

BLSTREAM PATRONAGE

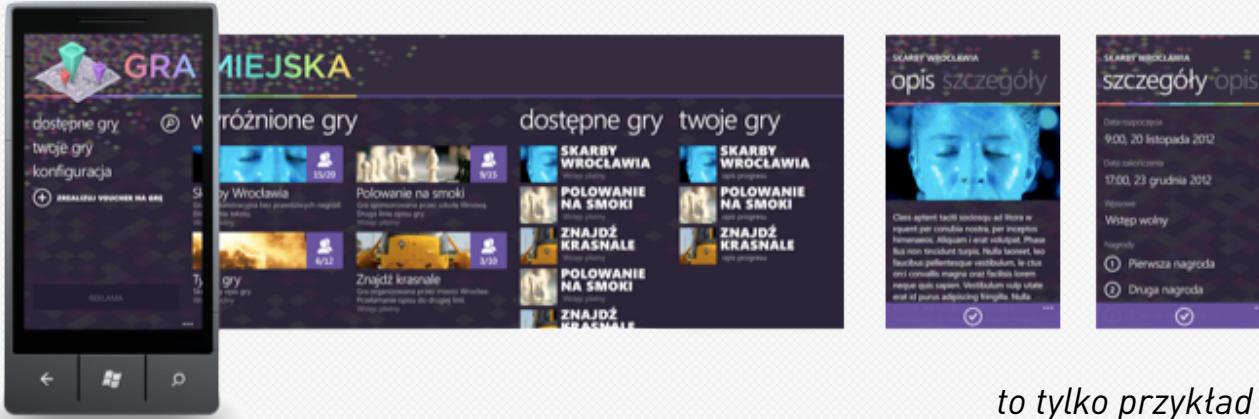
WROCŁAW, 14.03.2013



AGENDA

- # O projekcie
- # Wprowadzenie do Git/Github
- # Wprowadzenie do Jira
- # Spotkanie w grupach

PROJEKT: GRA MIEJSKA



System ułatwiający przeprowadzanie gier miejskich

PROJEKT: GRA MIEJSKA

Klient dla Windows Phone i Android

- # Obsługa kamery, GPS, QR Code,
- # Różne typy zadań,
- # Skóra zależna od gry...

Aplikacja webowa (strona w przeglądarce)

- # Kolejność zadań,
- # Osadzenie zadań na mapie,
- # Postęp gry,
- # Statystyki graczy...



CEL PROJEKTU

- # Przygotować w pełni działający system
- # Opublikować aplikacje mobilne w odpowiednich sklepach
- # Zorganizować i zagrać w grę – finalny test akceptacyjny

CEL PROJEKTU: SKŁADOWE

- # Stworzyć aplikację webową wspierającą organizację gier miejskich
- # Stworzyć aplikację mobilną wspierającą udział w grach miejskich
- # Zadbać o to aby aplikacje były użyteczne i dobrze wyglądały
- # Zadbać o wysoką jakość stworzonych aplikacji
- # Zaprojektować i wytworzyć aplikacje zgodnie z dobrze znanymi zasadami i praktykami
- # Zaangażować grupy UX i QA we wszystkie fazy procesu wytwarzania oprogramowania

DLACZEGO WARTO?

- # Zatrudnienie dla najlepszych
- # Projekt typu open source z możliwością dalszego rozwijania aplikacji
- # Specjaliści BLStream w roli mentorów
- # Praca na sprzęcie najnowszej generacji
- # Najnowsze technologie
- # Możliwość odbycia praktyk



GRUPY DEVELOPERSKIE

**PRZEMYSŁAW
POKRYWKA**

PROGRAMISTA JAVA/
SCALA



WWW

**ARTUR
RADOSZ**

PROGRAMISTA ANDROID



ANDROID

**ADAM
WALOCHA**

PROGRAMISTA WINDOWS
PHONE 8



WINDOWS PHONE

GRUPY UX i QA

**JAKUB
ANDRZEJEWSKI**

PROJEKTANT UX



**MACIEJ
WRZALIK**

PROJEKTANT UX/UI



**ŁUKASZ
PIETRUCHA**

QA ENGINEER



POZOSTAŁE OSOBY

**PAWEŁ
KOCZAN**

PRODUCT OWNER



**MARCELINA
LASOTA**

HR



**KAMILA
PODOLAŃCZUK**

HR



KOMUNIKACJA

- # Na spotkaniach - osobiście
- # Poza spotkaniami - wszelkie dostępne kanały: osobiście, Skype, e-mail, Jira ...
- # **Sygnalizowanie problemów, ryzyk – jak najszybciej**

SPOSÓB PROWADZENIA PROJEKTU – SCRUM/AGILE

Manifest Agile:

Wytwarzając oprogramowanie i pomagając innym w tym zakresie, odkrywamy lepsze sposoby wykonywania tej pracy. W wyniku tych doświadczeń przedkładamy:

Ludzi i interakcje ponad procesy i narzędzia.

Działające oprogramowanie ponad obszerną dokumentację.

Współpracę z klientem ponad formalne ustalenia.

Reagowanie na zmiany ponad podążanie za planem.

Doceniamy to, co wymieniono po prawej stronie, jednak bardziej cenimy to, co po lewej.

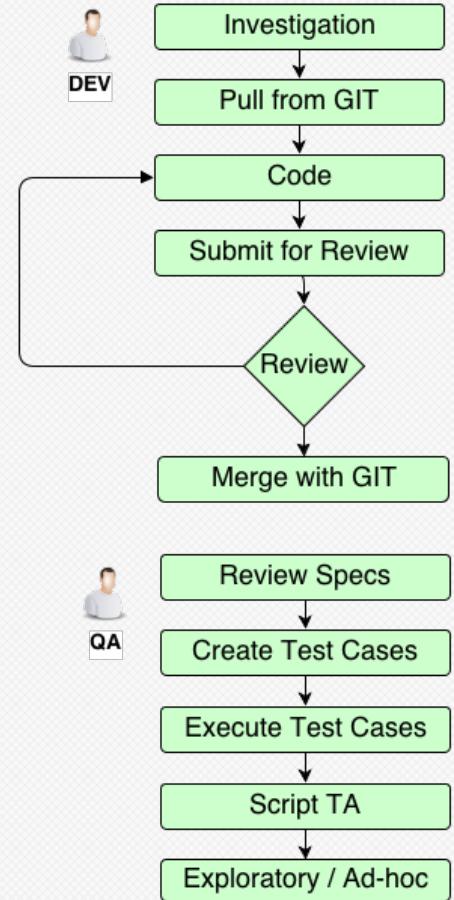
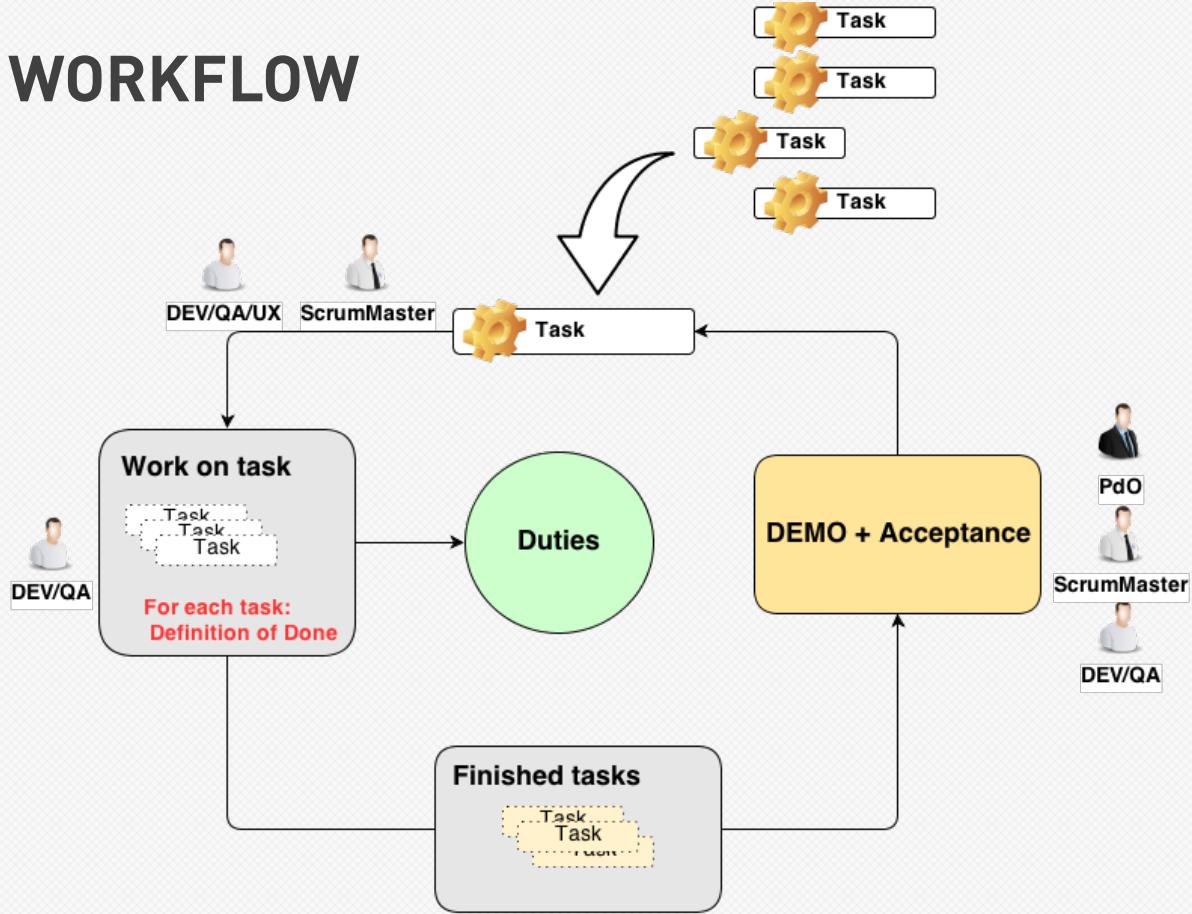
SPOSÓB PROWADZENIA PROJEKTU – SCRUM/AGILE

- # Role: Product Owner / ScrumMaster / Zespół
- # Ceremonie: Planowanie / Daily Scrum / Demo / Retrospektyna
- # Artefakty: Product Backlog / Sprint Backlog / Burndown Chart

SPOSÓB PROWADZENIA PROJEKTU – SCRUM/AGILE

- # Iteracyjność (Sprinty)
- # Przyrostowość
- # Definition of Done
- # Acceptance Criteria
- # Grooming

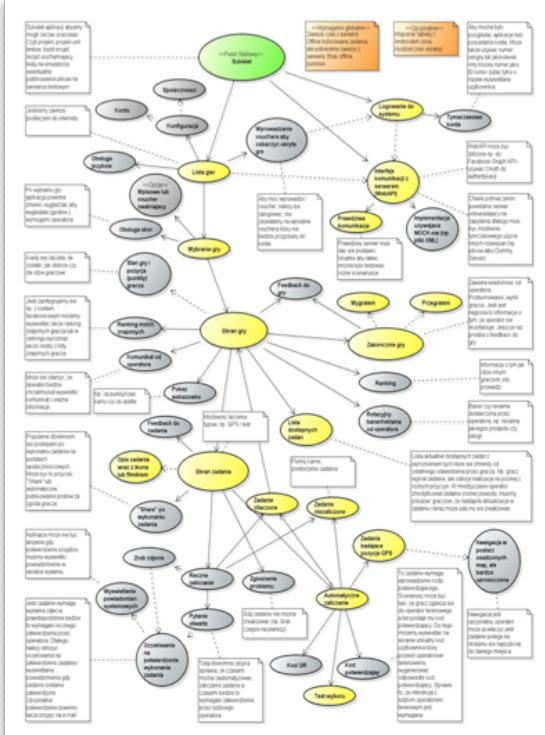
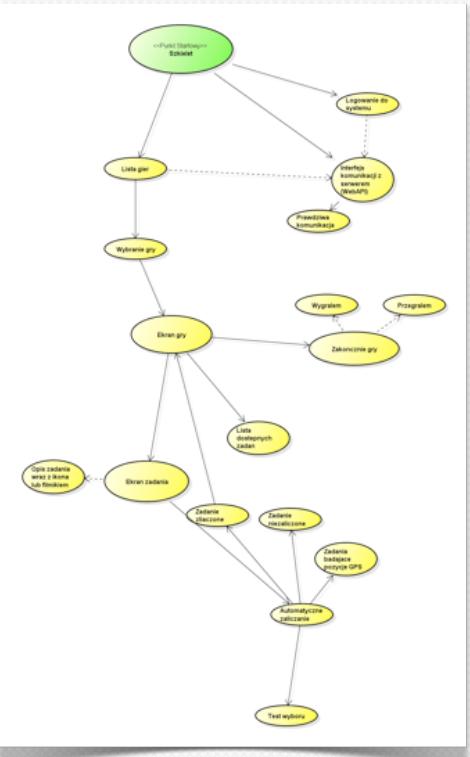
WORKFLOW



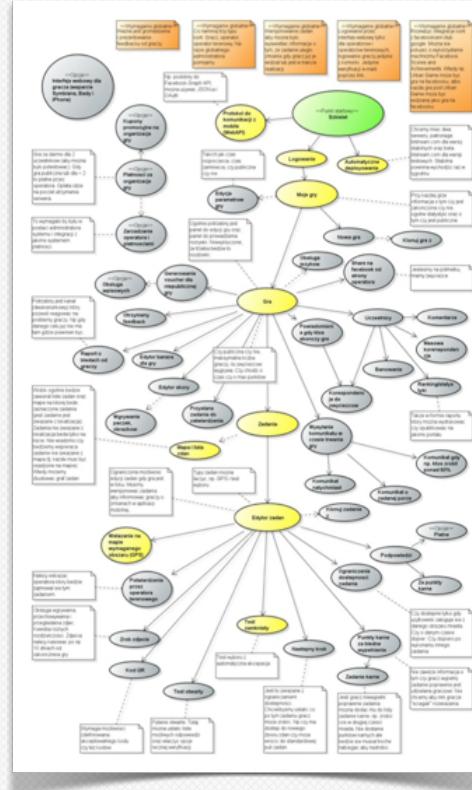
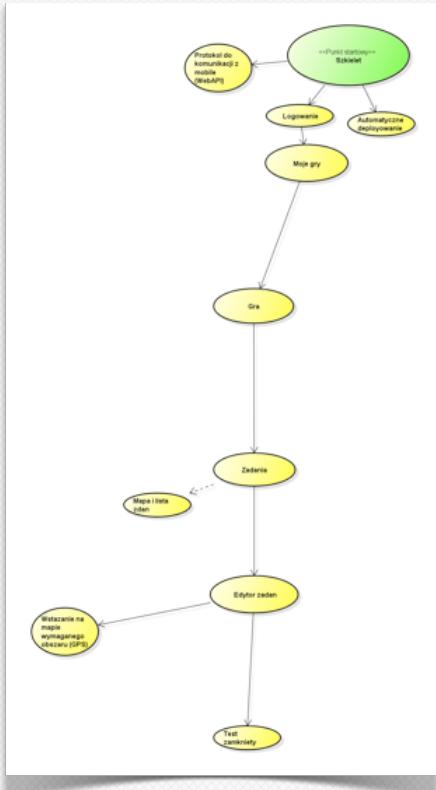
LINKI

- # **Github** <http://github.com/blstream/UrbanGame/>
- # **Wiki** http://github.com/blstream/UrbanGame/wiki/_pages
- # **Jira** <https://tracker.blstreamgroup.com/jira/browse/URBNG>
- # **Jenkins** <http://build.patronage.blstream.com/jenkins>

APLIKACJA MOBILNA: FUNKCJONALNOŚĆ



APLIKACJA WEBOWA: FUNKCJONALNOŚĆ



BACKLOG: WEB

- # URBNG-59 [Web] [WebBE] Application skeleton
- # URBNG-62 [Web] [WebBE] Automatic application deploy
- # URBNG-63 [Web] Basic operator panel
- # URBNG-75 [Web] Task manager panel
- # URBNG-83 [Web] Test with predefined answers - ABCD task editor
- # URBNG-82 [Web] GPS task editor
- # URBNG-84 [Web] Statistics for game and tasks
- # ...

<https://tracker.blstreamgroup.com/jira/browse/URBNG/component/15763>

BACKLOG: ANDROID i WINDOWS PHONE

- # URBNG-34 [All] [WebAPI] Protocol definition
- # URBNG-15 [Android] Application skeleton
- # URBNG-19 [Android] Available games view
- # URBNG-22 [Android] Available tasks view
- # URBNG-26 [Android] Test with predefined answers - ABCD task view
- # URBNG-50 [Android] Automatic deploy
- # URBNG-93 [Android] Handling notification about changes in game or tasks
- # URBNG-25 [Android] GPS task view
- # ...
<https://tracker.blstreamgroup.com/jira/browse/URBNG/component/15761>
<https://tracker.blstreamgroup.com/jira/browse/URBNG/component/15762>

BACKLOG: INTERFEJS KOMUNIKACJI - WebAPI

Urban Game (992005) / URBNG-34

[All] [WebAPI] Protocol definition

[Edit](#) [Assign](#) [Assign To Me](#) [Comment](#) [More Actions](#) [Resolve Issue](#) [Close Issue](#) [Workflow](#)

Description

Definition of protocol used to communication between server and mobile clients. It includes description of commands, frames

A/C:

1. Agreed with all development groups (server + mobile + QA)
2. Described in GitHub WIKI
3. The protocol can be extended in future
4. The protocol is safe - "another user cannot pretend to be me"
5. The protocol is fast:
 - a) it supports e.g. "version" tags so game/task attributes can be cached locally on mobile device
 - b) it supports local cache
 - 6. The protocol is testable - helpers tools are welcome
 - 7. Error cases are defined
8. At least following scenarios are defined:
 - a) login/logout
 - b) get list of games
 - c) get game details
 - d) join game/leave game - requires authentication
 - e) get game status for current player - requires authentication
 - f) support notifications
 - g) get list of tasks for selected game - requires authentication and current GPS position in request
 - h) get task details - required authentication
 - i) send solution for task - requires authentication
9. The mobile application does not know correct answers, never
10. All tasks verification and points assigning is done on server side

Please use details from URBNG-35 to see what fields are required.
Please also consider full list of features available on GitHub Wiki in order to make the protocol flexible enough:
<https://github.com/blstream/UrbanGame/wiki/Mobile-Application---Features>
<https://github.com/blstream/UrbanGame/wiki/Web-Application-Features>

Założenia wstępnie ustalone

Zaakceptowany przez wszystkie grupy

Testowalny

Implementacja typu MOCK w pierwszych iteracjach

Dokumentacja w GitHub

ITERACJA „0”

- # 2 tygodnie na rozruch
- # Analiza backlogu
- # Stworzenie szkicu protokołu
- # Zapoznanie się ze współpracownikami, procesem i narzędziami
- # Grooming

ITERACJA 1-4

- # Planowanie
- # Implementacja i testowanie
- # Grooming kolejnej iteracji
- # Demo
- # Retrospektyna

SPOTKANIA

- # W biurze - raz w tygodniu
- # Poza biurem - ...
- # Wirtualnie – Skype, e-mail, Jira, ...

PRACA W GRUPACH



PRACA W GRUPACH

- # Komunikacja między grupami
- # Przedstawiciel grupy
- # Lider grupy

HARMONOGRAM

- # Kick off: 14.03
- # Iteracja 0: Rozruch do 29.03
- # Iteracja 1: 02.04 – 26.04
- # Iteracja 2: 29.04 – 24.05
- # Iteracja 3: 27.05 – 14.06
- # Iteracja 4: 17.06 – 05.07
- # Zakończenie projektu – początek lipca

FORMALNOŚCI

- # Oświadczenie o przystąpieniu do programu dla każdego uczestnika
- # Umowa o praktykę studencką dla zainteresowanych odbyciem praktyk

GRA MIEJSKA - URZĄDZENIA



JENKINS

The screenshot shows the Jenkins dashboard at localhost:8090. The main area displays a summary of build status: one job is listed with 'Last Success' and 'Last Failure' both marked as 'N/A'. Below this, there are sections for the 'Build Queue' (empty) and 'Build Executor Status' (one idle master). On the left sidebar, links include 'New Job', 'Manage Jenkins', 'People', and 'Build History'. A search bar and a 'ENABLE AUTO REFRESH' button are at the top right. A legend at the bottom right indicates build status using colored icons: green for 'for all', red for 'for failures', and orange for 'for just latest builds'.

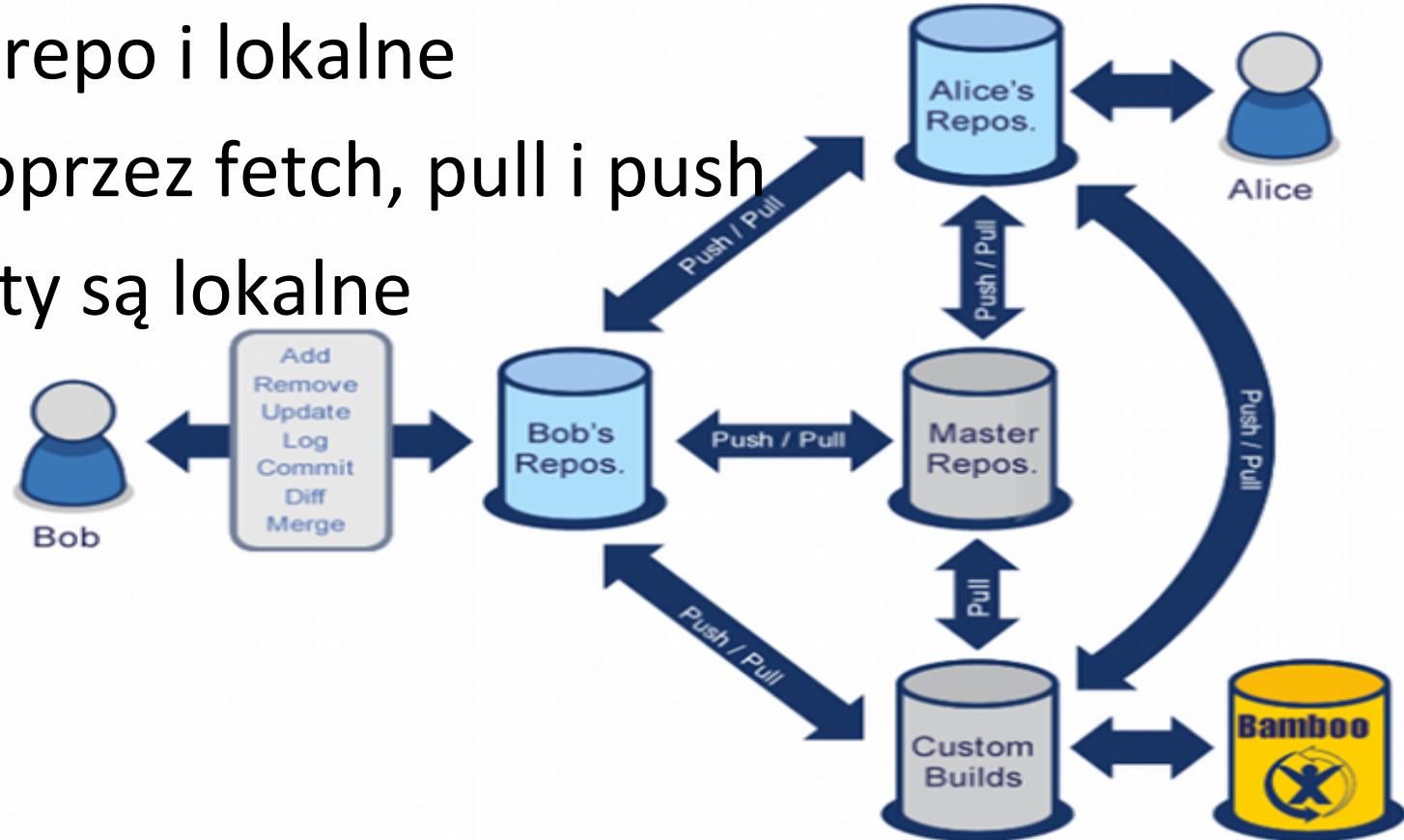
<https://wiki.jenkins-ci.org/display/JENKINS/Meet+Jenkins>

PYTANIA

?

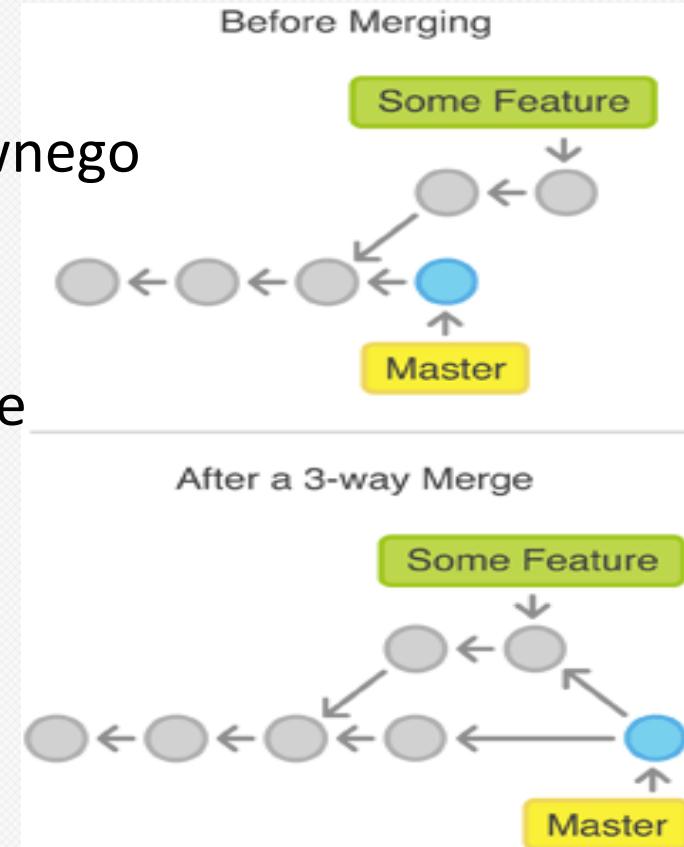
STRUKTURA GIT'A

- Zdalne repo i lokalne
- Sync poprzez fetch, pull i push
- Commity są lokalne

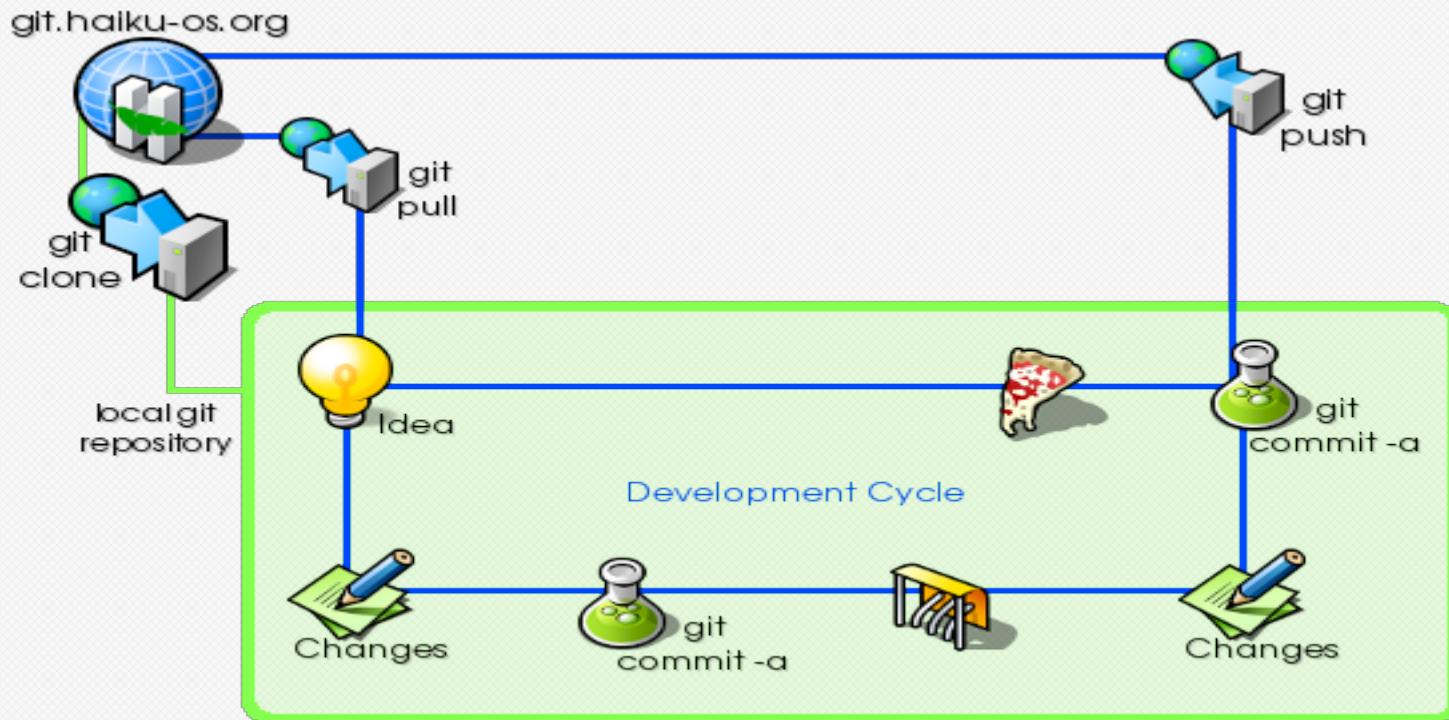


PODSTAWOWY WORKFLOW W GIT

- Git clone
- Praca na branchu utworzonym z głównego
- Commit zmian
- Switch na głównego brancha
- Pull zmian powstałych w międzyczasie
- Merge naszych zmian z brancha
 - 3 way merge
- Push brancha master



PODSTAWOWY WORKFLOW W GIT .2

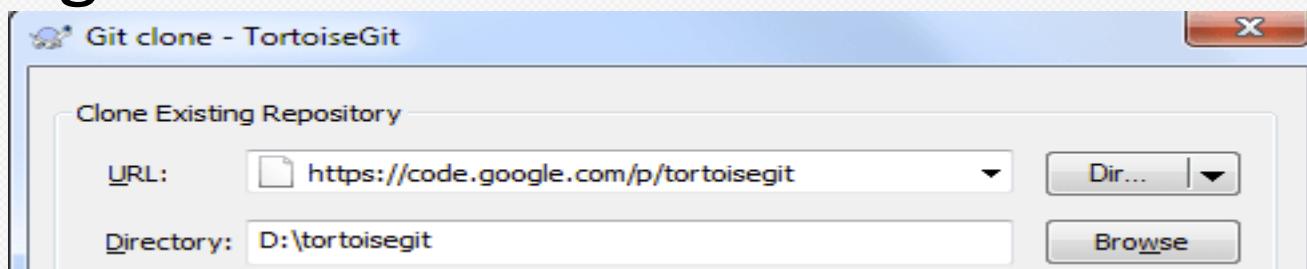


CO POTRZEBUJEMY DO URBAN GAME

- Konto github

A screenshot of a GitHub account page for 'Robert-ios'. The top navigation bar includes 'Explore', 'Gist', 'Blog', and 'Help'. On the left, there's a sidebar with 'Profile', 'Account Settings', and 'Emails'. The main content area shows a message about generating SSH keys and troubleshooting common SSH problems. Below this is a section titled 'SSH Keys' with a button 'Add SSH key'. A public key is listed: 'public (93:7c:aa:3c:6d:34:5c:d7:07:06:4d:aa:50:4b:fc:c2)' with a 'Delete' button next to it.

- Klient git'a: Tortoise



WORKFLOW W URBAN GAME

1. Github fork blstream/UrbanGame repo
2. Git clone naszego forka
3. Praca na lokalnym branchu naszej technologii
4. Lokalne commity zawierające nr taska, np.: URBNG-14
5. Ewentualne przepisywanie historii - tylko przed pushem
6. Push brancha ze skończonym zadaniem
7. Review -> pull request (implementacja taska + testy)
8. Powrót do kroku 4 :) lub idziemy dalej
9. Reviewer poprzez github'a merguje do brancha głównego
10. Done? jeszcze pozostało usunięcie forka
11. Done - obserwujemy Continous Integration



LIVE DEMO – TortoiseGit

DONE

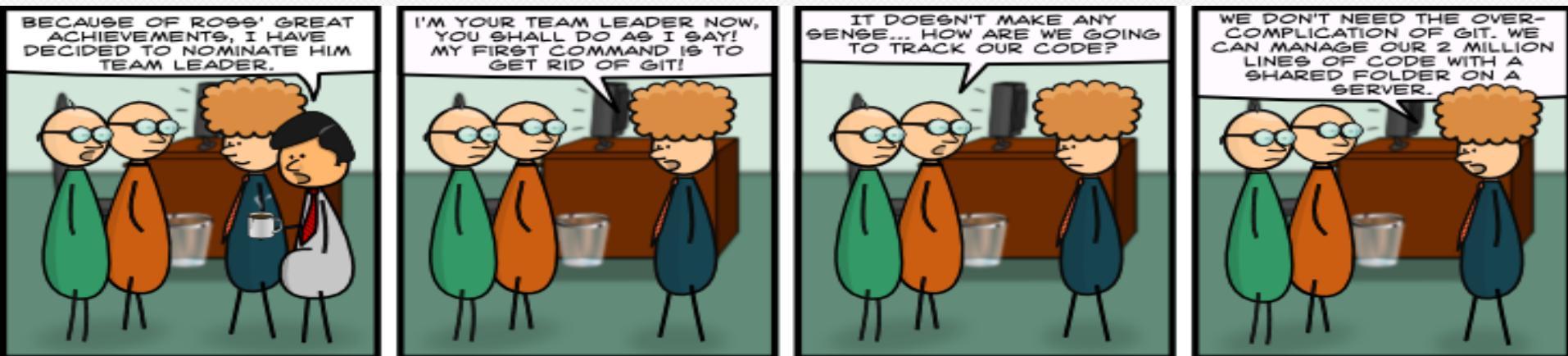


JAK UNIKNAĆ DUŻYCH MERGY

- Częste push'owanie zmian
 - dzielenie taska na mniejsze commits
 - ciągła praca nad taskiem
- Komunikacja, gdy możliwe konflikty
- Zgłaszanie nachodzących na siebie tasków
- Object Oriented Development

TROUBLESHOOTING

- Klient git'a zawsze nam podpowiada
- Jak czegoś nie rozumiemy pytamy Google'a
- Jak workflow niejasny pytam ScM
- Nie utrudniam pracy innym!



JIRA



JIRA

<https://tracker.blstreamgroup.com/jira/browse/URBNG>

Issue

Typ

Priorytet

Iteracja

Komponent

Status / Resolution

Ludzie

Opis / AC

Powiązane

JIRA

- # Dashboard
- # Wyszukiwanie/filtry
- # Strona projektu
- # Road Map