

RTU studiju kurss "Programmatūras attīstības tehnoloģijas"**33000 Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte****Vispārējā informācija**

Kods	DE0135
Nosaukums	Programmatūras attīstības tehnoloģijas
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles
Atbildīgais mācītājspēks	Oksana Ņikiforova - Doktors, Profesors
Apjoms daļās un kredītpunktos	1 daļa, 5.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	Pēc programmatūras krīzes apzināšanas fakta 1967.gadā programmatūras inženierija tika atzīta par inženierijas disciplīnu, kas ļāva apskatīt programmatūras attīstību kā inženierijas procesu. Kurss veltīts mūsdienīgajām programmatūras attīstības metodēm un tehnoloģijām, tās dzīves cikla dažādās fāzēs. Kopumā programmatūras izstrādi var sadalīt šādās komponentēs: dzīves cikls, process, analīzes un projektēšanas metodes un rīki, realizācijas vides, kvalitātes nodrošināšana, projekta vadība un tās labākās prakses, darbs komandā. Studiju kursā šīs komponentes ir apskatītas, parādot katram tipam tā evolūciju, mūsdienās lietojamākas metodes un prakses un atrisinātas un neatrisinātas problēmas. Kursā studenti ir iepazīstināti ar dažādām metodēm un praksēm, kas ir lietotas gan disciplinētā programmatūras izstrādē, gan spējā (angl. Agile) projektu organizācijā.
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	Mērķis ir apmācīt un virzīt studentus apgūt programmatūras izstrādes procesu modeļu attīstību un iepazīties ar dažādām tehnoloģijām, kas tiek lietotas programmatūras izstrādē un izstrādes procesa organizācijā. Uzdevumi: - Iemācīt noteikt programmatūras izstrādes procesa modeļa priekšrocības un trūkumus un izvēlēties projektam atbilstošo modeli. - Iemācīt definēt aktivitātes un artefaktus programmatūras izstrādes fāzēm. - palīdzēt studentiem iegūt praktisko pieredzi programmatūras izstrādes projekta realizācijā nelielā komandā nelielai sev ļoti pazīstamai problēmas videi. - Palīdzēt studentiem patrenēties uzstāties ar prezentāciju par aktuālo problēmu un tās risinājumiem programmatūras izstrādē, apskatot problēmu no dažādiem viedokļiem un pamatojot savus uzskatus uz problēmas risināšanu.
Patstāvīgais darbs, tā organizācija un uzdevumi	Studenti gatavojas praktiskajiem darbiem, atkārtojot lekcijās apgūto materiālu. Studenti sagatavo prezentācijas par izvēlētajiem zinātniskajiem rakstiem ar ko uzstājas semināros. Studenti gatavojas eksāmenam.
Literatūra	Obligātā/Obligatory Pressman R. S. Software Engineering A Practitioner's Approach, McGraw-Hill International Edition, 2019 Papildus/Additional Schach S. R. Object-Oriented & Classical Software Engineering, McGraw-Hill International Edition, 2011 Highsmith J. R. Agile Project Management: Creating Innovative Products, 2013 Campbell A. Agile: All You Need to Know about Agile Software Development. Team and Project Management using Scrum, 2020 Scientific papers on course topics for the last three years from scientific data bases IEEE, Scopus, WebOfScience, ScienceDirect, ACM, etc.
Nepieciešamās priekšzināšanas	Nav

Studiju kursa saturs

Saturs	Pilna un nepilna laika klātienes studijas		Nepilna laika neklātienes studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs
Ievads kursā un prasības studiju kursa apgūšanai.	2	0	0	0
Programmatūras izstrādes procesa organizācija un dzīves cikls.	8	6	0	0
Disciplinētas programmatūras izstrādes procesa modeļi.	8	8	0	0
Spējas programmatūras izstrādes procesa modeļi.	8	8	0	0
Ieskats programmatūras izstrādes kvalitātē un standartos.	4	4	0	0
Profesijas programmatūras izstrādē.	4	4	0	0
Komandas darba organizācija. Ieskats projekta pārvaldībā.	6	6	0	0
Semināri par zinātniskajiem rakstiem programmatūras attīstības tehnoloģiju jomā.	16	16	0	0
Eksāmens un pirmseksāmena konsultācija	4	8	0	0
Kopā:	60	60	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Spēj noteikt programmatūras izstrādes procesa modeļa priekšrocības un trūkumus un izvēlēties projektam piemērotāko modeli, izvēloties starp "disciplinētām" un spējām (Agile) metodoloģijām.	Praktiskā darba izpildes novērtēšana
Spēj definēt aktivitātes un artefaktus programmatūras izstrādes fāzēm un apzināties to priekšrocības un trūkumus.	Praktiskā darba izpildes novērtēšana
Spēj praksē pielietot tādas programmatūras izstrādes prakses kā uzdevumu sadalīšana, projekta sprintu plānošana, uzdevumu prioritizēšana, nepieciešama laika noteikšana uzdevuma izpildei	Piedalīšanas programmatūras izstrādes projekta spēlē un tās rezultāta novērtēšana. Novērtē gan studenti (pašnovērtējums un citu studentu novērtējums), gan mācītbspēks.
Spēj komandā uzprojektēt UI skices un definēt UX scenārijus un vispārējo programmatūras struktūru nelielai sev labi pazīstamai problēmas videi.	Piedalīšanas programmatūras izstrādes projekta spēlē un tās rezultāta novērtēšana. Novērtē gan studenti (pašnovērtējums un citu studentu novērtējums), gan mācītbspēks.
Spēj sagatavot un uzstāties ar prezentāciju par aktuālo problēmu un tās risinājumiem programmaatūras izstrādē, apskatot problēmu no dažādiem viedokļiem un pamatojot savus uzskatus uz problēmas risināšanu.	Uzstāšanas semināros, ko novērtē gan pārējie studenti, gan pasniedzējs. Citu studentu prezentāciju novērtēšanas un komentēšanas novērtēšana no mācītbspēka puses.
Spēj demonstrēt iegūtas zināšanas priekšmeta jomā, atbildot uz teorētiskajiem jautājumiem un risinot praktiskus uzdevumus.	Eksāmens

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Praktiskie darbi, kas ir izpildīti semestra laikā	20
Studiju projekta izstrāde komandā	20
Dalība savas komandas un citu komandu izstrādāto studiju projektu novērtēšanā	10
Prezentāciju izstrāde un uzstāšanas ar to semināros	20
Dalība citu studentu prezentāciju komentēšanā, jautājumu uzdošanā un novērtēšanā	10
Eksāmens	20
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbaudījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs
1.	5.0	40.0	20.0	0.0		*	