

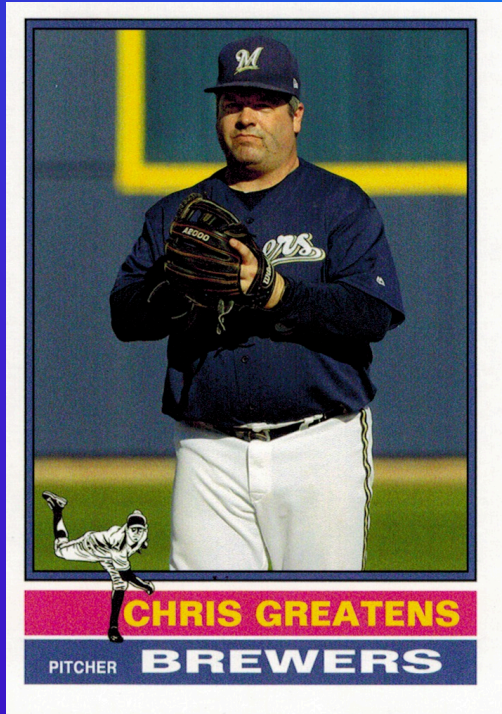
THE CURMUDGEON'S GUIDE TO BUILDING DRUPAL SITES

Chris Greatens

@cgreaten

bounteous

WHO AM I?



- Director, Digital Experience Platforms at Bounteous
- Working with Drupal and Bounteous since 2009
- Been a developer since 1994 and a curmudgeon since forever
- Currently half finished with a 9000 pc jigsaw puzzle

WHO IS BOUNTEOUS

We create big-picture digital solutions that help leading companies deliver transformational brand experiences.

The logo for Wilson, featuring the word "Wilson" in a red, serif font.The logo for the Commonwealth of Massachusetts, featuring a blue shield with a white cross and a figure, with the word "MASSACHUSETTS" in a grey, sans-serif font below it.The logo for Tesla, featuring the word "TESLA" in a black, sans-serif font with wide letter spacing.The logo for Ballad Health, featuring the word "Ballad" in a black, sans-serif font and "Health" in a smaller, black, sans-serif font below it, with a colorful graphic of three overlapping circles to the right.The logo for CHS Community Health Systems, featuring the letters "CHS" in a green, sans-serif font and "Community Health Systems" in a smaller, black, sans-serif font to the right, with a graphic of three stacked squares to the left.The logo for Zipcar, featuring a green circle with a white "Z" and the word "zipcar" in a black, sans-serif font to the right.The logo for Advance Auto Parts, featuring the words "Advance" and "Auto Parts" in a red, sans-serif font, with a checkered flag graphic to the right.The logo for Centura Health, featuring a blue cross icon and the words "Centura Health" in a black, sans-serif font.The logo for Caesars Entertainment, featuring a laurel wreath icon and the words "CAESARS ENTERTAINMENT" in a black, sans-serif font.The logo for PBS, featuring a black circle with a white head profile and the letters "PBS" in a black, sans-serif font to the right.

AGENDA

- 1. Overview**
- 2. Effective Drupaling**
- 3. Efficient Projects**
- 4. Growing As a Developer**
- 5. Wrap Up**

1. OVERVIEW

GOAL

To build a Drupal website where by the end of the project your client is happy.

To do this you need an efficient team,
effective plan, and clear communication.

1. EFFECTIVE DRUPALING

“

If you fail to plan, you are planning to fail

Benjamin Franklin

PREPARING THE BUILD – BE INVOLVED EARLY

- Help guide (not steer) the design
 - Look at designs with an eye toward leveraging Drupal as fully as possible.
- Determine what can go wrong
 - Identify the Risks (What Can Derail the Project?)
- If something goes wrong, know what you can change
 - Scope/Budget/Timeline – What Gives?
 - Understand the Feature Prioritization (What are the Must-Haves? Wants?)

PREPARING THE BUILD – DEFINE THE TECH STACK/ARCH

- Know the tech stack you are working with
 - Will the site be hosted on by a “Drupal” company? If not, what’s the tech stack?
 - Is there a need for a local development setup?
- Know the data flow through the system
 - What systems does the Drupal site need to integrate with?
 - Are you migrating content to the Drupal site?

PREPARING TO BUILD – DOCUMENT THE PLAN

- Create an Architecture Plan
 - Should have enough detail to give context to new team members.
 - Define the Development Workflow
 - List which theme and what modules will the Build depend on
 - Describe key points of the architecture (e.g. Migrations, Integrations)

“

Prolific developers don't always write a lot of code, instead they solve a lot of problems. The two things are not the same.

J Chambers

BUILDING THE SITE – GUIDING PRINCIPLES

If a developer needs to make a choice, what is guiding that choice?

- Use Drupal Core if you can, contrib modules if you wish, custom code if you must.
- Touch each application area as few times as possible.
- Don't hack core or contrib modules (but if you do, do it right)
- No “we will need this later” code. ((aka YAGNI)

BUILDING THE SITE – YOUR TOOLBOX (MODULES)

Contrib modules are the easiest way to be efficient. (Especially when you use the same ones over and over again.)

- Modules I use on nearly every project:
 - Layout Builder
 - Paragraphs
 - Views
 - PathAuto
 - Config Management
 - Google Tag
 - Migrate
 - Rabbit Hole
 - Display Suite
 - Devel
 - Shield
 - Metatag
 - Simple Sitemap

BUILDING THE SITE – YOUR TOOLBOX (MODULES)

But what happens if my toolbox doesn't have a module that will solve a specific problem?

Find a new module!

- Search on drupal.org or Google (or phone a friend).
- If you find a module that might work, research it and confirm.
- Assuming it will work, do you want to use it?
 - Is it covered the security team?
 - Who are the maintainers? When was the last commit?
 - How many people are using it?
 - What is in the issue queue?

BUILDING THE SITE – YOUR TOOLBOX

Building a Drupal site is more than just slinging code. A lot of work can and should be done with other tools

- Composer
- Drush
- Git (Source Control)
- Drupal Console

BUILDING THE SITE – EFFECTIVELY DEBUGGING PROBLEMS

While we all strive to building bug-free systems, sometimes bugs happen. Preparation and a solid strategy are key to debugging problems.

- Know what debugging tool you have available
 - Drupal: Logger, Devel, Web Profiler, Drupal Console, Drush scr, Twig debugger
 - PHP: print_r, var_dump, xdebug (integrate with your IDE)
- Have a strategy
 - Assess the problem
 - Form a hypothesis
 - Eliminate possibilities

2. EFFICIENT PROJECTS

KEYS TO HAVING EFFICIENT PROJECTS

The project team having the proper technical skills to complete a project is not all that is needed to assuring the client is happy with the results. Effective project teams have several traits:

- Strong leadership
- Clearly defined roles
- Psychological Safety
- Processes
- **Communication**

DAILY COMMUNICATION

Daily communication is a necessity. How do you communicate effectively?

- Daily Stand-ups (or Status Reports)
- **Peer Reviews**
- Ad-hoc (Using Communication Tools Effectively)

“

Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.

John F. Woods

WRITE THE CODE FOR FUTURE YOU

When you write code, be nice to yourself. Chances are it will be you that has to fix it.

- Don't Repeat Yourself
- Things Should Do One Thing
- Structure Code to be Self-Documenting Code
- Comment Where Appropriate
- How easy this to change?

DOCUMENTATION, PLEASE

Having well-documented code is great. But, there are times that documenting within the code is not enough.

- Where do others (e.g. PMs, BAs) look for and share information?
- Great for step-by-step instructions that others need to follow
- Documenting a process or part of the system is a great way to learn it
- If you are tired of repeating the answer, document it

3. GROWING YOURSELF

“

Continuous improvement is an intrinsic part of professionalism.

Robert Martin

BE CURIOUS

The best way to grow yourself is to remain curious. And then, act on that curiosity.

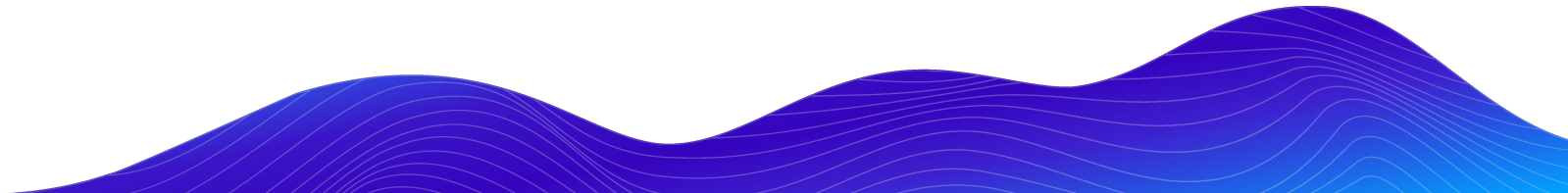
- Be Curious
 - Attend conferences (Hey! You're doing that already!), watch webinars
 - Read: newsletters, blogs, books
 - What are your friends doing?
 - Stretch yourself on projects

WHERE DO YOU GO FROM HERE?

What are actions you can take?

- Setup your IDE to allow interactive debugging
- Learn a new computer language
- Build a sandbox on your computer to experiment with.
- Pick out one topic you learned about this week and explain the concepts to a colleague.

QUESTIONS?

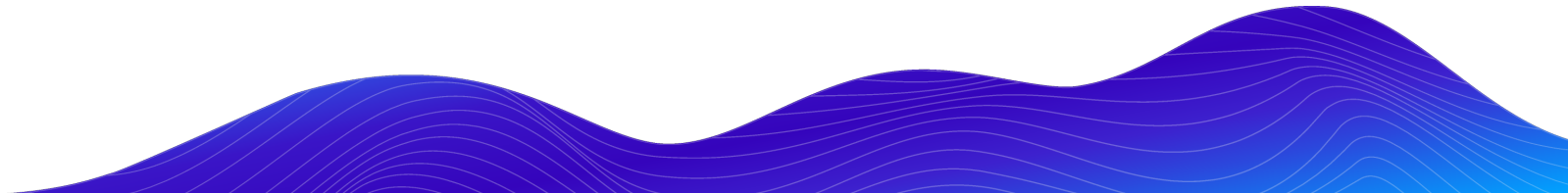


Thank You!

Chris Greatens

Twitter: @cgreaten

LinkedIn: <https://www.linkedin.com/in/christopher-greatens-9392638>



RESOURCES

- Drupal modules -- https://www.drupal.org/project/project_module
- PHP the Right Way – <https://phprightway.com/>
- Clean Code (PHP) – <https://github.com/jupeter/clean-code-php>
- Drupal VM -- <https://www.drupalvm.com/>
- Lando -- <https://docs.devwithlando.io/>