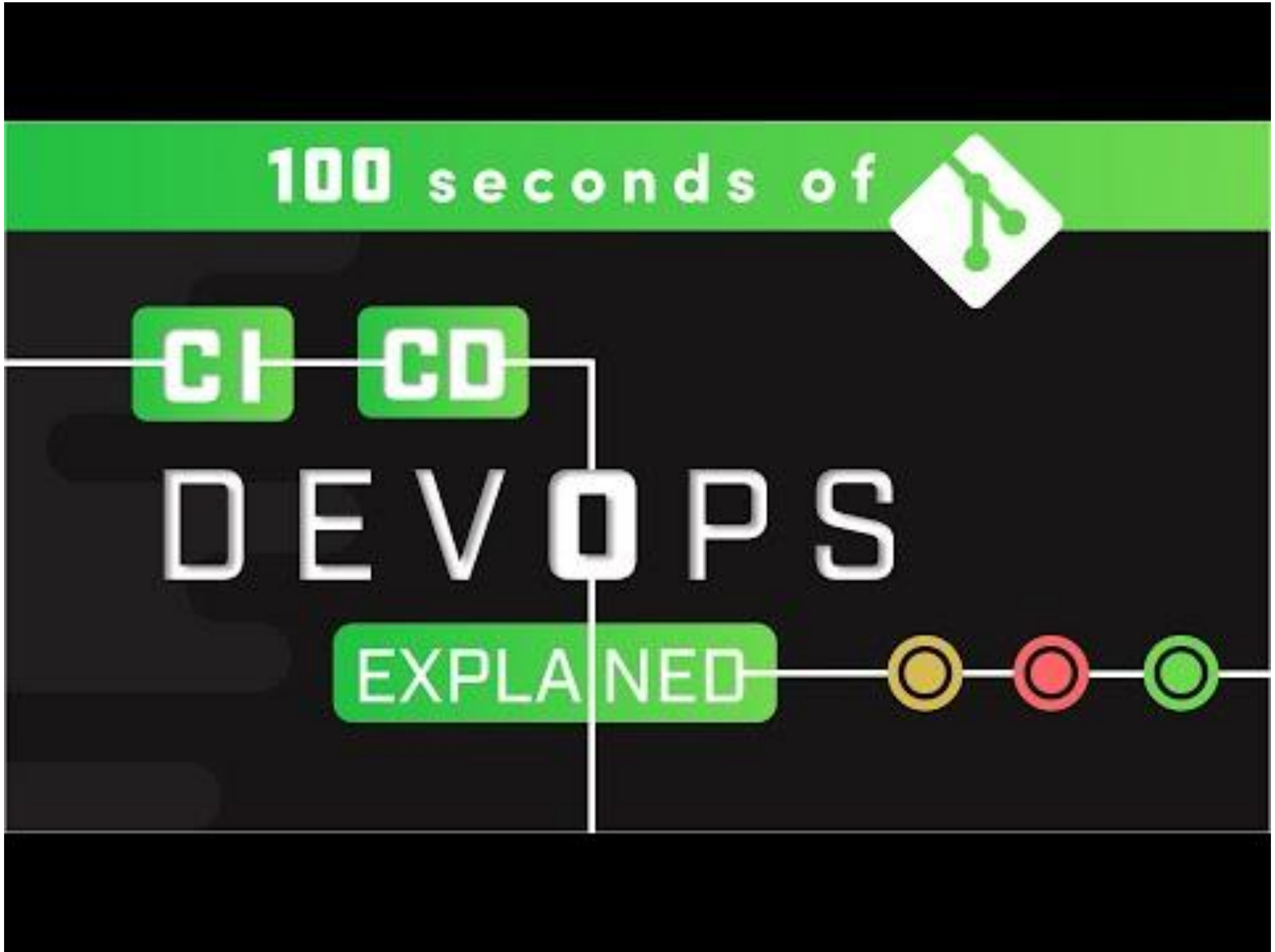
By automating the steps required for deployment, teams reduce issues tha may occur upon deployment and enable more frequent updates.

7.3 DevOps

When both practices are in place,
the resulting process is CI/CD,
which includes the full automation of all steps between code commit to production deployment.

- Implementing CI/CD allows teams to focus on building code and removes the overhead and potential human error in manual, mundane steps.
- CI/CD also makes the process of deploying new code quicker and less risky.
- Deployments then happen more frequently and in smaller increments, helping teams become more agile, more productive, and more confident in their running code.


AGILE DEVELOPMENT
Daily Standup
Product Stage
Backlogs
2 Weeks
Final Product
Process Flow Chart
User Inputs
Continuous Feedback
CONTINUOUS INTEGRATION
Commit
Build + Unit Test + Code Quality
DEV
0316 1121 0819
Code Repository
CI Server
Code Quality Metrics
Repository Manager
Agile DevOps
Daily Scrum
Sprint Review Retrospection
Continuous Integration
Continuous Testing
Continuous Delivery
Sprint Phrases
Continuous Feedback
Continuous Feedback
CONTINUOUS DELIVERY
Product
UAT
QA
Infrastructure as code
Provisioning Tools
Repository Manager
CI Server
Continuous Feedback
CONTINUOUS TESTING
Collaboration
Test Scripts
Test Suite
Auto Ticket Creation
Issue Tracking
CI Server
Testing Metrics
INT
UAT
QA


100 seconds of
CI
CD
DEVOPS
EXPLAINED

Wrapping Up

Bring questions to the class!

- Coding Standards
- Style Guides
- Code Reviews
- Pair Programming
- Collaborative Coding Tools
    - Playgrounds
    - Collaborative Text Editors
- Wikis
- DevOps