	UČNI NAČRT PREDMETA/COURSE SYLLABUS
Predmet **Course title**	**Razvoj in vzdrževanje programskih sistemov** **Development and Maintenance of Software Systems**

Študijski program in stopnja **Study programme and level**	**Študijska smer** **Study field**	**Letnik** **Academic year**	**Semester** **Semester**
Upravljanje poslovnih in informacijskih sistemov / 2. stopnja	Upravljanje in razvoj informacijskih sistemov	1. letnik	1.
Business and Information Systems Management / 2nd Cycle	Management and Development of Information Systems	1st year	1st

Vrsta predmeta/Course type	obvezni/obligatory
Univerzitetna koda predmeta/University course code	2_URIS_1_UN4

Predavanja **Lectures**	**Seminar** **Seminar**	**Sem. vaje** **Tutorial**	**Lab. vaje** **Laboratory work**	**Teren. vaje** **Field work**	**Samost. delo** **Individ. work**	**ECTS**
20			10		210	8

Nosilec predmeta/Lecturer:	doc. dr. Branko Kaučič Učni načrt pripravil prof. dr. Viljan Mahnič
Jeziki/ Languages: **Predavanja/Lectures:**	slovenski/Slovenian
Vaje/Tutorial:	slovenski/Slovenian

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:	**Prerequisites:**
• Vpis v prvi letnik študijskega programa. • Študent mora pred izpitom pripraviti in predstaviti ter zagovarjati projektno/raziskovalno nalogo.	• The prerequisite for inclusion is enrolment in the first year of study. • Student has to prepare, present and defend a project/research paper before the exam.

Vsebina:	**Content (Syllabus outline):**
• *Uvod*: Osnove programskega inženiringa, modeli razvoja programske opreme, kratek pregled faz. • *Menedžment projektov*: Podpora vodstva, tehnike vodenja projektov, grafa PERT in Gantt, metoda kritične poti, napovedovanje časovnega okvira, primeri uporabe in prototipi.	• *Introduction*: Fundamentals of software engineering, software development models, brief overview of phases. • *Project management*: support of management, project management techniques, PERT and Gantt graphs, critical path method, time prediction, use cases and prototypes.

• *Kakovost programske opreme*: Vzdrževanost, uporabnost, dostopnost, prenosljivost, interoperabilnost, preverljivost. • *Arhitektura*: Pomembnost izgradnje arhitekture, uporabniki arhitekture, pristopa od zgoraj navzdol in od spodaj navzgor, pomembnejši arhitekturni vzorci. • *Dobre prakse razvoja programske opreme*: Načrtovanje ortogonalnih sistemov, minimizacija kompleksnosti, sledenje trdnim načrtovalskim principom, upravljanje programerskih ekip. • *Razumevanje poslovnega področja*: Spoznavanje domene, zbiranje zahtev. • *Načrtovanje na visoki ravni*: Načrtovanje varnosti, strojne opreme, uporabniškega vmesnika, izgradnja arhitekture, izdelava poročil, dostop do podatkovnih baz. • *Jezik UML:* Osnovni koncepti, strukturni diagrami, diagrami obnašanja, diagrami interakcije. • *Načrtovanje na nizki ravni*: Objektno orientirano načrtovanje, identifikacija objektov, izgradnja hierarhij, načrtovanje podatkovnih baz, normalizacija podatkov. • *Razvoj*: Uporaba pravih orodij za razvoj, izbira algoritmov, nasveti in triki pri programiranju. • *Testiranje*: Cilji testiranja, razlogi za programske napake, ravni testiranja, pomembnejše tehnike testiranja, dobre prakse, ocenjevanje števila napak. • *Namestitev*: Načrtovanje namestitve, strategija prenosa v produkcijo, aktivnosti in napake pri nameščanju. • *Vzdrževanje*: Stroški vzdrževanja, ključne aktivnosti (odprava napak, prilagajanje spremembam, preventivno delovanje). • *Metodologije razvoja programske opreme*: Scrum, Ekstremno programiranje, Crystal, Feature-Driven Development.	• *Software quality*: Maintenance, usability, availability, portability, interoperability, verifiability. • *Architecture*: The importance of building architecture, architecture users, top-down and bottom-up approaches, important architectural patterns. • *Good software development practices*: orthogonal systems design, minimizing complexity, following solid design principles, managing teams. • *Understanding the business field*: Understanding problem domain, requirements gathering. • *High-level design*: Designing security, hardware, user interface, building architecture, developing reports, accessing databases. • *UML language*: Basic concepts, structural diagrams, behaviour diagrams, interaction diagrams. • *Low-level design*: Object-oriented design, object identification, object hierarchy construction, database design, data normalization. • *Development*: Using the right tools for development, choosing algorithms, programming tips and tricks. • *Testing*: Testing goals, reasons for programming errors, test levels, important testing techniques, good testing practices, estimation of errors. • *Deployment*: Deployment planning, transfer into production strategies, deployment activities and errors. • *Maintenance*: Maintenance costs, key activities (eliminating errors, adapting to changes, preventive operations). • *Software development methodologies*: Scrum, Extreme Programming, Crystal, Feature-Driven Development

Temeljna literatura in viri/Readings:

Temeljna literatura/Basic literature

- Stephens, R. (2015). *Beginning Software Engineering*. Sybex.
- Ingeno, J. (2018). *Software Architect's Handbook: Become a successful software architect by implementing effective architecture concepts*. Packt Publishing Ltd.

Priporočljiva literatura/Recommended literature

- Unhelkar, B. (2017). *Software Engineering with UML*. Auerbach Publications.
- Tilley, S., Rosenblatt, H. J. (2016). *Systems Analysis and Design, 11th Edition.* Cengage Learning.

Cilji in kompetence:	**Objectives and competences:**
Učna enota prispeva predvsem k razvoju naslednjih splošnih in specifičnih kompetenc: • usposobljenost za poglobljeno razumevanje najsodobnejših področij računalništva in informatike, • usposobljenost za uporabo pridobljenih znanj za samostojno reševanje strokovnih in znanstvenih problemov računalništva in informatike, • usposobljenost za samostojno in timsko raziskovalno in razvojno delo, za uporabo znanstvenih pristopov pri delu in za obvladanje sodobnih razvojnih postopkov na področju računalništva in informatike, • usposobljenost sodelovanja, dela v skupini in dela na projektih, • razumevanje temeljnih in razvojno raziskovalnih znanj računalništva in informatike ter obvladovanje zahtevnejših veščin obeh področij, • usposobljenost za aplikacijo klasičnih ter razvijanje novih konceptov in znanj računalništva in informatike, • razumeti in razvijati zakonitosti delovanja informacijskih sistemov, • poznavanje in modeliranje sodobnih tehnik zbiranja, pretvorbe, prenašanja in shranjevanja podatkov in informacij, • obvladovanje zahtevnejših metod zasnove informacijskih sistemov, • obvladovanje zahtevnih metod načrtovanja varnosti informacijskih sistemov.	*The learning unit mainly contributes to the development of the following general and specific competences:* • being qualified for in-depth understanding of the most contemporary areas of computer science and informatics, • being qualified to use the acquired knowledge to independently solve professional and scientific problems in computer science and informatics, • being qualified for independent and team research and development work, for the use of scientific approaches at work, and for mastering contemporary development procedures in the field of computer science and informatics, • being qualified for cooperation, group work and work on projects, • understanding basic and developmental research knowledge of computer science and informatics, and mastering more demanding skills of both fields, • being qualified for the application of classic and the development of new concepts and knowledge of computer science and informatics, • understanding and developing the laws of the information systems operation, • knowing and modelling modern techniques for collecting, converting, transferring and storing data and information, • managing more complex methods of information systems designing,

	• mastering complex methods for planning the security of information systems.

Predvideni študijski rezultati:	Intended learning outcomes:
Študent/študentka: • se usposobi za vodenje projektov s področja programskih sistemov, • razvije sposobnost razumevanja poslovnih problemov na višji ravni, • pozna ključne parametre kakovosti programske opreme, • pozna dobre prakse načrtovanja in razvoja programskih sistemov, • se usposobi za načrtovanje programske opreme na visoki in nizki ravni, • pozna in uporablja orodja za razvoj in testiranje programske opreme, • pozna različne strategije namestitve programske opreme v produkcijo, • je usposobljen za načrtovanje vzdrževanja na daljši rok, • pozna in uporablja diagramske tehnike jezika UML za objektno analizo in načrtovanje.	*Students:* • develop skills in the management of software systems projects, • develop the ability to understand business problems at a higher level, • understand key software quality parameters, • are familiar with good practice in design and development of software systems, • are trained in software design at a high and low level, • know and use software development and testing tools, • know the various strategies for software deployment in production, • are qualified to plan maintenance for the long run, • know and use UML language diagramming techniques for object analysis and design.

Metode poučevanja in učenja:	Learning and teaching methods:
• *predavanja* z aktivno udeležbo študentov (razlaga, diskusija, vprašanja, primeri, reševanje problemov), • *laboratorijske vaje:* refleksija izkušenj, praktično reševanje več tipičnih problemov na računalniku, predstavitev in zagovor programskih rešitev, diskusija, sporočanje povratne informacije.	• *lectures* with active student participation (explanation, discussion, questions, examples, problem solving), • *laboratory work:* reflection of experience, practical solving of several typical problems on a computer, presentation and defence of programming solutions, discussion, feedback.

Načini ocenjevanja:	Delež (v %) Weight (in %)	Assessment:
Načini: • izpit • izdelava, predstavitev in zagovor projektne/raziskovalne naloge Ocenjevalna lestvica: ECTS.	60 % 40 %	Types: • exam • preparation, presentation and defence of the project/research paper Grading scheme: ECTS.