



AI Driven SDLC Transformation

From Current State to Future State with AI Tools

Aug 25, 2025



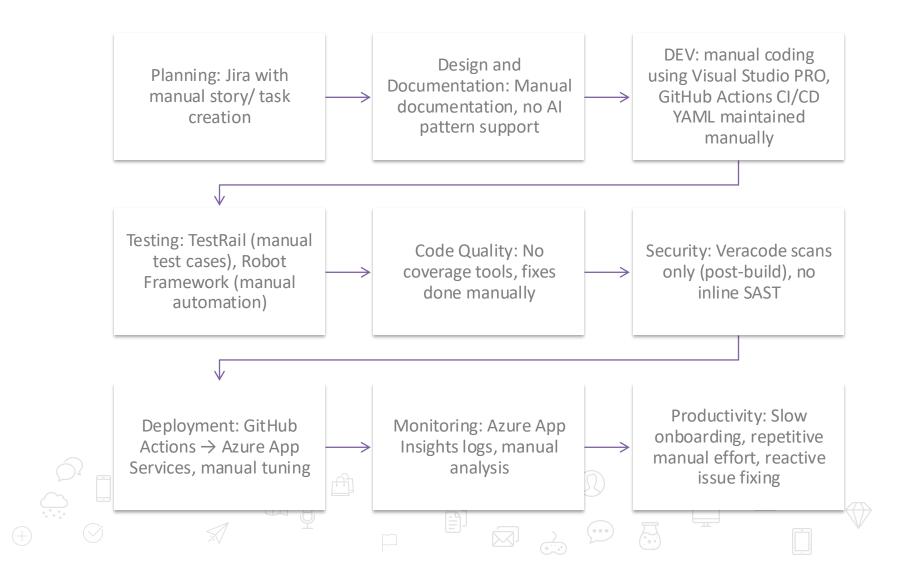
Agenda

- Current State and Pain Points
- SDLC Workflow
- Future State using Al tools
- Al Driven SDLC Workflow
- Estimated Time Savings with GitHub Copilot
- Existing Usage of GitHub Copilot Development Team
- Best Practices to Maximize Savings with Copilot
- Workflow Steps with Copilot
- Example: Gherkin User Story to Tests



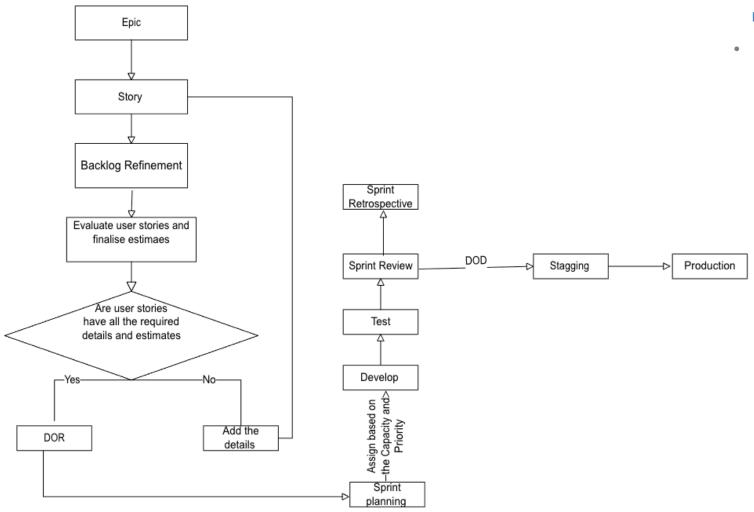


Current State and Challenges





SDLC Workflow



PR Approval process

DEV creates a PR →
 Gets approval
 from developers →
 QA reviews and
 approves → PR gets
 merged to the release
 branch

Release Plan

- Code Freeze: We follow a 2-week sprint cycle, with code freeze occurring every alternate Friday
- Rapid release: Biweekly deployments occur every alternate Tuesday (QA) and Wednesday (Staging → Production)
- Hotfix: Critical production issues are addressed immediately outside the regular cycle

Future State using AI tools

Requirements: Jira + Copilot/ Atlassian AI → auto-generate stories, acceptance criteria

Design: Copilot Chat → suggest patterns,
PlantUML + AI diagrams

DEV: Copilot in Visual Studio Pro → AI pair programming & secure code suggestions Peer Reviews: Copilot for PRs → summaries, test suggestions, secure fixes

Testing: Copilotgenerated tests, Alenhanced Robot Framework automation Code Quality: Coverlet (.NET) + Jest/ Istanbul (Angular) + SonarQube gates Security: Veracode + CodeQL + Semgrep + Gitleaks for PR inline scanning

Deployment: Copilotgenerated GitHub Actions YAML, Aloptimized pipelines

Monitoring: App Insights + AI anomaly detection + Copilot log summaries Performance: k6 load testing + Lighthouse CI (Angular) with AI reports Outcome: Faster delivery, proactive quality & security, higher dev productivity











































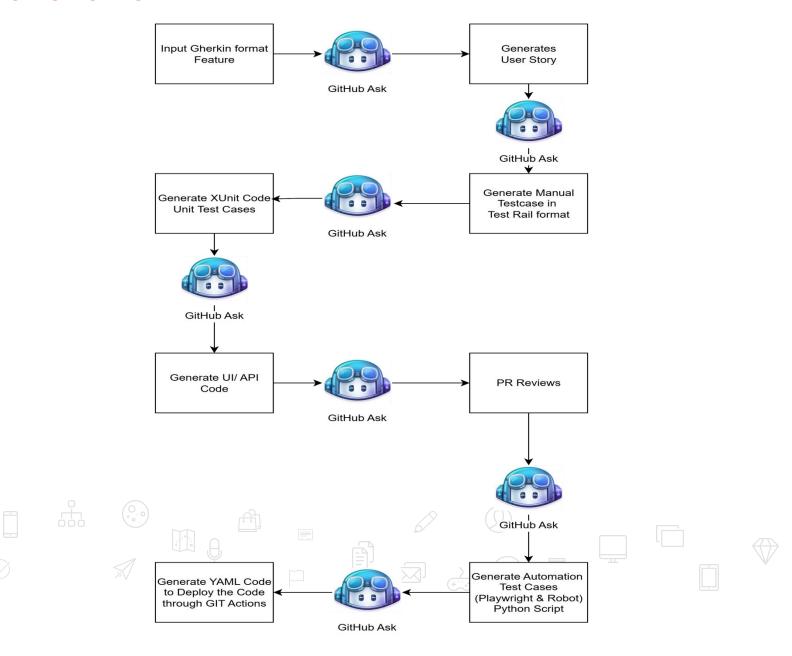








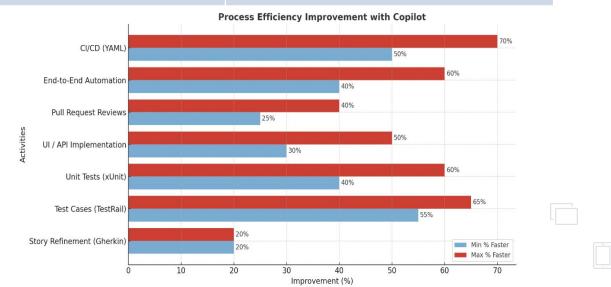
AI Driven SDLC Workflow





Estimated Time Savings with GitHub Copilot

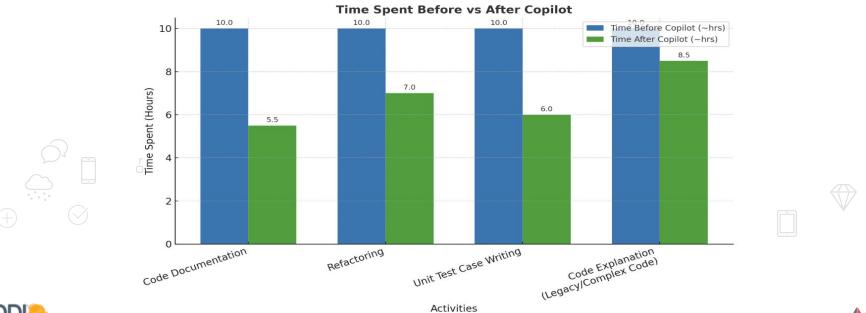
Activity	Productivity
Story Refinement (Gherkin)	~20% Faster
Test Cases (TestRail)	~55–65% Faster
Unit Tests (xUnit)	~40–60% Faster
UI / API Implementation	~30–50% Faster
Pull Request Reviews	~25–40% Faster
End-to-End Automation	~40–60% Faster
CI/CD (YAML)	~50–70% Faster





Existing Usage of GitHub Copilot – Development Team

Activity	Time Spent Prior Copilot	Time Spent After Copilot	Time Saved
Code Documentation	~10 hrs	~5 to 6 hrs	40 to 50% Faster
Code Refactoring	~10 hrs	~6 to 8 hrs	20 to 40% Faster
Unit Test Case	~10 hrs	~5 to 7 hrs	30 to 50% Faster
Code Explanation (Legacy/Complex Code)	~10 hrs	~8 to 9 hrs	10 to 20% Faster





Best Practices to Maximize Savings with Copilot

Activity	Best Practices
Standardize Prompts	Encourage consistent use of Copilot prompts across the team
Keep Humans in the Loop	Ensure design, security and domain logic decisions remain human-driven
Reusable Prompts	Maintain a /prompts folderwith shared, team-approved templates
Track Key Outcomes	Measure KOIs such as PRcycle time, review quality, test coverage, andpipeline reliability























Workflow steps with Copilot

Activity	Copilot Usage
Refine Gherkins	Convert into User Story
Generate Manual Test Cases	Document in TestRail
Write Unit Tests	Implement with XUnit
Develop Features	Implement UI / API Code
Pull Request Reviews	Validate quality and standards
Automation Testing	Execute with Playwright /Robot
CI/CD Pipeline	Configure in GitHub Actions (YAML)





Example: Gherkin User Story to Tests

Example:

- **User Story**
 - As a customer, I want to log in so I can access my dashboard
- Acceptance Criteria (Gherkin)
 - Given I am on the login page
 - When I enter valid credentials and click Login
 - Then I should see the dashboard with a welcome message

Using AI:

- Copilot: Jira Story with Acceptance Criteria
- TestRail case: preconditions, steps, expected Result
- Robot Framework script for login test
- PlantUML formatted design diagram script
- XUnit test case generation
- UI & API generation
- PR reviews
- Git Hub actions

































Thank you



