



# CONTINUOUS DEPLOYMENT

na przykładzie aplikacji w Django



**Wojciech Lichota** - STX Next  
Kwiecień 2016



# Wojciech Lichota



STX Next

[wojciech@lichota.pl](mailto:wojciech@lichota.pl)

[@wlichota](#)

<http://lichota.pl>



OTWIERAMY  
SIĘ NA  
ŁÓDŹ

**STX**NEXT  
agile / programming / teams



1. Wstęp

2. Workflow

3. Narzędzia

4. Wyzwania

1. Wstęp

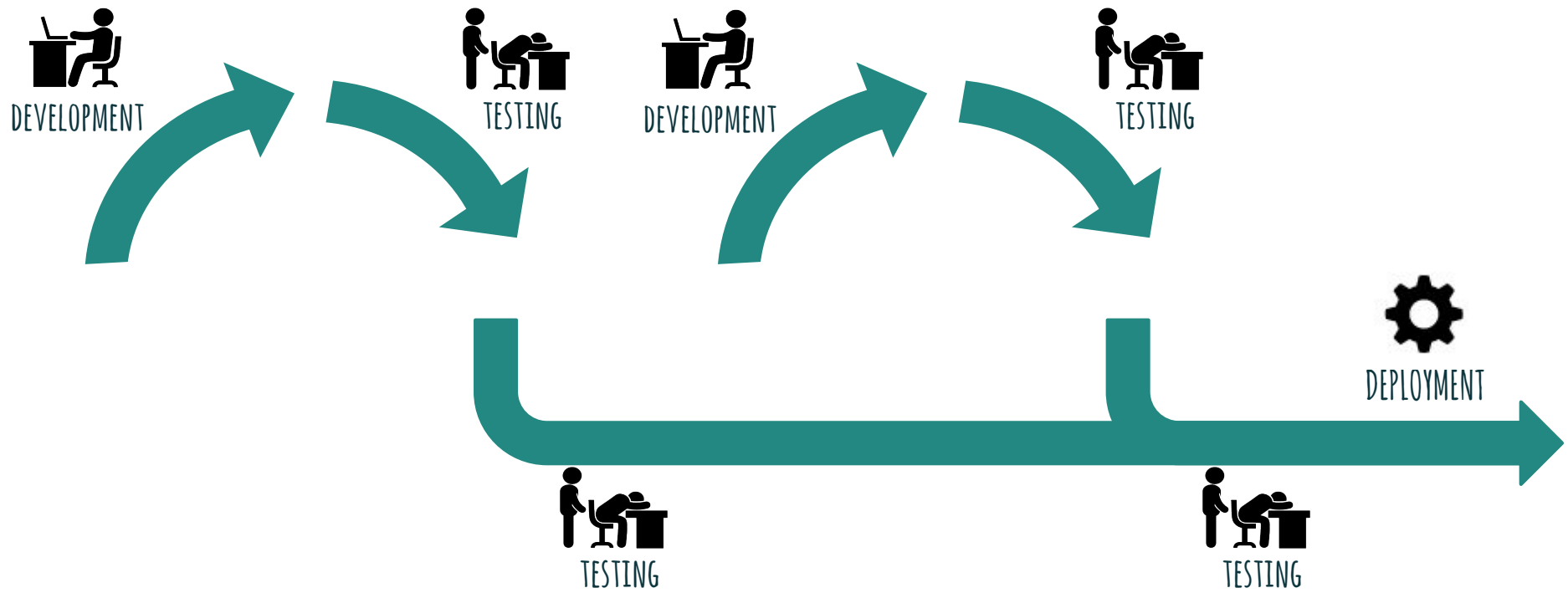
2. Workflow

3. Narzędzia

4. Wyzwania

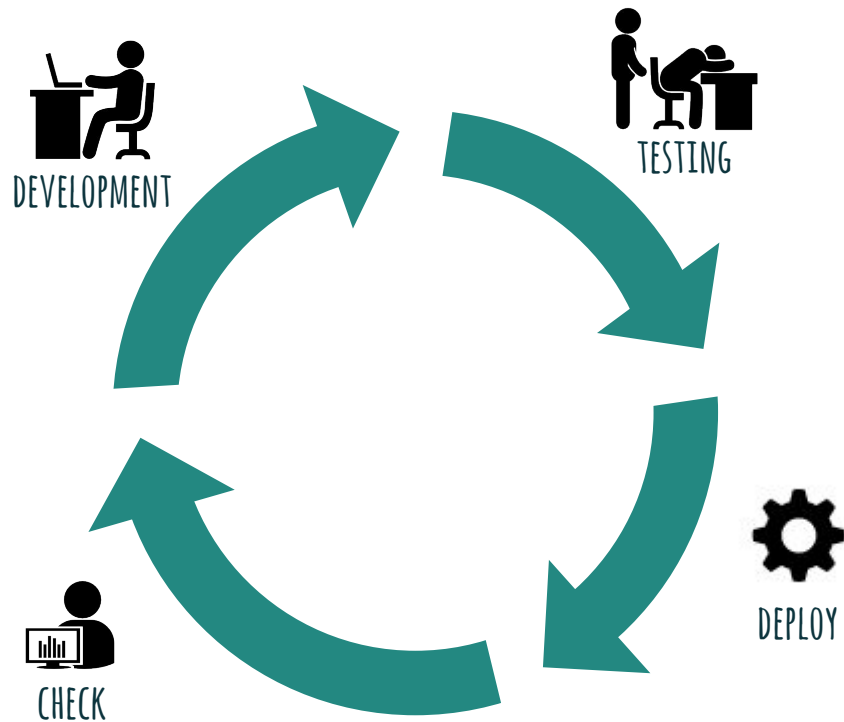


# Iterative Deployment





# Continuous Deployment



Tags

Applicant status

All vacancies

All cities

Age

All experiences

Current work

Latest activity date

Applicant source

clear

create report

Results: 81

12345»

All (507)										
Application (81)										
Confirmation (7)										
Verification (103)										
Interview (95)										
Decision (109)										
Employment (103)										
#	Name	E-Mail / Phone Number	City	Position (Status)	Source	Age	Experience	Creation Date	Latest Activity Date	
1.	Van Hernandez <div>Former Employee<div>django</div><div>css</div></div>	test480@stxnext.com 123 880 789	Łódź	Senior Python Developer APPLICATION	Recommendation	39	7	2014-09-30	2014-09-30	2
2.	Nadia Shupe <div>php</div> <div>Employee</div> <div>angular</div>	test478@stxnext.com 123 862 789	Poznań	Front End Developer APPLICATION	Recommendation	25	0	2014-09-30	2014-09-30	1
3.	Marion Bunker <div>php</div> <div>html</div>	test476@stxnext.com 123 179 789	Poznań	QA APPLICATION	Direct search	36	2	2014-09-30	2014-09-30	5
4.	Penny Perze	test465@stxnext.com 123 242 789	Piła	Front End Developer APPLICATION	CV database	39	5	2014-09-30	2014-09-30	1
5.	Patricia Padilla <div>python</div> <div>php</div> <div>html</div> <div>InBox</div> <div>django</div>	test444@stxnext.com 123 433 789	Łódź	Front End Developer APPLICATION	Job fair	21	3	2014-09-30	2014-09-30	2



STX Next RMS

typeahead.js  
Django  
CKEditor factory-boy  
South Django REST Framework  
Bootstrap python-linkedin  
bleach Selectize.js  
python-social-auth PostgreSQL  
Django-REST-Swagger  
nose AngularJS  
Underscore.js

# RMS Team

 **jakubjanuzik** #1  
470 commits / 18,397 ++ / 9,593 --

 **marek-leszczynski** #2  
188 commits / 5,841 ++ / 2,327 --

 **mnowakowska** #3  
157 commits / 6,905 ++ / 2,324 --

 **romanlevin** #4  
117 commits / 4,480 ++ / 2,291 --

 **dpolcyn** #5  
116 commits / 3,950 ++ / 3,667 --

 **KKrisu** #6  
113 commits / 20,790 ++ / 20,382 --


 **alubeck** #7  
107 commits / 21,785 ++ / 11,395 --

 **ArGroc** #8  
95 commits / 6,961 ++ / 4,232 --

 **mrpear** #9  
88 commits / 3,340 ++ / 2,390 --

 **ImEagle** #10  
82 commits / 3,782 ++ / 1,175 --

 **pkucmus** #11  
74 commits / 8,411 ++ / 21,515 --

 **add00** #12  
70 commits / 2,593 ++ / 1,341 --

 **mjelonek92** #13  
55 commits / 1,789 ++ / 1,068 --

 **lukaszjagodzinski** #14  
55 commits / 43,288 ++ / 22,084 --

 **bitrut** #15  
42 commits / 1,810 ++ / 947 --

 **avalanchy** #16  
38 commits / 11,246 ++ / 1,941 --

 **nihilifer** #17  
30 commits / 538 ++ / 136 --

 **Exorades** #18  
22 commits / 718 ++ / 835 --

 **romankierzkowski** #19  
19 commits / 2,086 ++ / 1,215 --

 **sargo** #20  
14 commits / 217 ++ / 129 --

 **Alkemic** #21  
10 commits / 339 ++ / 106 --

 **AcidWeb** #22  
7 commits / 162 ++ / 407 --

 **Gruszks** #23  
7 commits / 348 ++ / 125 --

 **msopocko** #24  
1 commit / 16 ++ / 5 --

 **dasm** #25  
1 commit / 88 ++ / 84 --



1. Wstęp

2. Workflow

3. Narzędzia

4. Wyzwania

# GIT Workflow

- ★ jeden branch główny (master)
- ★ feature branch (oddzielny branch na każdą historyjkę)
- ★ pull requesty

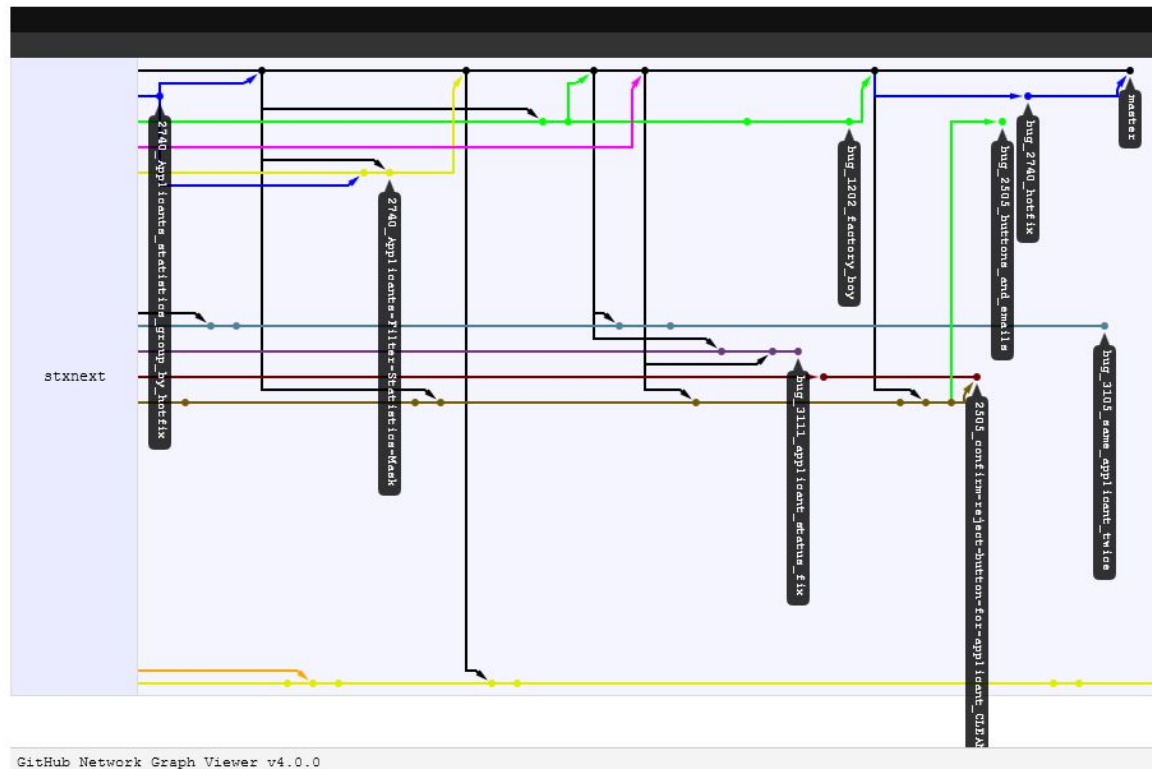
Graph Members

## The rms network graph

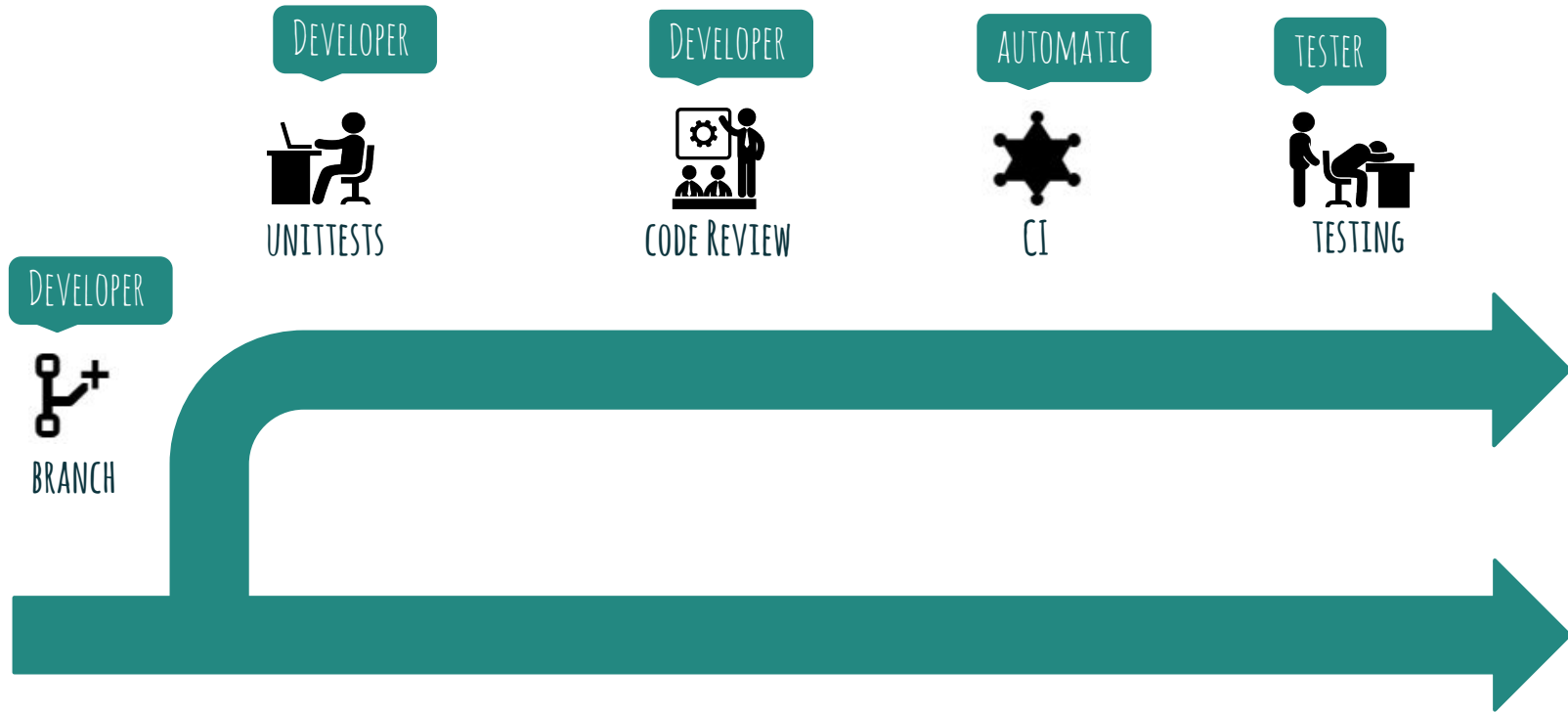
Keyboard shortcuts available

All branches in the network using stxnext/rms as the reference point. [Read our blog post about how it works.](#)

Show Help



# QA Workflow



# Deploy Workflow

TESTER



MERGE

AUTOMATIC



BACKUP

TESTER



DEPLOY

AUTOMATIC  
TESTER  
DEVELOPER



CHECK





1. Wstęp

2. Workflow

3. Narzędzia

4. Wyzwania

Konkurs



Is Python the best programming language?

♦ **A:** Yes

♦ **B:** No



# JENKINS



continuous  
integration

All		RMS				+
S	W	Name ↓	Ostatni Sukces	Ostatni Błąd	Ostatni Czas Trwania	
		<a href="#">RMS-live</a>	1 day 18 hr - <a href="#">#136</a>	—	3 min 43 sec	
		<a href="#">RMS-PR-testing</a>	23 hr - <a href="#">#65</a>	23 hr - <a href="#">#64</a>	1 min 24 sec	
		<a href="#">RMS-staging</a>	1 day 18 hr - <a href="#">#725</a>	—	2 min 21 sec	
		<a href="#">RMS-testing</a>	1 day 18 hr - <a href="#">#10829</a>	—	1 min 58 sec	

## Instancje:

- ★ testing - dla każdego z testerów, na feature branch
- ★ staging - z danymi testowymi
- ★ PR-testing - automatyczne budowanie każdego pull requesta
- ★ live - z danymi produkcyjnymi



# JENKINS



continuous  
integration

## Przydatne wtyczki Jenkinsa

- ★ Violations Plugin - pep8, pylint
- ★ Post build task Plugin - kroki po prawidłowym buildzie (np. uruchomienie Fabrica)
- ★ Matrix Authorization Strategy Plugin - uprawnienia (np. kto może wrzucać na produkcję)
- ★ Github Plugin - automatyczne tagowanie
- ★ GitHub Pull Request Builder - automatyczne budowanie PR

---

## Dodatki do Django wspomagające integrację z Jenkinsem:

- ★ django-nose - raporty odpalenia unittestów (opcja --with-xunit)
- ★ nosecover - raport pokrycia testami (opcja --with-xcover)



# BUILDOUT



build system

- ★ Tworzy reprodukowalne środowisko
- ★ Odseparowane sandboxy bez konieczności ich aktywacji/deaktywacji
- ★ Pozwala na uproszczenie komend (opcje uruchomienia w konfiguracji buildouta)

---

Przydatne recepty:

- ★ `djangorecipe`
- ★ `collective.recipe.template`
- ★ `buildout.recipe.uwsgi`
- ★ `mr.developer`



BOWER  
★  
GRUNT

Bower - zarządzanie zależnościami dla frontend

Grunt - wykonywanie operacji na zasobach dla frontend

---

Przykładowe taski Grunta:

- ★ łączenie plików JS/CSS
- ★ kopiowanie plików (np. obrazków z bibliotek do projektu)
- ★ kompresja JS/CSS
- ★ obfuskacja JS (tylko live)
- ★ uruchomienie jshint, compass





# FABRIC



remote  
execution

- ★ Spięcie kilku komend (np. odpalenie buildout, grunta, restarty serwerów) w jedno polecenie
- ★ Uruchomienie wybranych komend na zdalnych serwerach (poprzez ssh)



# GUNNERY

★  
remote  
execution

- ★ alternatywa dla Jenkins + Fabric
- ★ zarządzanie poprzez przeglądarkę różnymi instancjami na różnych serwerach

GUNNERY

VW UFO  
development

wojciech.lichota

Settings ▾

Help

Log Out

Environment *development*

Dashboard / VW UFO / development

Activity

View full history

development /  
update\_devel\_WIOLETTA  
Wioletta 18 minutes ago

development /  
update\_devel\_KAROLINA  
Karolina 36 minutes ago

development /  
update\_devel\_PIOTR  
Piotr 2 hours ago

development /  
update\_devel\_WIOLETTA  
Wioletta 3 hours ago

Servers +

stx_dev	10.93.1.30	app		
---------	------------	-----	--	--

Tasks +

update_devel_WIOLETTA	Execute
update_devel_KAROLINA	Execute
update_devel_PIOTR	Execute
update_devel_MARTA	Execute
update_VWL_FO_CLIENT	Execute



1. Wstęp

2. Workflow

3. Narzędzia

4. Wyzwania

# Restarty aplikacji

## Problem

Kilka instancji na wspólnym  
**Apache i mod\_wsgi.**  
Restart jednej instancji wpływa  
na chwilową niedostępność  
innych.

## Rozwiązanie

Osobny **uWSGI** dla każdej  
instancji.

### PROTIP

Użyj **Master FIFO** do zarządzania procesem **uWSGI**.

```
while lsof uwsgi_ctl; do
    echo Q > uwsgi_ctl
done
```

# Baza danych

## Problem

Dwa otwarte pull requesty i w obu migracja zmieniająca ten sam model.

Utrzymywanie wstecznych migracji, brak konfliktów.

## Rozwiązanie

Instancja testing zawsze na świeżej kopii bazy danych.  
Plik migrations/version.txt z ostatnim numerem migracji.

### PROTIP

```
# drop all tables
psql rms_testing -t -c "select 'drop table \"' || tablename || '\" cascade;'
from pg_tables where schemaname = 'public'" | psql rms_testing
# copy tables
pg_dump rms_staging | psql rms_testing
```

# Revert na live

## Problem

Jeżeli wrzutka na **live** spowoduje błędy, potrzebny jest szybki sposób przywrócenia i uruchomienia poprzedniej wersji.

## Rozwiązanie

- ★ Tagowanie kodu
- ★ Backup bazy danych
- ★ wrzutki tylko w godzinach gdy admin jest “pod ręką”
- ★ Automatyzacja revertu

### PROTIP

```
# backup db  
pg_dump -C rms > ~/rms_backup_$(date +"%Y%m%d%H%M").psql
```

Używaj narzędzia do monitoringu - np. **Sentry**, **Nagios**.



# Podsumowanie

- ★ Przedstawione rozwiązanie wystarczające dla niekrytycznych aplikacji
- ★ Szybciej nowa rzecz na produkcji
- ★ Większa ilość pracy dla testera, mniej dla dewelopera
- ★ Zawsze zakładaj problemy (jeżeli coś się może zepsuć - to na pewno się zepsuje)



QUESTIONS TIME

OTWIERAMY  
SIĘ NA  
ŁÓDŹ