**AKADEMIA GÓRNICZO-HUTNICZA**

**im. Stanisława Staszica w Krakowie**



**Informatyczne Narzędzia Pracy Grupowej**

**Projekt programu analizującego dane**

**Konrad Janczarski**

**Michał Pilch**

**Kamil Switek**

**WEAIiIB, AiR**

**Rok I**

Spis treści

[Cel projektu 3](#_Toc452860429)

[Scrum 6](#_Toc452860430)

[Sprint 1 6](#_Toc452860431)

[Cel 6](#_Toc452860432)

[Milestones 6](#_Toc452860433)

[Tickety i subtickety 6](#_Toc452860434)

[Problemy 6](#_Toc452860435)

[Podsumowanie 6](#_Toc452860436)

[Sprint 2 7](#_Toc452860437)

[Sprint 3 8](#_Toc452860438)

[Sprint 4 8](#_Toc452860439)

[Sprint 5 9](#_Toc452860440)

[Cel 9](#_Toc452860441)

[Milestones 9](#_Toc452860442)

[Tickety i subtickety 9](#_Toc452860443)

[Problemy 9](#_Toc452860444)

[Podsumowanie 9](#_Toc452860445)

[Wkład w projekt 10](#_Toc452860446)

[Wkład Konrada 10](#_Toc452860447)

[Sprint 1 10](#_Toc452860448)

[Sprint 2 10](#_Toc452860449)

[Sprint 3 11](#_Toc452860450)

[Sprint 4 11](#_Toc452860451)

[Sprint 5 11](#_Toc452860452)

[Wykresy 11](#_Toc452860453)

[Wkład Michała 12](#_Toc452860454)

[Sprint 1 12](#_Toc452860455)

[Sprint 2 12](#_Toc452860456)

[Sprint 3 12](#_Toc452860457)

[Sprint 4 12](#_Toc452860458)

[Sprint 5 13](#_Toc452860459)

[Wykresy 13](#_Toc452860460)

[Wkład Kamila 13](#_Toc452860461)

[Sprint 1 13](#_Toc452860462)

[Sprint 2 14](#_Toc452860463)

[Sprint 3 14](#_Toc452860464)

[Sprint 4 14](#_Toc452860465)

[Sprint 5 14](#_Toc452860466)

[Wykresy 14](#_Toc452860467)

# Cel projektu

Projekt miał na celu dostarczenie gotowego produktu jakim jest aplikacja wyszukująca podobieństwa w danej bazie danych.

Studenci zostali zobowiązani do wykonania tego zadania korzystając z opracowanej przez firmę NVIDIA technologii CUDA.

Istotnym elementem projektu była praca zespołowa członków zespołu.

Organizacja pracy miała przebiegać w oparciu o metodologię SCRUM oraz KANBAN.

Narzędzia pracy jakie wykorzystał zespół w celu osiągnięcia celu projektu to serwis internetowy GitHub wykorzystujący system kontroli wersji Git oraz Trello – serwis umożliwiający pracę w systemie KanBan.

# Wstęp teoretyczny

## Kanban

opracowana w latach pięćdziesiątych w Japonii metoda sterowania produkcją. Słowo kanban   
w wolnym tłumaczeniu można w poniższym przypadku oddać jako „spis widoczny”. W pierwotnym znaczeniu – w języku japońskim – oznacza: szyld, tabliczkę z napisem informującym, billboard.

Metoda KANBAN opiera się na poszczególnych kartach wyrobów, ich cyrkulacji i analizie. Kanban rozwijał się i dziś oznacza także system informacyjny, system planowania, rozdziału oraz kontroli czynności i zadań produkcyjnych. Polega na takim organizowaniu procesu wytwórczego, aby każda komórka organizacyjna produkowała dokładnie tyle, ile w danej chwili jest potrzebne. W metodzie tej za czynnik krytyczny zarządzania materiałami uznano sterowanie zapasami.

***Zasady metody Kanban***

• Wizualizacja

• Limit (Ograniczenie) prac w toku (WIP: Work In Progress)

• Zarządzanie przepływem (ang. flow)

• Jasne i jednoznaczne reguły

• Wprowadzenie pętli zwrotnej (ang. feedback)

• Wspólne usprawnianie i ciągła ewolucja.

**Tablica kanbanowa**

Tablica kanbanowa jest podzielona na kolumny. Na tablicy umieszczane są karteczki opisujące poszczególne zadania do wykonania.

W wersji podstawowej tablicę dzielimy na 4 kolumny:

* do zrobienia

– wszystkie sprawy, którymi w przyszłości chcemy się zająć. Gdy

przyjdzie nam do głowy jakiś nowy pomysł, to właśnie tu umieszczamy nową

karteczkę;

* priorytetowe

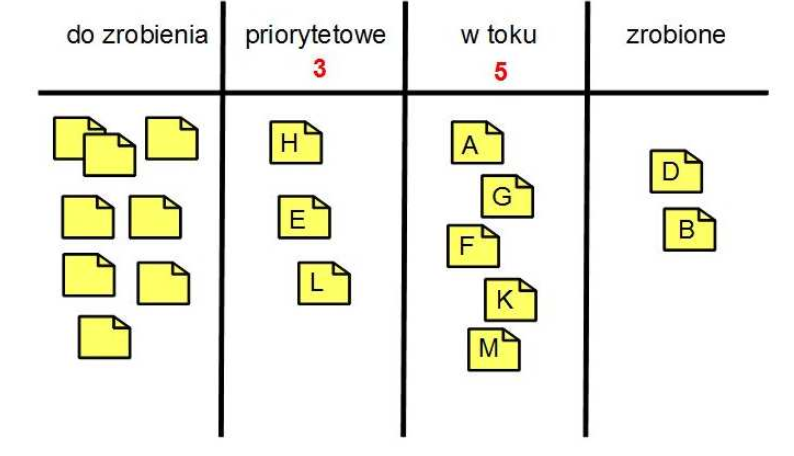
– to sprawy (karteczki) wybrane z pierwszej kolumny, którymi chcemy zająć się pierwszej kolejności;

* w toku

– sprawy, które obecnie realizujemy;

* zrobione

– tu przenosimy karteczkę z trzeciej kolumny, po zakończeniu danej sprawy.



## Scrum

Jest to iteracyjna i przyrostowa metodyka zarządzania procesami, zaliczana do metodyk zwinnych, zgodnych z manifestem Agile. W metodyce tej rozwój produktu podzielony jest na mniejsze, trwające maksymalnie jeden miesiąc kalendarzowy iteracje, zwane sprintami następującymi bezpośrednio po sobie. Po każdym sprincie zespół pracujący nad rozwojem produktu jest w stanie dostarczyć działającą jego wersję. Scrum jest często stosowany podczas tworzenia i rozwijania oprogramowania, nie jest jednak ograniczony tylko do tej dziedziny. Ogólne założenia podejścia zostały zaprezentowane przez Hirotakę Takeuchiego i Ikujiro Nonakę w artykule The New Product Development Game, opublikowanym w Harvard Business Review w styczniu 1986 roku. Definicja Scruma w zastosowaniu do produkcji oprogramowania została sformalizowana przez Kena Schwabera w 1995.

## Git

Jest to rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL  
 w wersji 2.

Pierwsza wersja narzędzia Git została wydana 7 kwietnia 2005 roku, by zastąpić poprzednio używany w rozwoju Linuksa, niebędący wolnym oprogramowaniem, system kontroli wersji BitKeeper.

**Najważniejsze cechy**

Dobre wsparcie dla rozgałęzionego procesu tworzenia oprogramowania: jest dostępnych kilka algorytmów łączenia zmian z dwóch gałęzi, a także można dodawać własne algorytmy.

Praca off-line: każdy programista posiada własną kopię repozytorium, do której może zapisywać zmiany bez połączenia z siecią; następnie zmiany mogą być wymieniane między lokalnymi repozytoriami.

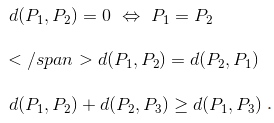
Wsparcie dla istniejących protokołów sieciowych: dane można wymieniać przez HTTP(S), FTP, rsync, SSH.

Efektywna praca z dużymi projektami: system Git według zapewnień Torvaldsa, a także według testów fundacji Mozilla, jest o rzędy wielkości szybszy niż niektóre konkurencyjne rozwiązania.

Każda rewizja to obraz całego projektu: w przeciwieństwie do innych systemów kontroli wersji, Git nie zapamiętuje zmian między kolejnymi rewizjami, lecz kompletne obrazy. Z jednej strony wymaga to nieco więcej pracy aby porównać dwie rewizje, z drugiej jednak pozwala np. na automatyczną obsługę zmian nazw plików.

## Metryka

W matematyce odległość, czyli metryka może być dowolną symetryczną funkcją (relacją) dwuargumentową (między dwoma punktami P1 i P2), d(P1,P2), przyjmującą nieujemne wartości   
i spełniającą tzw. warunek trójkąta. Innymi słowy, muszą zachodzić następujące trzy warunki:



Metryka euklidesowa i miejska są przykładami relacji spełniających powyższe warunki.

# Nasz Scrum

## Sprint 1

### Cel

Celem pierwszego sprintu było opracowanie sposobu pracy nad projektem. Zespół w trakcie trwania sprintu starał się wyszukiwać najlepsze rozwiązania komunikacji między członkami oraz starał się znaleźć najlepszy podział pracy. Najważniejszym elementem pierwszego sprintu było stworzenie środowiska pracy oraz opracowanie najlepszej dla nas tablicy KanBanowej.

### Milestones

Kamieniem milowy tego sprintu było znalezienie pierwszych pięciu metryk i osadzenie ich   
w środowisku pracy

### Tickety i subtickety

* ***Scrum Master*** – Scrum Masterem tego sprintu był Konrad.
* ***Daily Scrum*** . - każdy
* ***Zainstalować Google*** . - każdy
* ***Stworzyć repozytorium GitHub*** – osoba odpowiedzialna - Konrad
* ***Zapoznać się z poleceniami prowadzącego dotyczącymi pierwszego sprintu. -*** każdy
* ***Zainstalować Plus for Trello -*** każdy
* ***Znaleźć 5 metryk*** – osoba odpowiedzialna – Konrad

### Problemy

Kamil napotkał problem z instalacją Google Chrome. Problem został szybko rozwiązany.

### Podsumowanie

W czasie sprintu pierwszego zespół wypróbował metodykę „Daily Scrum”. Jednakże, codzienne poświęcanie czasu na projekt nie było możliwe, dlatego przestaliśmy praktykować „Daily Scrum”. Metodę tą zastąpiliśmy w następnych sprintach użyciem dodatkowej listy „Problemy” gdzie każdy wypisywał problemy które napotkał.

Sukcesem zakończyła się próba wybrania na sprint pierwszy „Scrum Mastera”, który na celu ma organizację tablicy Trello, rozwiązywanie sporów i innych problemów.

W trakcie trwania pierwszego tygodnia pracy założono repozytorium na GitHub’ie, gdzie znajdują się pliki na których zespół pracował. Dzięki możliwością jakie oferuje serwis GitHub była możliwa szybka wymiana danych i wykonanej pracy.

Dodatkowo każdy z zespołu zainstalował na swoim komputerze przeglądarki Google Chrome, które umożliwiły instalacje dodatku do Trello „Plus for Trello”, który znacznie ułatwił komunikację pomiędzy członkami zespołu i organizację pracy.

Ponieważ członkowie zespołu po raz pierwszy spotkali się z takim trybem pracy, sporo czasu zajęło zapoznanie się z celem projektu i zrozumienie go. To była główna bariera hamująca postępy pierwszego sprintu.

Znaleziono 5 metryk potrzebnych do napisania pierwszych prostych programów.

Celem sprintu pierwszego było zapoznanie się z środowiskiem pracy i uruchomienie odpowiednich środowisk programistycznych na własnych komputerach. Cel sprintu został zrealizowany bez większych trudności. Zadania mniej ważne również.

## Sprint 2

### Cel

Celem sprintu było wykonanie pierwszego właściwego fragmentu wykonywalnego programu. Zespół miał skompilować pierwsze obliczenia metryk

### Milestones

Skonfigurować do współpracy CUDA Toolkit, VisualStudio i emulator

### Tickety i subtickety

* **Scrum Master** – Michał Pilch
* **Zapoznanie się z technologią CUDA**
* **Błąd krytyczny systemu** – Konrad Janczarski

1. **Zainstalować Chrome, Plus for Trello, Tortoise Git** – Konrad – Ticket związany z błędem krytycznym systemu

* **Zainstalować NVIDIA CUDA Emulator** – Konrad Janczarski i Michał Pilch
* **Zainstalować i skonfigurować do współpracy VisualStudio i CUDA** – Kamil Switek
* **Skonfigurować do współpracy CUDA Toolkit, VisualStudio i emulator** – Konrad Janczarski i Michał Pilch
* **Skompilować pierwsze obliczenia metryk**- Kamil Switek
* **Znaleźć 5 metryk –** Konrad Janczarski

### Problemy

Napotkano istotny problem. Nie udało się skonfigurować emulatora. Uniemożliwia to pracę   
w architekturze cudy Konradowi i Michałowi

### Podsumowanie

Sprint 2 był już przejściem do poważniejszej pracy. Najpierw zapoznaliśmy się z narzędziami pracy grupowej i dostosowaliśmy je do siebie, zainstalowaliśmy również dodatek PlusforTrello dla przeglądarki Google Chrome. Udało się w pełni skonfigurować środowisko pracy (Nvidia CUDA, Visual Studio 2012). Kompilacja przykładowych programów podanych przez producenta zakończyła się powodzeniem. Następnie zaimplementowaliśmy pierwsze obliczenia metryk na karcie graficznej (tych metryk, które zostały znalezione w poprzednim sprincie). Podczas gdy jedna część zespołu zapoznawała się dalej z technologią CUDA, kompilacją programów i obliczeniami na karcie graficznej, pozostała część zajęła się znajdowaniem kolejnych metryk. Niepowodzeniem zakończyło się skonfigurowanie technologii CUDA na kartach graficznych innych niż Nvidia, ponieważ emulator odmawiał współpracy.

## Sprint 3

### Cel

Celem sprintu jest rozwinięcie fragmentu docelowego programu który do tej pory stworzyliśmy.

### Milestones

-Znaleźć metryki „Razem mamy 57 metryk”

### Tickety i subtickety

* Scrum Master – Kamil Switek
* „Razem mamy 57 metryk” – Konrad Janczarski
* Przerobić program z poprzedniego sprintu tak by działał na wielu wątkach – Kamil Switek
* Napisać program obliczający metryki bez technologii CUDA - Michał Pilch

### Problemy

-Nadal nie skonfigurowaliśmy emulatora CUDA

### Podsumowanie

Podczas sprintu 3 naszym głównym zadaniem było kontynuowanie zadań zaplanowanych   
w poprzednim sprincie. Zakończyliśmy znajdowanie metryk, udało się ostatecznie znaleźć ich ponad 50. Kontynuowaliśmy proces implementacji obliczeń metryk, tym razem jednak przydzielając każdej metryce osobny wątek na karcie graficznej. Niestety problem konfiguracji środowiska CUDA na kartach Radeon nie został rozwiązany. Rozpoczęliśmy implementację programu obliczającego to samo, ale na procesorze, aby sprawdzić, czy udało się osiągnąć jakiekolwiek przyspieszenie.

## Sprint 4

### Cel

Celem sprintu było napisanie programu który będzie liczyć metryki nie używając technologii CUDA oraz rozwijanie programu docelowego.

### Milestones

Napisać program liczący metryki nie używając CUDY

### Tickety i subtickety

* **Scrum Master** – Konrad Janczarski
* **Uporządkować plik „Metryki”** – Konrad Janczarski
* **Znaleźć źródła metryk** – Konrad Janczarski
* **Napisać dokumentację sprintu 1** – Konrad Janczarski
* **Dodać do programu kolejne metryki (36)** – Kamil Switek
* **Napisać program liczący metryki nie używając CUDY (11 metryk)** – Michał Pilch

### Problemy

Nie znaleziono źródła „metryki w X^N” która została już skompilowana.

Nie udało się również naprawić problemów związanych z błędnymi wprowadzeniami czasów do Plus for Trello (Konrad Janczarski).

### Podsumowanie

Sprint 4 był rozwinięciem poprzednich sprintów. Udało nam się przerobić program na zwykły język C, bez użycia CUDY, żeby móc później sprawdzić czy CUDA przyspiesza nam czas obliczeń. Do programu w CUDZie udało się dołożyć kolejne metryki, program działa obecnie na 36 metrykach. Zrobiliśmy również dokumentację poprzednich sprintów. Udało się także znaleźć źródła uprzednio znalezionych metryk. Nie udało się nam niestety znaleźć źródła metryki X^N. Kolejny problem który wynikł w tym tygodniu to naprawienie błędów przepracowanego czasu w Trello, niestety nie udało się go rozwiązać.

## Sprint 5

### Cel

Celem sprintu piątego była ogólnie pojęta finalizacja projektu. Składało się na to doprowadzenie aplikacji do ostatecznego kształtu oraz sporządzenie dokumentacji projektu oraz opracowanie prezentacji.

### Milestones

Napisać program który porówna szybkość wykonywania analizy danych za pomocą technologii CUDA i bez użycia tej technologii.

Uruchomić działający program analizujący dane w technologii CUDA.

Napisanie dokumentacji.

Napisanie newsletter’a

### Tickety i subtickety

* ***Dokumentacja Sprintu 2 – Kamil Świtek***
* ***Dokumentacja Sprintu 3 – Kamil świtek***
* ***Dokumentacja Sprintu 4 – Michał Pilch***
* ***Dokumentacja Sprintu 5 – Konrad Janczarski***
* ***Uzupełnienie kodu metryk nie wykorzystującego CUDy***
* ***Napisanie programu porównującego czas wykonywania analizy***
* ***Stworzenie bazy danych do analizy***
* ***Uruchomienie finalnego produktu***
* ***Udział w sprintach – Konrad Janczarski***
* ***Udział w sprintach – Kamil Switek***
* ***Udział w sprintach – Michał Pilch***
* ***Przygotować prezentację - Michał Pilch***

### Problemy

W sprincie tym nie zgłaszano żadnych problemów

### Podsumowanie

Sprint piąty to ostatni sprint w naszym Scrumie. W sprincie tym dokonaliśmy ostatnich prac finalizujących produkt.

Sprint piąty był dwa razy dłuższy od pozostałych sprintów. Mimo tego, że mieliśmy na wykonanie tego sprintu więcej czasu niż na inne to nakład pracy na jednostkę czasu w tym sprincie był największy spośród całego Scruma. Było to spowodowane złym oszacowaniem pozostałego czasu pracy.

Podczas tego sprintu sporządzono podsumowanie pracy jaką wykonał zespół w Sprintach 2,3,4 i 5.

Zdecydowanie były to najłatwiejsze zadania tego sprintu. Dzięki prowadzonym tablicom KanBanowym z łatwością można było przypomnieć sobie cele stawiane w tamtych okresach, wykonane zadania oraz występujące problemy.

W sprincie tym został uzupełniony kod programu wykonującego analizę bazy danych nie używając technologii CUDy. Należało przepisać napisane już kody w architekturze CUDY dokonując koniecznych zmian.

Istotnym elementem było stworzenie pliku tekstowego zawierającego przykładową bazę danych którą będzie analizował nasz program.

Najważniejszym i najtrudniejszym elementem Backlogu Produktu w tym sprincie było uruchomienie programu w finalnej formie. Należało napisać fragmenty kodu wykorzystujące metryki do analizy pliku.

Następnie każdy z członków zespołu dokonał sprawozdania swojego udziału w dokonywaniu przyrostów każdego sprintu.

# Wkład w projekt

## Wkład Konrada

### Sprint 1

W sprincie pierwszym pełniłem funkcję Scrum Mastera. Zadanie to zabrało mi najwięcej czasu spośród powierzonych mi na ten sprint zadań. Jako Scrum Master zajmowałem się organizacją tablicy Kanbanowej. Dokonywałem przypisywania osób do danych ticketów, nadawałem ticketom odpowiednie etykiety, zarządzałem Daily Scrumami.

W sprincie tym powierzono mi również zadanie założenia repozytorium GitHub. Zainstalowałem przeglądarkę Google Chrome i dodatek Plus for Trello.

### Sprint 2

W sprincie tym napotkałem problem w postaci błędu krytycznego systemu. Musiałem przeinstalować system na komputerze na którym pracuję. Oczywiście po reinstalacji systemu musiałem znowu zainstalować programy które instalowałem w sprincie pierwszym – Google Chrome, Plus for Trello, TortoiseGit.

Gdy uporałem się już z systemem operacyjnym napotkałem kolejny, bardzo istotny problem – moja karta graficzna nie obsługuje technologii CUDy.

Problem ten miał rozwiązać NVIDIA CUDA Emulator który zainstalowałem na swoim komputerze.

Następnie zainstalowałem Visual Studio oraz CUDA Toolkit. Bardzo dużo czasu zajęła mi próba skonfigurowania do współpracy CUDA Toolkit, Visual Studio i emulator. Niestety próba zakończyła się niepowodzeniem.

W Sprincie tym znalazłem 5 metryk.

### Sprint 3

Sprint trzeci rozpocząłem ponowną próbą skonfigurowania emulatora na swoim komputerze. Ponownie zajęło mi to dużo czasu i zakończyło się niepowodzeniem. Doszedłem do wniosku, że nie warto poświęcać w tym celu więcej czasu. Zająłem się szukaniem metryk.

W sprincie tym udało mi się znaleźć 47 metryk.

### Sprint 4

Sprint ten rozpocząłem od porządkowania pliku z metrykami. Należało sprawdzić czy któreś metryki się nie powtarzają oraz znaleźć źródła metryk. Ponieważ na bieżąco nie były notowane źródła, poszukiwania zajęły sporo czasu.

Nie udało mi się znaleźć źródła jednej z metryk która już została zaimplementowana.

Napisałem dokumentację sprintu 1.

Niestety w trakcie poprzednich sprintów wprowadziłem nieprawidłowe szacunki czasów oraz nieprawidłowe czasy spędzone nad konkretnymi zadaniami do Plus for Trello. W trakcie tego sprintu próbowałem naprawić błędne wprowadzenia czasów jednakże komendy które miały naprawić złe wpisy nie działały.

### Sprint 5

W sprincie piątym zajmowałem się sporządzaniem dokumentacji projektu.

Sporo czasu zajęło mi sporządzenie sprawozdania mojego wkładu w projekt .

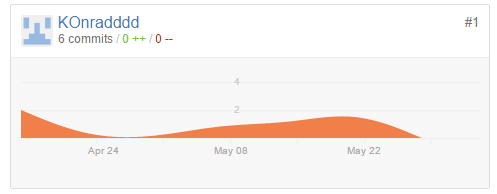
Napisałem również dokumentacje sprintu 5.

### Wykresy

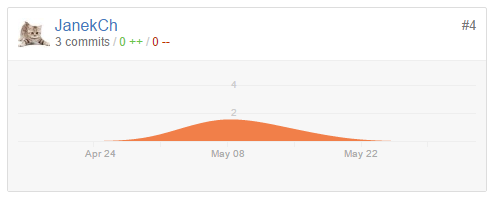
#### Trello

Niestety, najprawdopodobniej podczas próby naprawy błędnych wprowadzeń szacunków czasu, usunąłem wszystkie szacunki. Plus for Trello nie wygenerowało żadnego wykresu.

#### GitHub



Wykres commitów w czasie trwania scrum’a.



Commitów dokonywałem również z pracowni komputerowej, gdzie na stanowisku pracy TortoiseGit zalogowany był na konto JanekCh. Mimo że commitów dokonałem ja (logując się uprzednio) zostały zarejestrowane jako commity Janka.

## Wkład Michała

### Sprint 1

W tym sprincie moim zadaniem było zapoznać się z metodologiami, które będziemy używać   
w naszym programie, zawierało się w tym zaistalowanie dodatku Plus for Trello, w uprzednio zainstalowanym Google Chrome, dzięki niemu na bieżąco mogłem aktualizować postępy mojej pracy.

Dodałem również parę zadań na Trello.

### Sprint 2

W drugim sprincie moim głównym celem było przystosowanie mojego komputera do pracy   
w technologii NVIDIA CUDA. Żeby to zrobić musiałem zainstalować Visual Studio i NVIDIA Emulator for Every PC. Niestety miałem problemy z działaniem tego programu i nie mogłem używać tej technologii u siebie na komputerze.

Byłem również ScrumMasterem, więc dodawałem zadania do wykonania do tablicy na Trello.

### Sprint 3

W tym Sprincie miałem więcej do zrobienia niż w 2 poprzednich dwóch. Mając do dyspozycji program działający w CUDZIE zabrałem się do przerabiania go na język C. W tym celu musiałem pobrać CodeBlocksa i odpowiednio przystosować kod, by działał on na moim komputerze, który nie jest wyposażony w kartę graficzną firmy NVIDIA.

Rozpocząłem to zadanie w tym sprincie i poczyniłem w nim dość duże postępy, z nadzieją, że   
w następnym sprincie będzie dało się go skończyć.

### Sprint 4

W tym sprincie musiałem skończyć program, który zacząłem w poprzednim sprincie. Bez większych trudności udało mi się go skończyć. Program prosił o podanie ilości współrzędnych dla każdej z 2 danych, następnie o podanie tych współrzędnych. Program wyświetlał później korelacje podanych danych, czyli metryki, które zdefiniowane były jako funkcje. Wszystko kompilowało się i nie zwracało błędów.

Poza tym dodałem parę zadań do Trello.

### Sprint 5

W innych sprintach dużo się działo, jednak ten był najintensywniejszy ze wszystkich. Trwał on 2 tygodnie a nie tydzień jak większość pozostałych sprintów, lecz i tak nakład pracy na dzień był największy. Kamil skończył program w CUDZIE, który działał poprawnie, generował tablicę liczb   
a następnie losowo wybierał z niej liczby i liczył dla nich metryki, a następnie oceniał, czy są one większe czy mniejsze od założonego celu. Więc musiałem zrobić to samo w zwykłym języku C. Mieliśmy w programie dużo metryk, więc samo przerabianie ich zajęło dość dużo czasu. Oprócz tego musiałem zmienić maina, by nie wykorzystywał technologii CUDA, przy zachowaniu swojej funkcjonalności. Natknąłem się tutaj na problem, którego nie umiałem rozwiązać, program kompilował się, lecz w pewnym momencie działania wyłączał z błędem. Nie umiałem go rozwiązać, więc poprosiłem o pomoc Kamila, który zauważył błędne oznaczenie zmiennych i naprawił go.

Kolejną czynnością którą wykonałem była dokumentacja i finalizacja projektu. Napisałem dokumentację sprintu trzeciego, którą Konrad zawarł w tym sprawozdaniu. Następnie zająłem się prezentacją, spędziłem nad nią dość dużo czasu. W prezentacji przedstawiłem nasz projekt i co robiliśmy po kolei żeby go stworzyć. Oprócz tego opisałem całość swojej pracy nad projektem, sprint po sprincie.

### Wykresy

#### Trello

Niestety miałem problemy z wygenerowaniem wykresów w Trello, w niektórych sprintach nie dodałem czasu pracy, ale mimo tego w wykresach zawierało się go mniej niż dodałem.

#### GitHub

Wykres comitów ze Scruma

## Wkład Kamila

### Sprint 1

Podczas sprintu 1, z racji tego, że w Krakowie jako jedyny posiadałem kartę graficzną NVIDIA w swoim komputerze, zająłem się przygotowaniem środowiska do pracy. Niestety sprawiło mi to duży problem, ponieważ potrzebna była odpowiednia wersja Visual Studio (na początku ściągnąłem nieprawidłową, a program waży w okolicach 10 gigabajtów), później miałem problemy, aby połączyć Visuala z CUDą i ostatecznie tego celu nie udało się wykonać podczas pierwszego sprintu. Poza tym zapoznałem się z narzędziami pracy grupowej, jakich potrzebowaliśmy do pracy , zainstalowałem dodatek Plus For Trello, dzięki któremu na bieżąco mogłem śledzić czas swojej pracy.

### Sprint 2

W 2 sprincie powróciłem do problemu konfiguracji środowiska CUDA na komputerze. Na szczęście już na początku sprintu problem udało się rozwiązać i mogłem przejść do pracy. Najpierw skompilowałem przykładowe programy, a następnie wziąłem się za szkicową wersję naszego programu, która to liczyła odległości między dwoma n-wymiarowymi wektorami w różnych metrykach.

### Sprint 3

Głównym moim celem w sprincie 3 było przerobienie napisanego uprzednio programu, aby nie działał na jednym wątku, a na kilku, dzięki temu usprawni to jego działanie. Dodatkowo zaimplementowałem kolejne obliczenia metryk.

### Sprint 4

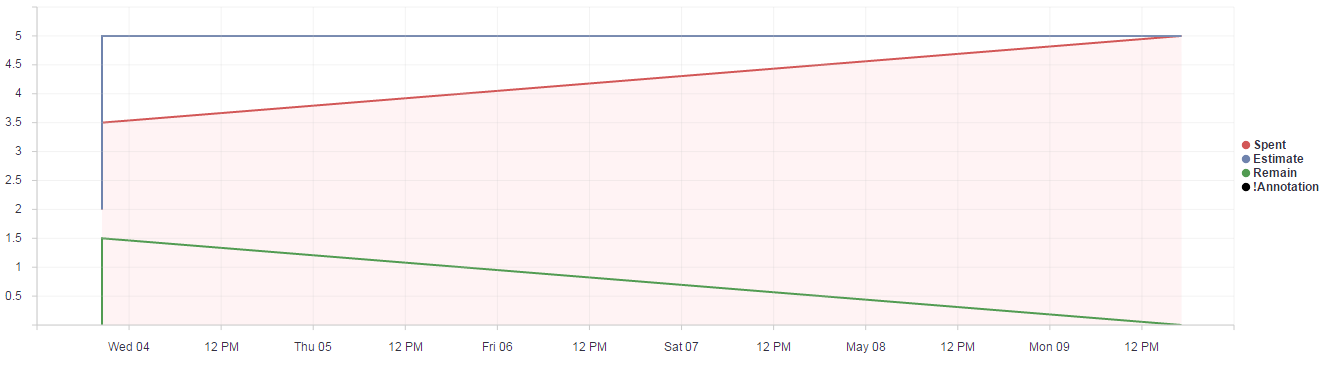
Podczas tego sprintu zakończyłem długi proces dodawania metryk. W sumie udało się ich zaimplementować 35. Kod programu wydłużył się do ok. 500 linijek, więc nie obyło się też bez debuggowania. Na szczęście program Visual Studio jest w tym bardzo pomocny i udało się to zrobić dużo szybciej niż korzystając ze zwykłego edytora.

### Sprint 5

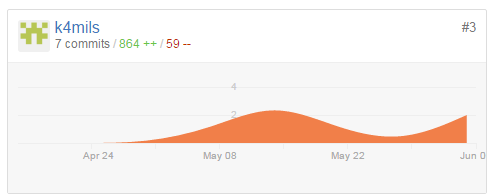
Sprint 5 był tym, w którym pracy było najwięcej. Trwał on dwa tygodnie i był tym najintensywniejszym, ponieważ wszystkie zadania trzeba było doprowadzić do końca i sporządzić dokumentację. Napisałem program, który losuje 90 000 liczb rzeczywistych, a następnie wstawia je do pliku tekstowego. Liczb miało być milion, niestety mój komputer nie przyjął takiej dużej ilości liczb i program obliczający metryki po uruchomieniu powitał mnie błędem. Do programu z metrykami dodałem również funkcjonalność polegającą na tym, że program ten z pliku tekstowego opisanego powyżej losuje 4 liczby, które to są numerami kolumn i wierszy, które się przetną. Przecięcie ich wyznacza nam dwa wektory, między którymi liczymy odległości w poszczególnych metrykach,  
 a następnie porównujemy z danymi użytkownika na wejściu (thresholdem). Zdebugowałem program Michała, który to samo robi w C. Napisałem dwa sprawozdania ze sprintów (2 i 3).

### Wykresy

#### Trello



#### GitHub



# Podsumowanie

Zespołowi udało się osiągnąć cel projektu.

Zespół stworzył program który analizuje bazę danych zawartą w pliku tekstowym. Do tego celu została użyta równoległa architektura obliczeniowa CUDA.

W trakcie trwania projektu stworzono również aplikację która porównuje szybkość wykonywania analiz szeregowo i równolegle – za pomocą cudy. Okazało się, że program wykonywany w cudzie działa dużo szybciej. Cuda zapewnia radykalny wzrost wydajności.

W celu wykonania przykładowej analizy danych wygenerowano plik wypełniony tablicami losowych liczb.

Program działa w te sposób, że dokonuje losowych wyborów wierszy zbudowanej macierzy. Następnie dla danych wierszy losowana jest liczba analizowanych współrzędnych i dane współrzędne.

Wybrane współrzędne są poddawane obliczeniom ich metryk, dzięki czemu można określić stopień ich korelacji.

Metodologia organizacji pracy „Scrum” zasadniczo usprawniła wykonanie projektu. Dzięki jednoznaczności obieranych celów udawało nam się osiągać systematyczne i efektywne przyrosty pracy, jednocześnie nie tracąc czasu na brak organizacji.

Scrum bardzo dobrze sprawdza się w dużym natłoku pracy. O ile w poprzedzających sprint piąty sprintach nie były zauważalne zalety Scruma, to w sprincie piątym Scrum sprawdził się wyjątkowo dobrze. Umożliwił skuteczną organizację pracy. Scrum sprawdza się bardzo dobrze w obszernych projektach, które należy wykonać w krótkim czasie. Znacząco zwiększa wydajność pracy.

W poszczególnych sprintach zespół posługiwał się metodą tablicy Kanbanowej. W celu organizacji sprintu tą metodą użyliśmy serwisu Trello.

Adaptacja do systemu Kanbanowego zajęła zespołowi jeden tydzień. W tym czasie zespół praktykował Daily Scrumy które okazały się nieskuteczne. Daily Scrum to bardzo dobra metoda podnoszenia wydajności pracy wykonywanej ciągle. Nasz zespół musiał zajmować się także innymi obowiązkami związanymi ze studiami, dlatego praca nad projektem nie była systematyczna i Daily Scrum nie miał dużego sensu stosowania.

Pracując nad programem korzystaliśmy z narzędzia GitHub wykorzystującego system kontroli wersji Git. W trakcie trwania całego Scruma nie skorzystaliśmy z możliwości systemu kontroli wersji. Pracą dzieliliśmy się w taki sposób, że nie było konieczne pracowanie na tych samych plikach. Nie wystąpiła żadna kolizja wprowadzenia danych.

Bardzo przydatnym okazał się dostęp do repozytorium projektu. Brak konieczności wymieniania się plikami z wynikami pracy poprzez rozsyłanie każdorazowo do członków projektu plików znacząco podnosi komfort pracy i pozwala uniknąć wielu nieporozumień.

Projekt umożliwił członkom zespołu nabyć wiele umiejętności związanej z pracą zespołową. Bardzo ważne jest zdobyte doświadczenie pracy nad wspólnym celem z innymi osobami.

Dzięki projektowi studenci poznali wiele metod i narzędzi których wcześniej nie znali, a które znacznie poprawiają organizację i wydajność pracy zespołowej.

# Źródła

## Metryki

### Wikipedia.pl

1. **Artykuł:” Przestrzeń metryczna”** (https://pl.wikipedia.org/wiki/Przestrze%C5%84\_metryczna), 22.05.201616r.)
2. **Artykuł „Odległość Hamminga”** (https://pl.wikipedia.org/wiki/Odleg%C5%82o%C5%9B%C4%87\_Hamminga, 22.05.2016r.:)

### MathDotNet

1. **Artykuł: Distance Metrics** (http://numerics.mathdotnet.com/Distance.html, 22.05.201616r.)

### MATHWORLD– WOLFRAM

1. **Artykuł: French Metro Metric** (http://mathworld.wolfram.com/FrenchMetroMetric.html), 22.05.2016r.:

Sung-Hyuk Cha**,**

1. **” *Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions”*** Issue 4, Volume 1, 2007: