

	<b>UČNI NAČRT PREDMETA/COURSE SYLLABUS</b>
<b>Predmet</b>	<b>Paradigme razvoja programske opreme</b>
<b>Course title</b>	<b>Software Development Paradigms</b>

<b>Študijski program in stopnja</b> <b>Study programme and level</b>	<b>Študijska smer</b> <b>Study field</b>	<b>Letnik</b> <b>Academic year</b>	<b>Semester</b> <b>Semester</b>
Poslovna ekonomija in upravljanje	Upravljanje in razvoj informacijskih sistemov	2.	3.
Business Economics and Management	Management and Development of Information Systems	2 <sup>nd</sup>	3 <sup>rd</sup>

**Vrsta predmeta/Course type**

izbirni / elective

**Univerzitetna koda predmeta/University course code**

3\_PEU\_IP\_UN4\_URIS

<b>Predavanja</b> <b>Lectures</b>	<b>Seminar</b> <b>Seminar</b>	<b>Sem. vaje</b> <b>Tutorial</b>	<b>Lab. vaje</b> <b>Laboratory work</b>	<b>Teren. vaje</b> <b>Field work</b>	<b>Samost. delo</b> <b>Individ. work</b>	<b>ECTS</b>
15	10				425	15

**Nosilec predmeta/Lecturer:**

doc. dr. Sebastian Lahajnar

**Jeziki/ Languages:**  
**Predavanja/Lectures:**

slovenski/Slovenian

**Vaje/Tutorial:**

slovenski/Slovenian

**Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:**

**Prerequisites:**

- Pogoje za vključitev v delo je vpis v drugi letnik študijskega programa.
- Študent mora pred izpitom pripraviti in predstaviti raziskovalno nalogo.

- The prerequisite for participation is enrolment in the second year of study.
- Student has to prepare, present and defend a research paper before the examination.

**Vsebina:**

**Content (Syllabus outline):**

- *Uvod:* Zgodovina metodologij razvoja programske, ključni koncepti in principi, temeljni življenjski cikel razvoja programske opreme (sistemov).
- *Razvojni modeli:* Osnove razvojnih modelov, ad-hoc model, zaporedni, iterativni, inkrementalni, prototipni, spiralni, raziskovalni, model ponovne uporabljivosti. Primerjava modelov,

- *Introduction:* The history of software development methodologies, key concepts and principles, the basic software (system) development life cycle.
- *Development models:* Development models basics, ad hoc model, sequential, iterative, incremental, prototypical, spiral, research, reusability model. Comparison of

<p>prednosti in slabosti, uporabnost glede na specifikke projekta.</p> <ul style="list-style-type: none"> <li>• <i>Strukturne metodologije:</i> Sistemska analiza in načrtovanje, informacijski inženiring.</li> <li>• <i>Objektne metodologije:</i> RUP (Rational Unified Process), Iconix.</li> <li>• <i>Agilne metodologije:</i> Scrum, XP (ekstremno programiranje).</li> <li>• <i>Primerjava metodologij:</i> Prednosti in slabosti metodologij, smotrnost uporabe v različnih situacijah.</li> <li>• <i>Strukturne tehnike za analizo in načrtovanje programske opreme:</i> Model ER, dekompozicijski diagram, diagram podatkovnih tokov, psevdo koda, strukturni diagram, diagram poteka, odločitvene tabele in drevesa.</li> <li>• <i>Objektne tehnike:</i> Osnovni koncepti objektnega pristopa, splošno o jeziku UML, diagramске tehnike za opis statičnega (razredni, paketni, komponentni, postavitveni diagram) in dinamičnega (diagrami komunikacije, diagram prehajanja stanj, diagram aktivnosti) vidika programske opreme.</li> <li>• <i>Razvojne strategije:</i> Razvoj znotraj podjetja, zunanje izvajanje, obravnava stroškov in koristi, postopek nabave programske opreme.</li> <li>• <i>Kakovost programske opreme:</i> Vzdrževanost, uporabnost, dostopnost, prenosljivost, interoperabilnost, preverljivost.</li> <li>• <i>Arhitektura:</i> Pomembnost izgradnje arhitekture, uporabniki arhitekture, pristopa od zgoraj navzdol in od spodaj navzgor, pomembnejši arhitekturni vzorci, izgradnja poslovno informacijske arhitekture.</li> <li>• <i>Dobre prakse razvoja programske opreme:</i> Načrtovanje ortogonalnih sistemov, minimizacija kompleksnosti, sledenje trdnim načrtovalskim principom, upravljanje programerskih ekip.</li> </ul>	<p>models, advantages and disadvantages, usability according to project specifics.</p> <ul style="list-style-type: none"> <li>• <i>Structural methodologies:</i> System analysis and planning, Information engineering.</li> <li>• <i>Object methodologies:</i> RUP (Rational Unified Process), Iconix.</li> <li>• <i>Agile methodologies:</i> Scrum, XP (Extreme programming).</li> <li>• <i>Comparison of methodologies:</i> Advantages and disadvantages, the expediency of usage in different situations.</li> <li>• <i>Structural techniques for software analysis and design:</i> ER model, decomposition diagram, data flow diagram, pseudo code, structural diagram, flow chart, decision table and decision tree.</li> <li>• <i>Object techniques:</i> Basic concepts of object-based approach, about the UML language in general, graphic techniques for describing static (class, packet, component, node diagram) and dynamic (communication diagrams, state transition diagram, activity diagram) aspect of the information system.</li> <li>• <i>Development strategies:</i> Internal development of the company, outsourcing, cost-benefit analysis, software procurement process.</li> <li>• <i>Software quality:</i> Maintenance, usability, availability, portability, interoperability, verifiability.</li> <li>• <i>Architecture:</i> The importance of building architecture, architectural users, top-down and bottom-up approaches, important architectural patterns, construction of enterprise architecture.</li> <li>• <i>Good software development practices:</i> orthogonal system design, minimizing complexity, following sound design principles, management of programming teams.</li> </ul>
---	--

## Literatura in viri/Readings:

### Temeljna literatura/Basic literature

- Tilley, S., Rosenblatt, H. J. (2016). *Systems Analysis and Design, 11th Edition*. Cengage Learning.
- Ahmed T. et al (2014). *Developing Information Systems: Practical guidance for IT professionals*. BCS, The Chartered Institute for IT.

### Priporočljiva literatura/Recommended literature

- Ingeno, J. (2018). *Software Architect's Handbook: Become a successful software architect by implementing effective architecture concepts*. Packt Publishing Ltd.
- Unhelkar, B. (2017). *Software Engineering with UML*. Auerbach Publications.
- Stephens, R. (2015). *Beginning Software Engineering*. Sybex.

## Cilji in kompetence:

*Učna enota prispeva predvsem k razvoju naslednjih splošnih in specifičnih kompetenc:*

- usposobljenost za poglobljeno razumevanje najsodobnejših področij računalništva in informatike,
- usposobljenost za kreativno in samostojno znanstveno raziskovalno in razvojno delo, reševanje zahtevnih in kompleksnih problemov in vodenje raziskovalnih in razvojnih projektov,
- usposobljenost za samostojno in timsko raziskovalno in razvojno delo v razvojnih in interdisciplinarnih skupinah, za uporabo znanstvenih pristopov pri delu in za obvladanje sodobnih razvojnih postopkov na področju računalništva in informatike,
- temeljito poznavanje in razumevanje upravljanja in razvoja informacijskih sistemov,
- sposobnost za reševanje poslovnih problemov z uporabo znanstvenih metod in postopkov,
- temeljito poznavanje in sposobnost za uporabo najsodobnejših kvantitativnih in kvalitativnih raziskovalnih metod,
- sposobnost za samostojno načrtovanje in izvedbo raziskovalnega dela, analizo in interpretacijo podatkov, oblikovanje in utemeljitev mnenj, stališč in predlogov ter pripravo raziskovalnega poročila,
- sposobnost analize, načrtovanja in izvedbe kompleksnih projektov razvoja programske opreme.

## Objectives and competences:

*The learning unit mainly contributes to the development of the following general and specific competences:*

- the ability to gain a deeper understanding of the most modern areas of computer science and informatics,
- the ability to carry out creative and independent scientific research and development work, to solve demanding and complex problems and to manage research and development projects,
- the ability to carry out independent and team-oriented research and development work in development and interdisciplinary groups, to apply scientific approaches to work and to master modern development processes in the fields of computer science and informatics,
- thorough knowledge and understanding of the management and development of information systems,
- the ability to solve business problems with scientific methods and procedures,
- thorough knowledge and the ability to apply the most modern quantitative and qualitative research methods,
- the ability to plan and conduct research independently, to analyse and interpret data, to formulate and justify opinions, positions and proposals and to prepare a research report,
- the ability to analyse, plan and implement complex software development projects.

**Predvideni študijski rezultati:****Študent/študentka:**

- se usposobi za vodenje kompleksnih projektov s področja razvoja programske opreme,
- analizira in kritično oceni obstoječe stanje programske opreme in informacijske tehnologije v organizaciji,
- načrtuje novosti in izboljšave z uporabo modernih metodoloških pristopov,
- načrtuje arhitekturo programske opreme na ravni organizacije kot tudi posameznih podsistemov,
- zasnuje poslovno informacijsko arhitekturo organizacije,
- oblikuje strategijo organizacije na področju informacijske tehnologije,
- presodi primernost uporabe načrtovalskih tehnik glede na podan kontekst in domensko področje,
- prilagaja splošno uveljavljene razvojne metodologije glede na specifične organizacij in projektov.

**Intended learning outcomes:****Students:**

- are trained to manage complex software development projects,
- analyse and critically evaluate the current state of software and information technology in the organisation,
- plan innovations and improvements using modern methodological approaches,
- design the software architecture on the organisation level as well as individual subsystems,
- design the enterprise architecture of the organisation,
- formulate the organization's strategy in the field of information technology,
- assess the suitability of the design techniques usage in relation to the given context and domain area,
- adapt the generally accepted development methods to the specific characteristics of the organizations and projects specifics.

**Metode poučevanja in učenja:**

- predavanja z aktivno udeležbo študentov (razlaga, diskusija, vprašanja, primeri, reševanje problemov),
- projektni seminar,
- individualne in skupinske konsultacije (diskusija, dodatna razlaga, obravnava specifičnih vprašanj),
- oblikovanje portfolija in samostojen študij (motiviranje, usmerjanje, samoopazovanje, samouravnavanje, refleksija, samoocenjevanje).

**Learning and teaching methods:**

- lectures with active student participation (explanation, discussion, questions, examples, problem solving),
- project work seminar,
- individual and group consultations (discussion, further explanation, addressing specific issues),
- designing a portfolio and independent study (motivating, directing, self-observation, self-regulation, reflection, self-assessment).

**Načini ocenjevanja:**

Delež (v %)

Weight (in %)

**Assessment:**

<b>Načini:</b> <ul style="list-style-type: none"> <li>• temeljna ali aplikativna raziskovalna naloga z zagovorom (obseg 30.000 znakov).</li> </ul>	100 %	<b>Types:</b> <ul style="list-style-type: none"> <li>• fundamental or applied research paper with defence (30,000 characters).</li> </ul>
--	-------	---

Ocenjevalna lestvica: uspešno, neuspešno.		Grading scheme: successful, unsuccessful.
--	--	--