

# Studium wykonalności i aktualny stan prac

Tomasz Kasprzyk, Daniel Ogiela, Jakub Stępak

Akademia Górniczo-Hutnicza  
Wydział Informatyki, Elektroniki i Telekomunikacji  
Katedra Informatyki

9 maja 2016

# Plan prezentacji

- 1 Studium wykonalności
  - Wymagania wobec produktu końcowego
  - Strategia testowania
  - Aspekt technologiczny
  - Plan
  
- 2 Aktualny stan prac
  - Zrealizowane zadania
  - Zadania do zrealizowania na najbliższe tygodnie

# Zadania produktu końcowego

- Poprawne obliczanie wyników wyborów w zadanym systemie wyborczym
- Obliczanie wyników wyborów w możliwie najkrótszym czasie
- Przystępna dla użytkownika prezentacja wyników wyborów

# Sposób działania produktu końcowego

- Aplikacja internetowa umożliwiająca zdefiniowanie wyborów
- Zdefiniowanie wyborów ma polegać na określeniu kandydatów, preferencji głosujących i rozmiaru zwycięskiego komitetu
- Użytkownik posiada różne sposoby zdefiniowania wyborów:
  - 1 Wczytanie pliku w odpowiednim formacie (.soc)
  - 2 Generacja z rozkładu normalnego
  - 3 Graficzna generacja preferencji wyborców
- System oblicza wyniki wyborów dla określonych przez użytkownika wartości parametru  $p$

# Rodzaje pisanych testów

- Testy jednostkowe do sprawdzenia poprawności działania kolejnych komponentów projektu
- Testy regresywne do sprawdzenia poprawności działania wcześniej dodanych elementów systemu po dodaniu nowych elementów
- Testy porównawcze do sprawdzenia skuteczności obliczania wyników wyborów

# Ocena poprawności działania produktu

- Testy porównawcze głównego algorytmu obliczającego wyniki wyborów z algorytmami innego typu
- Dla mniejszego rozmiaru danych wejściowych porównanie działania z algorytmem typu brute-force
- Dla większego rozmiaru danych wejściowych porównanie działania z innym algorytmem heurystycznym np. algorytmem zachłannym

# Wykorzystane technologie

- Python 2.7
- Django 1.9
- Bootstrap
- JavaScript (jQuery i Chart.js)
- Platforma Heroku do wdrożenia systemu
- Ciągła integracja: Travis + Coveralls

# Wybór technologii

- Doświadczenie części zespołu w pracy z wybraną technologią
- Przekonanie o możliwości zrealizowania projektu w wybranej technologii
- Zaoszczędzenie czasu na poznawanie nowych technologii
- Wydajność



# Szansa na powodzenie i przewidzenie trudności

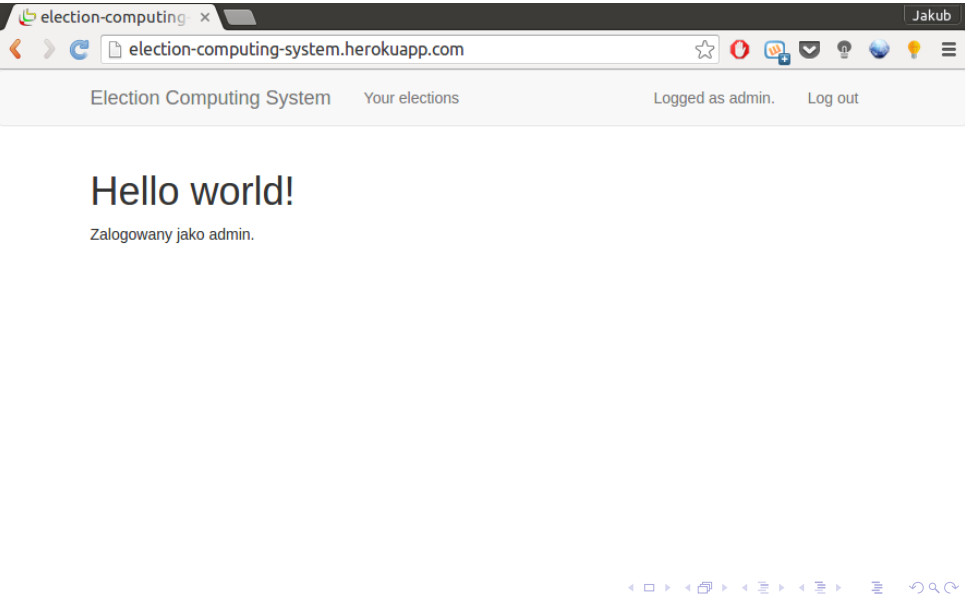
- Przekonanie o możliwości zrealizowania harmonogramu prac określonego w wizji i wykonania produktu na czas
- Główne problemy przewidywane przy projektowaniu i implementacji głównego algorytmu obliczającego wyniki wyborów – algorytmu genetycznego
- Zwiększone nakłady pracy całego zespołu w przypadku napotkania trudności i skupienie całego wysiłku na tym zadaniu

# Co już jest

- Implementacja algorytmu brute-force
- Generacja wyborów z rozkładu normalnego
- Wczytywanie wyborów z pliku wraz z ich walidacją
- Webowy interface
- Tworzenie wykresów 2D
- Konfiguracja wdrożenia systemu na Heroku

# Gdzie zobaczyć nasz produkt

- Adres repozytorium:  
`https://github.com/jakubste/election-computing-system`
- Działająca aplikacja:  
`https://election-computing-system.herokuapp.com`



Election Computing x

Jakub

localhost:8000/elections/list/

Election Computing System





Your elections

Logged as test.


Log out

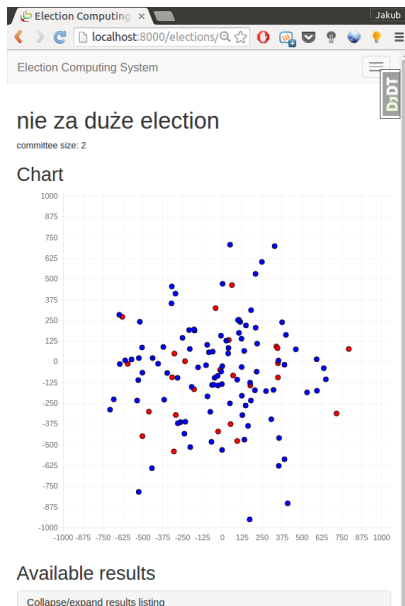
D3DT

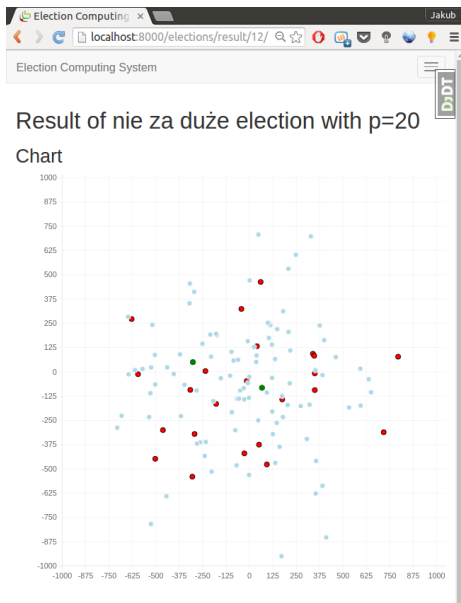
## Your elections

Name	Candidates	Voters	Results	Actions
sobotnie wybory	15	25	2	
mniejšie	10	30	3	
nie za duże election	25	100	2	
soc	5	4	0	

+ New election







# Co do zrobienia

- Algorytm zachłanny do obliczania wyników wyborów
- Algorytm genetyczny do obliczania wyników wyborów
- Polepszenie interfejsu i UX



# Dziękujemy za uwagę