MANAGING REGRESSION

Subject: Software Engineering: Software Processes

Presented by: Jenil Arvindbhai Paladiya

Matriculation Number : 4243558

Index

1. Managing Regression in Software Development
2. What is Regression?
3. Why Regression Happens
4. The Cost of Regression Bugs
5. Regression Testing: The First Line of Defense
6. Building a Regression Test Suite
7. Automating Regression Tests
8. CI/CD and Regression
9. Strategies for Preventing Regression
10. Conclusion: Embrace Regression Management

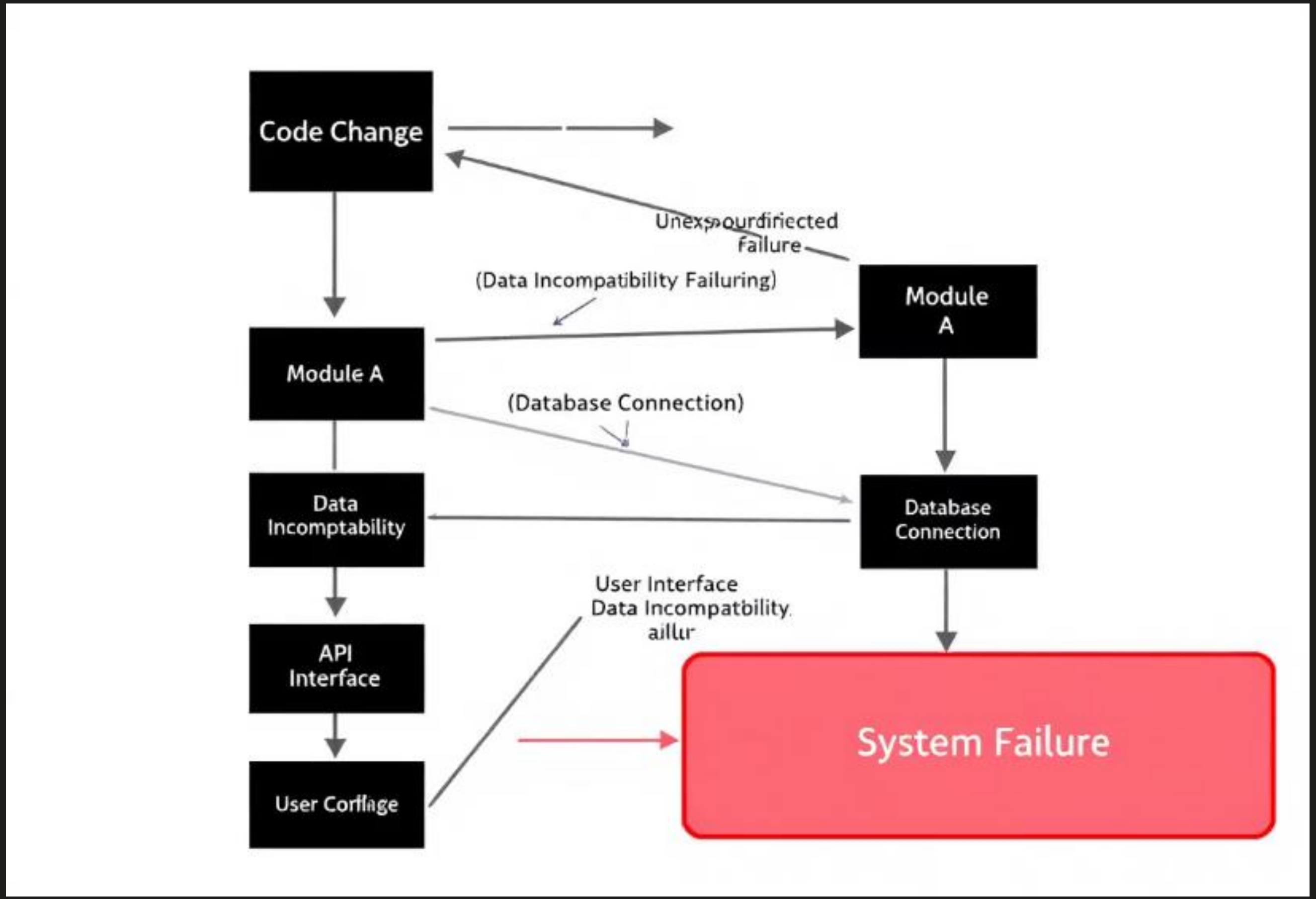
What is Regression?

Definition

Regression occurs when a new change introduces bugs in previously working functionality.

Example

Updating a WordPress plugin breaks the checkout page on an e-commerce site.



It's crucial to catch these regressions early in the development cycle to avoid costly fixes later.

Why Regression Happens

Code Changes

Changes affect unexpected areas of the codebase.

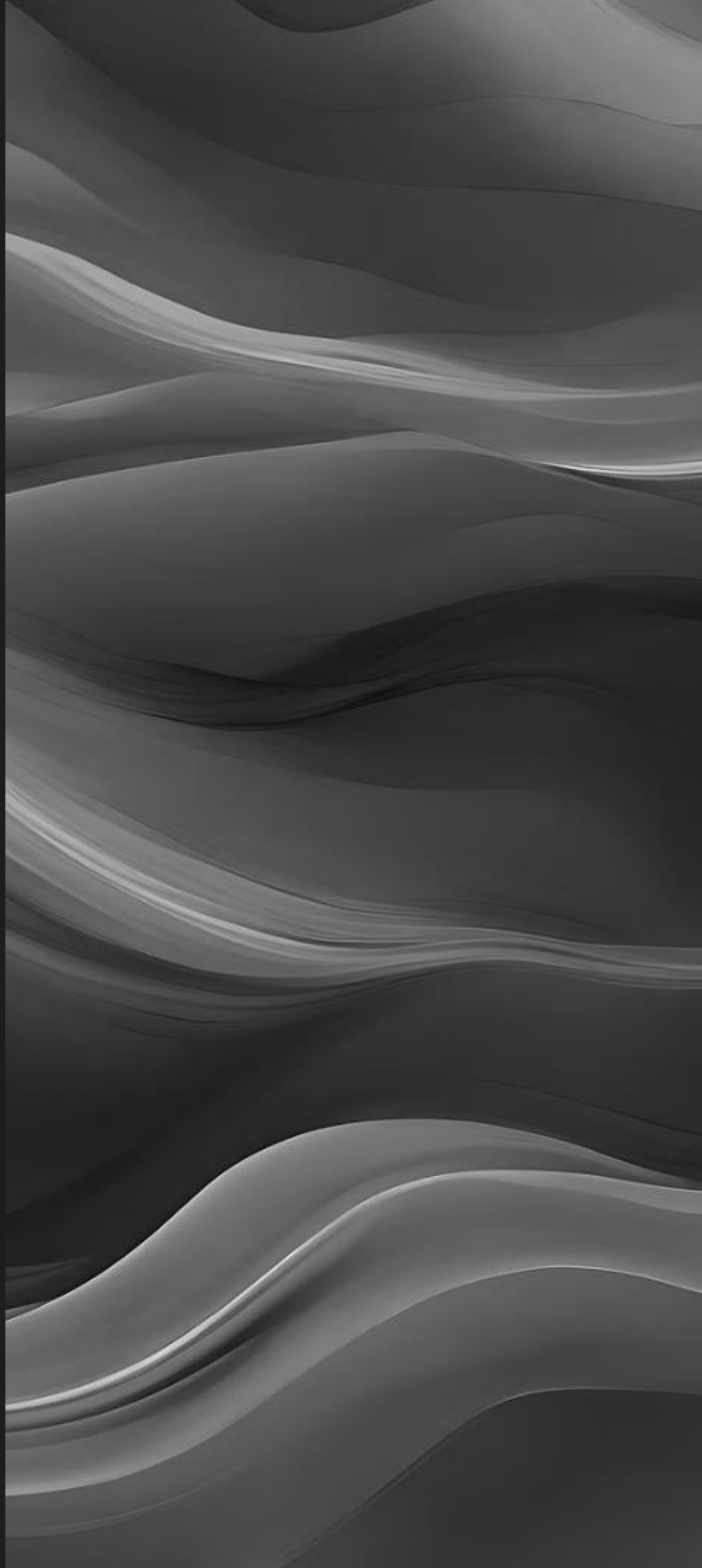
Lack of Testing

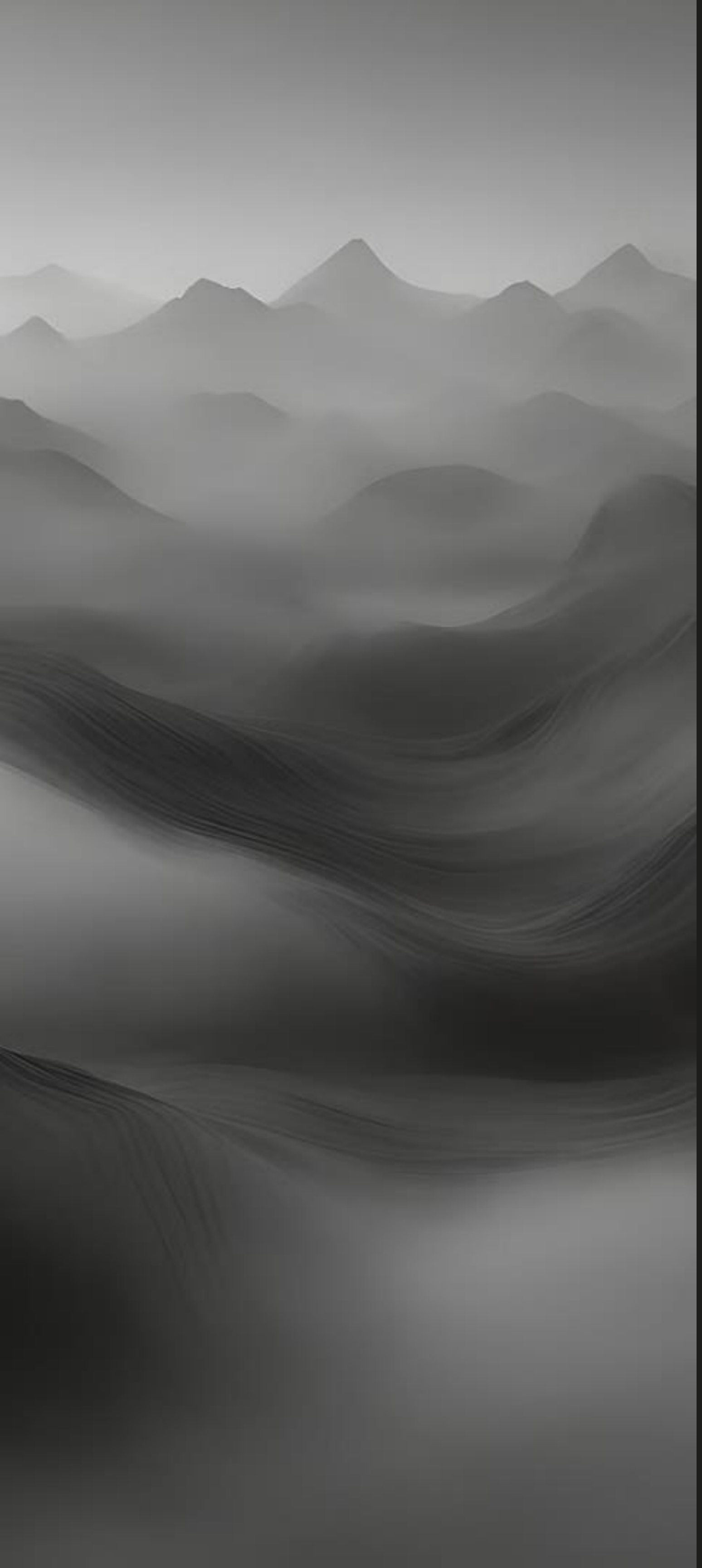
Inadequate testing leads to missed regressions.

Tight Deadlines

Time constraints result in development shortcuts.

Poor communication, technical debt, and complexity all contribute to regression issues.





The Cost of Regression Bugs

Financial Impact

Revenue loss and expenses for bug fixes.

Reputation Damage

Loss of user trust due to poor software quality.

Time Wasted

Debugging and fixing issues delays releases.

The 2017 Equifax breach, due to a regression bug, cost over $1.4 billion.

Regression Testing: The First Line of Defense

1. ### Definition

   Re-running tests after code changes.

2. ### Goal

   Verify existing functionality still works correctly.

3. ### Methods

   Manual vs. Automated testing strategies.

The primary focus is to identify regressions early in the development cycle.

Building a Regression Test Suite

Critical Functionalities

Identify and prioritize tests for core features.

Risk and Impact

Focus on high-risk areas that impact users.

Test Coverage

Aim for broad coverage of scenarios and edge cases.

Example: E-commerce suite: login, search, add to cart, checkout.

Automating Regression Tests

1 Benefits

Speed, repeatability, and improved efficiency.

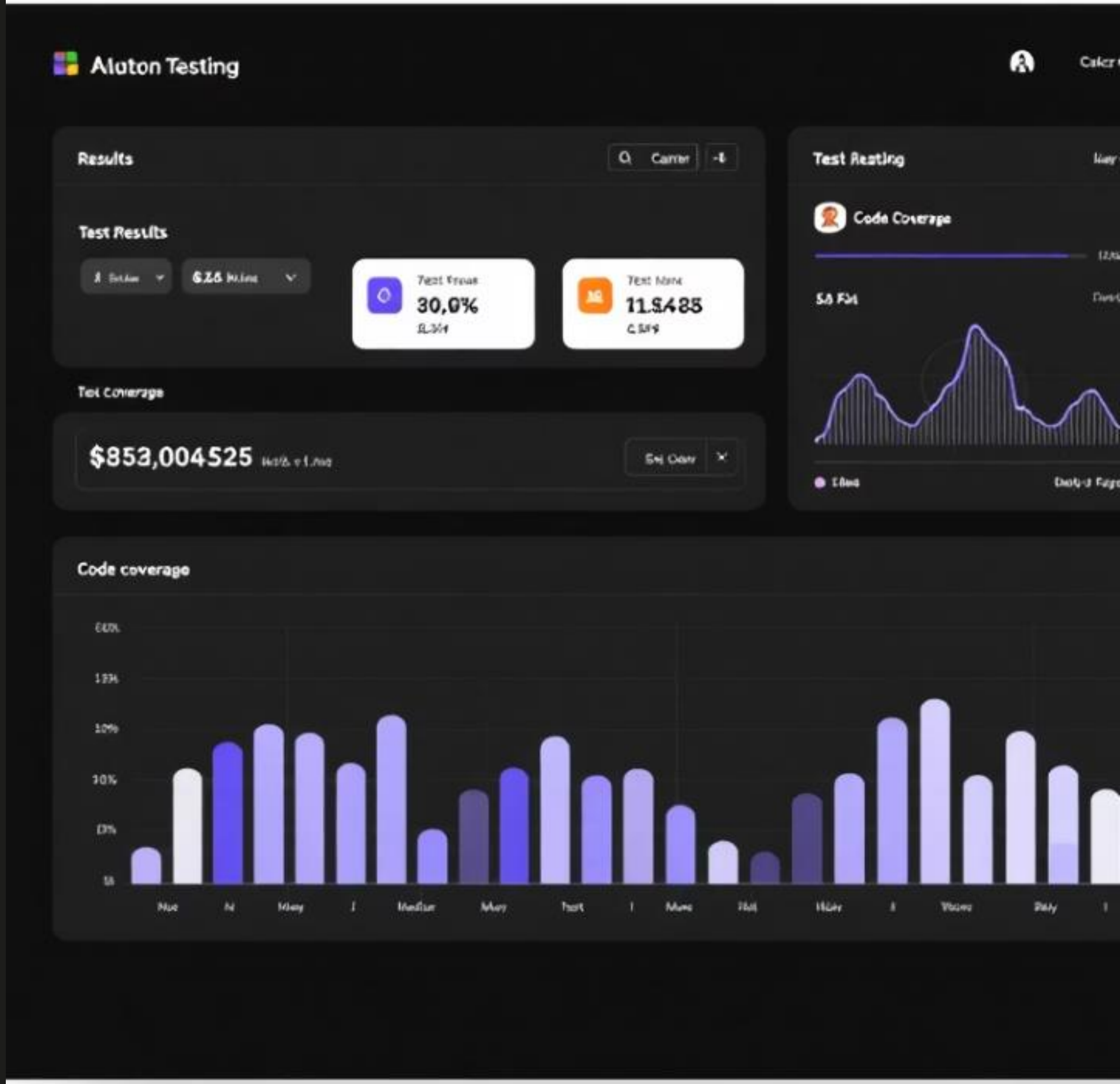
2 Tools

Selenium, Cypress, JUnit, TestNG.

3 CI/CD

Continuous Integration/Continuous Deployment.

Automated tests can be set to run on every code commit.



CI/CD and Regression

Automated Tests

Run tests at each build in the pipeline.

Immediate Feedback

Detect regressions early in the process.

Catch Issues

Prevent bugs from reaching production.

Tools like Jenkins, GitLab CI, and CircleCI facilitate faster development cycles.

Strategies for Preventing Regression

1

Code Reviews

Peer reviews by other developers.

2

Unit Testing

Test individual components.

3

Integration Testing

Test between modules.

Clear requirements, specifications, and detailed documentation are crucial for prevention.

Conclusion: Embrace Regression Management

100%

Regression is inevitable.

High

Invest in testing and automation.

Continuous

Improve processes regularly.

Effective management leads to high-quality software and user satisfaction.

Thank You

Presented by Jenil Arvindbhai Paladiya.