UC SANTA BARBARA
Information Technology

AI-Driven Unit Testing for Drupal Development: Automation with WebOps & CI/CD Pipelines

Sanil Katula
Supervisor - Elda Schaffield

UC SANTA BARBARA
Student Services
Information Systems
Information Technology Services

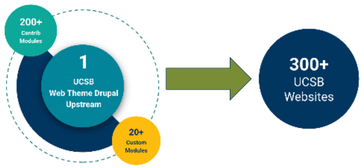
The Challenge: Ensuring Stability

UCSB's web ecosystem relies on UCSB web theme to manage 300+ websites, each with a combination of custom and contributed modules. However, with every update:

1. Manual testing is limited – Only the updated features are reviewed, while other site functionalities remain untested.

2. Undetected bugs surface post-deployment, leading to site crashes, broken pages, and frustrated users.

3. Fixing issues after release is costly – Debugging takes time, delaying future updates.

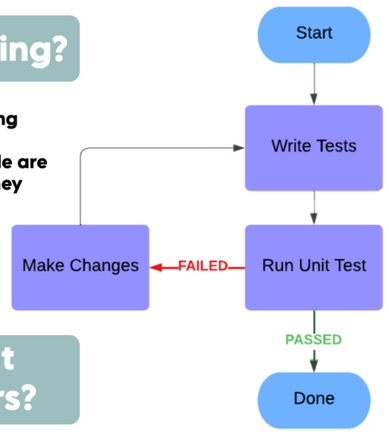


Challenge: new code or code updates can break 300+ sites.

The Solution: Automating WebOps with AI-Driven Unit Testing

What is Unit Testing?

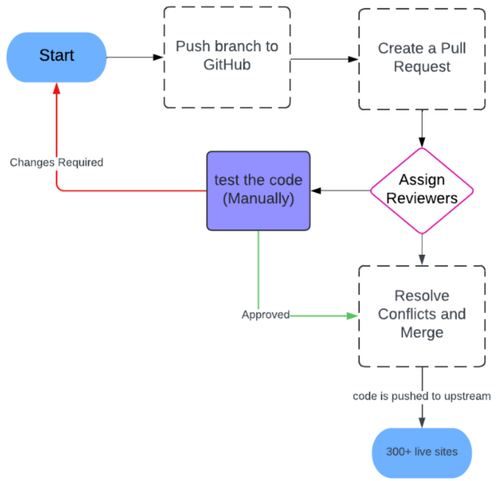
Unit testing is a software testing technique where individual components (or "units") of code are tested in isolation to ensure they function correctly.



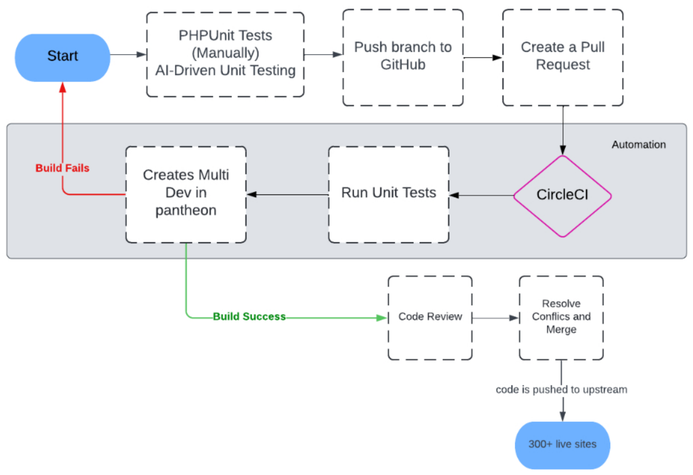
Why does unit testing matters?

Unit Testing matters because it helps to catch out bugs early in development before it's too hard to catch them

WebOps - Current Workflow



WebOps - Proposed Workflow



Proof of Concept: AI-Driven Unit Testing for the Annual Report custom Drupal module

The Annual Report Module is a standalone custom module developed specifically for Student Affairs (SA) Annual Reports. It manages departmental reports across multiple years, providing a backend system for report owners to draft, review, and publish reports.

This module follows a complex workflow—starting as a draft created by the Annual Report Manager, then progressing through multiple review and approval stages before publication. Ensuring the accuracy and stability of this workflow is crucial, as any disruption can impact data integrity and report accessibility.

To address this, AI-driven unit testing was implemented using Amazon Q and GitHub Copilot, automating test generation and validation. This eliminates the need for time-consuming manual testing every time a new feature is deployed, ensuring that the entire report lifecycle functions correctly with each update.

The implementation of unit tests in the Annual Report Module significantly improved system reliability, prevented deployment-breaking errors, and set a foundation for scalable automated testing across UCSB's WebOps.

Next Steps: Scaling Up Automation Across UCSB's WebOps Infrastructure

As the project progresses, the next phase will focus on expanding automated testing across multiple modules and fully integrating unit tests into the CI/CD pipeline.

Key Future Goals:

1. Expand Testing Coverage – Apply AI-driven unit testing to UCSB Web Theme and additional UCSB Drupal modules.

2. Integrate Seamlessly with WebOps – Automate tests at each stage of the WebOps deployment cycle.

Acknowledgement

I would like to express my sincere gratitude to Elda for her invaluable mentorship and guidance throughout this project. A special thanks to Rebecca Candy for her continuous support and insights. I also appreciate the SSIS team for fostering a collaborative and inclusive environment. Finally, I am grateful to ITS for providing this internship opportunity, allowing me to explore AI-driven unit testing and WebOps automation.