CSCI427/927 Service-Oriented Software Engineering

Revision

Outline

- Revision
- Exam preparation

Revision

- Don’t look at the slides, list 20 (or more) things you learn from this subject
- Now, share the list with your friends and discuss what you have missed and what they have missed.

Topics (2nd half)

- Service-oriented architectural patterns
- Service analytics (Process mining and analytics)
- Service change management
- Service compliance management
- Case studies + Modern industry trends (LLM)

Service-oriented architectural patterns

- **Purpose:** Implement scalable, adaptable, and reliable systems using service-oriented architecture.
- **Key Topics:**
  - **Microservices Architecture:** Independent, stateless services, each responsible for a single functionality, own **database** and UI **management code**
  - Service Communication: Synchronous vs. asynchronous interactions, direct vs. indirect service communications (e.g., using message brokers).

- Microservice data design
  - **isolate data + most sharing** is '**read-only**'
  - Inconsistency management=>Eventual consistency
- Service coordination (workflow)
- Service failure
  - Timeouts and circuit breakers
- **RESTful Services:** Use HTTP methods (GET, POST, PUT, DELETE) to interact with resources (advantage and disadvantage).
- **Deployment & Management:** Continuous deployment practices in microservice architectures (DevOps).

Service analytics

- **Purpose**: Analyze and improve service/process execution using analytics techniques.
- **Key Topics:**
  - **Process Mining:** Reverse engineer process models from process/event logs, focusing on identifying patterns and inefficiencies.
  - **Predictive Analytics**: Forecast outcomes like cost, performance, and resource consumption based on historical data.
  - **Prescriptive Analytics**: Recommend actions to optimize process flows and resource allocation.

Service change management

- **Purpose:** Manage inevitable changes in software systems due to evolving business, infrastructure, and technology environments.
- **Key Topics**:
  - **Change Propagation:** Ripple effects in systems requiring coordinated changes across UI, database, and logic layers.
  - **Consistency Constraints:** Use of Object Constraint Language (OCL) to maintain system consistency during changes.

- **Repair Plans:** Triggering events and context conditions (with plan body)
  - **Semantic Process Networks (SPNet)**
  - minimal-disruption of service choreography (i.e. UML activity diagram)

Service Compliance Management

- **Purpose:** Ensure organizational activities comply with rules from legal frameworks and internal policies.
- **Key Topics:**
  - **Compliance Management Process:** Discovery, modeling, checking, resolution, and managing changes in compliance obligations, compliance-by-design, and monitoring
  - **Semantic Process Monitoring**: Use process annotations to track compliance during execution.
  - **Semantic Compensation**