No Red Shoes

How ***Not*** To Lead A Team Of Drupal Developers

AUGUST 23, 2018

Introduction

Tobby Hagler

DIRECTOR OF ENGINEERING

Email: thagler@phase2technology.com
Drupal.org: **tobby**

Architect for projects such as

- NBA.com
- Weight Watchers
- Memorial Sloan Kettering Cancer Center

Recent DrupalCon presentations

- Cthulhu Drupal: Coding with Lovecraft
- Building NBA.com on Drupal 8
- Dungeons & Dragons & Drupal

© 2016 Phase2

© 2016 Phase2

ABOUT ME

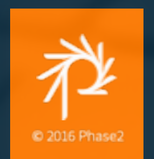
Grew up in Gravel Ridge, Arkansas

No formal education

Self Taught - Learned a lot from others

Learned about Programming through books, IRC, and later, the Web

Worked for dial-up ISPs and online newspapers in the 1990s



© 2016 Phase2

MORE ABOUT ME

Newspaper job turned into Director of Online Services

Managed several developers, responsible for revenue, no clue how to manage

Moved to corporate

Eventually became a Technical Manager

© 2016 Phase2

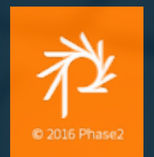
SO WHY THE AUTOBIOGRAPHY?

Self taught, but lots of mentors along the way

Lots of Trial and Error

Had to make some mistakes along the way

I didn’t get here alone, neither should anyone else



© 2016 Phase2

Good decisions come from experience, and experience comes from bad decisions.

~ Unknown

© 2016 Phase2

SOME OF THE BAD ADVICE I'VE BEEN GIVEN

© 2016 Phase2

No red shoes in the workplace;

Establish an arbitrary rule (as a fireable offense) to establish your authority.

© 2016 Phase2

Lead With Purpose, Not Rules

Everything with purpose, nothing arbitrary

Authority comes from experience, and sharing that experience is more valuable than enforcement of rules

Managers already *have* authority by default

Don’t separate the leader from the team; this makes you the focus instead of the team

Drupal developers value community, and respond to inclusion and collaboration

© 2016 Phase2

Rules Without Purpose Have Consequences

Everything with purpose, nothing arbitrary

“No red shoes” is arbitrary, but the consequences are real:

- Men are less likely to wear red shoes than women, so now this rule becomes unintentionally exclusionist
- Red shoes are a way to show personality; banning these things diminish individual passion within the team
- People resent being told they *can’t* do something more than being told they *must* do something

These rules set a precedent that important decisions will be made in a vacuum or without regard to consequence

© 2016 Phase2

People are resources, cogs;

Put them in the right place, and
everything will just run smoothly.

© 2016 Phase2

Projects Plans Are Only The Start

People aren’t just interchangeable resources

My project needs 2 back end developers, a front-end dev, and a migration specialist. As long as I have those, my project should run on schedule, right?

Other factors to consider:

- Specific people perform differently, have different skill levels
- Team dynamics and participation
- Collaboration is the Dark Matter that holds a team together

© 2016 Phase2

People Perform Differently

People aren’t just interchangeable resources

Drupal is an expansive framework

- Not every developer can do all the things
- Some challenges require collaboration

It’s not always *just* your team

- The Drupal community plays an important role in Drupal projects
- Client stakeholders can add valuable resources, insights, or even challenges

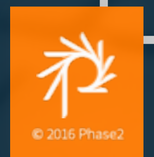
© 2016 Phase2

Drupal Developers And Specialization

People aren’t just interchangeable resources

Drupal developers tend to specialize in certain areas

- Front-end
  - Theme layer
  - Twig templates
- Back-end
  - Module development
  - SQL queries
- Site building
  - Content types, fields, Views
  - Placing blocks, layout



© 2016 Phase2

Anyone can ‘do’ Drupal;

Don’t worry about the perfect team, just get the cheapest people, and train them.

© 2016 Phase2

Hire The Right People For Your Team

Not everyone ‘does’ Drupal the same

Drupal, like may Open-Source projects, are full of self-taught developers

- Varying degrees of habits
- Different participation levels

Look for additional things, like:

- Community participation
- Code (core, module, theme) contributions
- Blog posts, issue queues, and other contributions
- Amount of time registered on drupal.org

© 2016 Phase2

Interviewing Drupal Developers

Not everyone ‘does’ Drupal the same

Ask Drupal-related questions

- How do hooks work?
- What’s the difference between services and plugins
- Explain Dependency Injection

Ask problem-solving questions

- How do you resolve a WSOD?
- How do you fix a bug in Drupal core or a contrib module?
- How many ways can you do … ?

© 2016 Phase2

Only hire Computer Science majors;

A college degree (or accreditations, or certification…) is the only thing worth looking at on a résumé.

© 2016 Phase2

Hiring Requires Interviewing

Hire based on the interview, not the resume

Obvious: Someone with a Computer Science degree has more software development training than those without

Experience replaces education over time

- Changes in technology require continued experience and education
- Developers get rusty when they're not actively using certain technologies or languages

Certifications and accreditations show that a candidate passed a test; how have they used that knowledge since?

© 2016 Phase2

Hiring Requires Interviewing

Hire based on the interview, not the resume

Other ways to learn

- Drupal, PHP, and other Open-Source technologies are easy to self-learn
- People can evolve into being developers as part of other jobs
- Plenty of good online courses available

It’s important to know what candidates understand and how they think, and to not make assumptions based on résumés or a list of certifications.

© 2016 Phase2

That's how the CEO/CTO wants it;

Measurable results, collecting trophies, and staying within the lines is the best way to be successful.

© 2016 Phase2

Fear-Based Management

Work with upper management to do things right

“Fear-Based Management” doesn’t refer to managing a team through fear. It’s about managing in a way that reduces their own fears, even to the detriment of the team or project.

Some managers lead their team in ways designed to abate their own fears.

- Worried about how stakeholders will perceive their performance
- Concerned about past failures or edge cases
- Too afraid to step outside their comfort zones

© 2016 Phase2

Out Of Date Information And Practices

Work with upper management to do things right

Staying in touch with best practices is tough. The “old way” of doing things don’t always hold up.

- People used to coiled languages see the world differently than interpreted languages
- Leads to wildly different approaches to project planning, resource allocation (both people and hardware)

Should still listen to their advice and experience, even if things have changed

Asking “why” is not about questioning authority

© 2016 Phase2

Managing Up

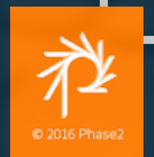
Work with upper management to do things right

“Managing Up” is one method to provide value for your boss and your organization.

It *can* be about setting expectations so that you and your team can work with minimal interruption.

It *can* mean regular communication, being proactive, and keeping upper management up-to-date with relevant and timely information.

It’s about investing in a worthwhile relationship.



© 2016 Phase2

Work (them) harder;

If you put their nose to the grindstone,
you can have a nice car like this one day.

© 2016 Phase2

More Hours Isn’t More Productivity

Work smarter, not harder

Developers often work odd hours that hits their ideal time

Working extended hours, weekends during crunch time (just before a launch) is common, and fine as long as there is downtime

Extended periods of long hours lead to burnout for people and teams

© 2016 Phase2

More Hours Isn’t More Productivity

Work smarter, not harder

Long hours with no end in sight:

- Leads to resentment, people leave
- Effectiveness is reduced when people are tired
- Productivity isn’t linear — Diminishing returns

Managers who work too hard or multitask splits their focus

- Neglecting the team keeps them blocked
- Spend time up front to plan
- Discard unnecessary tasks

© 2016 Phase2

Long Hours Requires Prioritization

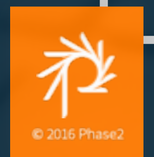
Work smarter, not harder

Prioritize with **JOY**

- **Juniors** first
- **Others** second
- **Yourself** last

“The needs of the many outweigh the needs of the few”

Spending time unblocking others up front means more people are in a productive state than one person being able to work while others are spinning their wheels



© 2016 Phase2

LEARNING TO LEAD DEVELOPERS

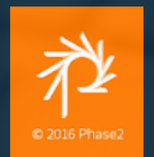
© 2016 Phase2

LEADING DEVELOPERS

Managing bits is easier than managing people

Software does what it’s told; code is a set of instructions

People are unpredictable; results will always vary



© 2016 Phase2

MANAGING SOFTWARE

Code is a set of instructions and patterns; clearly defined rules

Computers will (almost) always interpret their instructions the same

Subject to:

- Hardware limitations
- Interpreter or compiler versions
- Data, content, and `$variables`
- Programmer's ability to convert business logic to code

© 2016 Phase2

LEADING PEOPLE

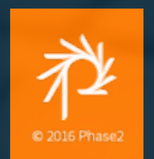
People don’t respond to instruction like code

People respond to different stimuli:

- Experience or Skill Level
- Perspective
- Motivation or Fears
- Passion or Drive

Different people respond differently to the same thing

People *are* the `$variable`



© 2016 Phase2

Know The Team

Spend time learning who your people are

- How do they respond to stimuli?
  - Pressure, deadlines, stress
  - Do they like rules? Freedom? Flexibility? Structure?
- Are they creative or linear thinkers?
- How much of the big picture can they see at once?
- What drives someone to do their job well?
  - Or at all?
  - 9-5? Night owls? Early risers?

Be the person you needed when you were there.

© 2016 Phase2

QUESTIONS?

© 2016 Phase2