**Nazwa przedmiotu:**

Reliable, Scalable and Maintainable IT Systems

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Fakultatywny dowolnego wyboru

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - zaawansowane

**Kod przedmiotu:**

ERSMS

**Semestr nominalny:**

3 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. liczba godzin kontaktowych – 47 godz., w tym
obecność na wykładach 30 godz.,
spotkania projektowe 15 godz.,
konsultacje 2 godz.,
2. praca własna studenta – 70 godz., w tym
przygotowanie do kolokwium 25 godz.,
przygotowanie do zadania projektowego 15 godz.,
realizacja projektu 30 godz.
Łączny nakład pracy studenta wynosi 117 godz., co odpowiada 4 pkt. ECTS.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1,61 pkt. ECTS, co odpowiada 47 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2,05 pkt. ECTS, co odpowiada 15 godz. spotkań projektowych plus 45 godz. przygotowania i realizacji projektu

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 15h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Osoby uczęszczające na przedmiot powinny przede wszystkim mieć wiedzę dotyczącą systemów operacyjnych i sieci komputerowych oraz praktyczną umiejętność programowania w nowoczesnych językach programowania wysokiego poziomu. Realizacja projektu informatycznego wymaga również znajomości co najmniej podstawowych zagadnień dotyczących inżynierii oprogramowania.

**Limit liczby studentów:**

36

**Cel przedmiotu:**

Rozwój i wdrażanie niezawodnych, skalowalnych i zarządzalnych systemów informatycznych niesie ze sobą szereg wymagań wynikających ze specyfiki tych systemów. Celem przedmiotu jest przekazanie wiedzy niezbędnej do sprostania tym wymaganiom.

**Treści kształcenia:**

Wykład:
1. Introduction to Large-scale Systems
Distributed Systems
Transparency
Openness
Scalability
Decentralized Algorithms
Scaling Techniques
Vertical and Horizontal Distribution
2. Reliability, Scalability and Maintainability
Reliability
Scalability
Maintainability
3. Data Models and Query Languages
Relational and Object Data Model
Document Data Model
Query Languages for Data
Graph-Like Data Models
4. Datastores
Data Structures in Databases
Data Warehousing
Column-Oriented Storage
5. Dataflows and Communication
Protocols
Networking Review
Middleware Protocols
Data Encoding Formats
Dataflow: Databases
Dataflow: RPC and REST
Dataflow: Message-Passing
Sockets
MPI
Message-Queuing Systems
6. Synchronization
Physical Clocks
Clock Synchronization Algorithms
Lamport’s Algorithm
Vector Timestamps
Causal Delivery of Messages
7. Replication
Synchronous vs Asynchronous Replication
Persistent and Transient Communication
Replication Lag
Multi-Leader Replication
Leaderless Replication
Mutual Exclusion
8. Partitioning
Partitioning of Key-Value Data
Partitioning and Secondary Indexes
Partitions Rebalancing
Election Algorithms
9. Transactions
Transaction Concept
Weak Isolation Levels
Serializability
Two-Phase Locking
10. Fault Tolerance
Unreliable Networks
Unreliable Clocks
Byzantine Faults
11. Consistency and Consensus
Linearizability
CAP Theorem
Ordering and Causality
Lamport Timestamps
Distributed Transactions and Consensus
The Impossibility of Consensus
Two-Phase Commit
Three-Phase Commit
12. Batch Processing
Batch Processing with Unix Tools
Map Reduce Workflows
Graphs and Iterative Processing
13. Stream Processing
Stream-Oriented Communication
Messaging Systems
Partitioned Logs
Stream Processing
Quality of Service
Reasoning About Time
Zajęcia projektowe:
Student realizuje zadanie projektowe w grupie 4-6 osobowej.
Każda grupa projektowa ma za zadanie
a. zaprojektować,
b. zaimplementować,
c. przetestować,
d. zademonstrować odbierającemu projekt
zadaną i zdefiniowaną przez prowadzącego projekt z przedmiotu usługę.
System powinien być uruchamialny w środowisku zdalnym bądź chmurowym, do którego można się odwołać z komputerów klienckich pracujących pod kontrolą systemu Linux. Dla części serwerowej wymagane jest również wykorzystanie środowiska systemu Linux.
Efektem końcowym projektu powinny być:
- własne działające oprogramowanie,
- zautomatyzowana procedura prezentacji możliwości zrealizowanego oprogramowania (obligatoryjnie w postaci skryptu sh, z podziałem na etapy),
- pełna dokumentacja projektu w postaci elektronicznej zawierająca co najmniej następujące pozycje:
o uporządkowane wstępne opracowania,
o doprecyzowaną względem treści szczegółową koncepcję zaproponowanego
o rozwiązania,
o opis organizacji grupowego wytwarzania oprogramowania, przypisanych ról w projekcie, sprawozdanie z prac nad projektem, czyli listę
o spotkań roboczych wraz z datami i ustaleniami,
o dokumentację techniczną proponowanego rozwiązania,
o dokumentację testów,
o dokumentację uruchamiania i konfiguracji systemu,
o opis przygotowanych skryptów testowych i prezentacyjnych.
W każdej grupie projektowej powinny być zdefiniowane następujące role:
- kierownik projektu,
- architekt - osoba odpowiedzialna za wytworzenie architektury rozwiązania,
- osoba odpowiedzialna za przygotowanie i zarządzanie repozytorium projektu oraz przechowywanie aktualnej kopii zapasowej projektu,
- dokumentalista - osoba odpowiedzialna za zarządzanie dokumentacją projektu, zbieranie i ujednolicanie dokumentacji technicznej tworzonej przez pozostałych członków zespołu oraz m.in. protokołowanie ustaleń z poszczególnych spotkań roboczych zespołu projektowego,
- tester - osoba odpowiedzialna za przygotowanie planu testów, w tym testów obciążeniowych, niezawodnościowych i synchronizacyjnych, ich przeprowadzenie oraz opisanie,
- handlowiec - osoba odpowiedzialna za przygotowanie powtarzalnej, maksymalnie zautomatyzowanej demonstracji końcowej prototypu oraz samo sprawne przeprowadzenie końcowej demonstracji działania efektu pracy zespołu projektowego.
Z dokumentacji musi jasno wynikać, kto i w jakim stopniu uczestniczył w realizacji poszczególnych elementów projektu.

**Metody oceny:**

Wykład prowadzony w tradycyjnej formie.
Realizacja projektu będzie rozbita na kilka etapów (wyniki uzyskane po zakończeniu każdego z nich będą prezentowane i dyskutowane z prowadzącym w trakcie 2-3 godzinnego spotkania):
1. analizę biznesową przedstawionego problemu,
2. przygotowanie technicznych założeń i projektu architektury systemu,
3. implementację systemu realizującego wymagania projektu,
4. wytworzenie i demonstrację docelowego rozwiązania.

**Egzamin:**

nie

**Literatura:**

1. “Designing Data-Intensive Applications”, Martin Kleppmann, O’Reilly Media Inc. 2017
2. „Distributed Systems”, Maarten van Steen, Andrew S. Tanenbaum, CreateSpace Independent Publishing Platform; 3.01 edition, 2017

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103A-INISY-MSP-ERSMS

**Uwagi:**

(-)

## Charakterystyki przedmiotowe