Drupal Quality Initiative—Instructions

Document	Description
https://drupal.org/project/dqi	Main project page with issues etc.
The Big Checklist	Spreadsheet with checklists for each category.
Best Practices	Detailed instructions for each step; this document and the spreadsheet link into the Best Practices document.
Introduction Video	From BADCamp 2018.

Table of Contents

Introduction

The goals of the Drupal Quality Initiative are:

- Gather industry best-practices in one place.

- Establish Quality Levels similar to the Carnegie Mellon Capability Maturity Model: the DQI Level ("Drupal Quality Initiative Level"). For example, ultimately we want to be able to say, "That module is DQI Level 3—definitely ready for production sites."
- Create a community structure for contributed module maintainers to self-assess the quality of their modules.
- Give module maintainers a quality ladder to climb.

Key Work Products

Who is the target for? Lead developer, site builder/technical team lead

Overall goal: A set of documents that will guide a lead developer to produce a successful project/contributed module.

Insert detail about iron triangle.

Suggested products of the initiative are:

1. A master checklist of quality items to consider in each major area: Team Management, Requirements, Coding, etc.
2. A standard for bugs; likely use the same levels used on Drupal.org i.e. minor, major, etc.
3. Creating Drupal Quality Initiative Levels. Something like the following (*not* attached to this!):
    a. **DQI Work Product Levels** i.e. the actual code and documentation
        i. DQI Level 1: Basic quality. Passes linting tests, code formatting, consistent object oriented structure, basic security auditing.
        ii. DQI Level 2: Moderate quality. Adds basic unit testing, major bugs known, included in Drupal Security Review.
        iii. DQI Level 3: Advanced quality. Adds functional testing, load testing, automated security testing, low number of major bugs and there is evidence of the code being used in many production sites.
    b. **DQI Team Levels** i.e. processes the team uses, similar to the Capability Maturity Model:
        i. DQI Level 1: Initial. Ad hoc, reactive, chaotic.
        ii. DQI Level 2: Repeatable. Many processes are repeatable, possibly with consistent results, even during times of stress.
        iii. DQI Level 3: Defined. The processes are used sufficiently for teams to become competent or the process to be validated in a range of situations.
        iv. DQI Level 4: Managed (Capable). High quality is repeatable.
        v. DQI Level 5: Optimizing (Efficient). Teams continually improve process performance through incremental and innovative technological changes/improvements.

Is this initiative opinionated?

Yes and no. There are clearly different ways to accomplish the same goal, particularly when choosing tools. Plus what works for one team may not work for another. Thus flexibility is the name of the game when implementing "best practices."

However, there likely will be some degree of opinion when we start creating the DQI Work Product levels.

Resources

- List of static analysis tools
- Coder + PHPCS

Requirements

Design

Code

Install Coder module

Coder includes PHP Code Sniffer.

Use instructions here to install Coder in your ~/.composer/vendor directory. Be sure to add the Drupal and DrupalPractice standards or phpcs won't be able to do anything. Add the location of phpcs and phpcbf to your path (the README.md explains how).

Package Management

Drupal has aligned on Composer.

Code Linting

Linting provides formatting and hints when there are potential problems with the code. You can lint for problem highlighting and use a different code formatter, if you wish, or have the linter perform both functions for you.

- PHP Code Sniffer
- PHP Lint (runs multiple lints at one time)
- ESLint (Javascript)

Code Linting for Core

Code formatting

Using a code (re-)formatter makes codes easier to read and makes code consistent.

Recommended Formatter

The code formatter gaining tremendous traction is Prettier. It handles PHP, Javascript and more. Learn more from their description page "Why Prettier?"

Security Auditing

- RIPS Tech
- PHP Malware Finder (PMF)
- SonarSource PHP
- Exakat
- PHPStan
- Psalm
- PHP Vulnerability Hunter
- Grabber
- Symfony Security

Documentation

Testing

- Testing Starter kit

Action items:

- ChrisW: Create Drive document for testing https://docs.google.com/document/d/1n3feyWe1LcbsvLG4zQXTv5Romw0Oq2AUBKtbUZ3t0l0/edit?usp=sharing
- Ladder:
    - Level 1: Read through the docs provided. Be able to alter the provided tests for your site. Be able to test if critical pages and text is present on the site. No behavior testing.
    - Level 2: Be able to duplicate the kinds of tests we provide to cover super important pages of your site.
    - Level 3: Be able to test the critical/complex behavior of a key interaction on the site.