ISSSR 2017-2018

Program

Software measurement. Theory of measurement. Introducing to software measurement. Types of Software Measurements Models (MM). The MM as a product. Developing a MM: the MM Life-cycle. Practical examples: size measurements, functional measurements and non-functional measurements. Function Point. Constructive Cost Model and its evolutions.

IT supports for organizations management. ***The "Goal Question Metrics plus Strategies" (GQM+S) approach for measurement-based alignment of goals and strategies in an organization***. GQM+S in Nutshell (Part1 1 e 2). ***IT Infrastructure Library*** (***ITIL***): Practices for IT service management that focuses on aligning IT services with the needs of business. **Balanced Scorecards**(BSC).

Introducing measurements in industrial organizations. ***Measurement Ontologies***. From basic software measurement conceptual models and related evolutions (Cantone, Donzelli *et al*. 1995, 2009, 2016) to a Software measurement model ontology (Garsia, Piattini *et al*. (2006) to a Reference domain ontology for developing a software measurement strategy for high maturity organizations (Barcellos *et al*. 2012, 2013, 2016).

Empirical/Experimental Software Engineering. Experimental Models for Validating Sw. Technology. From Surveys to Case studies. Software Engineering Experimentation. Basic Principia. The experimental process. Definition phase. Planning phase. Design for one or more factors/treatments. Blocking on undesired factors. Validity evaluation. Operation. Data analysis and Interpretation. Big data: report on some nowadays experiences in a major company. Assigned project (eligible): Improving accuracy of learning AI-based ticket classifiers.

Software development. ***Process, and Higher Level Disciplines***. The Unified Process (UP). The Agile approach. Scrum. Requirement Management. Project Management.

Software Architecture, and software development-supporting Technologies. ***Service-oriented Architecture***. REST services, their invocation, and answering. Spring MVC. Bus-based architecture. ***Designing & implementing the backend***: Model-View-Control, Presentation-View-Control. Spring Data. Hibernate. ***Designing & implementing the frontend***: Angular. ***Cloud.*** Cloud Computing, and DevOps. Characteristics and usage of the IBM Cloud.

(Capstone course) Development projects (to be developed by using what has been shown above). ***P1***: Agile team-conducted - 7 pp, an iteration per week,- project for the integration and evolution of a GQM+Strategies software support software tool such as developed by three previous academic years' classes. ***P2***: Agile team-conducted - 6 teams, 6 pp/team in the average - development from the scratch of a system for managing the acquisition and handling of Tickets.