



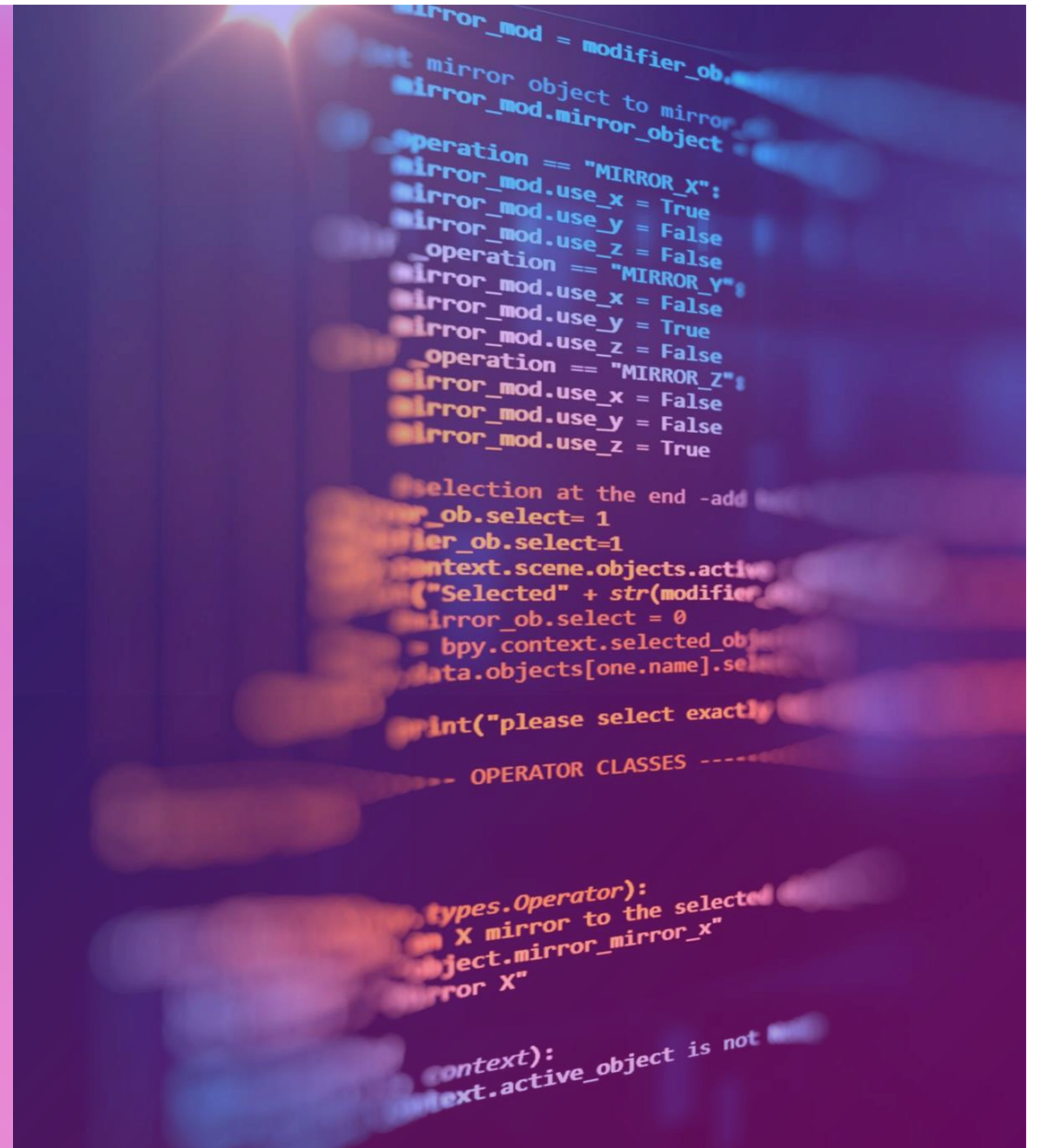
# Leveraging AI in Code Review: Reducing Human Bias and Improving Quality

Team 8: Komeetta

Karoliina I., Oleg I., Eetu O., Miika A.

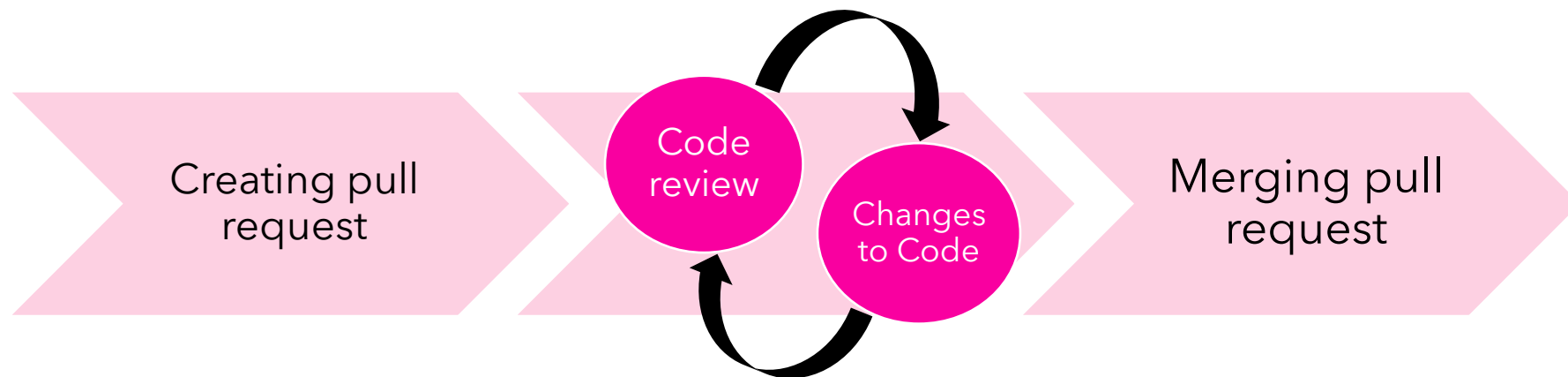
# Agenda

1. What is Code Review?
2. Challenges with Code Review
3. Advantages of AI for Code Review
4. AI vs Traditional Code Reviews
5. Limitations and Concerns
6. Trending Tools
7. Case Studies
8. Productivity Impact
9. Summary

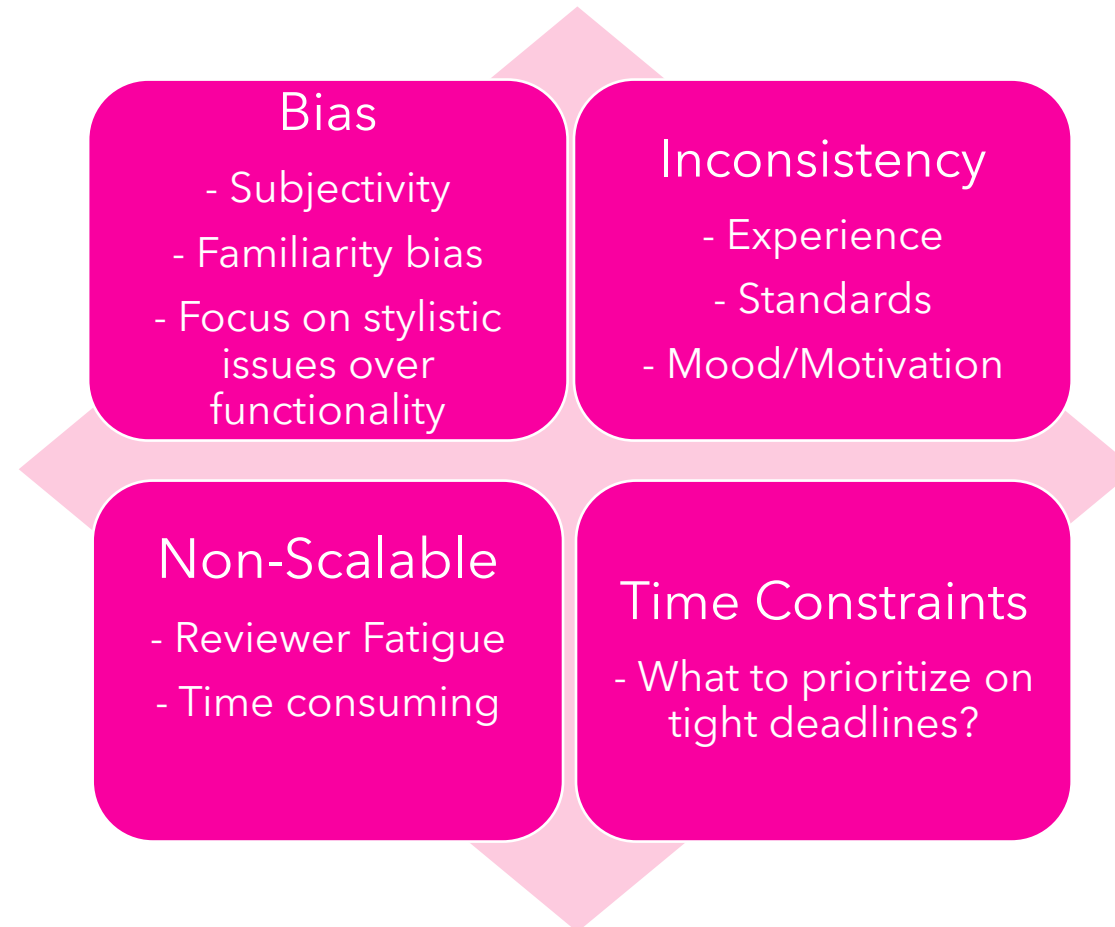


# What is Code Review?

- Code reviews evaluate code to detect bugs and improve quality
- They help developers understand the codebase and ensure compliance with company standards
- Often done as a peer review, using tools like GitHub or GitLab



# Challenges with Code Review



# Advantages of AI for Code Review



## Efficiency & Speed

AI can drastically reduce the time required for code review by quickly analyzing code and identifying issues, allowing teams to focus on development.



## Consistency & Accuracy

AI ensures that code reviews are consistent and accurate, eliminating human biases and fatigue that can lead to oversight, regardless of code complexity.



## Detection of Hard-to-Find Errors

AI excels at identifying subtle errors that manual reviews may miss, providing deeper analysis and suggesting fixes to prevent bugs.



## Enhanced Learning & Skill Development

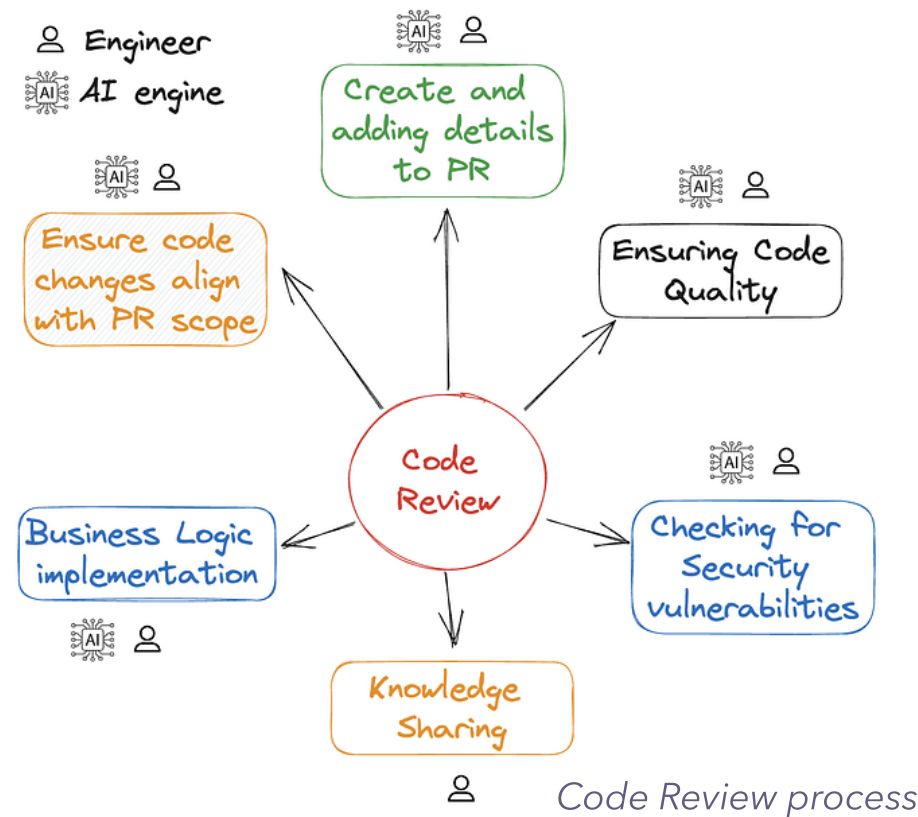
AI offers instant feedback and recommendations, helping developers learn and improve their coding skills in real-time as they work.



# AI vs Traditional Code Reviews

## Traditional Code Review Process

- Focuses on correcting syntax and immediate errors in the code.
- Involves a manual review by a developer, emphasizing line-by-line analysis.
- Can miss broader issues like application behavior under different scenarios.



## AI-Based Code Review Advantages

- Analyzes code as part of a larger system, identifying subtle issues.
- Improves code quality by highlighting potential vulnerabilities and logic flaws.
- Accelerates development cycles, allowing for faster feedback and iteration.

# Limitations and Concerns



## Over-Reliance on AI Tools

There is a risk that developers may become overly dependent on AI tools, compromising their judgment and expertise. This reliance can hinder their understanding of the code, making debugging more challenging.



## Limitations in Understanding Context

AI tools lack the ability to grasp the context and intent behind code, often leading to misinterpretation. This can result in false positives and missed issues that a human reviewer would catch.



## Handling of False Positives and Negatives

False positives create unnecessary work and can erode trust in AI tools, while false negatives can allow critical vulnerabilities to slip through. Continuous learning from mistakes is essential for improvement.



# Trending Tools



## CodeRabbit

- Step-by-step reviews
- Customizable review rules



## PullRequest

- Security reviews
- Human in-the-loop validation



## GitHub CoPilot

- Real-time code suggestions
- Automatically reviews



## Codacy

- CI/CD
- Automated reviews



# Case studies



## GitHub copilot

- **Key finding:** Developers completed tasks 26% faster



## Ericsson Pineapple

- **82,6% of users:** Supports prioritization of code review requests
- **56,5% of users:** Has a positive impact on review lead time
- **Key benefits:** Automates repetitive coding tasks and improves efficiency

# Productivity impact

- Reduction in repetition
- Developers focus on meaningful work
- Shorter task completion and review lead times
- Improved collaboration with clearer prioritization
- Enhanced code quality



# Summary

## Pros of AI:

- Provides instant feedback
- Eliminates human bias
- Identifies hard-to-find logic flaws
- Saves time and reduces repetitive work

## Cons of AI:

- Can lead to false positives and negatives
- May weaken developers' judgement

**Conclusion:** *AI enhances code reviews by reducing bias and improving efficiency, but human oversight remains essential for context and quality control.*

# Questions for the audience

1. If you would not use AI for code review, what would be the biggest reason?
2. Do you think AI-driven code review can fully replace human reviewers in the future? Why or why not?

# Sources:

- <https://usetrag.com/blog/best-code-review-tools>
- <https://www.pullrequest.com/>
- <https://www.hatica.io/blog/ai-code-review-tools/>
- <https://dx.doi.org/10.2139/ssrn.4945566>
- <https://arxiv.org/abs/2102.05916>
- <https://www.turing.com/blog/ai-code-review-improving-software-quality>
- <https://medium.com/aiguys/ai-in-code-reviewing-f7740eb86b27>
- <https://swimm.io/learn/ai-tools-for-developers/ai-code-review-how-it-works-and-3-tools-you-should-know>
- <https://www.sonarsource.com/learn/code-review/#what-is-an-automated-code-review>
- <https://about.gitlab.com/topics/version-control/what-is-code-review/>
- <https://www.atlassian.com/agile/software-development/code-reviews>
- <https://www.thebehavioralscientist.com/glossary/familiarity-bias>
- <https://eon.pubpub.org/pub/3ana9ey0/release/2>