Continuous Delivery

Almost Continuous Almost Delivery

Sergej Kurakin

Context

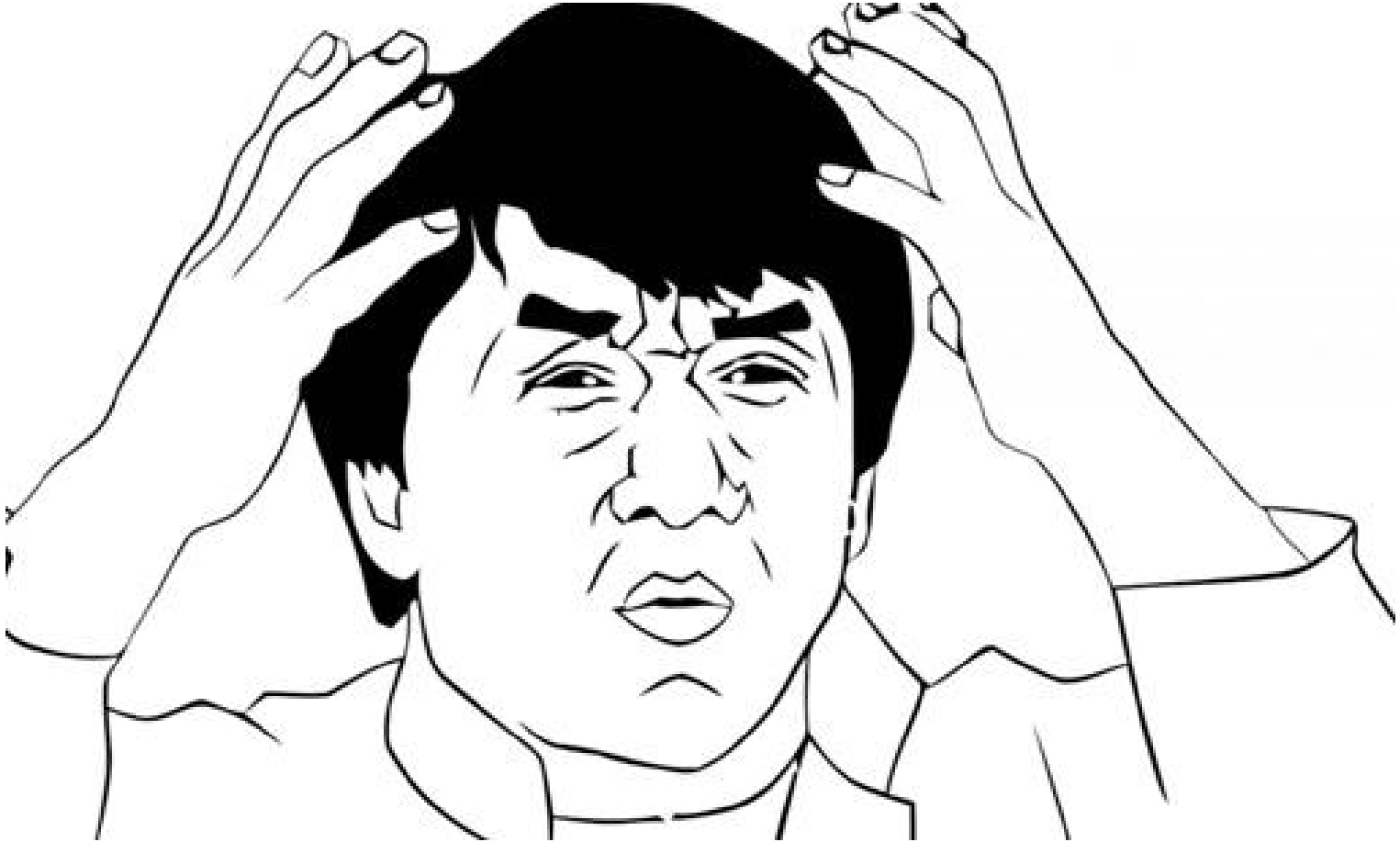
- It is an experience of a small team
- We ain’t live (still in private alpha)
- More challenges to come soon

Everything based on jenkins-php with adoption to our needs.

Checklist for Manual Delivery

Checklist Example

- ❑ Check-out code
- ❑ Run tests
- ❑ Build project
  - ❑ Install non-dev vendors only
  - ❑ Optimize Class Maps
  - ❑ Build SASS/SCSS/LESS
  - ❑ Build CoffeeScript/TypeScript
  - ❑ Build Sprites
  - ❑ Combine and Minimize
  - ❑ Encode files
- ❑ Prepare for upgrade
  - ❑ SQL database backup
  - ❑ NoSQL storage backup
- ❑ Upload changes to server
- ❑ Verify that upload was successful
- ❑ Extract database migrations
- ❑ Run database migrations
- ❑ Warmup application cache
- ❑ Switch version
- ❑ Flush PHP and NGINX cache
- ❑ Restart NodeJS nodes



Server Access

Any team developer should be able to deliver changes. Or a part of the team.

What if anyone fails and panics? What about human mistakes?

Continuous Delivery

Wikipedia: Continuous Delivery is a design practice used in software development to automate and improve the process of software delivery.

http://en.wikipedia.org/wiki/Continuous_delivery

It is all about Automation

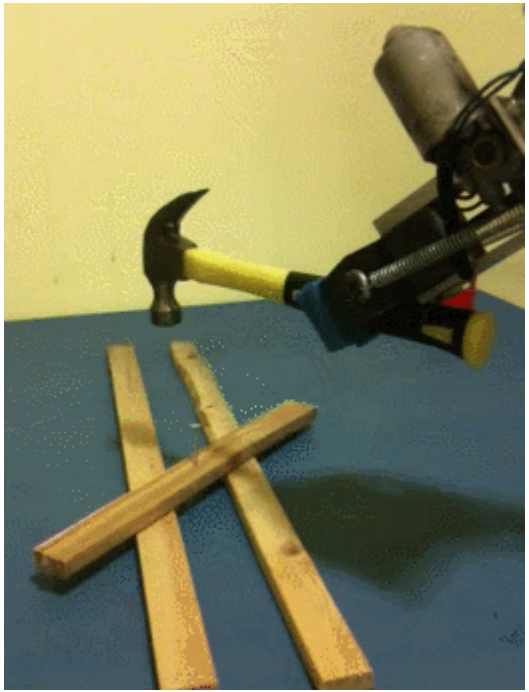
- Automated testing
- Continuous integration
- Automated deployment

- Automated static code analysis

Yes, it scares

http://devopsreactions.tumblr.com/post/101664094339/the-fear-of-automated-deployments

Our first step



http://devopsreactions.tumblr.com/post/112866727315/first-stages-of-automation

Some of our steps

http://devopsreactions.tumblr.com/post/68152221988/trying-to-automate-things-sometimes

Automation base: Jenkins CI

- http://jenkins-ci.org/
- Open Source
- MIT License
- JDK 1.6 Required

Jenkins CI Roles

- Check VCS for updates
- Check-out code
- Tasks triggering
- Scheduled tasks
- Access rights inside team
- Process orchestration

Automated Testing

Unit Tests

- PHPUnit
- Prophecy
- Jasmine

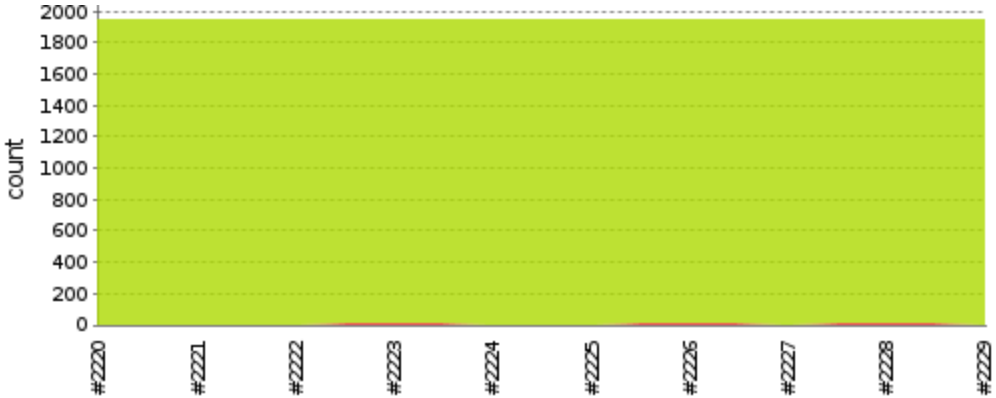
Functional Tests

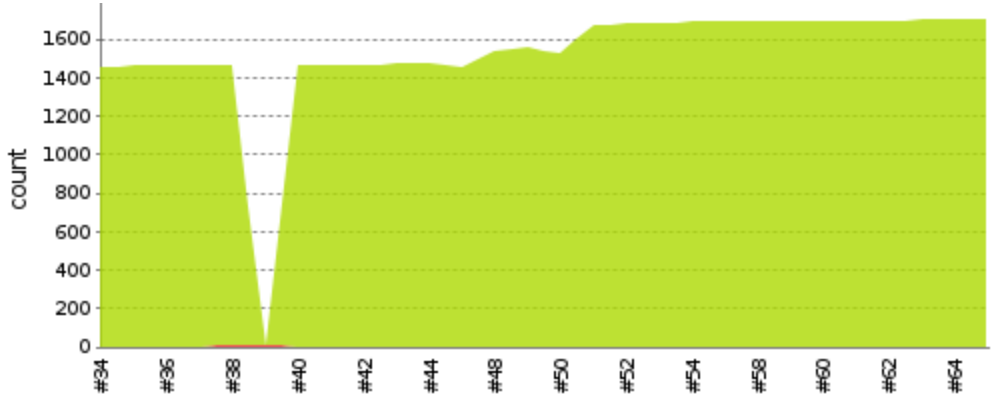
- PHPUnit + fastest
- Symfony 2 standard toolset
- Doctrine Fixtures

Continuous Integration

- Checkout code
- Install dependencies and tools
- Run phplint
- Run DB migrations
- Run all tests
- Run delivery test

You need trends images





Automated Deployment

- Build process
- Code upload
- Database schema sync
- Cache warmup
- Switch to new Version

Build process with Phing

- Full copy of project's code
- Dump/minimize of all assets
- Optimize autoload
- Remove any dev-related stuff

Code upload with Capistrano v2

- SSH + tar.gz (:deploy_via :copy)
- Multi-stage deployment
- Custom tasks added

Database schema sync

- Doctrine Migrations
- Capistrano custom task
- No backups...

Cache warmup

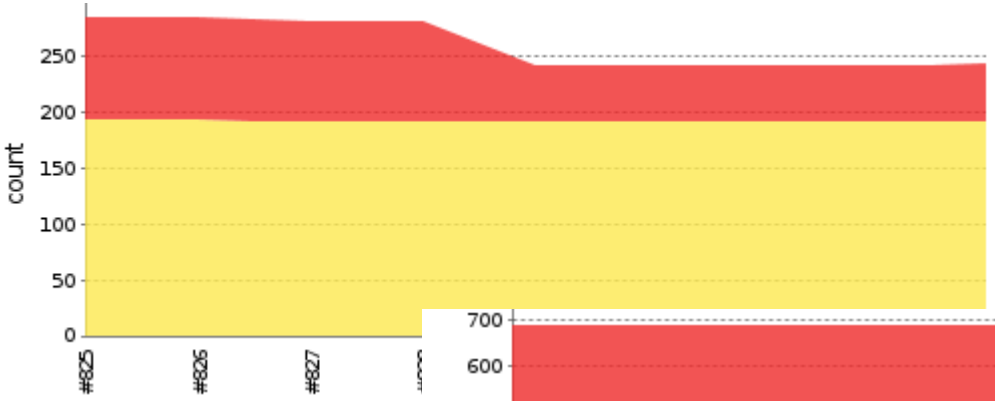
- Standard Symfony 2 cache warmup
- Custom Capistrano task

Version switch

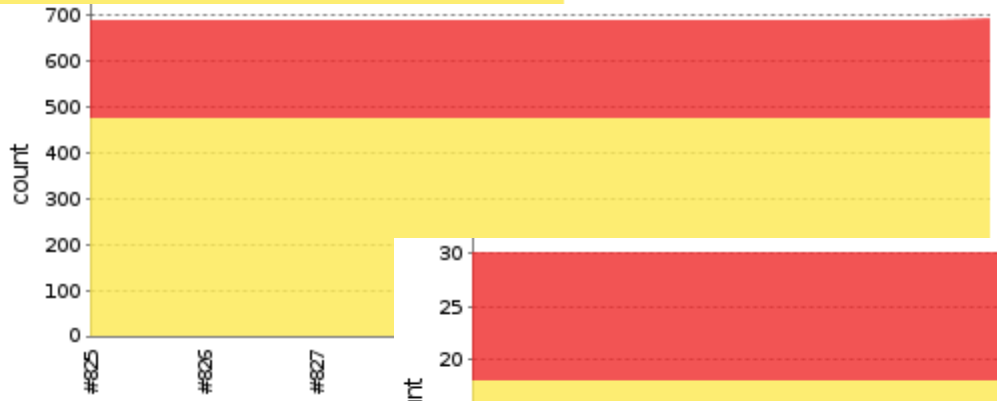
- Standard Capistrano task
- php-fpm reload (special sudo command)
- nodejs reload (special sudo command)
- nginx reload (special sudo command)

Automated static code analysis

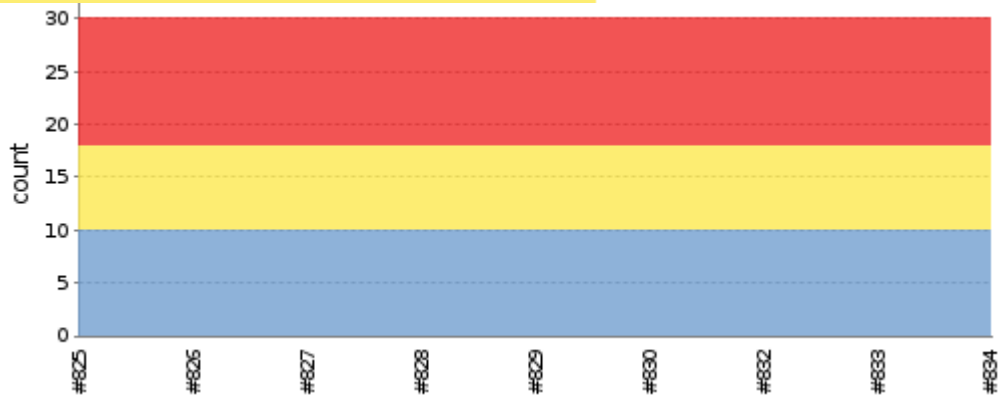
- PHP Code Documentation (phpdox)
- API Documentation (raml)
- PHP Code Sniffer (phpcs)
- PHP Mess Detector (phpmd)
- PHP Copy/Paste Detector (phpcpd)
- Reports published by Jenkins CI



PHP Code Sniffer



Mess Detector



Copy Paste

PHP Mess Detector

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
690	1	0

Summary

Total	High Priority	Normal Priority	Low Priority
690	217	473	0

Details

Packages | Files | Categories | **Types** | Warnings | Details | New | High | Normal

Type	Total	Distribution
BooleanArgumentFlag	62	
CamelCaseVariableName	1	
CouplingBetweenObjects	24	
CyclomaticComplexity	40	
DepthOfInheritance	2	
ElseExpression	98	
ExcessiveClassComplexity	20	
ExcessiveClassLength	3	
ExcessiveMethodLength	24	
ExcessiveParameterList	9	
ExcessivePublicCount	8	
ExitExpression	1	
LongVariable	120	
NPathComplexity	32	
NumberOfChildren	3	
ShortVariable	49	
StaticAccess	26	
TooManyFields	13	
TooManyMethods	21	
UnusedFormalParameter	49	
UnusedLocalVariable	85	
Total	690	

PHP Code Sniffer

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
243	2	0

Summary

Total	High Priority	Normal Priority	Low Priority
243	51	192	0

Details

Folders | Files | Categories | **Types** | Warnings | Details | New | High | Normal

Type	Total	Distribution
ContentAfterOpenBracket	1	
NotAtEnd	8	
SpaceAfterFunction	1	
SpacingAfterOpenBrace	41	
TooLong	192	
Total	243	

Process

- develop
  - Tests, Build, Deploy to Testing server
  - PHP QA Tools
  - PHP and API documentation
- release
  - Tests, Build, Deploy to Preview server
- master
  - Tests, Build, Deploy to Live server

If you want to start

- Write unit tests
- Write functional tests
- Automate deployment
- Install Jenkins CI and try

Use virtualization for your first steps. Use something that is easy to deploy.

Jenkins Plugins

Jenkins PHP Template

Phing

Email-ext plugin

ANSI Color

Green Balls

Gravatar

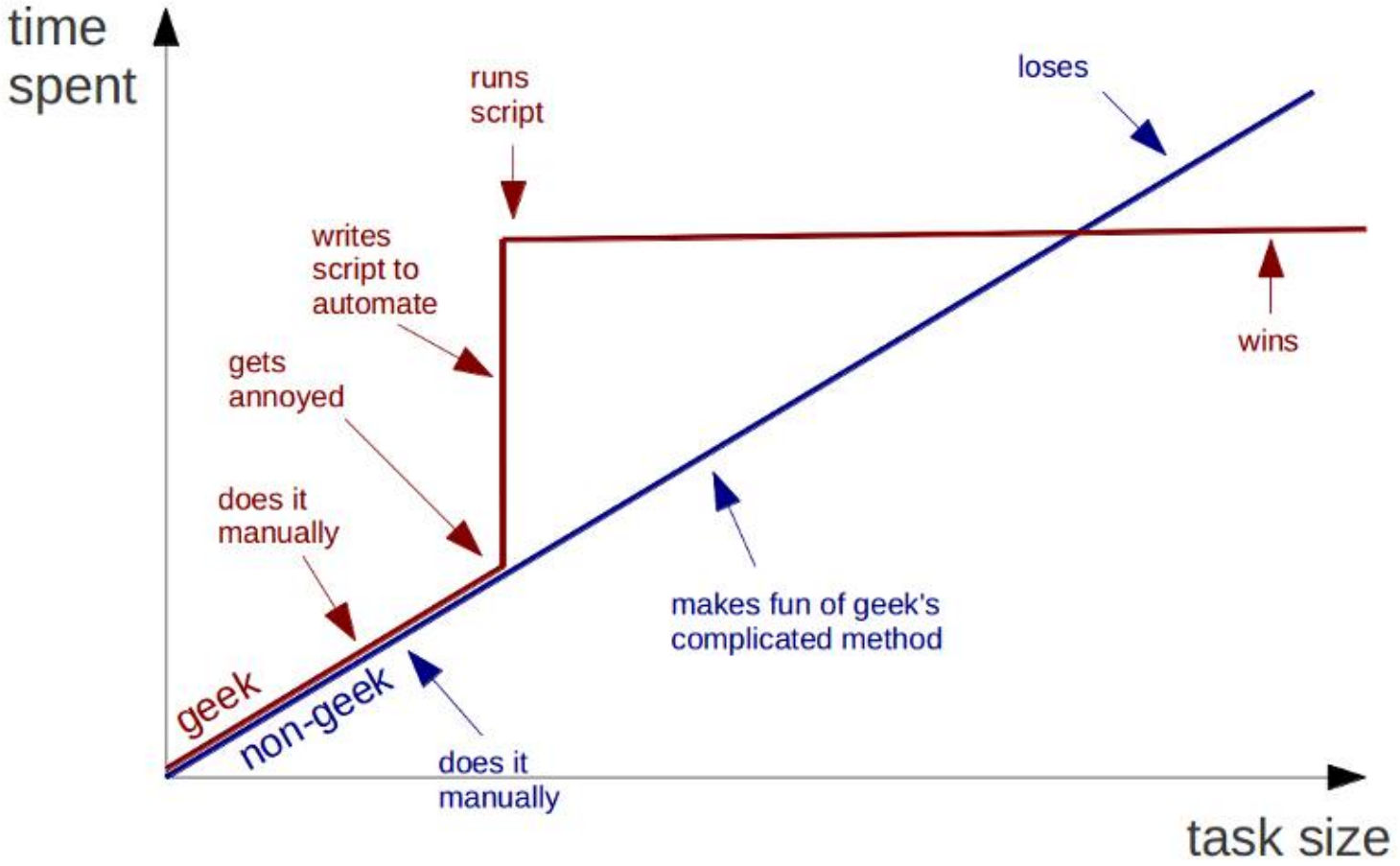
Benefits

- Higher software quality
- Higher code quality
- Machines do most routine tasks
- Standardized deploy process
- You deploy faster and more often

Challenges

- Find time to learn
- Find server to run
- Get used to fail at start
- Your teammates
- Your manager

Geeks and repetitive tasks



Prepare for dirty job

Sometimes you will need

```
$ sudo -i
# su - jenkins

$ cd /var/lib/jenkins
```

Security

- Don’t run Jenkins under “root”
- Enable at least “Matrix-based Security”
- Update often or run LTS
- Update plugins regularly
- Put NGINX with SSL in front of Jenkins
- Use instance per-team or per-project

When everything works

http://devopsreactions.tumblr.com/post/48601970709/whole-suite-of-automated-tests-passed

How sometimes it looks like

http://devopsreactions.tumblr.com/post/99634417776/demonstrating-end-to-end-automation-to-new

Our Upcoming Challenges

- Do something with Migrations
- Reduce Fixtures memory usage
- Tests per feature-branch
- Organize Back-ups
- Launch it Live

Questions?

Sergej Kurakin

Work Email: sergej.kurakin@nfq.lt
Personal Email: sergej@kurakin.info
https://www.linkedin.com/in/sergejkurakin

Special thanks to authors of all pictures used in this presentation.