DM TOOLS

Progress Review and Testing Overview

SEP Team 5

Progress Review

MVP Progress Review

MVP:

- Catalog view of monsters, player characters, encounters, and campaigns
- CRUD for monsters, player characters, encounters, and campaigns
- Basic running through an encounter of a campaign

Left to Do and Scheduling Plan:

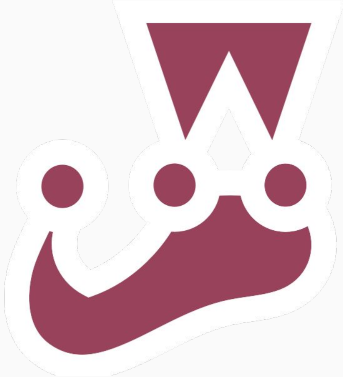
- UD Front end integration for Encounters and Campaigns **[U - Sprint 5, D - Sprint 6]**
- CRUD integration for player characters **[Sprint 6]**
- Catalog view of player characters **[Sprint 6]**
- Basic running through an encounter **[4/7 tasks for Sprint 5, remaining 3 tasks for Sprint 6]**

Testing Overview

Testing Tools

TS-Enzyme:

- React Integration
- Front End Testing



TS-Jest:

- API Testing Integration
- Backend and Scripting Testing

Travis CI and CodeCov:

- Continuous Integration
- Ensures 80%+ code coverage on all pull requests

Nock:

- Mocks HTTP requests and their returns

Testing Responsibilities

Feature developer(s) are responsible for implementing all code-based tests for their feature

Code reviewer are responsible for verifying that the code-based tests are accurate and comprehensive for the feature

Product owner is responsible for (official) acceptance testing

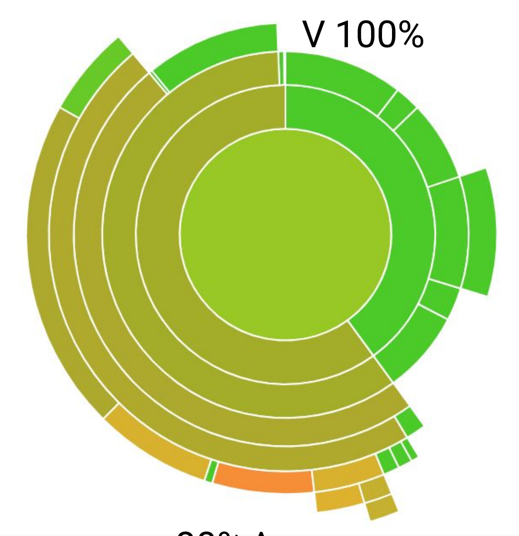
Testing Standards (Unit & Regression Testing)

Unit Testing

- At least 80% of all feature code has to be covered
  - *Usually close to 90%, with a few 80%'s*

Regression Testing

- At least 80% of codebase must be covered at all times
  - *Consistently has been around 93%*
- 100% of the tests for the codebase must pass at all times



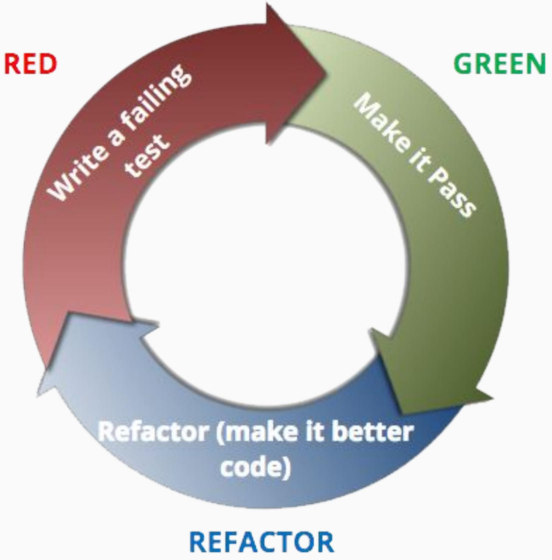
80% ^

Testing Standards (Acceptance Testing)

- External Product Owner pilots feature to local DMs currently involved in campaigns
- Piloting observed by scrum master and reported back to team
- All features must be accepted by the external product owner and (within reason) local DMs
    - Procedure for conflict between DMs and team/product owner documented on Wiki

Testing Implementation Strategy

- Snapshot Testing
- Happy/Sad Path Testing
- Pseudo-Red-Green Refactor
- Isolate Components and Interactions via Mocking when not implementing integration testing



Product Demo

Questions? Comments? Concerns?