Shift Left: Implementing Performance Testing

Enhancing Performance with a "Shift Left" approach.

Shift Left: Introduction

- Shift Left - Performance Testing.
- Focus on minimising:
  - Bugs
  - Defects
  - Other potential issues.
- The broader objectives:
  - Reducing overall project costs
  - Enhancing the end-user experience
  - Improving product quality and reliability
  - Expediting time to market.

How did we get here?

- Why is there no pre-defined release cycle or schedule for the application being built?

- Why has Performance Testing only been conducted on demand and at such a late stage in the development process?

- What are the types of performance issues that've been encountered due to the current approach?

- What are the performance issues that were discovered late in the process, and what steps, if any, have been taken to address similar issues?

Creating a Performance Testing Strategy...

- How do we Shift Left our Performance Testing activities?
  - Define SRLC
  - Planning
  - Clear Objectives
  - A Comprehensive Scope
- How can we measure our testing efforts?
  - KPI's
  - Baseline and Target (NFR's)
  - Feedback
  - Cost Efficiency

Introducing Collaborative Testing...

- No "i" in Team!

- The importance of cross-functional collaboration.
  - Sharing Expertise
  - Holistic Understanding
  - Team-Based KPI's
  - Effective Communication
  - Collaborative Scenario Design

Implement CI/CD and Automation

- What are the advantages of automation in this context?
    - Automation Tools
    - Consistency (no human errors)
    - Frequent Testing
    - Immediate feedback (metrics) and Regression Testing etc.
    - Cost Effective
- What types of tests can we automate?
    - Load
    - Stress
    - Smoke

In Conclusion…

- My proposal for the team to implement Shift Left is as follows:

  - Testing Strategy
    - Thorough planning with clear and concise objectives
  - Collaborative Testing
    - Holistic understanding with clear communication and shared expertise
  - CI/CD - Automation
    - Cost effective frequent and consistent automated testing

Thank You For Listening!