

# Using Grunt

## to Manage Drupal Build and Testing Tools

BAD Camp 2014

11/9/2014

# Joe Turgeon

Director of Engineering

**Drupal.org/GitHub/Twitter:** @arithmetic



# Why Build and Test Drupal?



# Why build Drupal?

- (or: How I Learned to Love Drush Make)
- End anti-pattern of committing core/contrib code
- Improve transparency through declarative build process
- Simplify use of external libraries and resources
- Enforce reproducibility



# Why test Drupal?

- Beyond QA and “clicking around”
- Empower devs to produce quality code
- Improve objectivity of requirements
- Collect and track metrics



# Why Grunt?

- Community-supported, widely-adopted, flexible
- Requires only Node.js
- Declarative configuration
- 3,000+ contributed plugins



# Drupal, Grunt, Oh My?



# Introducing Grunt Drupal Tasks

- Started as R&D project in January
- Since March, used on 6 client projects
- Becoming integrated in our distributions
- Released on GitHub in September





# Best Practices

- Enforces consistent practice across team and CI env
- Including Drush make, Behat, code style checks
- Manages tool dependencies for projects
- Tools are ready to use and discoverable



# Dev Tool Orchestration

*Sass*



# Let's Do This!



# Follow Along

- <https://github.com/phase2/grunt-drupal-tasks>
- Requires:
  - npm (included with Node.js)
  - Grunt CLI (npm install -g grunt-cli)
  - Recommended: Composer (global install)



# Setting Up a Project



# Project Set Up

- Initialize a new project with:

```
npm install grunt grunt-drupal-tasks
```

```
cp -rf node_modules/grunt-drupal-tasks/example/* .
```

- See: [grunt-drupal-tasks / CONFIG.md](#)

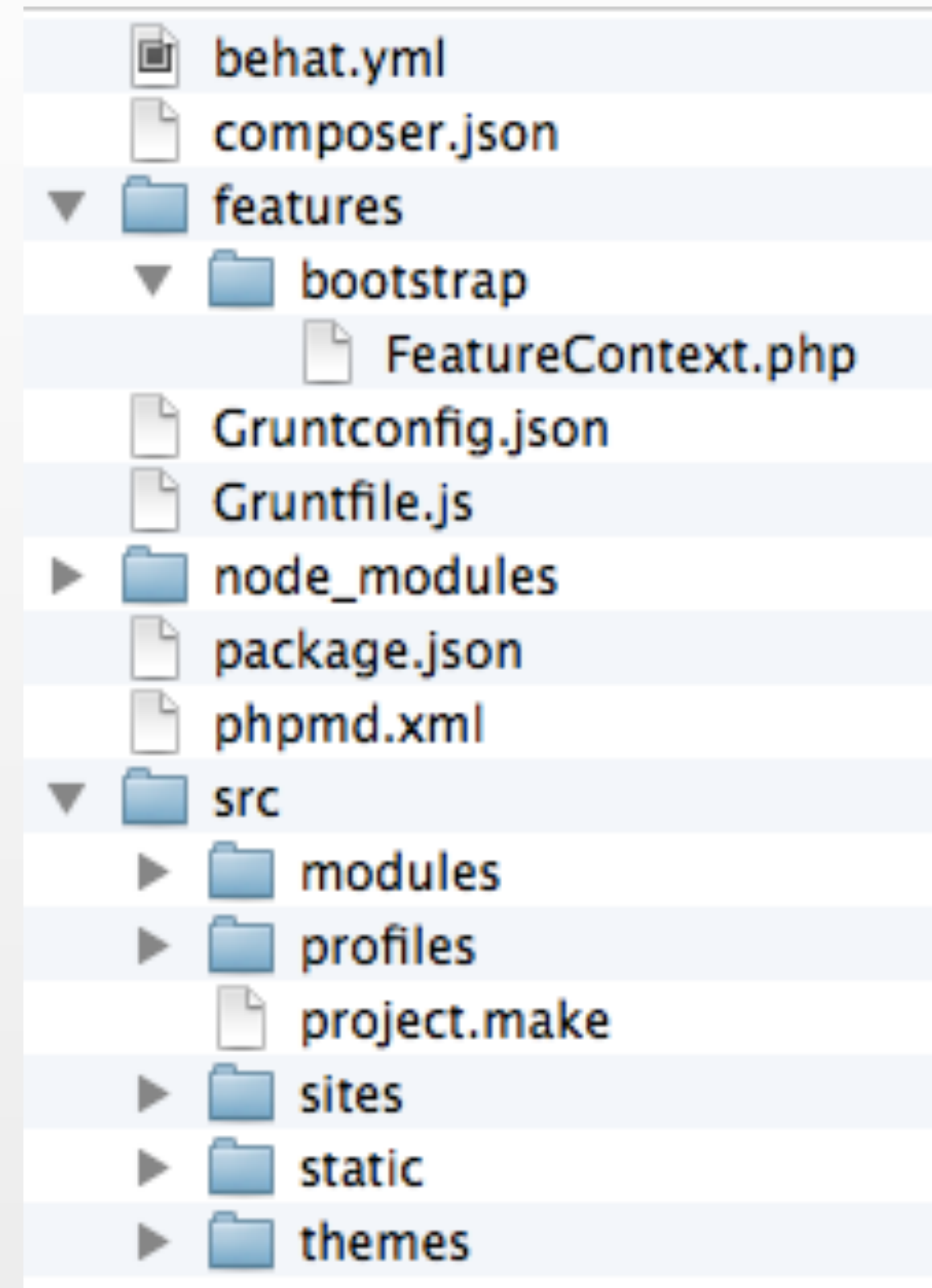


11. bash  
pleasant-air:gdt-demo Turgeon\$



# Drupal Scaffolding

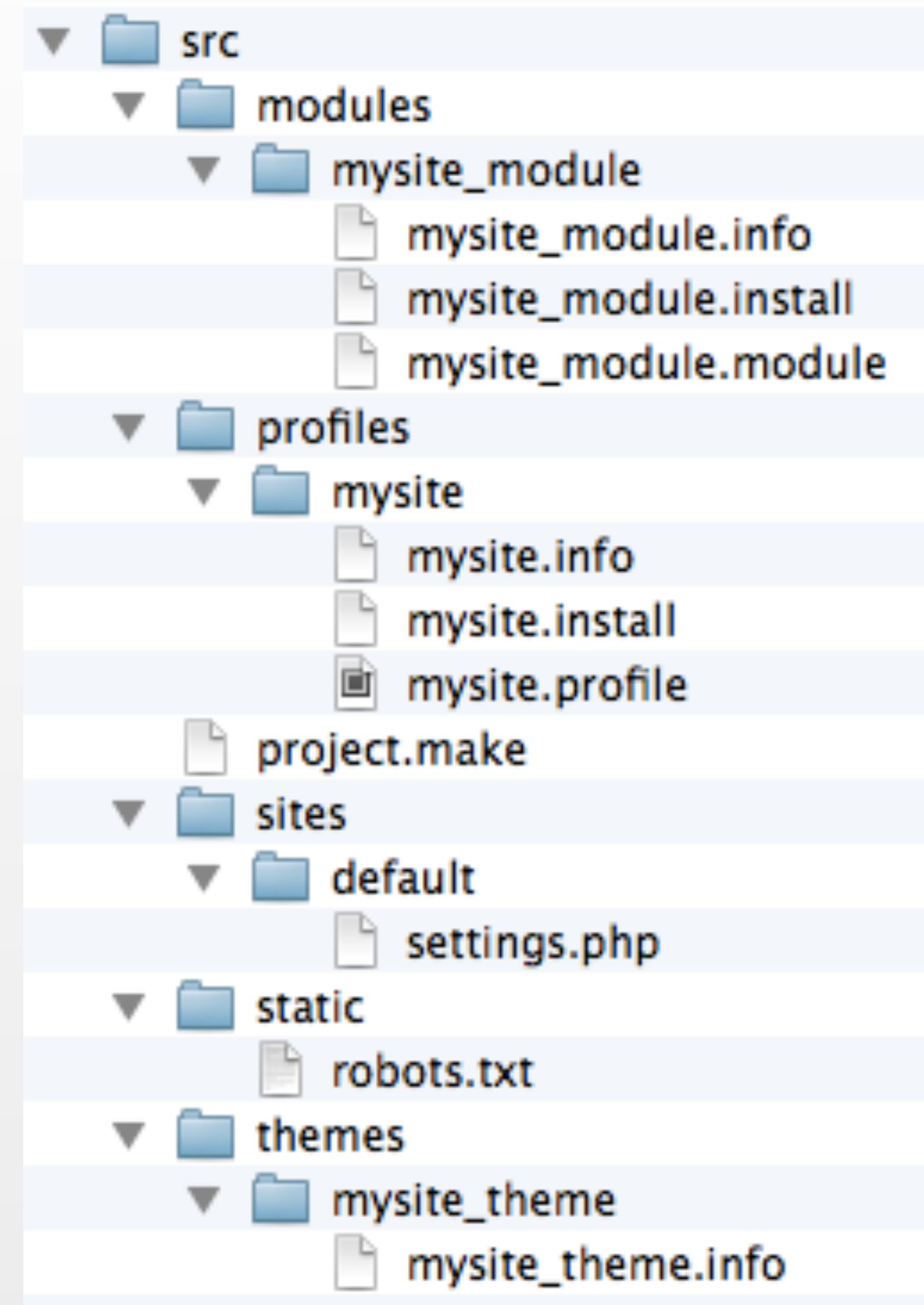
- Scaffolding accommodates custom project code and supporting tools
- This structure is committed as the initial project code base





# Add Project Code

- src/modules: Custom modules
- src/profiles: Custom install profiles
- src/project.make: Drush make file
- src/sites: Single- or multi-site config
- src/static: Static file replacements
- src/themes: Custom themes



# Building the Project



# Project Build

- Build the project with:

`grunt`

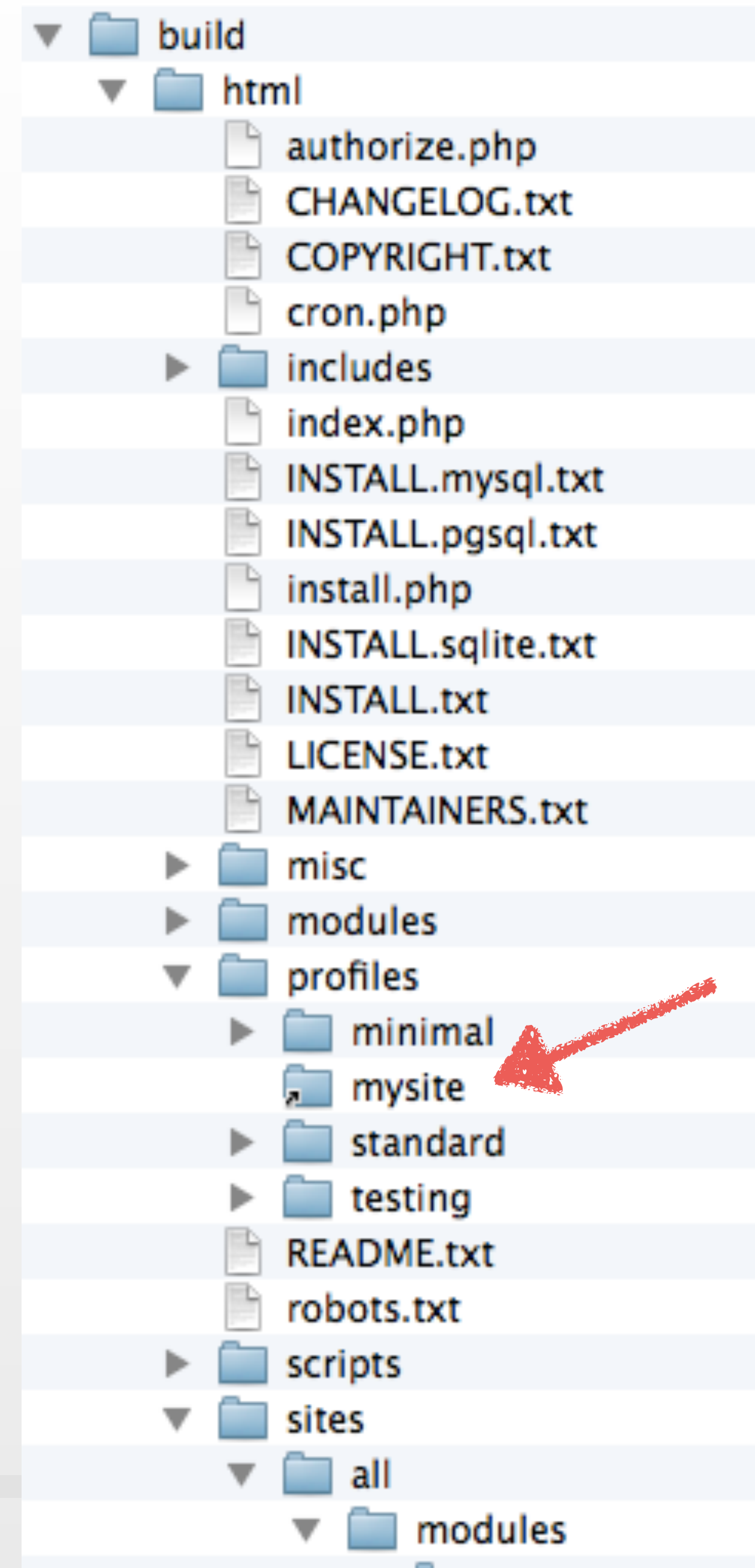


11. bash  
pleasant-air:gdt-demo Turgeon\$



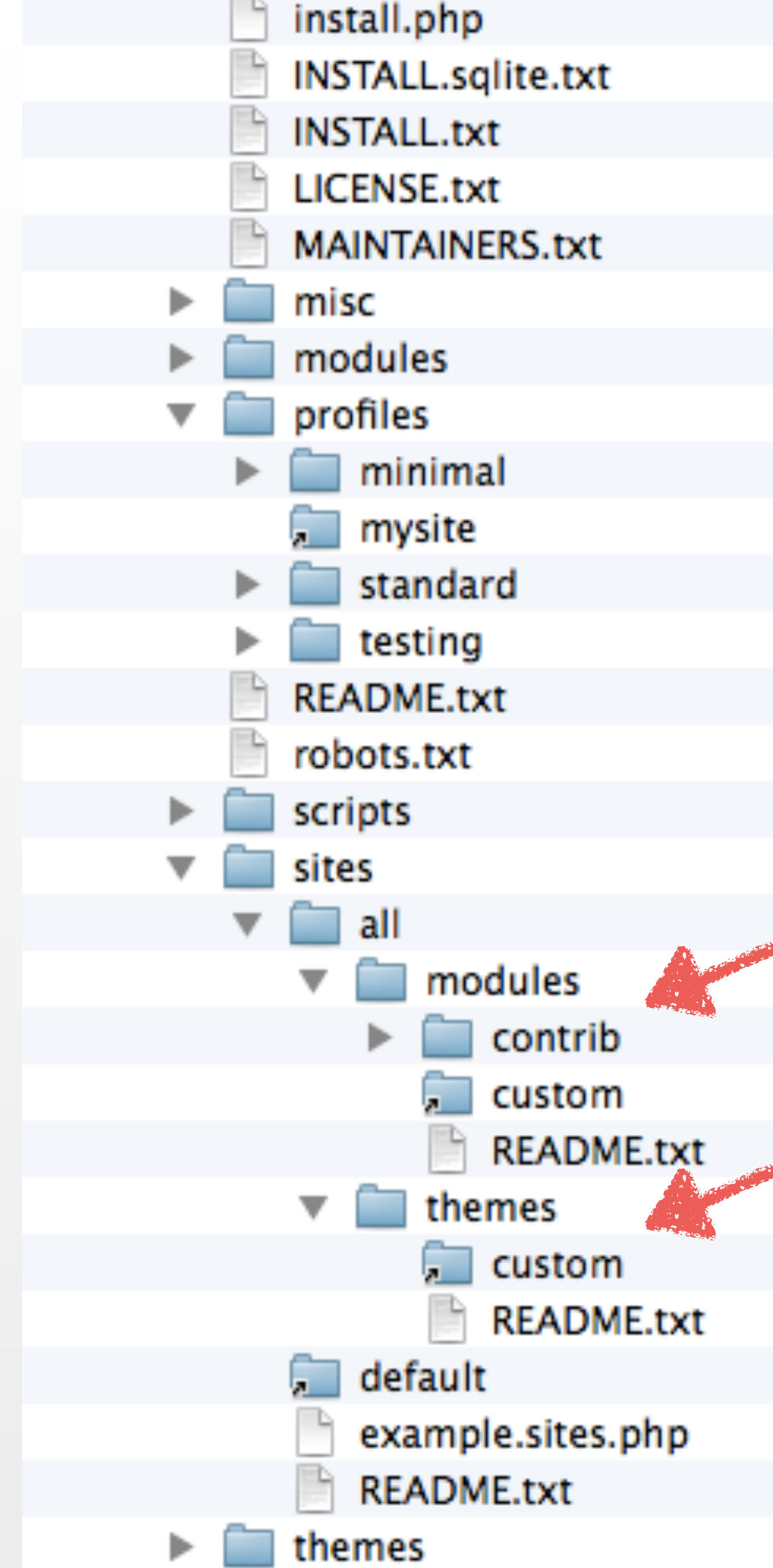
# Build Results

- build/html: Fully-assembled Drupal site
- build/html/profiles/mysite: Symlink to src/profiles/mysite



# Build Results

- build/html/sites/all/modules/contrib:  
Contrib modules from make file
- build/html/sites/all/modules/custom:  
Symlink to src/modules
- build/html/sites/all/themes/contrib:  
Contrib themes from make file
- build/html/sites/all/themes/custom:  
Symlink to src/modules



# Symlinks, really?

- Yes, really!
- Relative links within the scaffold structure are portable
- The build process ensures the links are set up correctly
- You do not need to re-build after each change to a source file
- XDebug/IDEs work for both core/contrib and custom code





# Validating Code





# Validating Code Quality

- Run PHP Lint and PHP Code Sniffer (with Drupal coding standards) with:

```
grunt validate
```



# Validating Code Quality

- PHP Lint verifies syntax and more

```
pleasant-air:gdt-demo Turgeon$ grunt validate
Running "phplint:all" (phplint) task
Warning:
Parse error: parse error in src/modules/mysite_module/mysite_module.install on line 4
Errors parsing src/modules/mysite_module/mysite_module.install Use --force to continue.

Aborted due to warnings.
```



# Validating Code Quality

- PHP Code Sniffer verifies Drupal's code standards

```
Running "phpcs:validate" (phpcs) task
```

```
Starting phpcs (target: validate) in src/**/*.php src/**/*.module src/**/*.inc s  
rc/**/*.install src/**/*.profile src/**/*.css !src/**/*.features.*inc !src/sites  
/**
```

```
FILE: ...utomation/grush/gdt-demo/src/modules/mysite_module/mysite_module.module
```

```
-----  
FOUND 8 ERROR(S) AFFECTING 6 LINE(S)  
-----
```

```
 2 | ERROR | Missing file doc comment  
 4 | ERROR | Missing function doc comment  
 5 | ERROR | Expected 1 space before "="; 0 found  
 5 | ERROR | Expected 1 space after "="; 0 found  
 9 | ERROR | More than 2 empty lines are not allowed  
10 | ERROR | More than 2 empty lines are not allowed  
11 | ERROR | Files must end in a single new line character  
11 | ERROR | More than 2 empty lines are not allowed
```



# Validating Code Quality

- Encourage devs to validate code before committing
- Focus peer code review on architecture
- Fail integration builds or reject pull requests at certain thresholds



# Testing Drupal



# Testing Drupal Functionality

- Provides Behat, the Drupal Extension, other dependencies
- Testing with Behat requires an installed Drupal site accessible by URL
- Currently, this requires manual set up of a local env
- Future feature: Using “drush runserver” for a test env



# Testing Drupal Functionality

- Run Behat for any included test features with:

```
grunt behat
```



11. bash  
pleasant-air:gdt-behat-test Turgeon\$





# Getting Help



# Getting Help

- See a list of available commands with:

```
grunt help
```



```
pleasant-air:gdt-demo Turgeon$ grunt help
Running "availabletasks:help" (availabletasks) task
```

### Testing & Code Quality

**analyze** => Static codebase analysis to detect problems. Outputs Jenkins-compatible reports. (e.g., PHPMD)  
**behat** => Parallel behat  
**validate** => Quick code health check for syntax errors and basic practices. (e.g., PHPCS w/ Drush Coder rules)

### Utilities

**clean** => Clean files and folders. (default|sites|temp)  
**copy** => Copy the contents of the static files directory to the Drupal docroot. (static|tempbuild)  
**mkdir** => Make directories. (init|files)  
**newer** => Run a task with only those source files that have been modified since the last successful run.

### Dependency Management

**composer** => Wrapper for Composer commands  
**drushmake** => Run "drush make" if the makefile is newer than the dest directory.

### Build Process

**default** => Alias for "composer:install", "validate", "newer:drushmake:default", "symlink:profiles", "symlink:modules", "symlink:themes", "clean:sites", "symlink:sites", "mkdir:files", "copy:static" tasks.

### Operations

**package** => Package the operational codebase for deployment.



# Taking It Further



# Extending

- Overriding the steps in the default build process
- Defining new steps or replacing existing ones



# Future Development

- Drupal 8 support
- Compatibility with other testing tools
- Configurable deployment steps



# How to Help

- Try implementing Grunt Drupal Tasks on your next project
- File issues on GitHub to share how it could work better
- Patches welcome! (Submit pull requests on GitHub.)



# Thanks! Questions? Suggestions?

- Slides:  
<http://www.slideshare.net/arithmetic/using-grunt-with-drupal>
- GitHub repo:  
<https://github.com/phase2/grunt-drupal-tasks>







[PHASE2TECHNOLOGY.COM](https://phase2technology.com)

---