Inżynieria Oprogramowania 2 - Kalendarz Metodologia: Unified Process								
Tydzień	Michał N	tierzyński Komentarz	Michai Plan	Padzik Komentarz	Łukasz Plan	Napora Komentorz	Kam Plan	il Żak Komentarz
1	Płon Zapoznanie się z dokumentacją (3h)	Romentorz Projektowanie architektury wspólnej komunikacji (Network Adapter używany w komponentach poza serwerem)	Zapoznanie się z dokumentacją (3h)	Romentorz Pytania odnośnie dokumentacji i podział prac w grupie. Projektowanie biblioteki dla całego zespołu (3h)	Zapoznanie się z dokumentacją (3h)	Zapoznanie się z dokumentacją, wyjaśnienie niejasność. Podział prac w grupie (3h).	Zapoznanie się z dokumentacją (3h)	Romentorz Po zapoznaniu się z dokumentacją, konieczne byłk wyjaśnienie paru niejasności z prowadzącymi przedmiot (1h) Przedyskutowanie problemu z grupą (2h).
2	Wstępna implementacja i projektowanie komunikacji - Task Manager (3h)	Stworzenie szkieletu Task Managera.	Stworzenie szkieletu projektu serwera i klas odpowiadających za rejestrację komponentów (3h)	Stworzenie szkieletu serwera i dokończenie biblioteki (3h)	Wstępna implementacja i projektowanie komunikacji - Computational Node (3h)	Stworzenie szkieletu Computational Noda(3h).	Wstępna implementacja i projektowanie komunikacji - Client (3h)	Założenie repozytorium zgodnie z załeceniami (2h). Stworzenie szkieletu klienta (1h).
3	Dopracowywanie komunikacji - Task Solver (3h)	Eksport kodu do nowego repozytorium svn. Projektowanie i impementacja Task Managera, Netrok Adaptera.	Dopracowanie komunikacji - Server (3h)	Implementacja odbierania i wysylania wiadomości (3h)	Dopracowanie komunikacji - Computational Node (3h)	Projektowanie i implementacja Computational Noda(3h).	Dopracowanie komunikacji - Client (3h)	Stworzenie całego klienta (wersja niedziałająca). Ponadto opracowanie części projektu Common wspólnego dla wielu komponentów (całość 3h).
4	Dopracowywanie komunikacji - Task manger (3h)	Projektowanie i impementacja Task Managera, Netrok Adaptera.	Dopracowanie komunikacji - Server (3h)	implementacja prostego algorymu rozdzielającego zadania pomiędzy komponenty (3h)	Dopracowanie komunikacji - Computational Node (3h)	Projektowanie i implementacja Computational Noda(3h).	Dopracowanie komunikacji - Client (3h)	Dopracowanie clienta – wersja poprawnie nawiązująca połączenie i wysyłająca cokolwiek (3h). Nadprogramowe (2h) na poprawki związane ze zmianą sposobu komunikacji.
5	Testowanie komunikacji - Task Manager (3h)	Zmiana w sposobie komunikacji w serwerze, spowodowana ustaleniami z innymi grupami. Unit testy.	Testowanie komunikacji – Server (3h)	Wprowadzanie poprawek zauważonych podczas testów (3h)	Testowanie komunikacji - Computational Node (3h)	Wprowadzenie poprawek zwiazanych ze zmianą sposobu komunikacji(dodatkowe 4h).	Testowanie komunikacji - Client (3h)	Dopracowanie clienta - wersja działająca także z serwerami innych zespotów (3h). Nadprogramowe (6h) związane z ponowną zmianą sposobu komunikacji. Kolejne (4h) więcej na stworzenie dokumentacji oraz Lieitfactiwa.
6	Działająca komunikacja pomiędzy innymi komponentami aplikacji	Refaktoryzacja kodu.	Oddanie etapu komunikacji (th), szkielet klas odpowiadających za algorytmy rozwiązywania problemów (zh)	Refaktoryzacja kodu	Oddanie działającej komunikacji pomiędzy innymi komponentami aplikacji (3h)	Refaktoryzacja kodu.	Oddanie działającej komunikacji pomiędzy innymi komponentami aplikacji (1h), przygotowanie do implementacji - Client (2h)	Ostatnie poprawki, zwłaszcza w wyglądzie kodu. Poprawienie drobnych błędów. (Całość 3h)
7	Implementacja i integracja algorytmu - Task Solver (3h)		Dopracowanie algorytmów rozwiązywania problemów (3h)	Refaktoryzacja kodu	Implementacja i integracja algorytmu - Computational Node (3h)	Praca nad szabionem algorytmu.	Implementacja i integracja algorytmu - Client (3h)	
8	tmplementacja i integracja algorytmu - Task Solver (3h)		Testowanie działania algorytmów rozwiązywania problemów (3h)	Refaktoryzacja kodu	Dopracowanie algorytmu - Computational Node (3h)	Praca nad algorytmem.	Poprawki w algorytmie - Client (3h)	
9	Działający algorytm		Oddanie etapu algorytmów (1h), dopracowanie poprzednich etapów (2h)	Refaktoryzacja kodu	Oddanie działającego algorytm (3h)	Testowanie algorytmu.	Oddanie działającego algorytmu (Ih), ewentualne poprawki w kodzie (2h)	
10	Testowanie i implementacja pozostałych funkcjonalności - Task Manager (3h)		Dopracowanie poprzednich etapów (3h)	Refaktoryzacja kodu	Testowanie i implementacja pozostałych funkcjonalności - Computational Node (3h)	Dodanie brakujacych funkcjonalności.	Testowanie i implementacja pozostałych funkcjonalności - Client (3h)	
11	Testowanie - Task manger (3h)		Testowanie działania całego programu serwera (3h)	Refaktoryzacja kodu	Testowanie gotowego Computational Node (3h)	Testowanie połączenia z innymi grupami.	Testowanie calego modulu - Client (3h)	
12	Testowanie - Task Solver (3h)		Testowanie całego projektu (3h)	Refaktoryzacja kodu	Testowanie całego projektu (3h)	Testowanie połączenia z innymi grupami.	Testowanie całego projektu (3h)	
13	Oddanie projektu (3h)		Odďanie projektu (3h)	Refaktoryzacja kodu	Odďanie projektu (3h)	Refaktoryzacja kodu.	Oddanie projektu (3h)	
14	Championship (3h)		Championship (3h)		Championship (3h)		Championship (3h)	