DevOps

MODERN TRENDS

Mykola Marzhan

WHAT SHOULD BE SOLVED BY DevOps?

Source: https://www.fom.be/compos/view/103

DevOps 1.0

> DevOps 1.0 has been heavily centered around harmonizing the interplay of development, QA, and operations. The primary goal has been to institutionalize continuous delivery, while also creating more flexible and stable application infrastructure.
>
> *- Justin Baker*

Source: https://dzone.com/articles/devops-20

DevOps 1.0

10+

DEPLOYS PER DAY

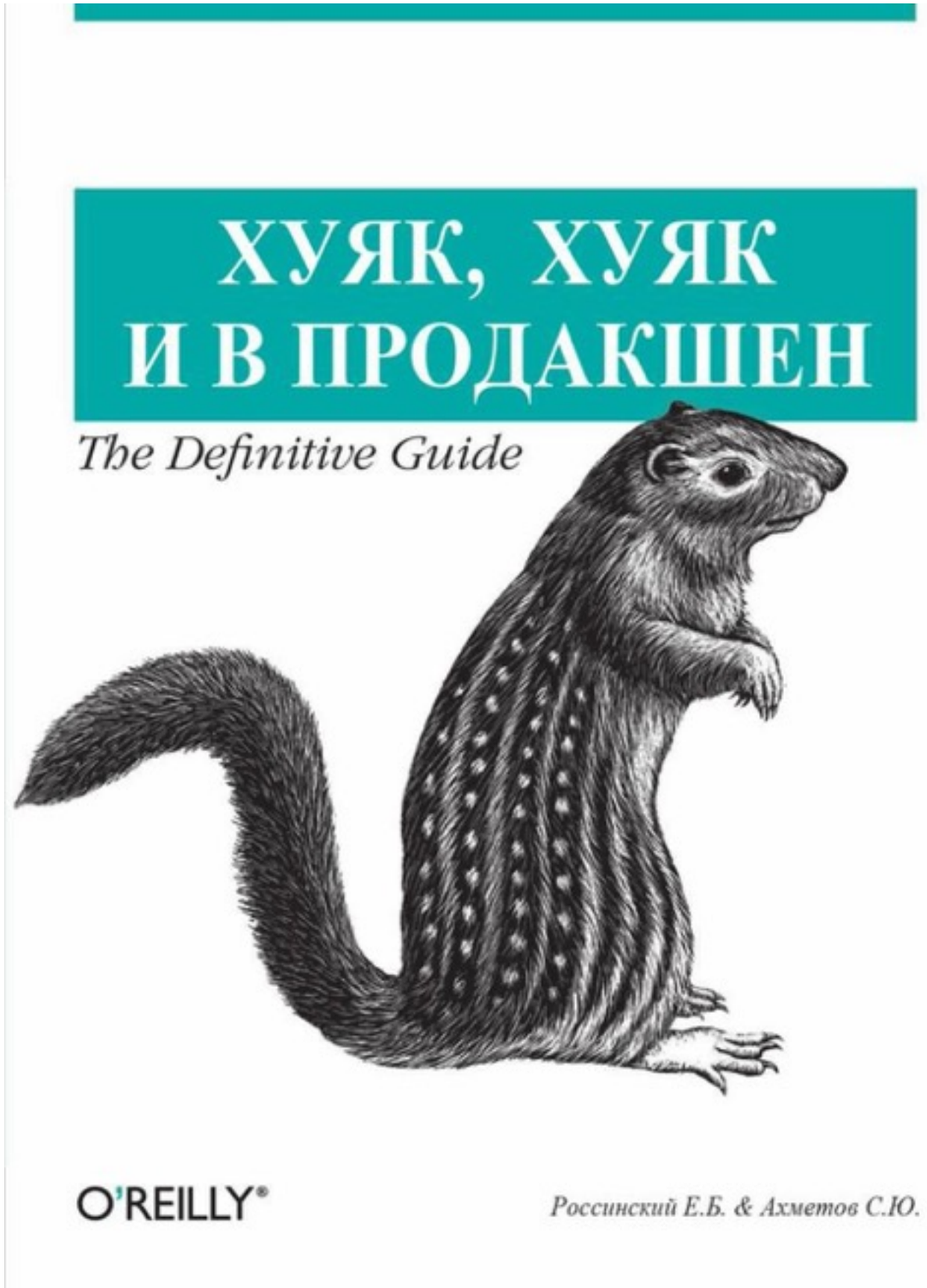
Dev and Ops Cooperation at Flickr
Velocity 2009

Source: http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr

DevOps 1.0

10+ DEPLOYS PER DAY

Dev and Ops Cooperation at Flickr
Velocity 2009



Source: https://goo.gl/zirxUQ

THE PROBLEM

> Through 2016, a lack of effective release management will contribute up to 80% of production incidents in large organizations with complex IT.
>
> *- Gartner research*

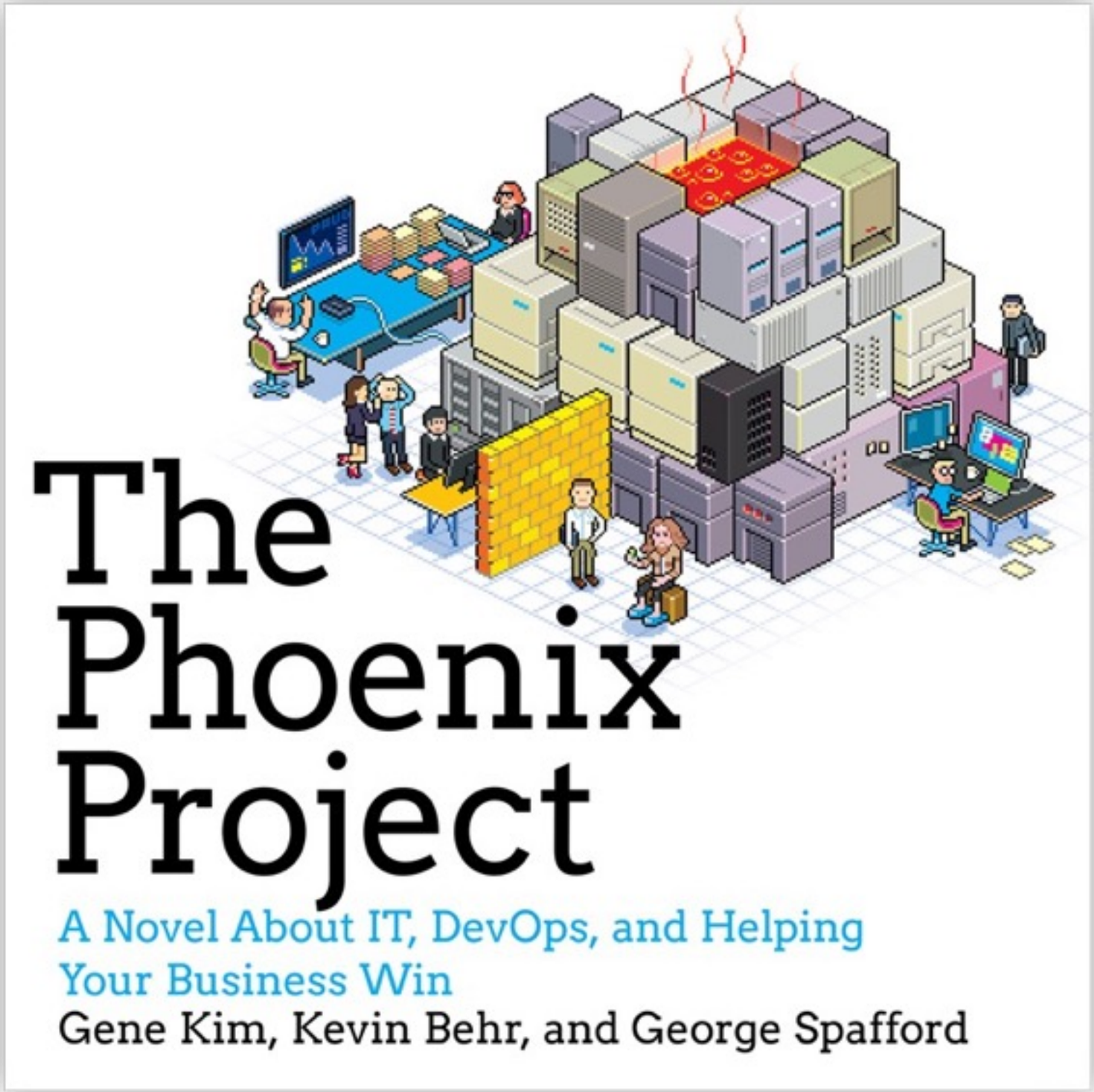
Source: https://www.gartner.com/doc/2504615/it-operations-set-effective-centralized

PREDICTION: DevOps 2.0

> With DevOps 2.0, we see the emergence of adaptive feature delivery as a critical component for successful software releases.
>
> *- Justin Baker*

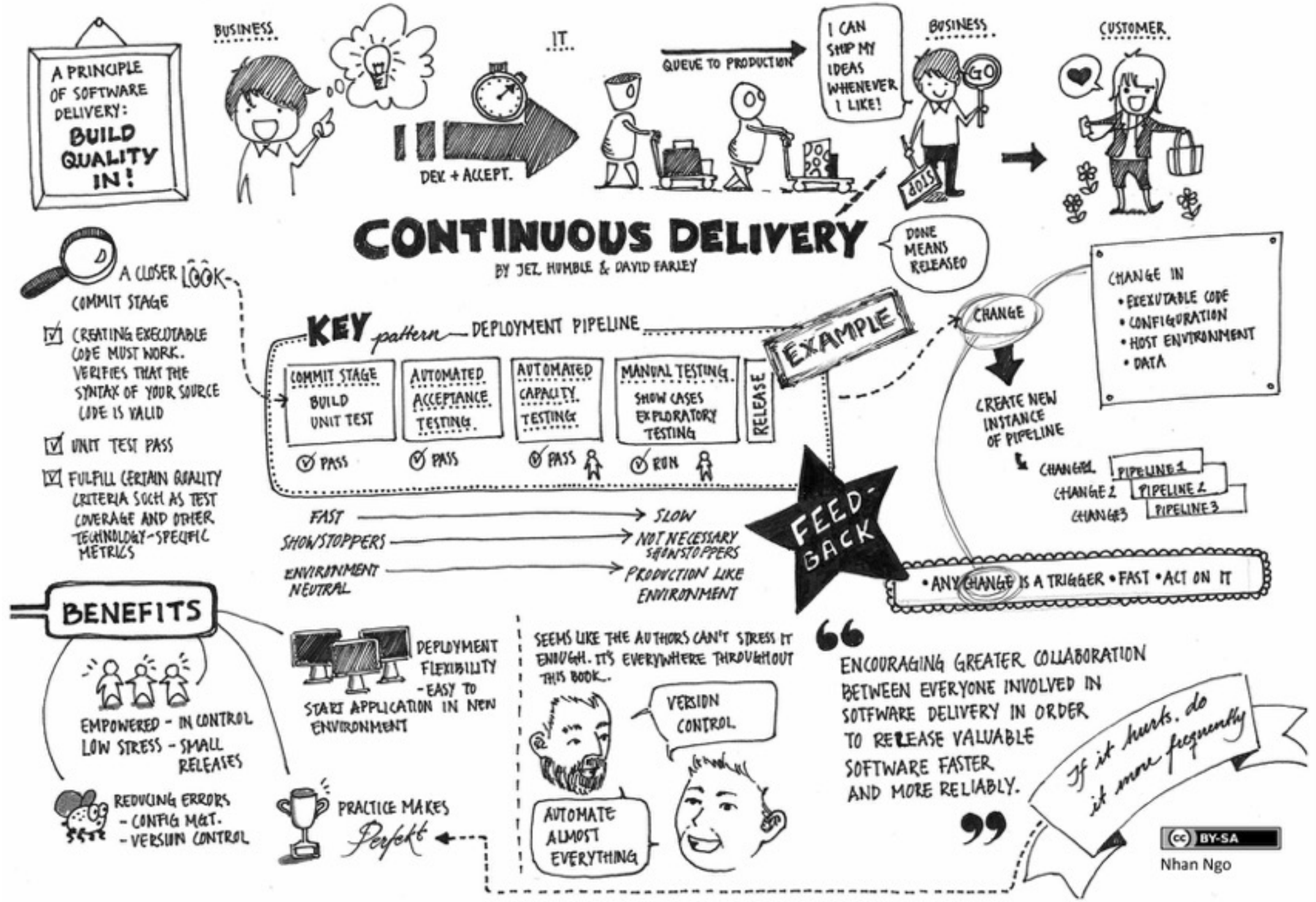
Source: https://dzone.com/articles/devops-20

THE PHOENIX PROJECT BOOK



Source: https://www.amazon.com/Phoenix-Project-DevOps-Helping-Business/dp/0988262592

CONTINUOUS DELIVERY



Source: https://continuousdelivery.com/2014/02/visualizations-of-continuous-delivery/

DONE MEANS RELEASED

The definition of "Done" and "Working Software" changes from something that is coded and tested and ready to demo to something that is working in production – now.

- *Feature should be covered by automatic tests*
- *Update and Rollback tools are inseparable part*

Source: https://dzone.com/articles/devops-isnt-killing-developers

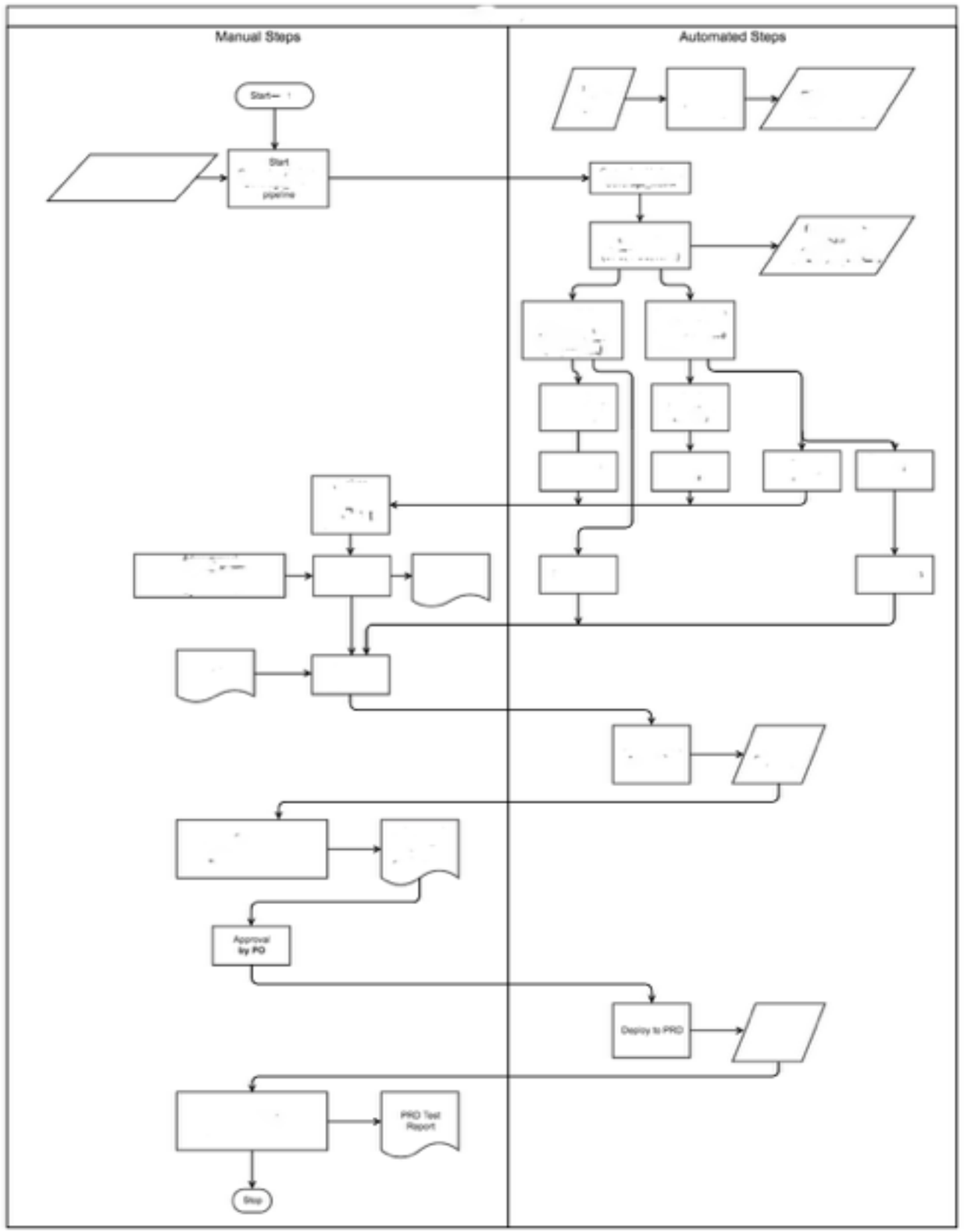
OLD SCHOOL

VALUE STREAM MAPPING

is a lean-management method for analyzing the current state and designing a future state for the series of events that take a product or service from its beginning through to the customer.

Source: https://en.wikipedia.org/wiki/Value_stream_mapping

VALUE STREAM MAPPING



DOCUMENT AS-IS, NOT AS YOU WANT IT TO BE

If you're already doing something but there is no formal process for it, don't try to create a process around it and change it at the same time.

For one, you'll be surprised how often how you think things are done and how they are actually done are different.

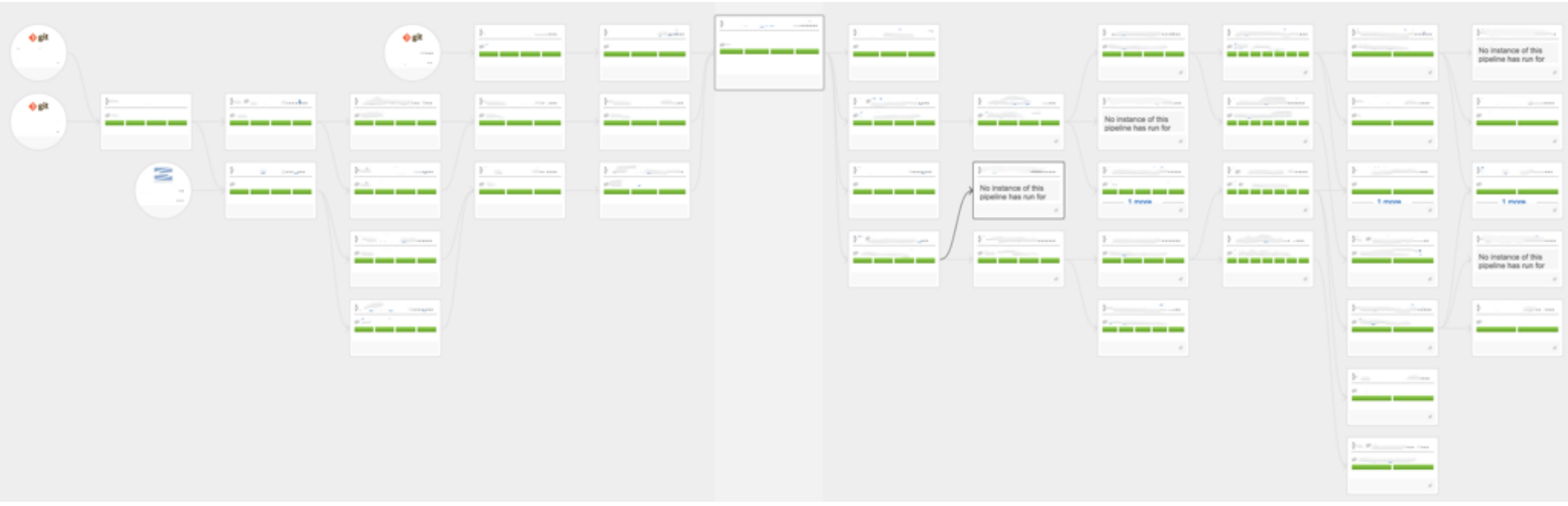
Source: https://pilot.co/blog/minimum-viable-process/

CONWAY'S LAW

> Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.
>
> *-Melvin Conway, 1968*

Source: https://en.wikipedia.org/wiki/Conway%27s_law

DELIVERY PIPELINE



DELIVERY PIPELINE TOOLS

GOCD

JENKINS PIPELINES

Source: https://wiki.jenkins-ci.org/download/attachments/2916393/logo.png
https://api.go.cd/current/images/logo.png

CONTINUOUS TESTING

is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate.

Source: https://en.wikipedia.org/wiki/Continuous_testing

AUTOMATE EVERYTHING

really refers to error-prone manual efforts around deployment, and the provisioning, cloning, and sharing of environments. Automating these frees up countless hours for manual efforts that positively impact quality like exploratory testing and collaboration.

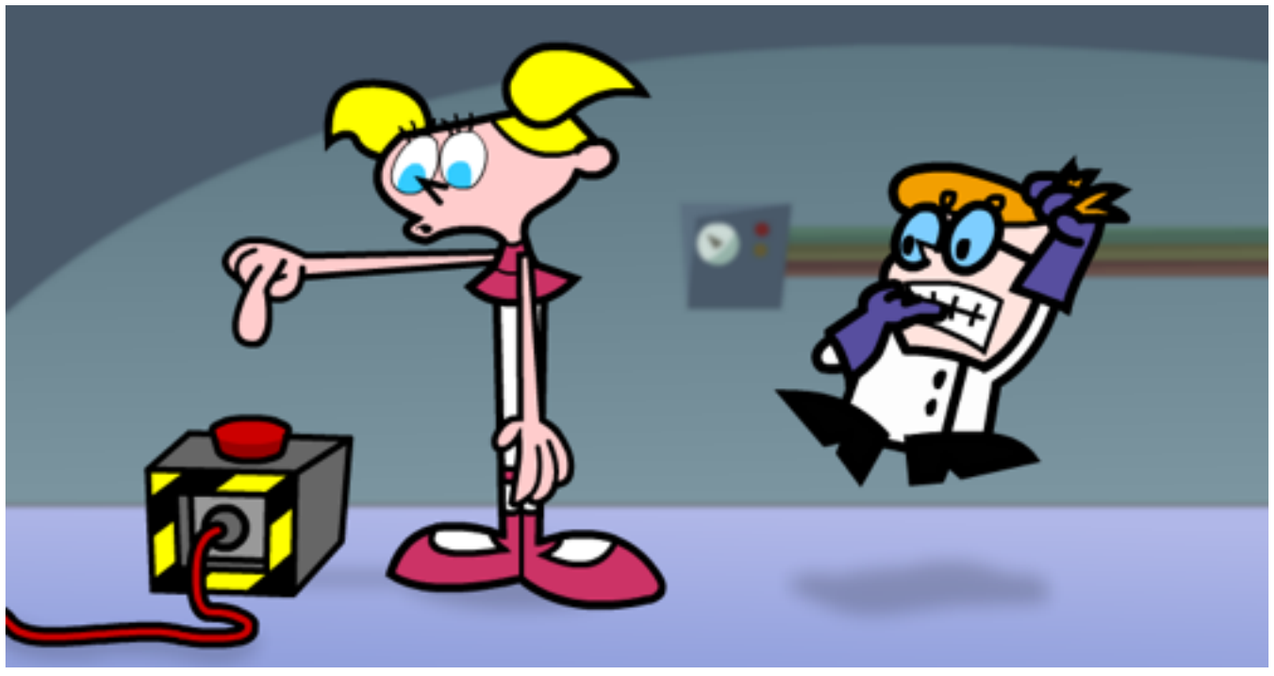
Source: https://www.skytap.com/blog/automate-everything-challenged-in-recent-devops-com-article/

MINIMUM VIABLE PROCESS

If you're introducing an entirely new process, start with something simple. You'll spend less time monitoring it and it will be adopted quicker. Simple processes are also easier to learn from, because they are easier to analyze.

Source: https://pilot.co/blog/minimum-viable-process/

BUILD-TEST-DEPLOY IN ONE STEP



Source: https://100ro.blogspot.com/2009/11/uu-what-does-this-button-do_27.html

ChatOps

is a collaboration model that connects people, tools, process, and automation into a transparent workflow. This flow connects the work needed, the work happening, and the work done in a persistent location staffed by the people, bots, and related tools.

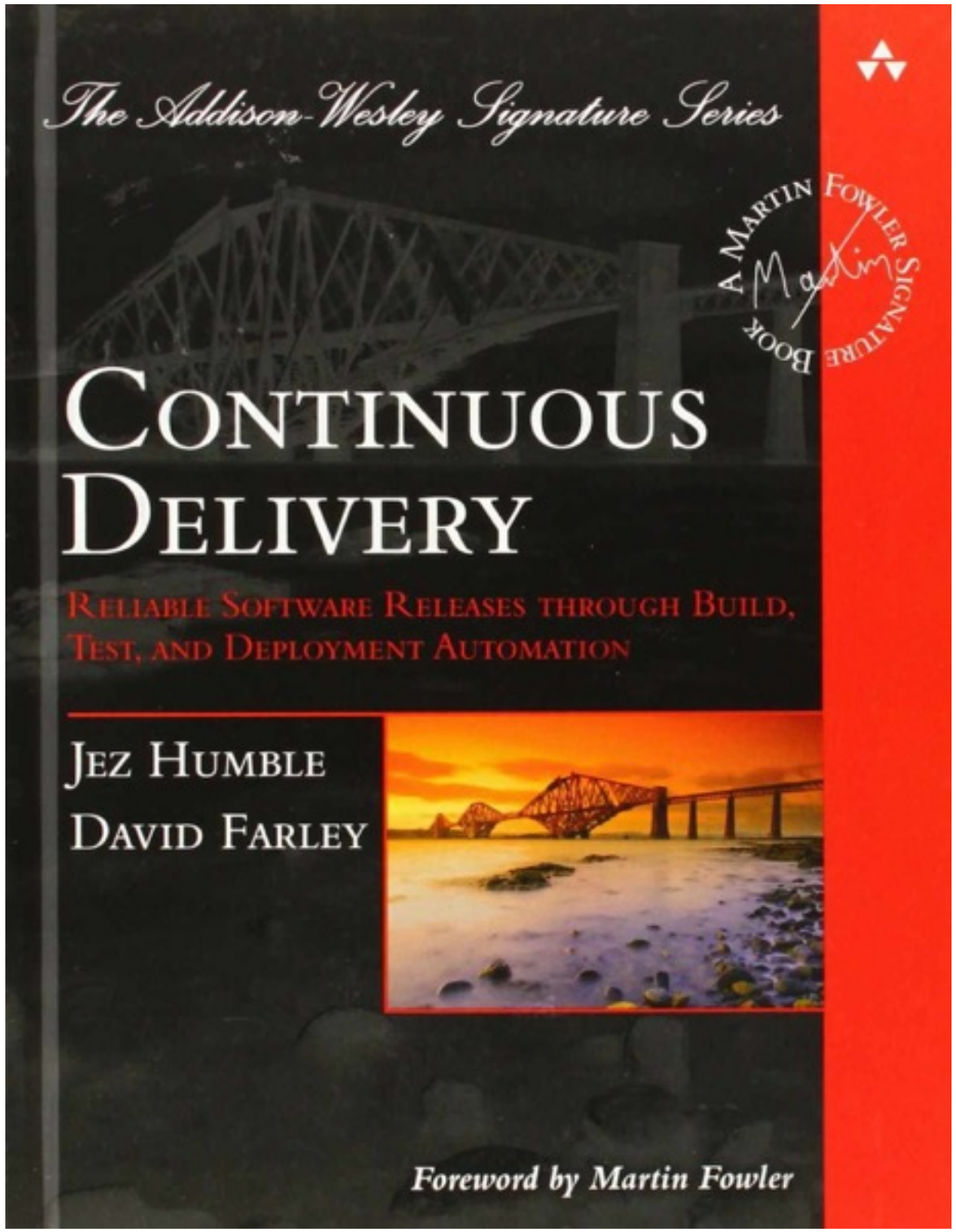
Source: https://blogs.atlassian.com/2016/01/what-is-chatops-adoption-guide/

ChatOps TOOLS

LITA COG HUBOT

Source: https://victorops.com/chatops/

CONTINUOUS DELIVERY BOOKS



Source: https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912/

CLOUDS

Source: http://all-silhouettes.com/vector-clouds/

CLOUDS

> Ideally test environments can be spun up and down quickly and are allocated on-demand. This gives the lowest cost and the highest throughput. Hosting these environments in the cloud is therefore ideal.

Source: https://www.ravellosystems.com/blog/continuous-integration-deployment-test-automation/

CLOUDS ARE KILLING DEVOPS

Amazon Web Services and other managed service providers have allowed for a dramatically simplified way of working, reducing complexity on the developer end and, thus, allowing them to focus on software development instead of installing databases and ensuring processes like backup, redundancy and uptime.

In other words, managed services removed a lot of headaches with which DevOps teams were forced to deal.

Source: https://techcrunch.com/2016/04/07/devops-is-dead-long-live-devops/

CLOUDS

Source: https://plus.google.com/+amazonwebservices/posts/dnqT2UBbfzG

INFRASTRUCTURE AS CODE

is the process of managing and provisioning computing infrastructure (processes, bare-metal servers, virtual servers, etc.) and their configuration through machine-processable definition files, rather than physical hardware configuration or the use of interactive configuration tools.

Source: https://en.wikipedia.org/wiki/Infrastructure_as_Code

INFRASTRUCTURE AS CODE: TERRAFORM

- AWS
- Azure
- CloudStack
- DigitalOcean
- Docker
- Google Cloud
- Heroku
- OpenStack
- Parallels
- QEMU
- VMware

Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently.

Configuration files describe to Terraform the components needed to run a single application or your entire datacenter.

The infrastructure Terraform can manage includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc.

Source: https://www.terraform.io/docs/providers/index.html

IMAGE CREATION: PACKER

- Amazon EC2
- Azure
- DigitalOcean
- Docker
- Google Cloud
- OpenStack
- Parallels
- QEMU
- VirtualBox
- VMware

Packer is a tool for building identical machine images for multiple platforms from a single source configuration.

Source: https://github.com/mitchellh/packer

TEST-DRIVEN INFRASTRUCTURE

In software development, Test Driven Development (TDD) is well recognized for improving design, increasing code quality, and allowing refactoring and better knowledge sharing.

Similar benefits can be gained in infrastructure projects when infrastructure is treated as code, driving that code development with tests.

Source: https://spin.atomicobject.com/2014/10/28/test-driven-infrastructure-tdi/

TEST-DRIVEN INFRASTRUCTURE

```
monitor::url { "Url-Example42_TestDatabase":
    url      => "http://www.example42.com/testdb.php",
    port     => '80',
    target   => "${fqdn}",
    pattern  => 'Database OK',
    enable   => "true",
    tool     => "${monitor_tool}",
}

monitor::mount { "/var/www/repo":
    name    => "/var/www/repo",
    fstype  => "nfs",
    ensure  => mounted,
    options => "defaults",
    device  => "nfs.example42.com:/data/repo",
    atboot  => true,
}
```

Source: http://www.example42.com/2011/02/28/test-driven-puppet-infrastructures-deployment/

TEST-DRIVEN INFRASTRUCTURE: TEST KITCHEN

- Amazon EC2
- Blue Box
- CloudStack
- Digital Ocean
- Rackspace
- OpenStack
- Vagrant
- Docker
- LXC containers
- Bats
- shUnit2
- RSpec
- Serverspec
- Berkshelf
- Librarian-Chef
- runit

Test Kitchen is an integration tool for developing and testing infrastructure code and software on isolated target platforms.

Source: https://github.com/mitchellh/packer

VERSION EVERYTHING

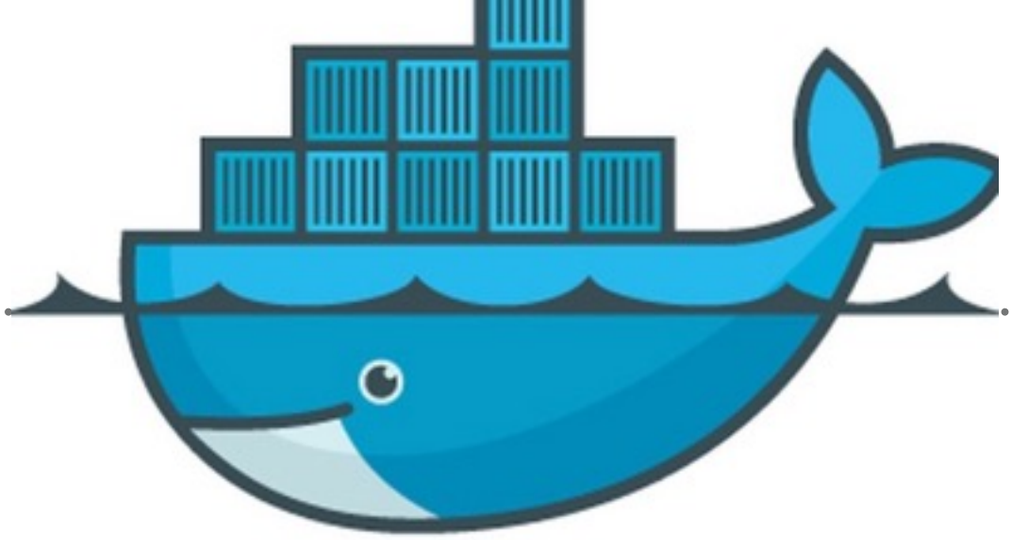
Yes, everything: infrastructure, configuration, application code, and your database. If you do, you have a single source of truth that enables you to view the software system — and everything it takes to create the software — as a holistic unit.

Source: https://www.ibm.com/developerworks/java/library/a-devops6/index.html

ISSUE: MONOLITHIC ARCHITECTURE

Microservices *is a specialization and implementation approach for service-oriented architectures used to build flexible, independently deployable software systems.*

Source: https://en.wikipedia.org/wiki/Microservices



DOCKER: SECURITY IS PROBLEM #1

- *Kernel exploits*
- *Denial-of-service attacks eat shared kernel resources*
- *Container breakouts - root user*
- *Poisoned images*
- *Hardcoded API keys or username/passwords*

Source: https://www.oreilly.com/ideas/five-security-concerns-when-using-docker

CHAOS MONKEY

randomly terminates virtual machine instances and containers that run inside of your production environment. Exposing engineers to failures more frequently incentivizes them to build resilient services.

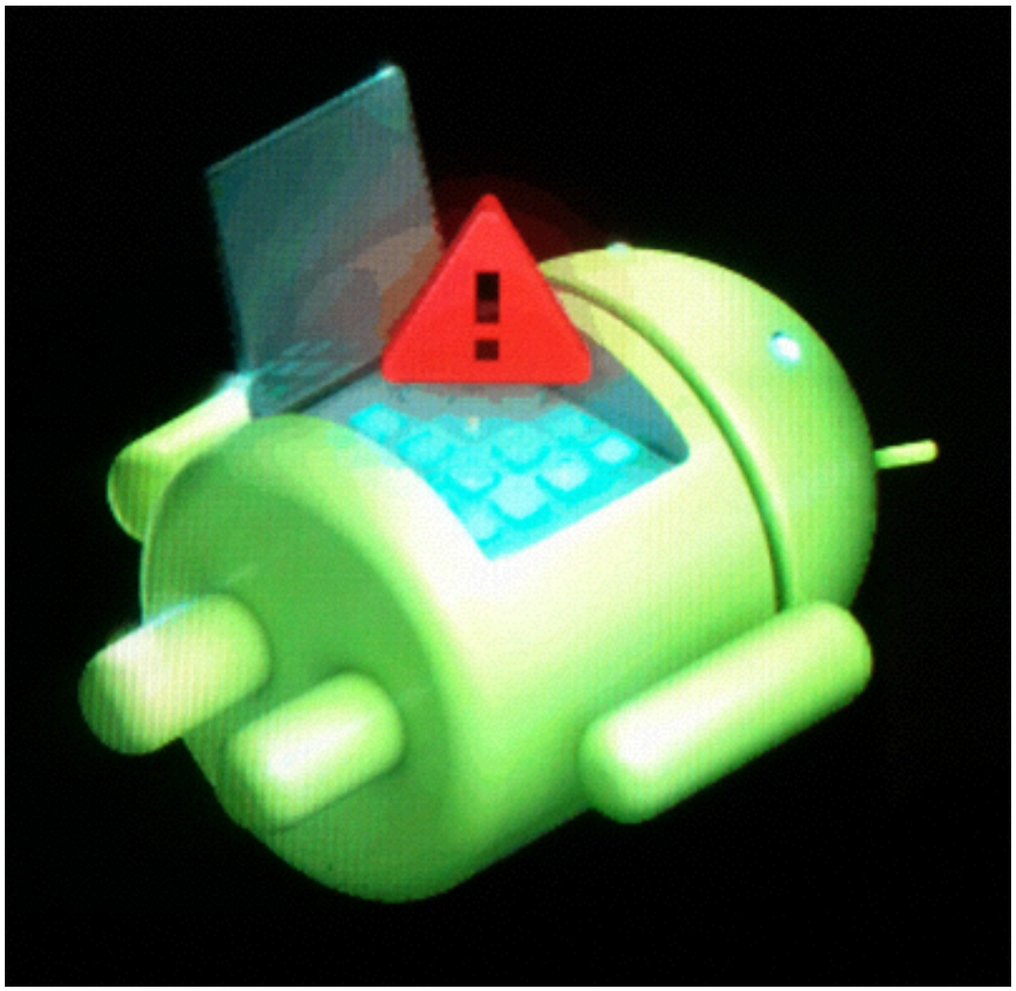
Source: https://github.com/netflix/chaosmonkey

DEVELOPER SELF-SERVICE

in order to achieve success in short-cycled sprints, it is important to have certain processes like building your code or creating a new prototype environment fully automated

Source: https://www.elastic.io/self-service-agile-software-development/

UPDATE STRATEGIES: SOFTWARE



Source: http://forum.xda-developers.com/transformer-prime/help/rck-picture-android-exclamation-mark-t3368018

FAIL FORWARD & ROLLBACK

Learn to accept that failure will happen. Often spending your effort decreasing MTTR (mean time to recovery) as opposed to increasing MTBF (mean time between failures) is a much better investment. Failure is not a question of 'if' but a question of 'when'.

Source: http://blog.lusis.org/blog/2011/10/18/rollbacks-and-other-deployment-myths/

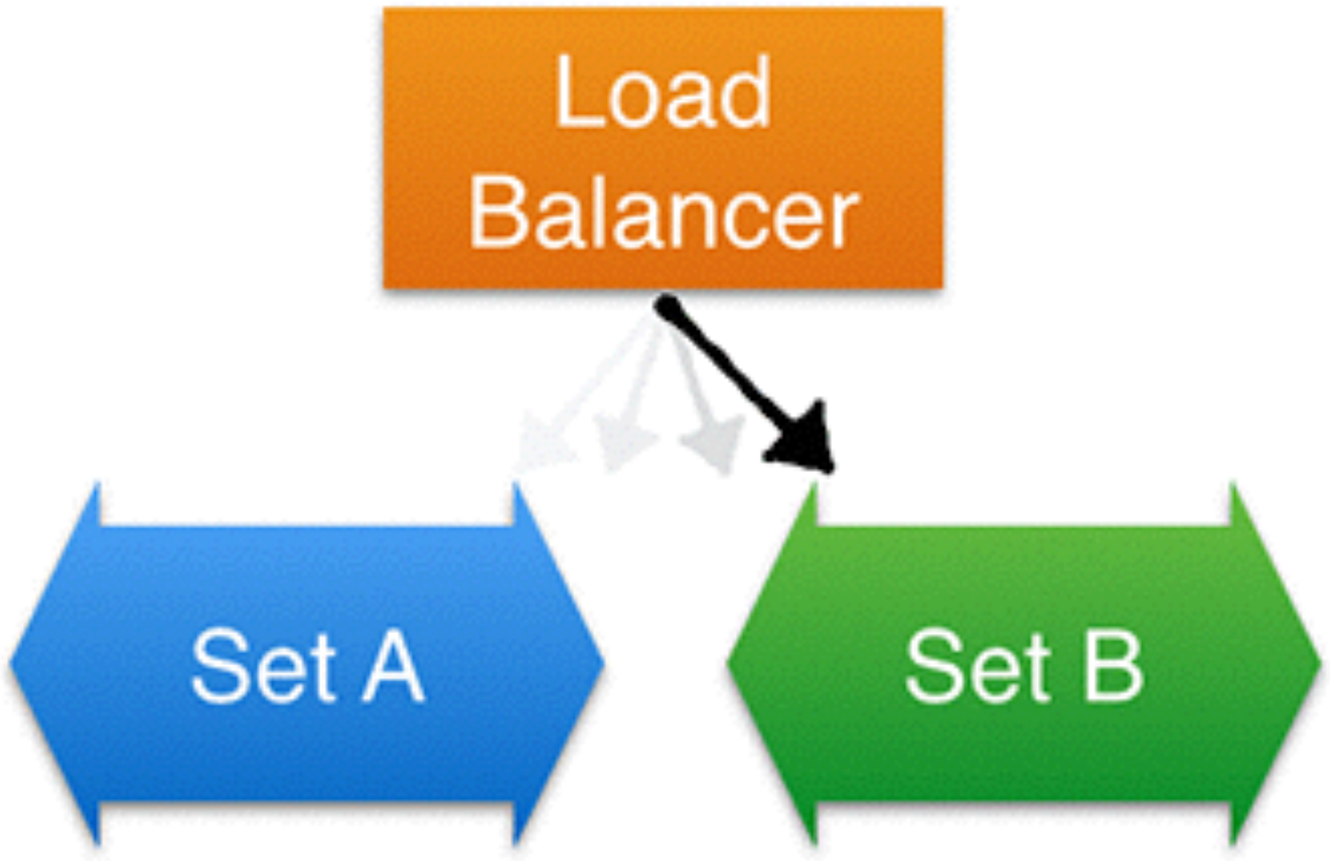
BLUE-GREEN DEPLOYMENT

Blue-green deployment is a release technique that reduces downtime and risk by running two identical production environments called Blue and Green.

At any time, only one of the environments is live, with the live environment serving all production traffic. For this example, Blue is currently live and Green is idle.

Source: https://docs.cloudfoundry.org/devguide/deploy-apps/blue-green.html

BLUE GREEN DEPLOYMENTS



Source: http://depcon.containx.io/docs/bluegreen-deployment

ROLLING UPDATE

To update a service without an outage, updates one pod at a time, rather than taking down the entire service at the same time.

Source: http://kubernetes.io/docs/user-guide/rolling-updates/

FEATURE TOGGLE

is a technique in software development that attempts to provide an alternative to maintaining multiple feature branches, such that the feature can be tested, even before it is completed and ready for release. Feature toggle is used to hide, enable or disable the features, during run time.

Source: https://en.wikipedia.org/wiki/Feature_toggle

DARK LAUNCHING

is a process where software is gradually or stealthily released to consumers in order to get user feedback and test performance. Code is wrapped in a `feature_flag==true` *which is used to control who gets to see the new feature and when.*

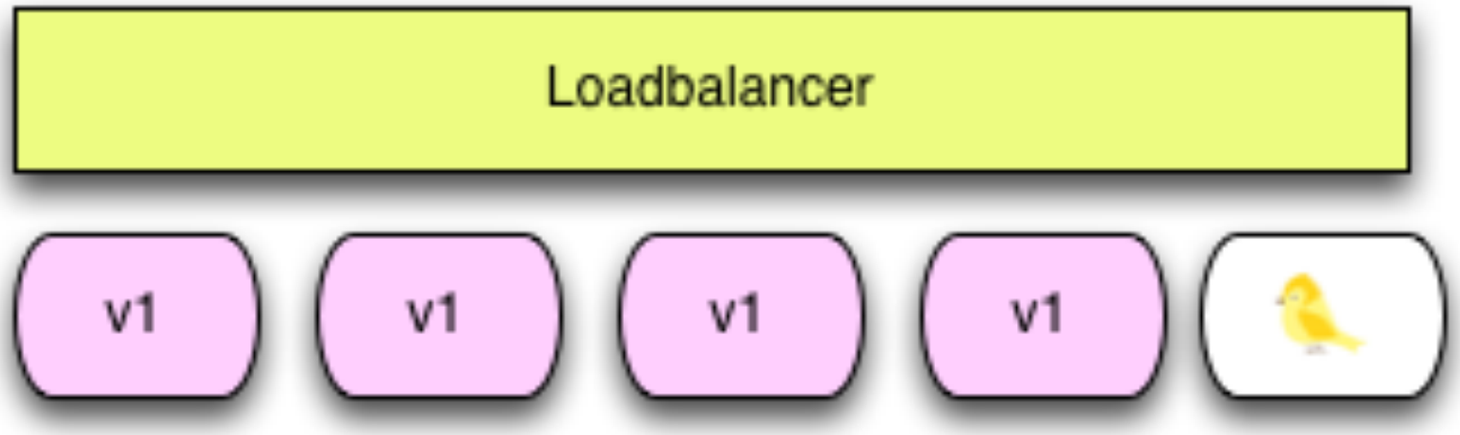
Source: https://www.quora.com/What-is-a-dark-launch-in-terms-of-continuous-delivery-of-software

CANARY RELEASE

A go-live strategy in which a new application version is released to a small subset of production servers and heavily monitored to determine whether it behaves as expected. If everything seems stable, the new version is rolled out to the entire production environment.

Source: https://blog.xebialabs.com/2016/03/21/essential-devops-terms/

CANARY RELEASE



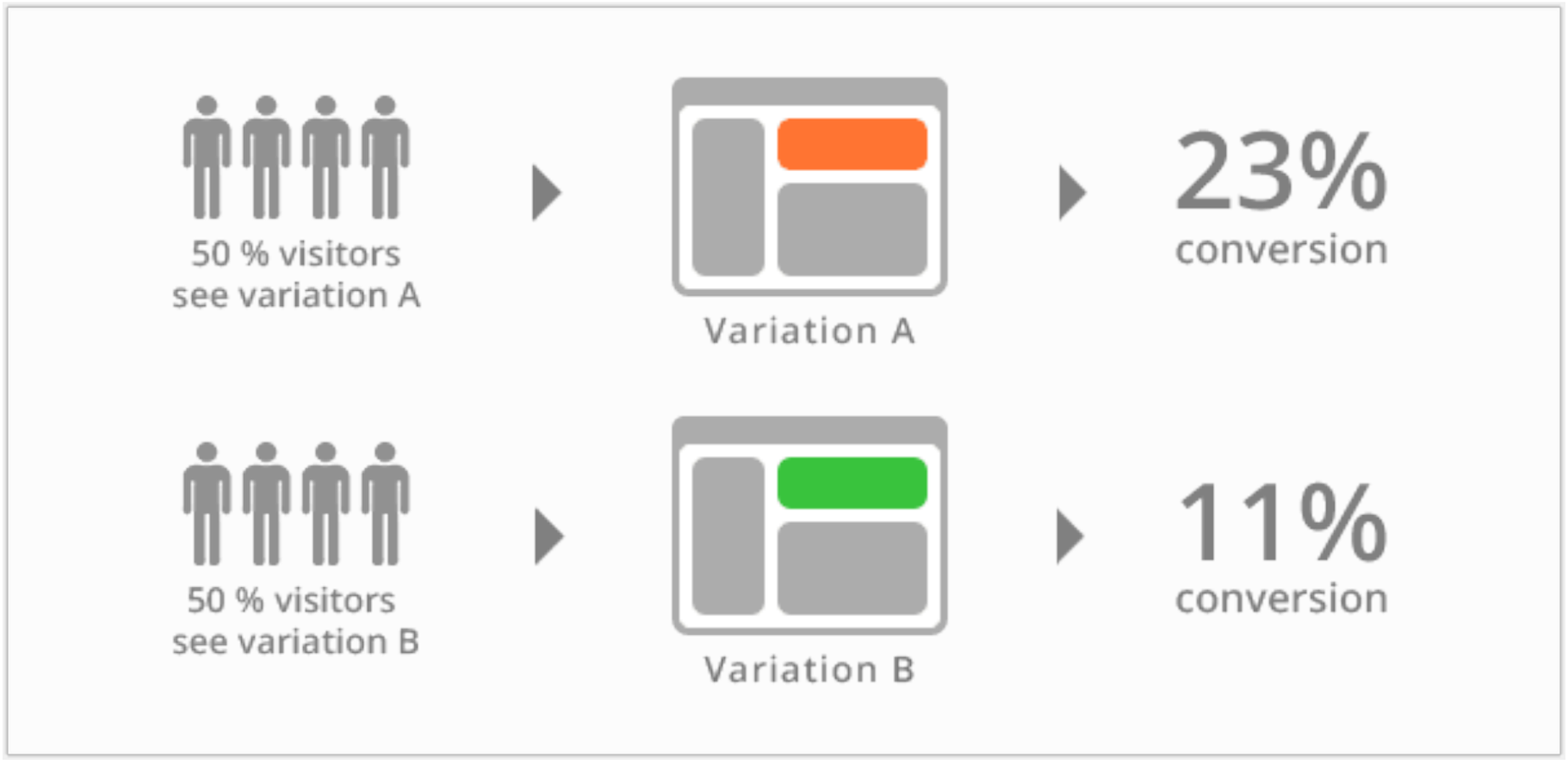
Source: http://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/

A/B TESTING

A technique in which a new feature, or different variants of a feature, are made available to different sets of users and evaluated by comparing metrics and user behavior.

Source: https://xebialabs.com/glossary/a/b-testing/

A/B TESTING

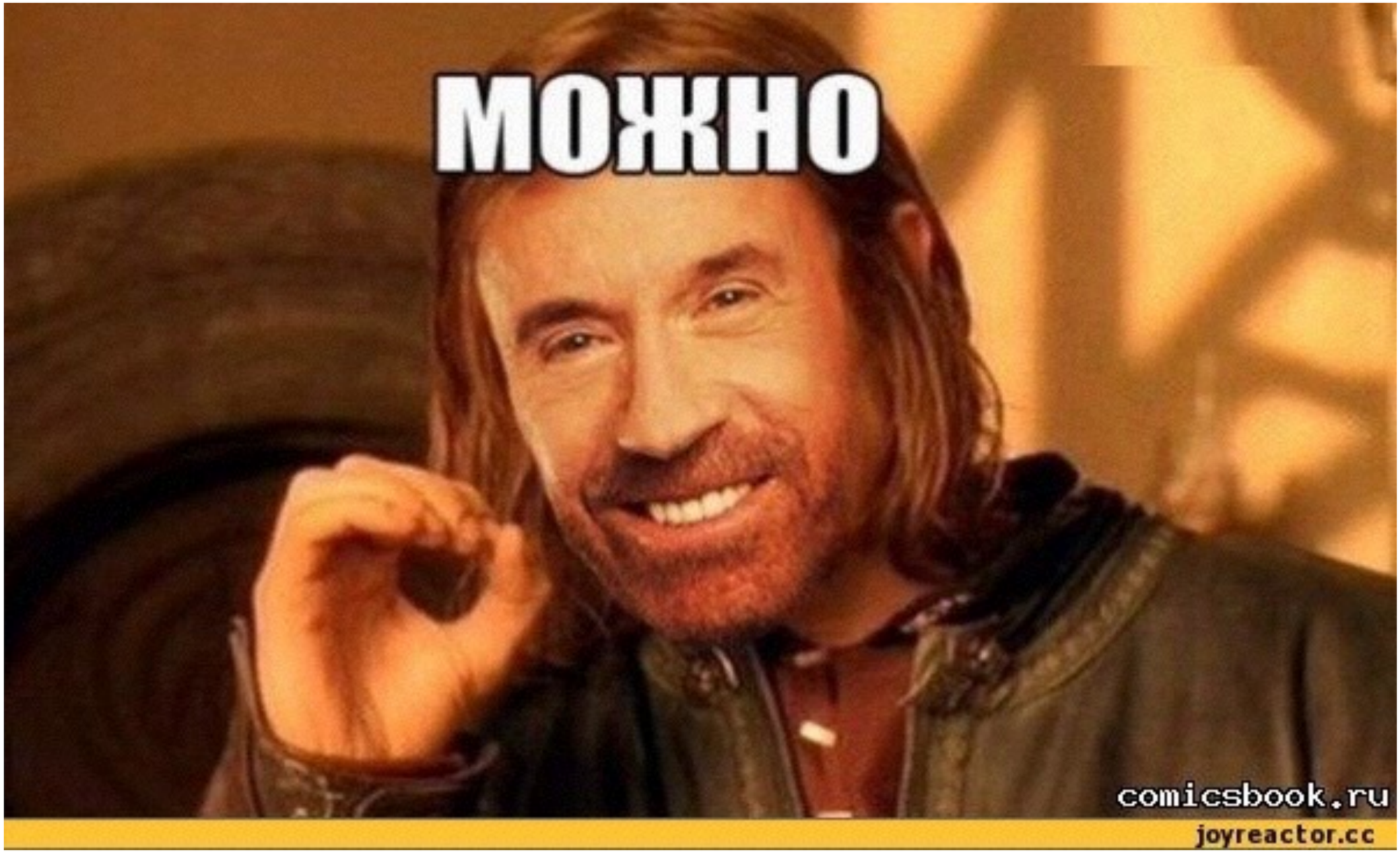


Source: http://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/

UPDATE STRATEGIES: DATABASE

Source: http://buben.biz.ua/buben-oracle.html

CHANGE TYPE/FORMAT OR RENAME A COLUMN



Source: http://cs.pikabu.ru/images/big_size_comm/2013-04_4/13662072079688.jpeg

CHANGE TYPE/FORMAT OR RENAME A COLUMN

- Software version A doesn't know about a new column
- Add new column with NULL value
- Update software to version B:
  writes values into both columns, reads values from an old one
- Fill column with computed value in small batches:
  `UPDATE table SET column=(...) WHERE id<1000;`
- Update software to version C:
  writes values into both columns, reads values from a new one
- Update to version D uses new column only
- Delete column before/after update to version E

ADD A COLUMN

- Software version A doesn't know about a new column
- Add new column with NULL value
- Update software to version B:
  writes values into the column, doesn't read values from it
- Fill the column with computed value by small batches
  UPDATE table SET column=(...) WHERE id<1000;
- Update to version C writes and reads values into/from the column

DELETE A COLUMN

- Software version A uses an old column
- Update software to version B:
  writes values into an old column, doesn't read values from it
- Update to version C doesn't use old column
- Delete column before/after update to version D

UPDATE DATABASE TOOLS

- Liquibase
- Flyway
- DBDeploy
- Ruby on Rails Active Record
- Play Framework Evolutions
- Django South

DEVOPS IS KILLING

Source: https://s3.amazonaws.com/rapgenius/1362449233_tumblr_m5bdwuz6wZ1rvzu9do1_500.jpg

DEVOPS IS KILLING THE OPERATIONS TEAM

- *Starting to worry about your OPs job?*
- *Yes, you should worry.*

Source: https://blog.csanchez.org/2012/03/07/is-devops-killing-the-operations-team/

DEVOPS IS KILLING DEVELOPERS

“DevOps” is meant to denote a close collaboration and cross-pollination between what were previously purely development roles, purely operations roles, and purely QA roles. Because software needs to be released at an ever-increasing rate, the old "waterfall" develop-test-release cycle is seen as broken. Developers must also take responsibility for the quality of the testing and release environments.

Source: https://jeffknupp.com/blog/2014/04/15/how-devops-is-killing-the-developer/

DEVOPS ISN'T KILLING DEVELOPERS — BUT IT IS KILLING DEVELOPMENT

Done Means Released! Operational risks become more important than project risks, and operational metrics become more important than project metrics. System uptime and cycle time to production replace Earned Value or velocity. The stress of hitting deadlines is replaced by the stress of firefighting in production and being on call.

Source: https://dzone.com/articles/devops-isnt-killing-developers

DEVOPS IS KILLING QA

Rapid deployment to production doesn't leave time for manual testing or for manual testers, which means developers are responsible for catching all of the bugs themselves before code gets to production – or do their testing in production and try to catch problems as they happen

Source: https://dzone.com/articles/devops-isnt-killing-developers

DEVOPS IS KILLING OUTSOURCING

"To be most effective, enterprises need to own the transformation, and it has to be a truly collaborative effort across disciplines–business leadership, engineering, system administration, security & compliance," says Thomas Enochs, vice president of customer success at Chef. "Collaboration and transformation are difficult to achieve with outside third parties or vendors. They need to own and drive the change themselves, and not be dependent on others."

Source: https://devops.com/features/will-devops-kill-it-outsourcing-part-1/

DEVOPS IS KILLING IT

Waterfall require large QA departments. Old School QA department requires many Manual QA Engineers which is easy start for every new guy in IT.

When you start off by having your engineers run operations you never allow new ops people to start from ground up and develop their skills, learning the pain points as the system grows thus ensuring when you grow to the point that you need a operations engineer the is a shortage of trained people available.

Source: http://mrmiller.nonesensedomains.com/2014/04/15/how-devops-is-killing-the-operations-engineer/

KEEP CALM

DevOps IS TAKING OVER THE WORLD

Mykola Marzhan