Spring 2024

# IS 663-852: System Analysis & Design

Vassilka Kirova

New Jersey Institute of Technology
Universeity Heights
Newark, NJ 07102
http://njit.edu

# IS 663 – System Analysis & Design – S2024 (Preliminary)

## Part I: Course and Instructor Information

| | |
|---|---|
| **Semester:** | Spring 2024 |
| **Course name:** | **System Analysis & Design** |
| **Course number:** | **IS 663** |
| **Instructor name:** | Vassilka Kirova |
| **Course location:** | Hybrid class - **WebEx** and **CKB 313** |
| **Course meeting time:** | 6:00 – 8:50 PM ET |
| **Office hours & location**: | Monday, 6:30 – 8:30 PM ET – online (by appointment)<br>Wednesday, 8:50-9:50 PM – online or in class, based on class mode |
| **Email:** | vassilka.kirova@njit.edu<br>**For all emails use the following subject: IS663-<additional info>** |

## Part II: Course Description

### 1. Course description:

This course develops the skills necessary to analyze, design and manage the development of enterprise-scale information system solutions, incorporating contemporary methods and effective organizational and global project management practices. It focuses on technical business system analysis and design techniques, and covers key software engineering principles, methods, and frameworks, including process models, agile and lean principles, project and risk management, estimation, requirements elicitation and analysis, modeling, system and software architecture, design patterns, and quality systems. Students will actively participate in discussions, review selected articles, participate in team exercises and collaborate on projects involving analysis and prototyping of applications addressing real-world problems and integrating current and emerging technologies.

For the latest course information go to: IS663101-System Analysis and Design
The information below should help you plan and organize your preparation during the semester.

### 2. Prerequisite courses and knowledge:

- Pre or corequisite: IS 601
- Optional pre or corequisite: IS 631
- Required background:
  - The students are required to have knowledge of key information systems concept, software development life cycle activities, and project management issues.
  - Good understanding of data modeling techniques and database fundamentals is expected as well.
  - Good understanding of modern trends in business and information analysis, information technology, data modeling, object-oriented principles and agility are a plus
  - Undergraduate software engineering courses provide a good foundation to build on in this class.

### 3. Outcomes expected upon the completion of the course

Upon the completion of the course the students will be able to:

- Understand the structure, activities, flows, control loops, value creation and best practices related to classical and modern lifecycle models, including Agile methods such as Scrum and SAFe
- Analyze, contrast and compare different process models for suitability to project characteristic, needs and context
- Select, customize, or define hybrid process models for specific classes of development projects
- Perform market analysis, formulate alternatives and blueprint project ideas
- Run agile (simulated) start-up projects using Agile frameworks and performing the roles of a product owner, scrum master and team member.
- Build Product Backlogs (PBLs), writing user stories
- Do analysis, specification and design of software applications using techniques such as use cases, scenarios, design patterns and design heuristics
- Perform essential project management functions, including planning and risk management in global and collaborative projects
- Estimate projects using classical and agile estimation techniques, such as White band Delphi
- Build prototypes, understand DevOps concepts and CI/CD pipelines
- Build core (UML, agile UML) models
- Understand different architectural models, levels and types of architecture and perform comparative analysis
- Understand the impact, and opportunities provided by modern architectures and infrastructures such as Microservices architecture (MSA), containers, Kubernetes, open APIs, IaaS, PaaS, SaaS and serverless computing
- Understand the fundamentals of quality assurances and quality frameworks

**4. Ways that students will be assessed throughout the course:**

- Team project execution and deliverables - content, mastery of methods discussed in class and creativity; teamwork; research and analysis skills
- Discussions -- active participation and moderation of discussions; sharing of ideas and information related to the discussion topics; systematic progress with paper reading assignments
- Individual assignments – content, understanding of methods discussed in class and their effective use in the assignments, research and analysis skills
- Class participation – contribution to the discussions and exercises, sharing, collaboration
- Exam/Quiz – understanding of the course material and demonstrated effective application of the acquired knowledge and skills to solving practical problems.

**5. Required & Recommended texts:**

- *Lecture Notes*
  Lecture notes are the basic course material for this class. The notes are made available on Canvas every semester.
- *Textbook*: "Software Engineering," Ian Sommerville, Addison-Wesley, England, 10th Edition, 2018 (paperback).
  - NJIT Bookstore or
  - Amazon.com – Books
  - Check also for fee options on the internet.

- *Articles and Discussion Supporting Materials*
  For the list of readings check the Course Outline available on Canvas as well as Canvas Discussions forum.
- *Extra Reading (Books)*
  - Software Engineering at Google: Lessons Learned from Programming Over Time, by Titus Winters, Tom Manshreck, Hyrum Wright, O'Reilly, 2020.

- "Modern Software Engineering: Doing What Works to Build Better Software Faster," David Farley, 2022.
- "The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition)," Frederick P. Brooks, 1995.
- "The Design of Design: Essays from a Computer Scientists," Frederick P. Brooks, 2010.
- "Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum," Craig Larman and Bas Vodde, Pearson Education Inc, 2009.
- "Practices for Scaling Lean & Agile Development: large, Multisite and Offshore Product Development with Large-Scale Scrum," Craig Larman and Bas Vodde, Pearson Education Inc, 2010.
- "Agile Software Development with Scrum," Ken Schwaber and Mike Beedle, Prentice Hall, 2002.
- Scaled agile framework (SAFe) - https://www.scaledagileframework.com/
- "Design Patterns / Elements of Reusable Objet Oriented Software," Erich Gamma, Richard helm, Ralph Johnson, and John Vlissides (known as the "Gang of 4" of "GOF"), 1994.
- "UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)," Martin Fowler, 2008.
- UML Documentation & White Papers: http://www-01.ibm.com/software/rational/uml/
- Beyond the 12-Factor App, Kevin Hoffman (Free eBook on cloud-native development)

6. **Required software/hardware:**
   Free and open-source software; NJIT supported tools and hosting environments.

7. **Web resources:**
   See Class information on Canvas (https://njit.instructure.com/courses/33722)

8. **Instructor's contact information:**
   **Email:** vassilka.kirova@njit.edu
   **In the subject include *"IS-663"*** plus any details you would like to add.

   **Mailing Address:**
   Dr. Vassilka Kirova
   Department of Informatics, CCS
   NJIT, University Heights
   Newark, NJ 07102

**Part III: Mapping Learning Outcomes to Course Assessment**

| **Course Learning Outcomes** | **Measure** (i.e., exam, homework, rubric) |
|---|---|
| Good understanding of classical and modern lifecycle models, including agile methods | In class and online discussions; team project |
| Hands on analysis and specification skills, using methods such as use cases, scenarios, user stories and Product Backlogs | Team project; final exam |
| Good understanding of project management functions, including risk analysis in organizational, global and collaborative projects | Team project |
| Practical knowledge of estimation techniques | Team project; in class exercises |
| Understanding of architecture and design activities as well as the impact an opportunity provided by modern architectures and infrastructures such as cloud computing and SOA | Individual assignment, based on literature analysis; in class and online discussion |
| Hands on modeling skills (UML, agile UML) | Exam/Quiz |
| Understanding the fundamentals of Quality Assurances and quality frameworks | In class and online discussion; team project |

**Part IV: Course Outline** (Note: this course outline is preliminary and subject to change)

| Week | Lecture/Activity/Discussion |
|---|---|
| Week 1<br>Jan. 17 | Course logistics and introductions<br>Course overview – topics, objectives, latest developments in SAD/SE, industry trends and ongoing research<br><br>Project: project overview; teams start to form; discussion about topics initiated |
| Week 2<br>Jan.24 | Process models: fundamentals<br>Review of traditional process models and comparative analysis<br><br>Project start:<br>1) All groups-teams finalized,<br>2) Teams work together to select and align on a topic,<br>3) Project proposals posted in Canvas. |
| Week 3<br>Jan. 31 | Iterative-incremental process models<br>RUP and Scrum – introduction and comparative analysis<br>Agile software development<br>Scrum values, principles, and practices; Scrum teams<br><br>Project:<br>1) Discussion and approval of project topics<br>2) Market research and competitors<br><br>Presentations/Discussion rules explained |
| Week 4<br>Feb. 7 | Product backlogs (PBL) structure and management; agile estimation<br>Large scale Agile<br><br>Exercise: Agile estimation<br><br>Project meetings - progress review |
| Week 5<br>Feb. 14 | Requirements engineering concepts, methods, and standards;<br>Functional and Nonfunctional requirements (availability & reliability, performance, security and safety, usability, etc.)<br><br>Presentations/Discussion (topics – tbd)<br><br>Project meetings- progress review |
| Week 6<br>Feb. 21 | Use Cases – definitions, modeling, and analysis.<br><br>Presentations/Discussion (topics – tbd)<br><br>Project meetings - progress review |
| Week 7<br>Feb. 28 | Use Cases specification approaches<br>Use Case vs. User Stories – correlation and usage guidelines<br>Project PBLs - grooming and annotations.<br><br>Project progress updates – shared with the class |

| | |
|---|---|
| Week 8<br>March 6 | Project use case model and use case specification – team presentations.<br><br>**Exam/quiz (tbd)**<br><br>**Individual assignment - start** |
| Week 9<br>March 13 | Spring Recess<br>No class |
| Week 10<br>March 20 | Project management: planning, risk management, estimation<br><br>Presentations/Discussion (topics – tbd)<br><br>Project meetings - progress review |
| Week 11<br>March 27 | Model-driven engineering: introduction to Object Oriented Analysis & Design (OOAD) concepts<br>Exercises: Simple OOA models explained<br><br>**Individual assignment - due date**<br><br>Project check-list – clarifications, Q&A |
| Week 12<br>April 3 | OOAD – dynamic models<br>Exercises: OO Modeling<br><br>Project meetings - progress review |
| Week 13<br>April 10 | Object Oriented Design heuristics and patterns - invited talk<br>UML Overview<br><br>Presentations/Discussion (topics – tbd)<br><br>**Project documentation due date!**<br><br>Project presentation template |
| Week 14<br>April 17 | Architecture and design concepts, patterns, and specification; software product lines<br>Design quality – coupling and cohesion.<br><br>Selected topics: Trends in software technology, and delivery models: DevOps/CICD, Cloud, AI/ML, Gen AI, LLMs, Security, Quality and continuous improvement frameworks (e.g., Lean Six Sigma, ISO 9000, CMMI, PSP, TSP, TQM, etc.)<br><br>**Course summary and open discussion**<br><br>**Quiz/exam (tbd)** |
| Week 15<br>April 24 | Course project presentations, prototype demos and retrospective<br><br>**Prototype and presentation submission**<br><br>**Course closure** |

**Part V: Assignment Weighting (How Your Final Grade is Being Calculated?)**

| Assessment Item | Percentage of final grade |
|---|---|
| Team Project | 40% |
| Individual Assignment | 10% |
| Exam/Quizzes | 30% |
| Reading/presentations/discussion | 15% |
| Participation (posts on Canvas, attendance, class participation, initiative) | 5% |

**Grade policy (approximate): A (95%-100%), B+ (85%-94.9%), B (75%-84.9%)**

## Part VI: Delivery Mechanism

The following delivery mechanisms will be utilized:

☒ Canvas: IS 663 102
☒ NJIT WebEx
☒ In-class meetings – room: CKB 313.

## Part VII: Plagiarism and Academic Integrity

The approved "University Code on Academic Integrity" is currently in effect for all courses. Should a student fail a course due to a violation of academic integrity, they will be assigned the grade of "XF" rather than the "F" and this designation will remain permanently on their transcript.

All students are encouraged to look over the University Code on Academic Integrity and understand this document. Students are expected to uphold the integrity of this institution by reporting any violation of academic integrity. The identity of the student filing the report will be kept anonymous.

NJIT will continue to educate top tier students that are academically sound and are self-disciplined to uphold expected standards of professional integrity. **Academic dishonesty will not be tolerated at this institution.**

## Part VIII: Getting Help - General

The IST Helpdesk is the central hub for all information related to computing technologies at NJIT. This includes being the first point of contact for those with computing questions or problems.

How to contact the Helpdesk?

1. Call 973-596-2900. Monday - Friday 8 am - 7 pm.
2. Go to Student Mall Room 48. Monday - Friday 8 am - 7 pm
3. Log a Help Desk Service Request online - IST Service Desk (njit.edu)