

RTU studiju kurss "Nepārtraukta programmatūras serveru testu automatizācija"

33000 Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte

Vispārējā informācija

Kods	DE0282
Nosaukums	Nepārtraukta programmatūras serveru testu automatizācija
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles; Brīvās izvēles
Atbildīgais mācītbspēks	Ērika Nazaruka - Doktors, Asociētais profesors
Apjoms daļās un kredītpunktos	1 daļa, 6.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	Studiju kursa ietvaros tiek sniegtas padziļinātas zināšanas nepārtrauktas piegādes programmatūras serveru (servera puses) testu automatizācijas risinājumos. Tiek apskatīti dažādi servera puses testu automatizācijas rīki, sākot no pavisam vienkāršiem līdz pilnīgi pielāgotam testu automatizācijas ietvaram funkcionālo un nefunkcionālo testu veikšanai. Studentiem ir iespēja veikt servera puses testu automatizācijas ietvara izstrādi, iepazīt virtualizācijas vides un veikt ietvara iekļaušanu nepārtrauktās programmatūras piegādes plūsmā. Studiju kursa ietvaros studenti izmanto labajās praksēs un tendencēs balstītus risinājumus.
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	Studiju kursa mērķis ir sniegt teorētiskas un praktiskas zināšanas nepārtrauktas programmatūras piegādes serveru testu automatizācijā, īstenojot un pārraugot dažādas metodes, rīkus un risinājumus. Studiju kursa uzdevumi: 1. Pilnveidot studentu izpratni par servera puses testu automatizācijas nepieciešamību. 2. Attīstīt prasmes veikt automatizēto testpiemēru izstrādi un integrēt automatizācijas risinājumus virtualizācijas vidēs un nepārtrauktās programmatūras piegādes plūsmā. 3. Pilnveidot studentu prasmes zinātnē un labajās praksēs balstītu risinājumu jēgpilnā lietošanā. 4. Attīstīt spējas argumentēti diskutēt par programmatūras serveru testu nepieciešamību, principiem, tehnoloģiju un rīku tendencēm nepatraktas piegādes procesā.
Patstāvīgais darbs, tā organizācija un uzdevumi	Patstāvīgais darbs ietver sagatavošanos laboratorijas darbu izpildei un eksāmenam. Studenti sagatavosies, lai laboratorijas darbos praktiski izmantotu iegūtās zināšanas serveru funkcionālo un nefunkcionālo testu automatizācijā, testu iekļaušanā nepārtrauktajā programmatūras piegādes plūsmā, kā arī virtualizētas vides izveidošanā un pārvaldīšanā, izmantojot atbilstošus rīkus un tehnoloģijas.
Literatūra	Obligātā/Obligatory: A. K. Jena, H. Das, and D. P. Mohapatra, Eds., Automated Software Testing: Foundations, Applications and Challenges. Springer Singapore, 2020. doi: 10.1007/978-981-15-2455-4. M. A. J. Richardson, Automating and Testing a REST API: A Case Study in API testing using: Java, REST Assured, Postman, Tracks, cURL and HTTP Proxies, 1st edition. Compendium Developments Ltd, 2017. J. Arundel and J. Domingus, Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud. Beijing; Boston: O'Reilly, 2019. J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, 1st edition. Upper Saddle River, NJ: Addison-Wesley Professional, 2010. J. Nickoloff and S. Kuenzli, Docker in Action, 2nd edition. Shelter Island, NY: Manning Publications, 2019. I. Molyneaux, The Art of Application Performance Testing: From Strategy to Tools, 2nd edition. Beijing: O'Reilly Media, 2015. Papildu/Additional: N. Forsgren, J. Humble, and G. Kim, Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations, 1st edition. Portland, Oregon: IT Revolution Press, 2018. R. Lawrence and P. Rayner, Behavior-Driven Development with Cucumber. Addison Wesley Professional, 2019. M. Fowler, "Continuous Integration," martinowler.com, Jan. 05, 2010. https://martinowler.com/articles/originalContinuousIntegration.html (accessed May 13, 2021). I. Miell and A. H. Sayers, Docker in practice. Shelter Island, New York: Manning Publications, 2016. K. Beck, Test Driven Development: By Example, 1st edition. Boston: Addison-Wesley Professional, 2002. J. Turnbull, The Docker Book: Containerization is the new virtualization, 18092nd edition. James Turnbull, 2014.
Nepieciešamās priekšzināšanas	Programmatūras izstrādes pamati, objektorientētās programmatūras izstrādes pamati.

Studiju kursa saturs

Saturs	Pilna un nepilna laika klātienes studijas		Nepilna laika neklātienes studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs
Ievads (testu automatizācijas nepieciešamība, testu automatizācijas salīdzinājums ar manuālo testēšanu).	2	0	0	0

Testu virzītas izstrādes (TDD) princips.	2	0	0	0
Serveri un to saskarnes, lietotņu komunikācija ar serveriem (teorija, HTTP pieprasījumi, RESTful API, SOAP).	4	0	0	0
REST API testēšanas rīki, to izvēle un uzstādīšana.	6	6	0	0
Vienkāršu REST API testpiemēru manuāla izpilde un vienkāršu testu automatizācija (Postman, Newman).	4	6	0	0
Pielāgota testu automatizācijas ietvara realizācija funkcionālo testpiemēru automatizācijai.	8	20	0	0
Nefunkcionālo testu automatizācijas nepieciešamība, plānošana.	2	0	0	0
Slodzes testu izveide (JMeter).	4	5	0	0
Veiktspējas testu izveide (JMeter).	2	5	0	0
Nepārtrauktas integrācijas un piegādes loma programmatūras izstrādes ciklā.	2	0	0	0
Nepārtrauktās integrācijas un piegādes (CI/CD) rīki, to izvēle un uzstādīšana (Git, Jenkins, GitLab CI).	4	4	0	0
Programmatūras piegādes konveijera izveide dažādos rīkos (rīku funkcijas, izmantojamība dažādās reālās situācijās).	8	10	0	0
Izstrādāto testu ieviešana nepārtrauktās integrēšanas procesā.	4	6	0	0
Virtualizētas vides, to izmantojamība testu automatizācijā un nepārtrauktas integrācijas un piegādes plūsmā.	8	12	0	0
Testu automatizācijas rīku uzstādīšana, startēšana no virtualizētas vides (Docker).	4	6	0	0
Konsultācijas.	14	0	0	0
Eksāmens.	2	0	0	0
Kopā:	80	80	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Spēj pamatoti izvēlēties serveru testu automatizācijai nepieciešamos rīkus, balstoties uz servera specifiku, prot veikt to uzstādīšanu un pārvaldību.	Sekmīgi aizstāvēti laboratorijas darbi un eksāmens. Kritēriji: spēj izskaidrot prasības serveru testēšanai; spēj identificēt rīkus serveru testēšanai; prot uzstādīt un pārvaldīt rīkus serveru testēšanai.
Prot veikt funkcionālo testu automatizāciju kvalitātes nodrošināšanas nolūkos ar apgūtajiem testu automatizācijas rīkiem.	Sekmīgi aizstāvēti laboratorijas darbi. Kritēriji: prot projektēt funkcionālos testus; prot realizēt uzprojektētos funkcionālos testus ar rīka palīdzību.
Prot veikt nefunkcionālo testu automatizāciju kvalitātes nodrošināšanas nolūkos ar apgūtajiem testu automatizācijas rīkiem.	Sekmīgi aizstāvēti laboratorijas darbi. Kritēriji: prot projektēt nefunkcionālus testus; prot realizēt uzprojektētos nefunkcionālos testus ar rīka palīdzību.
Spēj izmantot apgūtās teorētiskās un praktiskās zināšanas servera testu automatizācijas ietvara izveidē un tā iekļaušanu nepārtrauktajā programmatūras piegādes plūsmā.	Sekmīgi aizstāvēti laboratorijas darbi, eksāmens. Kritēriji: prot izveidot servera testu automatizācijas ietvaru; prot integrēt izveidoto ietvaru nepārtrauktajā programmatūras piegādes plūsmā; spēj izskaidrot servera testu automatizācijas ietvara izmantošanas principus.
Prot izveidot un pārvaldīt operētājsistēmas līmeņa virtualizēto vidi izmantojot Docker, pamatot uzstādīto rīku izvēli.	Sekmīgi aizstāvēti laboratorijas darbi. Kritēriji: spēj pamatoti izvēlēties rīkus operētājsistēmas līmeņa virtualizētai videi; spēj izskaidrot Docker izmantošanas principus; prot izveidot un pārvaldīt izveidotu virtualizētu vidi.

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Laboratorijas darbi	80
Eksāmens	20
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbaudījumi			Brīvās izvēles pārbaudījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs	Ieskaite	Eksām.	Darbs
1.	6.0	20.0	0.0	60.0		*		*		