



Ubuntu Touch Release Process

Łukasz 'sil2100' Zemczak <lukasz.zemczak@canonical.com>

IRC: sil2100

Plan prezentacji



- Czym jest Ubuntu Touch?
- Krok pierwszy: kod, launchpad, bzr i merge request
- Krok drugi: dostarczenie kodu do ekosystemu
 - Paczki. deb i. click
 - Archiwa Ubuntu i Click App Store
 - System lądowania: CI Train
- Krok trzeci: testowanie
- Krok czwarty: budowa obrazu, kanały ubuntu-touch
- Krok piąty: awansowanie obrazu do kanałów stabilnych
 - Testy regresyjne
 - Koordynacja i podjęcie decyzji
- Krok szósty: OTA i phased upgrades

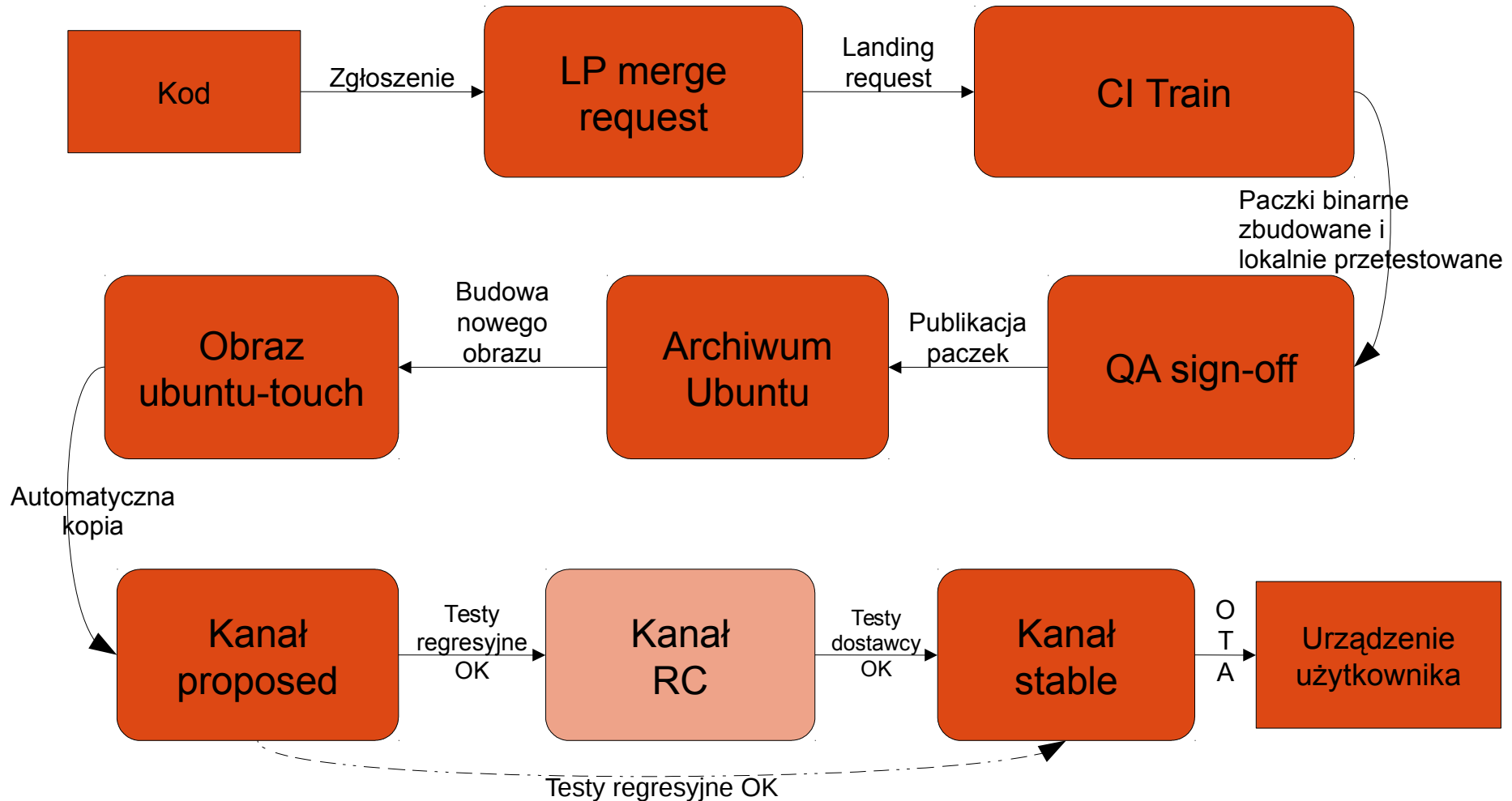
Czym jest Ubuntu Touch?



- Ubuntu do dotykowych urządzeń mobilnych (tablet, smartphone)
- Dalej Ubuntu w podstawie
- Unity8 jako powłoka
- Confinement aplikacji (apparmor)
- Wspierane urządzenia deweloperskie: Nexus 4, Nexus 7 i Nexus 10 (tak jakby)
- Oficjalne urządzenia: BQ Aquaris 4.5, wkrótce telefon Meizu



Droga od kodu do urządzenia Touch



Krok pierwszy: kod, launchpad, bazaar



- Launchpad: system do zarządzania i rozwoju oprogramowania
- Bazaar: główny wspierany system kontroli wersji launchpad
- Merge request: propozycja scalenia naszego kodu do wybranej gałęzi – zazwyczaj trunk
- Recenzje, poprawki, zatwierdzenie, gotowe do lotu



<https://launchpad.net/>

Krok drugi: publikacja kodu



- Ubuntu korzysta z systemu pakietów Debian'a (.deb)
- Bazowe oprogramowanie przechowywane w archiwach (repozytoriach)
- Aplikacje końcowe dostarczane w formie pakietów click (.click)
 - Prostszy format niż .deb
 - Aplikacje bezpieczniejsze (confinement za pomocą apparmor)
 - Zależne od framework'a
- Pakiety click publikowane do Ubuntu Click App Store – szybki proces
- Szybki rozwój touch: nagła potrzeba szybkiego budowania i publikacji tradycyjnych paczek .deb



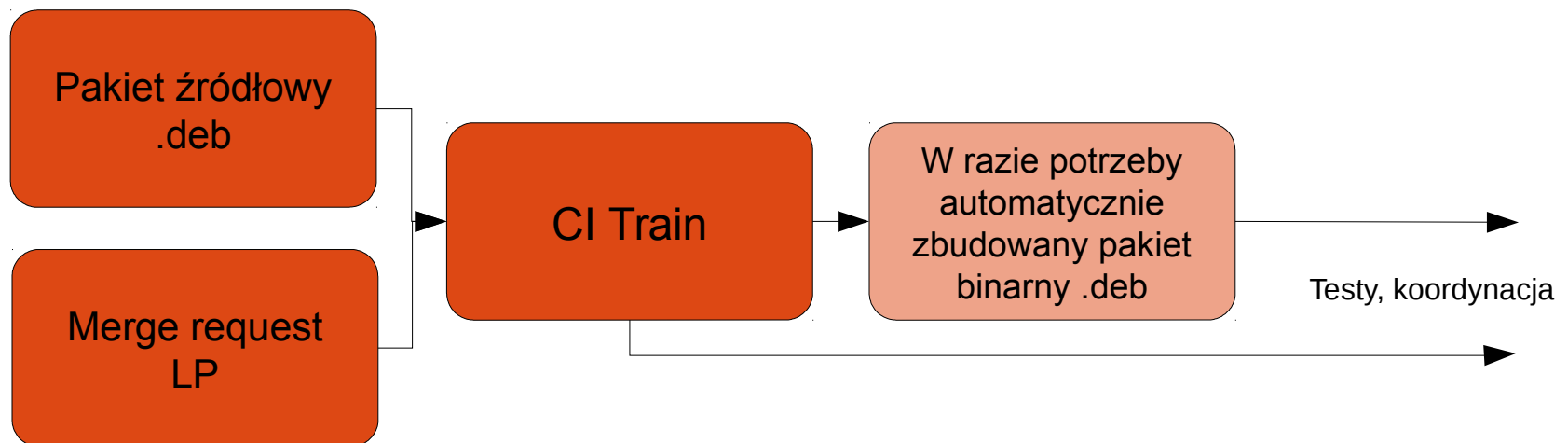
ubuntu^o



Krok drugi: publikacja kodu – CI Train



- System 'CI' (continuous integration) dla pakietów Debiana
- Implementacja 'tymczasowa' (proof-of-concept):
 - Kod bazowy napisany w Python 3.4 (dawniej 2.7)
 - Jenkins
 - Baza i część frontendu – zautomatyzowany Google Spreadsheet (!)
 - Graficzny frontend w JavaScript (AngularJS)
 - Poprawna, pełna implementacja w trakcie prac
- Koordynacja co i kiedy jest publikowane dla Ubuntu Touch
- Gating pakietów bazowany na testach QA



Krok trzeci: testowanie



- Automacyjne testy funkcjonalne wykonywane dla każdego MP
 - Używany framework: autopilot
- Testy manualne dla każdego zgłoszenia w CI Train
 - Proces nazywany QA sign-off
 - Maksymalne zmniejszenie ryzyka wprowadzania regresji
 - Minus: naturalne spowolnienie i wąskie gardło zależne od zasobów QA

Role:

- Zespół QA (Quality Assurance) zatwierdza albo odrzuca zgłoszenie
- LT (Landing Team) koordynuje i publikuje

Krok trzeci: testowanie



QA Testing Requests - for questions ping EU: jibel, US: jfunk - or ubuntu-qa on #ubuntu-ci-eng ☆ Public Calendar < Show

Need QA Sign-off	Ready for Testing	Under Testing	Passed	Failed
<p>ubuntu-rtm/landing-004 - lxc-android-config : ogra 1</p> <p>add signon-apparmor-extension to the seed : dbarth</p> <p>ubuntu-rtm/landing-003 - dialer-app : boiko 1</p> <p>Vivid final release Testing</p> <p>ubuntu/landing-007 - webbrowser-app : oSoMoN</p> <p>ubuntu/landing-007 - webbrowser-app : oSoMoN</p> <p>ubuntu/landing-009 - location-service : mandel</p>	<p>ubuntu/landing-027 - indicator-location : charles 1</p> <p>ubuntu/landing-009 - location-service : mandel 1</p> <p>ubuntu/landing-023 - unity-webapps-qml : alex-abreu 1</p> <p>ubuntu/landing-025 - telephony-service : boiko 1</p>		<p>ubuntu/landing-030 - location-service : tvoss 4</p> <p>ubuntu/landing-020 - net-cpp : tvoss 3</p> <p>ubuntu/landing-002 - qtbase-opensource-src,qtbase-opensource-src-gles : Mirv, kgunn 3</p> <p>OTA rtm→vivid testing 5</p> <p>ubuntu/landing-029 - network-manager : cyphermox, awe 2</p>	<p>ubuntu/landing-000 - system-setting 5</p> <p>ISO Testing te 2</p> <p>calendar-app: 7</p> <p>ubuntu/landing-000 - messaging-ap 2</p> <p>Sanity Tests - I touch/ubuntu-r krillin / 269</p>

Krok czwarty: obraz Ubuntu Touch



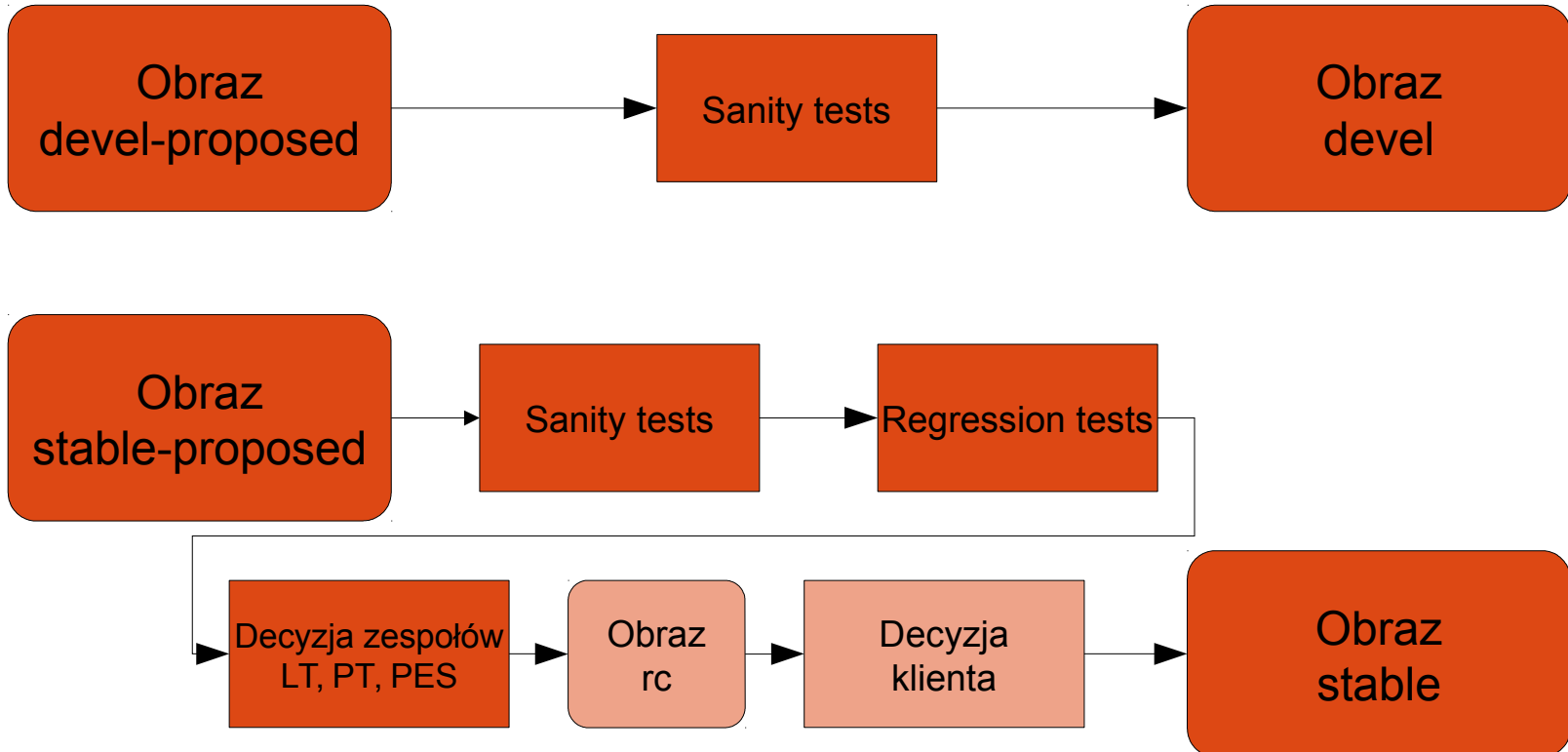
- Dystrybucja Ubuntu Touch odbywa się za pomocą obrazów
- Obrazy z odpowiednią numeracją przechowywane w wybranych kanałach
- Najważniejsze kanały:
 - ubuntu-touch/devel-proposed
 - ubuntu-touch/devel
 - ubuntu-touch/stable-proposed
 - ubuntu-touch/rc
 - ubuntu-touch/stable
- Zmiany teoretycznie dostępne dla użytkowników tylko po znalezieniu się w obrazie



Krok piąty: 'awansowanie' obrazu



- Obrazy spełniające pewne kryteria są kopiowane (awansowane) do kolejnych kanałów w łańcuchu
- Kanał stabilny mocno skoordynowany: planowo 1 wydanie na miesiąc



Krok szósty: OTA i phased upgrades



- Użytkownicy danego kanału widzą nowe uaktualnienie z każdym nowym obrazem
- OTA – Over The Air:
 - Uaktualnienia bazowane na obrazach
 - Możliwe dzięki podziałowi na części read-only i read-write
 - Uaktualnianie za pomocą delty generowanej po stronie serwera
- System uaktualnień fazowych (phased upgrades):
 - Nie wszyscy użytkownicy dostają automatycznie uaktualnienie
 - Dystrybucja fazowa
 - Zwiększone bezpieczeństwo na wypadek krytycznych problemów
- Awans do kanału stabilnego → phased upgrades → nowe uaktualnienie na telefonach użytkowników

Mamy nowe wydanie Ubuntu Touch



...i użytkownicy urządzeń końcowych widzą nowe uaktualnienie

\o/



Pytania?

Dziękuję!

Łukasz 'sil2100' Zemczak

lukasz.zemczak@canonical.com

canonical.com

ubuntu.com