HUSTEF
HUNGARIAN SOFTWARE TESTING FORUM

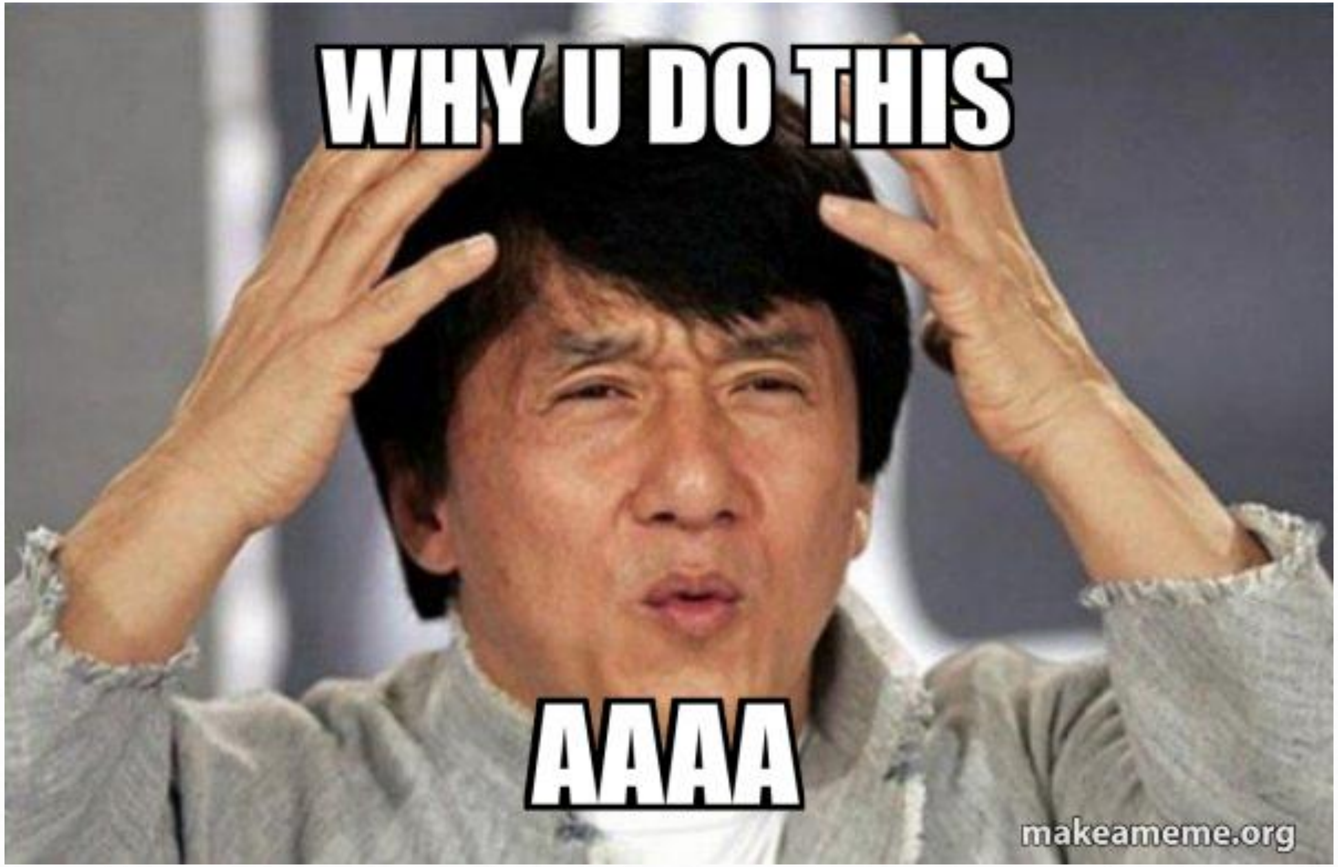
Test Craftsmanship: **Clean Code Practices** for Test Automation

Manoj Kumar Kumar
Practice Lead - QE&A - Cognizant Australia

Quick show of Hands?

- Is your test automation prepared for major application changes without extensive rework?
- When did you last review your test code for duplications or inefficiencies?
- Does your testing practices contribute to code cleanliness or add complexity?
- Have outdated tests caused false results?

How Intimidating?



Optimizing Test Automation

Test Authoring

Crafting precise and effective tests from the outset

Test Maintenance

Ensuring tests remain relevant and up-to-date

Test Review

Assessing and refining test quality

Test Execution

Running tests efficiently and analyzing results

Test → Test Code/Script*

Test Authoring

The foundation of test automation, where quality tests are crafted.

Strategies

Descriptive and Meaningful Test Case Names (SRP)

Emphasize clear and descriptive names for test cases to convey intent.

Fast Feedback Loops

Stress the importance of quick feedback during test authoring for rapid iterations.

Small Functions and Modularization

Promote writing short, focused functions and modularizing test code.

The Campsite Rule

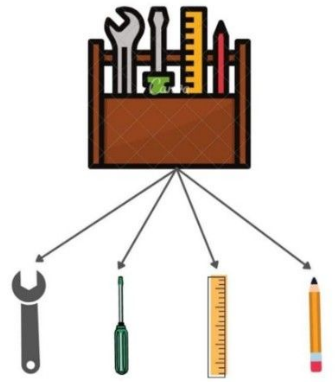
Emphasize the practice of leaving test code in a better state than found.

Use TDD

Encourage Test-Driven Development (TDD) for quality test case design.

Descriptive and Meaningful Test Case Names (SRP)

Use human-readable titles
Be unique
Be concise
Make test cases reusable

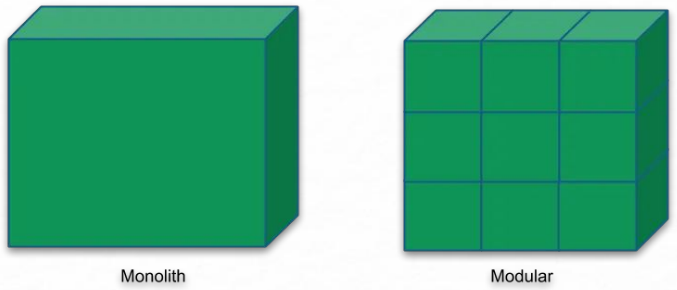


"A class should only have one reason to change"

"A class or method should have only a single responsibility"

Small Functions and Modularization

Promote writing short, focused functions and modularizing test code.

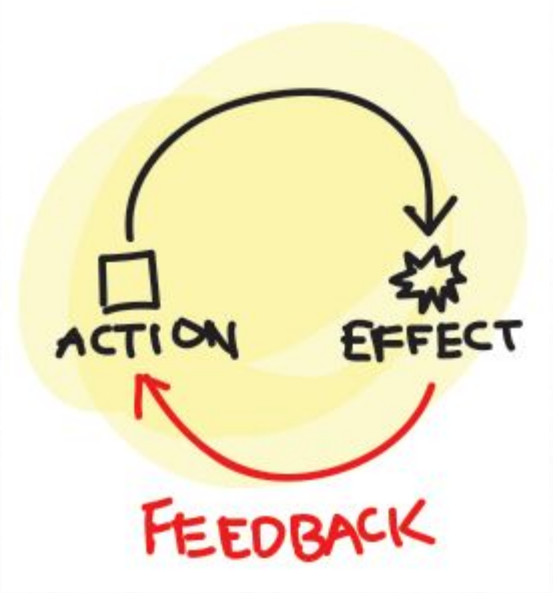


Fast Feedback Loops

Write tests to enable fast feedback!

Fast Feedback Loops are like the accelerators of our development process.

Use models? - Test Pyramid? Trophy?



The Campsite Rule

After camping, sweep the leaves, douse the fire, and make it pristine for the next adventurer!

↓

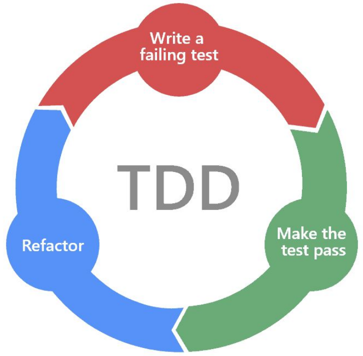
Always leave a codebase in a better state than when you found it.

Use Test Driven Development (TDD)

Ensuring that code is thoroughly tested and reliable from the outset.

Encouraging modular and maintainable code.

Last step of TDD is "refactoring" to reduce duplication, aligning with the Campsite Rule.



https://unsplash.com/photos/SqjhKY9877M

Test Maintenance

Ensuring test automation remains relevant and effective over time.

Strategies

Version Control (Trunk-Based Development)

Discuss version control systems and trunk-based development for managing test code efficiently.

Continuous Integration (CI/CD)

Mention CI/CD practices for automated test maintenance.

Refactoring and DRY

Highlight refactoring to reduce duplication (DRY principle) during test maintenance.

Design Patterns

Introduce design patterns for creating maintainable tests, such as Page Object and others.

Design Patterns

Page Object Model

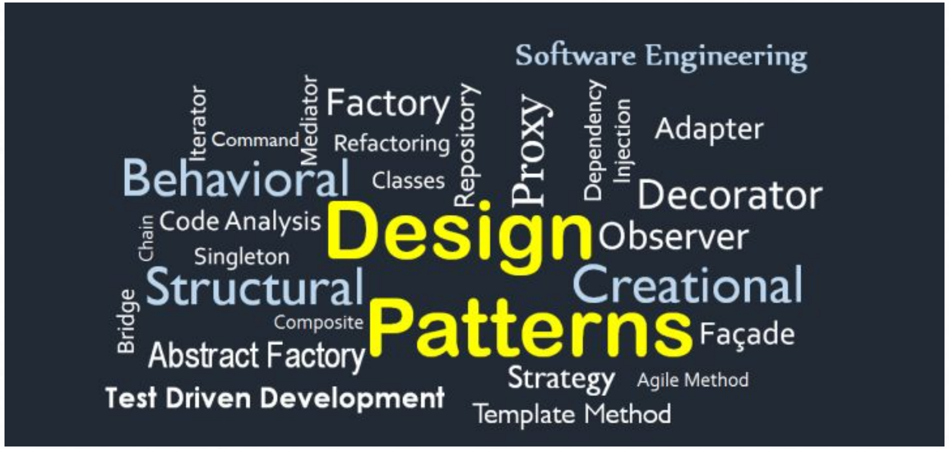
Business-Layer Page Object

Factory Design Pattern

Facade Design Pattern

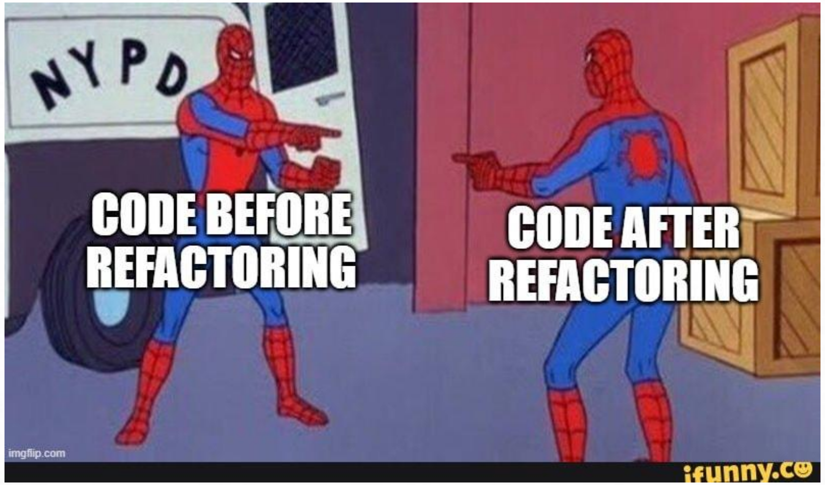
Singleton Design Pattern

Fluent Page Object Model



Refactoring and DRY

Refactoring to reduce duplication (DRY principle)



Refactor vs Rewrite

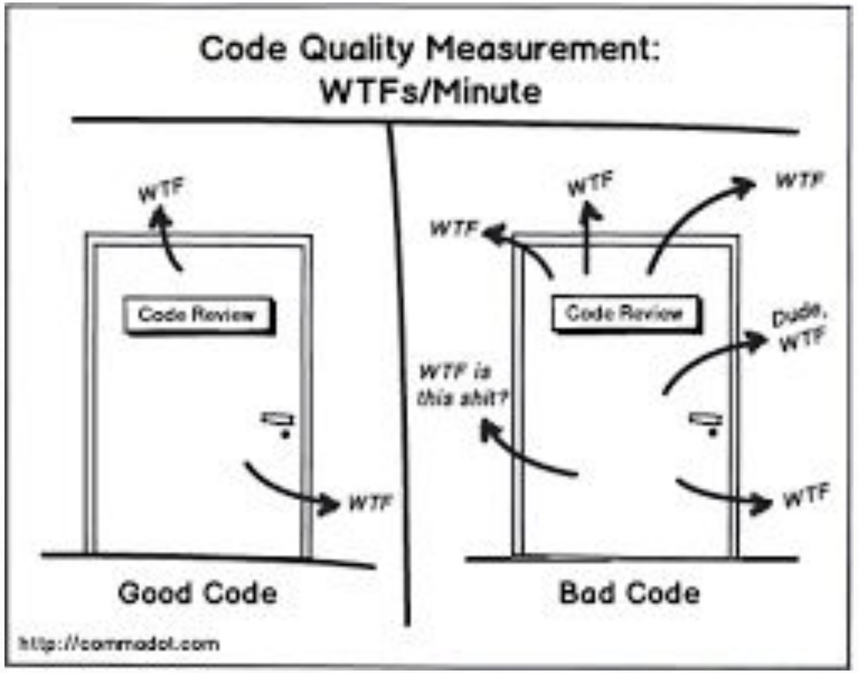
Refactor when you can improve the code's quality and maintainability without changing its core functionality.

Rewrite when the existing code is unmaintainable, outdated, or incompatible with modern requirements and technologies.

Test Review

Ensuring the integrity and effectiveness of test scripts

Application Code Vs Code Quality



Strategies

Code Review

Emphasize the need for peer code reviews for test scripts.

Code Smells

Discuss recognizing and addressing code smells during test reviews

Law of Demeter (LoD)

Minimize coupling between parts of test code.

Effective Comments and Documentation

Encourage clear comments that explain the 'why' behind the code

Code Smells

Code Smells are patterns of code that suggest there might be a problem, that there might be a better way of writing the code or that more design perhaps should go into it.

Int

Use Code linting tools…

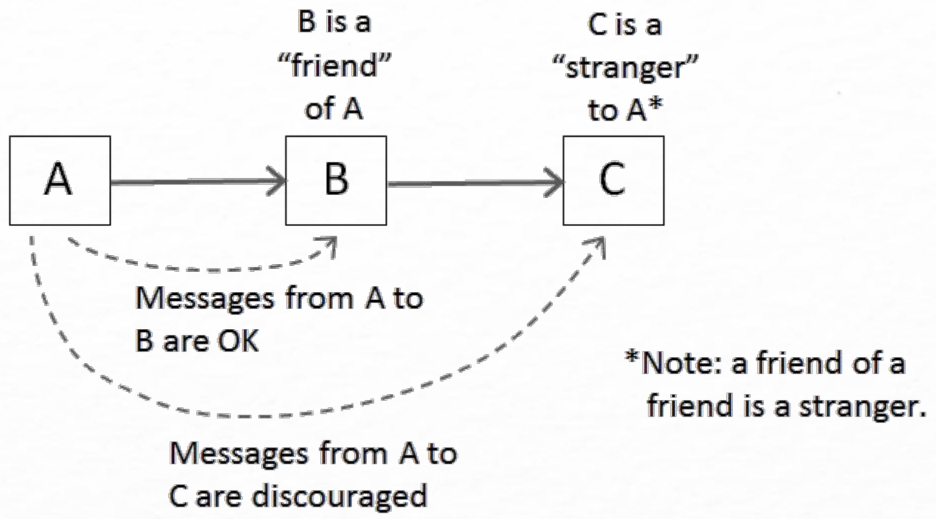
Law of Demeter (LoD)

“Only talk to your immediate friends.”

and

“Don’t talk to strangers.”

Page Object should encapsulate the details of the web page or component it represents


B is a “friend” of A
C is a “stranger” to A*
A
B
C
Messages from A to B are OK
Messages from A to C are discouraged
*Note: a friend of a friend is a stranger.

Test Execution

Running tests and deriving actionable insights

Strategies

Parallel Execution

Explain how parallel execution can save time during test execution

Continuous Monitoring

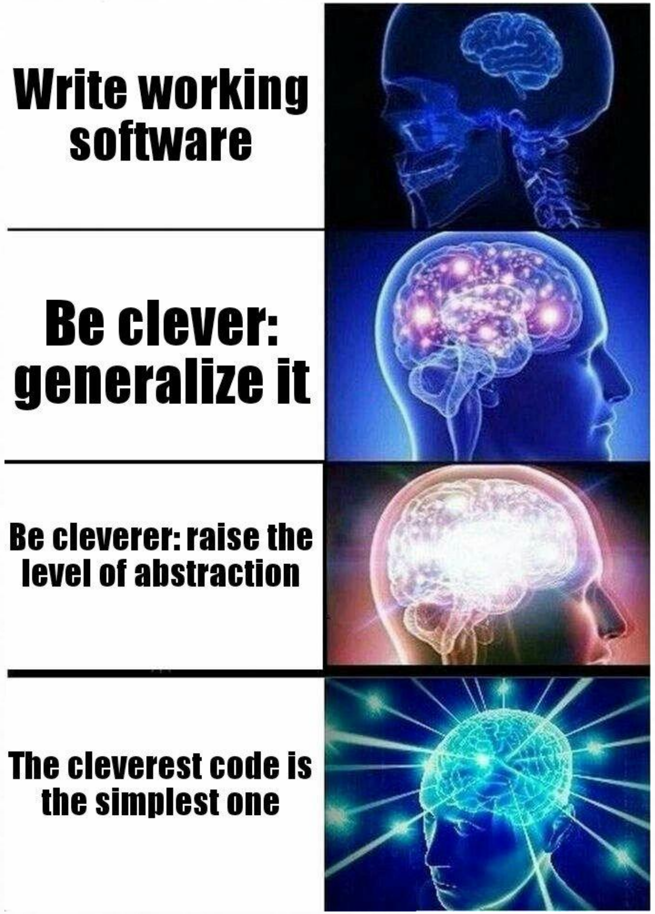
Discuss monitoring test execution in CI/CD pipelines

Test Reporting and Logging

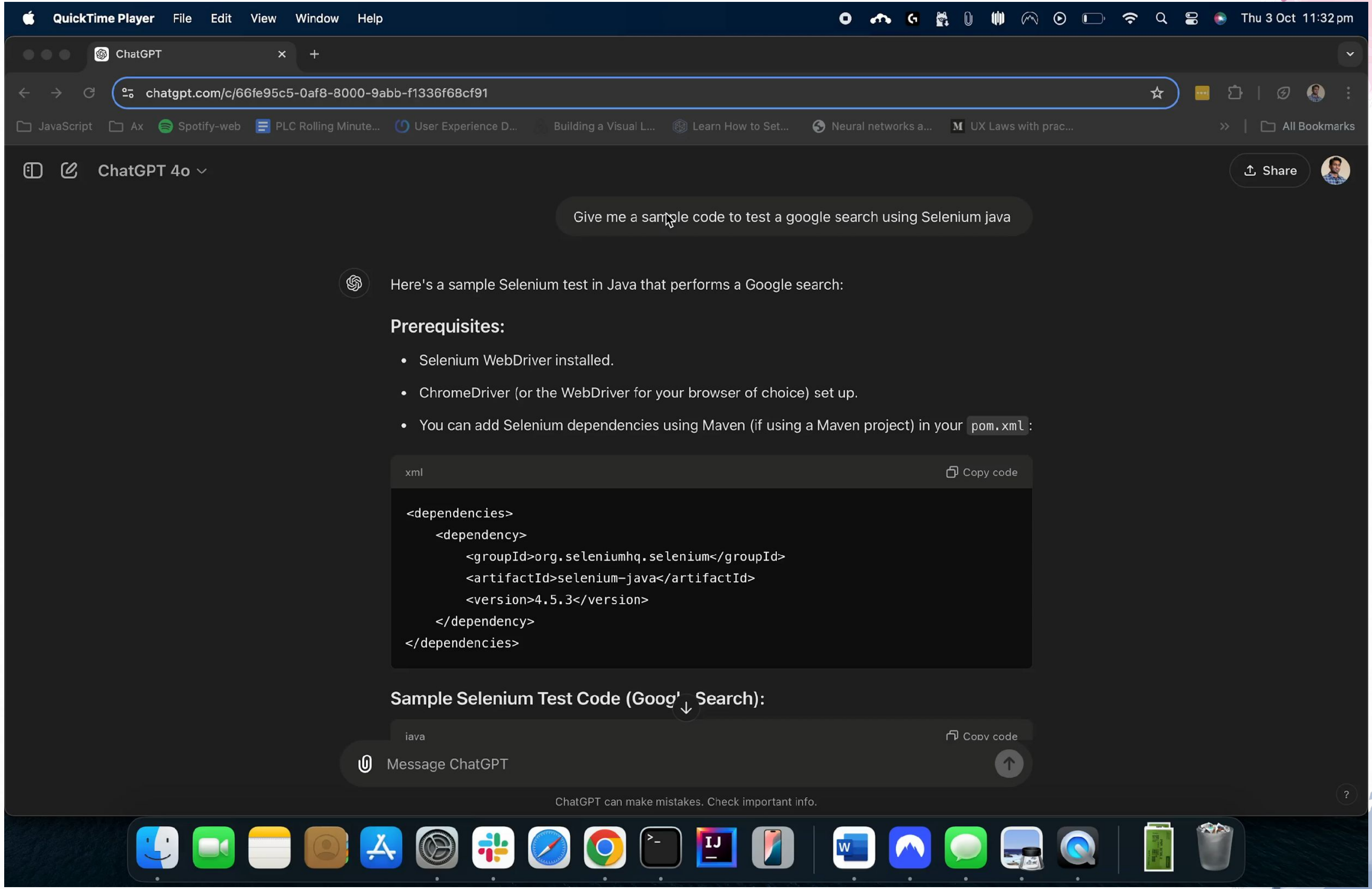
Highlight tools and practices for reporting and analyzing test results

Avoid Cleverness

The cleverest solution to a problem is actually the simplest one that just works


Write working software
Be clever: generalize it
Be cleverer: raise the level of abstraction
The cleverest code is the simplest one

Let’s Ask AI



Give me a sample code to test a google search using Selenium java

Here's a sample Selenium test in Java that performs a Google search:

Prerequisites:

- Selenium WebDriver installed.
- ChromeDriver (or the WebDriver for your browser of choice) set up.
- You can add Selenium dependencies using Maven (if using a Maven project) in your `pom.xml`:

xml Copy code

```xml
<dependencies>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.5.3</version>
    </dependency>
</dependencies>
```

Sample Selenium Test Code (Goog Search):

java Copy code

AI's Impact on Test Automation: From Authoring to Execution

AI: Test Authoring

Automated code suggestions (e.g., Copilot, ChatGPT)
Scenario-based test generation from natural language
Enhanced collaboration across teams

AI:Test Maintenance

Self-healing test locators for resilient automation
Predictive maintenance for proactive fixes

AI: Test Review

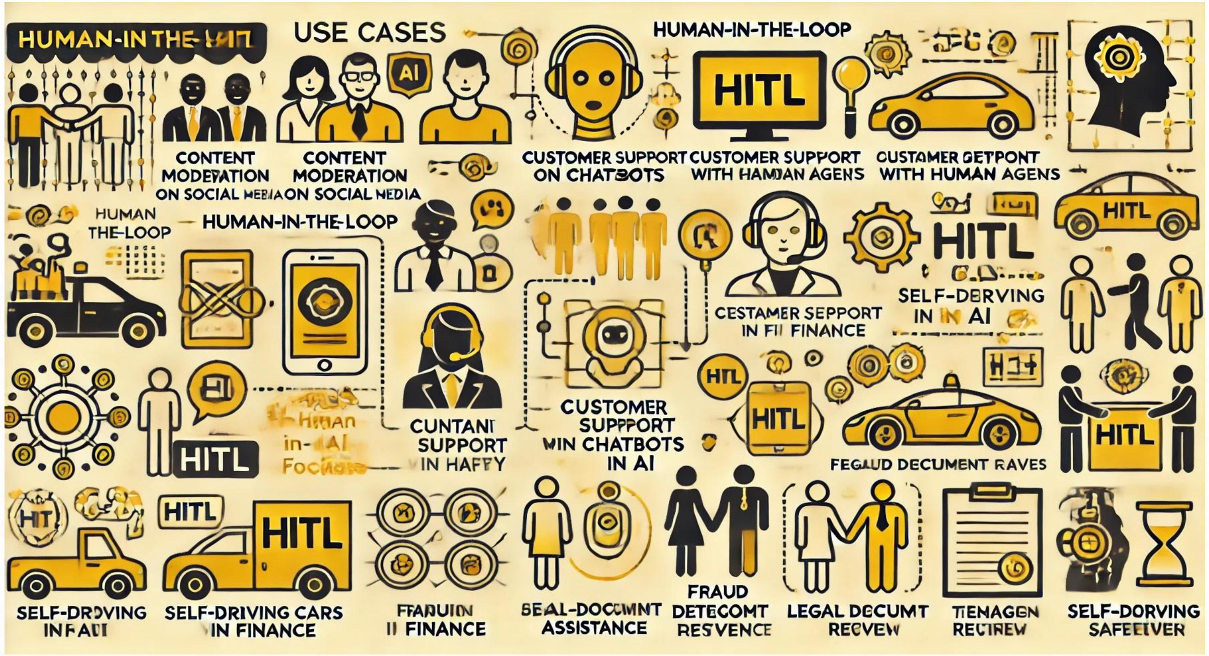
First-level AI code reviews (pattern recognition, clean code enforcement)
Automated quality metrics and flaky test identification

AI: Test Execution

Intelligent test selection based on code changes
Autonomous testing for exploratory coverage

Hallucinations and human-in-the-loop

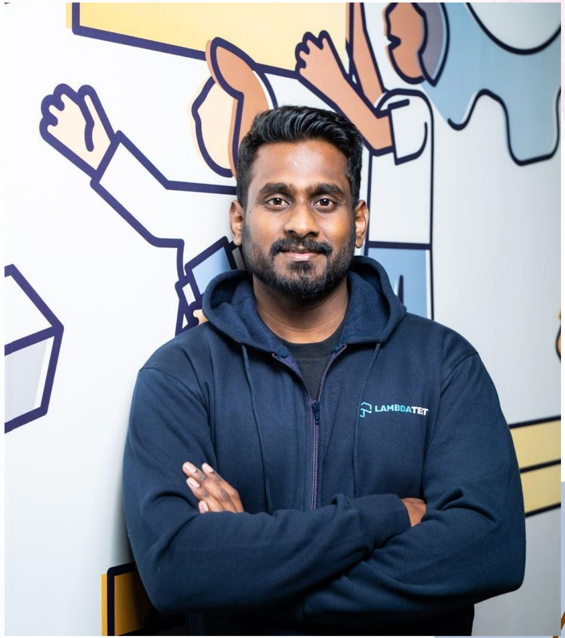
- AI hallucinations refer to instances where AI systems, particularly language models, generate incorrect or nonsensical information.



Together, let's craft a future where the code we write is efficient and enduring. Because, In the world of Test Craftsmanship, we understand that test code is every bit as good as production code.

About Me

- Practice Lead - QE&A - Cognizant Australia
- Distinguished Speaker by ACM
- FOSS Fan
  - *Project Leadership & Committer - Selenium.dev*
  - *Software Freedom Conservancy*
  - *Committer - Appium.io*
  - *K8s-Selenium*
  - *O11Y-Selenium*
- DevOps & Accessibility Advocate
- MotoGP and BasketBall fan



THANKS