

Inżynieria Oprogramowania 2 - Kalendarz								
Metodologia: Unified Process								
Tydzień	Michał Mierzyński		Michał Padzik		Łukasz Napora		Kamil Żak	
	Plan	Komentarz	Plan	Komentarz	Plan	Komentarz	Plan	Komentarz
1	Zapoznanie się z dokumentacją (3h)	Projektowanie architektury wspólnej komunikacji (Network Adapter używany w komponentach poza serwerem)	Zapoznanie się z dokumentacją (3h)	Pytania odnośnie dokumentacji i podział prac w grupie. Projektowanie biblioteki dla całego zespołu (3h)	Zapoznanie się z dokumentacją (3h)	Zapoznanie się z dokumentacją, wyjaśnienie niejasności. Podział prac w grupie(3h).	Zapoznanie się z dokumentacją (3h)	Po zapoznaniu się z dokumentacją, konieczne było wyjaśnienie paru niejasności z prowadzonymi przedmiot (1h). Przedyskutowanie problemu z grupą (2h).
	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 zaleceniami (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 wysyłania 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).
	Dopracowywanie komunikacji - Task manger (3h)	Projektowanie i impementacja Task Managera, Netrok Adaptera.	Dopracowanie komunikacji - Server (3h)	Implementacja prostego algorytmu rozdzielającego zadania pomiędzy komponenty (3h)	Dopracowanie komunikacji - Computational Node (3h)	Projektowanie i implementacja Computational Noda(3h).	Dopracowanie komunikacji - Client (3h)	Dopracowanie klienta - 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 związanych ze zmianą sposobu komunikacji(dodatkowe 4h).	Testowanie komunikacji - Client (3h)	Dopracowanie klienta - wersja działająca także z serwerami innych zespołów (3h). Nadprogramowe (6h) związane z ponowną zmianą sposobu komunikacji. Kolejne (4h) więcej na stworzenie dokumentacji oraz UnitTestów.
	Działająca komunikacja pomiędzy innymi komponentami aplikacji	Refaktoryzacja kodu.	Oddanie etapu komunikacji (1h), szkielet klas odpowiadających za algorytmy rozwiązywania problemów (2h)		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)		Implementacja i integracja algorytmu - Computational Node (3h)		Implementacja i integracja algorytmu - Client (3h)	Implementacja algorytmu - metoda Solve (3h)
	Implementacja i integracja algorytmu - Task Solver (3h)		Testowanie działania algorytmów rozwiązywania problemów (3h)		Dopracowanie algorytmu - Computational Node (3h)		Poprawki w algorytmie - Client (3h)	Poprawienie metody Solve algorytmu DVRRP (3h)
9	Działający algorytm		Oddanie etapu algorytmów (1h), dopracowanie poprzednich etapów (2h)		Oddanie działającego algorytm (3h)		Oddanie działającego algorytmu (1h), ewentualne poprawki w kodzie (2h)	Integracja algorytmu z komponentami aplikacji (3h)
	Testowanie i implementacja pozostałych funkcjonalności - Task Manager (3h)		Dopracowanie poprzednich etapów (3h)		Testowanie i implementacja pozostałych funkcjonalności - Computational Node (3h)		Testowanie i implementacja pozostałych funkcjonalności - Client (3h)	Testowanie działania algorytmu wraz z komponentami w zakresie grupy oraz naniesienie odpowiednich poprawek (3h).
11	Testowanie - Task manger (3h)		Testowanie działania całego programu serwera (3h)		Testowanie gotowego Computational Node (3h)		Testowanie całego modułu - Client (3h)	Poprawienie klienta w zakresie działania algorytmu - poprawny timeout oraz sprawdzanie rodzaju otrzymanego rozwiązania.
	Testowanie - Task Solver (3h)		Testowanie całego projektu (3h)		Testowanie całego projektu (3h)		Testowanie całego projektu (3h)	Testowanie z inną grupą algorytmu wraz z usprawnieniem klienta na podstawie błędów z aplikacją po podłączeniu do innej grupy (3h)
13	Oddanie projektu (3h)		Oddanie projektu (3h)		Oddanie projektu (3h)		Oddanie projektu (3h)	Ostatnie poprawki błędów wykrytych w działaniu aplikacji klienta podczas testowania z inną grupą. (2h) + oddanie projektu (1h)
	Championship (3h)		Championship (3h)		Championship (3h)		Championship (3h)	