GitHub

Lecture #1, 2024

University American College Skopje
School of Computer Science and Information Technology
Course: Programming Languages

Goal of our lecture

Create a robust enterprise architecture to lay the groundwork for our upcoming projects with Java Spring Boot, enabling the development of custom APIs that can be hosted on any cloud service provider.

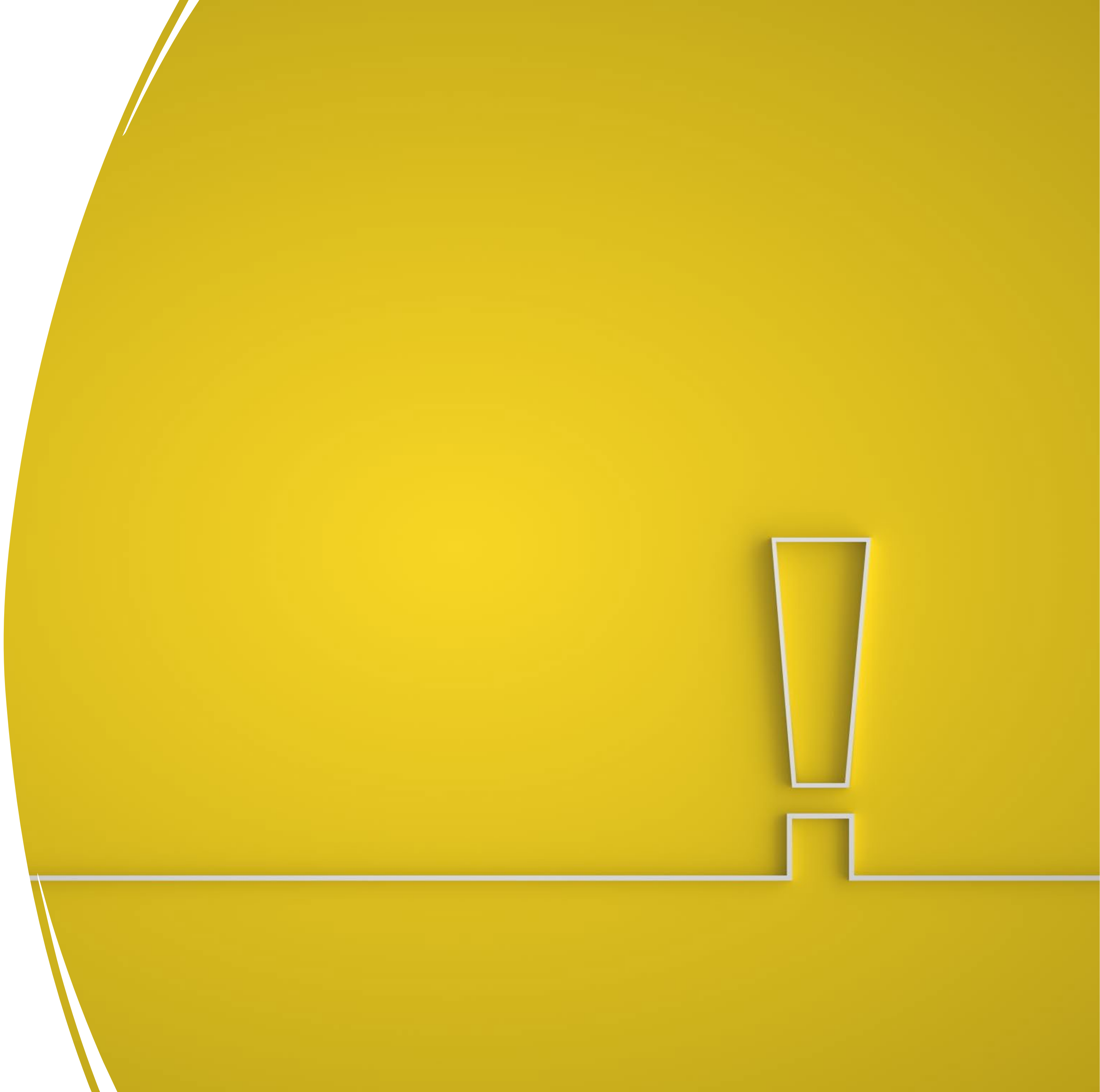
Discover Git: Unlocking the Power of Collaborative Development

Meet Alice and Bob

- Alice and Bob need a platform to collaborate, share code, and track their project's progress.
- Alice Chooses GitHub
- Bob Chooses Dropbox
- As the project evolves, Alice and Bob face challenges that highlight the differences between their chosen platforms.

Version Control

- Alice easily manages her code using Git, a built-in feature of GitHub that tracks changes and keeps a history of all revisions.
- Bob finds it difficult to keep track of changes and ends up with numerous versions of the same file, leading to confusion and wasted time.



Code Reviews and Collaboration

- GitHub's pull request feature allows Alice to easily review, discuss, and approve changes made by other team members.
- Bob struggles to collaborate efficiently, as Dropbox lacks features like code review, comments, and inline discussions.
- Alice takes advantage of GitHub's project management tools and integrations with popular apps, ensuring her team stays organized and efficient.
- Bob misses out on advanced project management tools, limiting his team's productivity and causing potential delays.

Choose GitHub for a better collaboration experience

In the end, Alice's team enjoys a smooth, efficient workflow with GitHub, while Bob's team struggles with the limitations of Dropbox.

Be like Alice for our project ;)

The Development Dilemma

- Imagine working on a project with multiple contributors
- Different features, bug fixes, and deadlines
- How do you manage versions and collaboration?

Why Git? The Game Changer

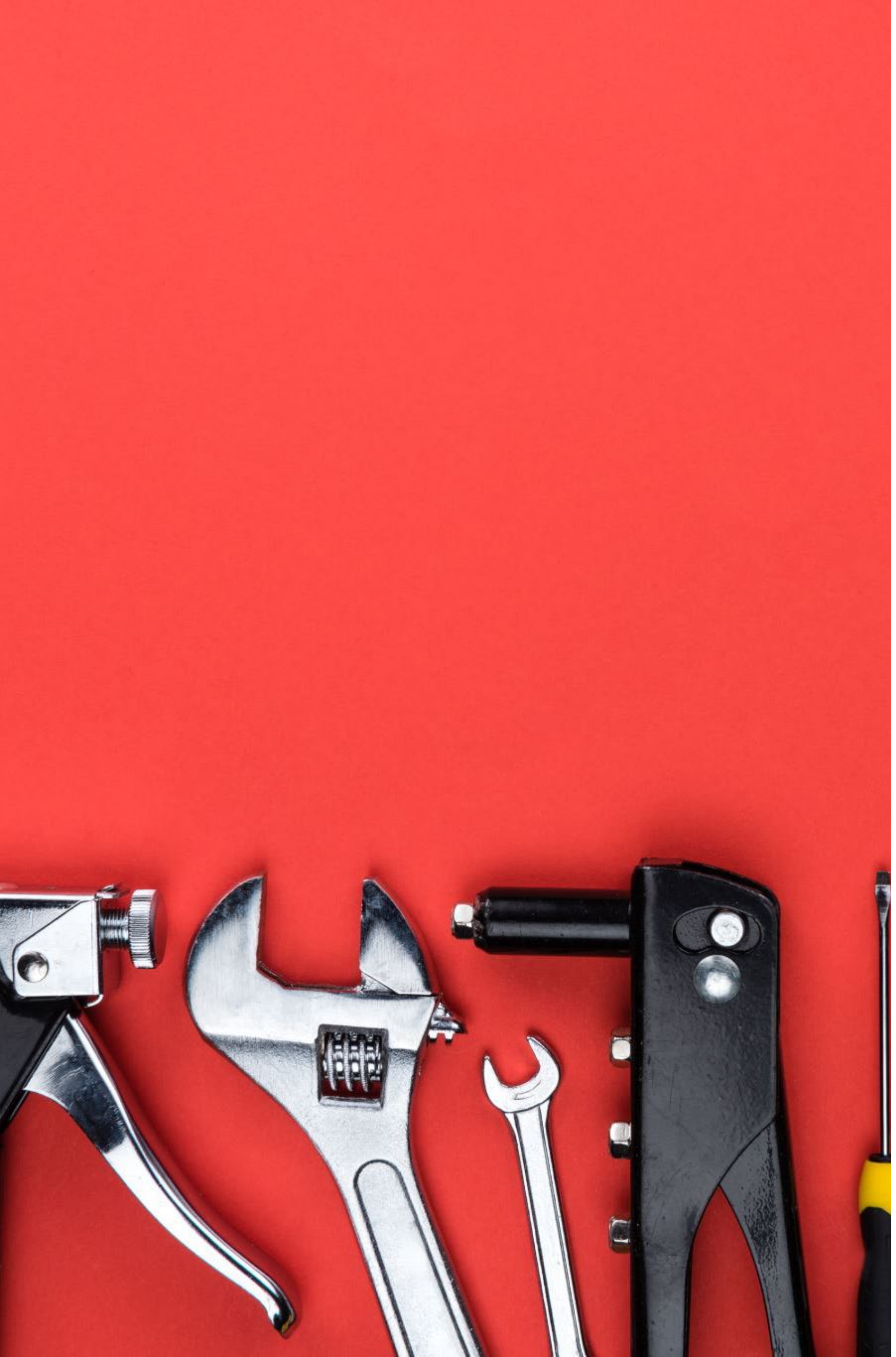
- Distributed Version Control System: No central point of failure
- Fast and efficient: Designed for large projects
- Collaborate with ease: Ideal for open source and team projects
- Non-linear development: Branch and merge effortlessly

A Brief History: From Linux to Everywhere

- Created by Linus Torvalds in 2005
- Developed to manage the Linux Kernel project
- Addressed limitations of other version control systems
- Rapid adoption and growth in the open source community

Git and GitHub: The Dynamic Duo

- **Git**: The powerful version control system
- Track changes, coordinate work, and develop in parallel
- **GitHub**: The social hub for Git
- Host repositories, collaborate, and manage projects with ease



Starting Your Journey: Repositories and Basic Operations

- **Repository**: Your project's home, history, and files
- **Create with confidence**: git init or git clone
- **Add, commit, and push**: The essentials of version control
- **Undo mistakes**: Git has your back with git revert

Collaborate Like a Pro with GitHub

- **Fork**: Own your copy of any repository
- **Branch out**: Develop features or fix bugs in parallel
- **Pull Requests**: Propose, review, and merge changes
- **Resolve conflicts**: Merge with ease and confidence

Real-World Success Stories

- Linux Kernel: The project that started it all
- TensorFlow: Google's open source machine learning library
- Bootstrap: The popular CSS framework
- Countless open source projects powered by Git

Links

- GitHub Guides: https://guides.github.com/
- Git Documentation: https://git-scm.com/docs
- GitHub Learning Lab: https://lab.github.com/