

DEVOPS IS MORE THAN A LICENSE PLATE

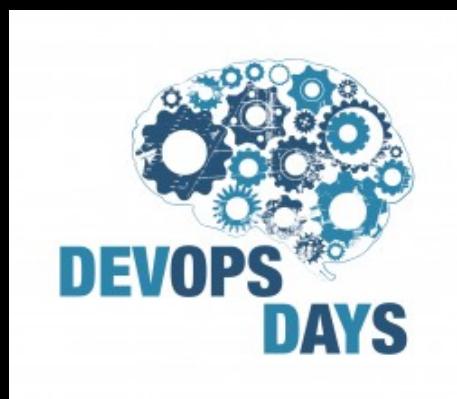
Matt Stratton
Customer Architect, Chef Software, Inc

mattstratton@chef.io
@mattstratton

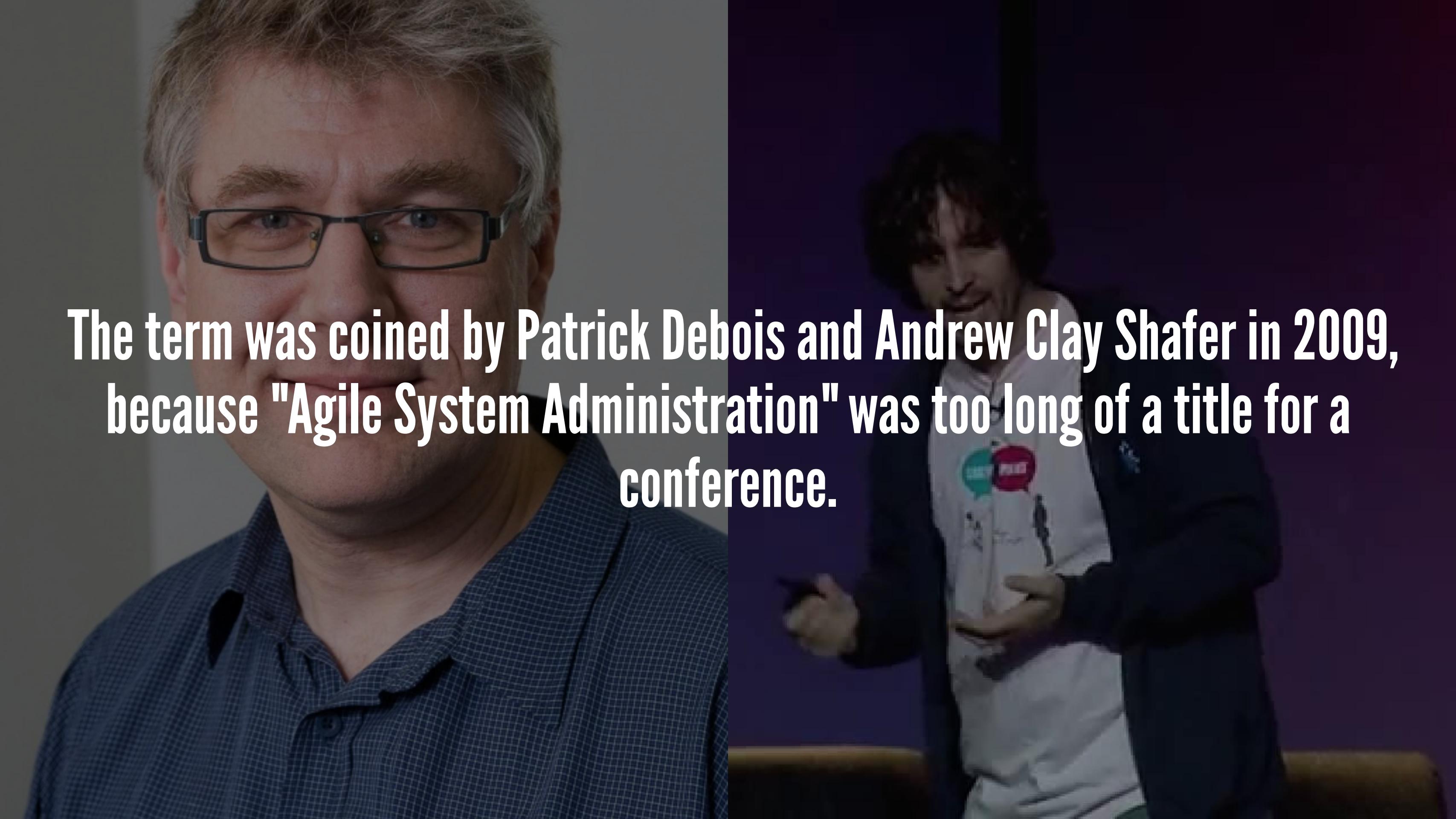


WHAT IS DEVOPS?

Etsy



A brief history...



The term was coined by Patrick Debois and Andrew Clay Shafer in 2009, because "Agile System Administration" was too long of a title for a conference.

CALMS

Culture, Automation, Learning, Measurement, and Sharing



After the first US based Devopsdays in Mountainview 2010 Damon Edwards and John Willis coined the acronym CAMS, which stands for Culture, Automation, Measurement and Sharing. Jason Hand later added an L, standing for Lean, to form CALMS.

There are several different models/definitions of DevOps, but for purposes of this discussion, I'm going to use CALMS to specify different focus areas on what DevOps can be.

**"A cultural and professional movement, focused
on how we build and operate high velocity
organizations, born from the experiences of its
practitioners."**

-- Chef Style DevOps¹



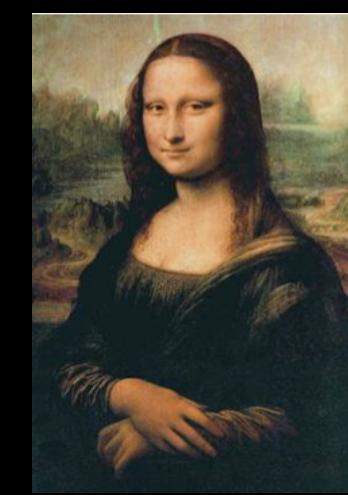
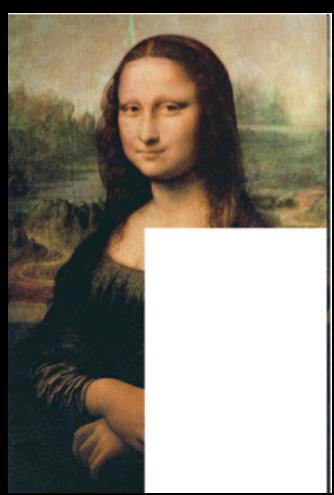
¹ https://youtu.be/_DEToXsgrPc

PATTERNS FOR SUCCESS



**DEVELOP FEATURES
ITERATIVELY, NOT
INCREMENTALLY**

Incrementally



Iteratively



SOCIAL CODING

Version control is nothing more than a communication tool for
developers
– Jez Humble

CODE REVIEW

- ▶ "The law of four eyes"
- ▶ Code review can be done by a maintainer, or by a peer.
- ▶ Have some type of sanity check prior to code review.

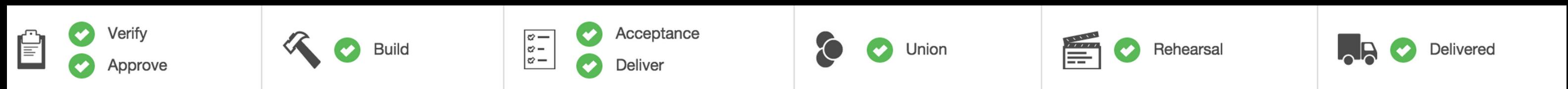




**DEMO. DEMO.
DEMO**

demo weekly, even if you don't have much to show. Even if it's just a process. Record the demos and share them.

CONSISTENT WORKFLOW



The way change moves is fixed, but is flexible at the level of execution

CONTINUOUS INTEGRATION

- ▶ Remember, it's iterative
- ▶ Short-lived branches, merged into master frequently
- ▶ Stop the build when it goes red

TESTING

- ▶ Focus on testing outcomes, not functions
- ▶ Perfect is the enemy of good
- ▶ Beware of over-testing²

² <https://coderanger.net/overtesting/>



SAFETY

- ▶ Varies from organization to organization
 - ▶ Just another aspect of quality
 - ▶ Trust, but verify

End to end, integrated workflow

HOW THIS WORKS WITH CHEF



End to end, integrated workflow



Application in Prod



OTHER PATTERNS

- ▶ Build autonomous actors
- ▶ Define alerts properly
- ▶ Blameless post-mortems

CHEF SPECIFIC SUCCESS PATTERNS

- ▶ Use Test Kitchen to develop and test cookbooks.
 - ▶ Manage dependencies using Berkshelf.
- ▶ Download community and internal cookbooks using Berkshelf.
 - ▶ Pin dependencies in your cookbook.
- ▶ Write tests using InSpec to verify cookbook functionality.

CHEF SPECIFIC SUCCESS PATTERNS

- ▶ Generate custom boilerplate cookbooks.
- ▶ Use linting tools (`foodcritic`, `cookstyle`).
- ▶ Commit each cookbook to its own git repo.
 - ▶ Build cookbooks in a pipeline.
- ▶ Lint, test, and upload cookbooks with a pipeline.

CHEF PROVISIONING & DEPLOYMENT PRACTICES

- ▶ Have a plan for evaluating, testing, and adopting new version of the chef-client.
- ▶ Can bootstrap nodes on all operating systems in your environment.
- ▶ Include bootstrapping the chef-client in your provisioning process.
- ▶ Managing aspects of the chef-client across live nodes, including interval and splay.

CHEF PROVISIONING & DEPLOYMENT PRACTICES

- ▶ Upload new versions of cookbooks and all of their dependencies to your Chef Server with a pipeline.
- ▶ Upload new versions of data bags, environments, and roles with pipeline.
 - ▶ Manage run_lists.

CHEF PROVISIONING & DEPLOYMENT PRACTICES

- ▶ Verify SSL on nodes with Chef Server in production.
- ▶ Pin cookbook versions in environments, and put nodes in environments in order to create predictable chef-client runs.

OPEN SOURCE PRACTICES WORK GREAT FOR ENTERPRISE PROJECTS TOO

- ▶ Have a **CONTRIBUTING.md** file in your repo(s)
 - ▶ Keep the conversation friendly
- ▶ Require tests and other automation for pull requests

<https://opensource.guide/best-practices/>

QUESTIONS?

