Sylwia Nowak, Błażej Bobko, **Patryk Kujawski**, Jakub Cieślik

|  |  |
| --- | --- |
| Technologia | |
| Język programowania: | C# |
| Framework: | .net 4.5 |

|  |
| --- |
| Metodologia |
| SCRUM - agile software developmentCore Roles  * Product owners - prowadzący: Janusz Rafałko, Agnieszka Jastrzębska, Michał Okulewicz, * Development team - nasz zespół, * SCRUM master - w naszym przypadku przyjmie postać team-leader'a - Sylwia Nowak.  Dlaczego ta metodyka?  * uwzględnia ona możliwość szybkiej komunikacji i wymiany danych online, * szybkie spotkania (standupy, daily SCRUM meetings) na uczelni, * nie planujemy 100% przebiegu implementacji - nasze pomysły poza szkieletem z dokumentacji mogą prowadzić do zmian w implementacji poszczególnych modułów, * nie musimy od samego początku skupiać się na całości rozwiązania - na bieżąco myślimy o tym co jest aktualnie najważniejsze do poprawnego działania projektu.  Nasze zadanieSprint Czyli 1 iteracja naszego projektu - każdy sprint odnosić się będzie do kolejnych milestone' ów zadanych w postaci terminów do realizacji kolejnych części projektu.  Spełnione przy tym są założenia metodologii:   * sprint trwa od 2 do 4 tygodni, * na końcu każdego spritnu oddajemy działający program, przetestowany oraz udokumentowany, * po oddaniu gotowej części projektu musimy ustalić co będzie wykonywane w kolejnej iteracji, co jest najważniejsze i musi być wykonane w pierwszej kolejności.  Meeting  * uzgadniamy plany tygodniowe:   + co jest do wykonania na teraz,   + uzgodnienie, co jesteśmy w stanie zrobić,   + odbywają się poza czasem zajęć na uczelni.  Daily scrum meeting (stand-up meeting)  * szybkie spotkania na uczelni (na stojąco).  Artefakty  * Product backlog - tworzymy sprint backlogi (kto co zrobił, robił i na jakim jest etapie), * WPS - work break down structure - rozbicie całego projektu na części elementarne w postaci drzewa. |

## Organizacja pracy

Zbudowana przy wykorzystaniu 3 pierwszych kroków metody WPS. Kolejne podziały będziemy uzgadniać na bieżąco analizując nasze aktualne problemy.

|  |
| --- |
| Sprint 1 - 18.03.2015 Wstępna implementacja architektury komponentów systemu. Sprint 2 - 25.03.2015 Zaimplementowana pełna komunikacja między komponentami systemu. Testowanie - 25.03.2015 do 08.03.2015Milestone - communication - 08.04.2015 |

|  |
| --- |
| Sprint 3 - 15.04.2015 Wstępna wersja algorytmu (działająca, nie w pełni zoptymalizowana). Sprint 4 - 22.04.2015 W pełni zoptymalizowana wersja algorytmu. Testowanie - 22.04.2015 do 29.04.2015Milestone - algorithm - 29.04.2015 |

|  |
| --- |
| Sprint 5 - 06.05.2015 W pełni zintegrowane środowisko obliczeniowe klastra. Sprint 6 - 13.05.2015 Ukończony system obliczeniowy zoptymalizowany pod względem szybkości i wydajności komunikacji i obliczeń. Testowanie - 13.05.2015 do 27.05.2015Milestone - final application - 27.05.2014 |