

## INFORMAČNÝ LIST PREDMETU

<b>Vysoká škola:</b> Univerzita Konštantína Filozofa v Nitre	
<b>Fakulta:</b> Fakulta prírodných vied a informatiky	
<b>Kód predmetu:</b> KI/SI/22	<b>Názov predmetu:</b> Softvérové inžinierstvo
<b>Druh, rozsah a metóda vzdelávacích činností:</b> <b>Forma výučby:</b> Prednáška / Cvičenie <b>Odporúčaný rozsah výučby ( v hodinách ):</b> <b>Týždenný:</b> 2 / 2 <b>Za obdobie štúdia:</b> 26 / 26 <b>Metóda štúdia:</b> prezenčná	
<b>Počet kreditov:</b> 6	
<b>Odporúčaný semester/trimester štúdia:</b> 1., 3.	
<b>Stupeň štúdia:</b> II.	
<b>Podmieňujúce predmety:</b>	
<b>Podmienky na absolvovanie predmetu:</b> Celková záťaž študenta: 150 hodín. Denné štúdium: prednášky: 26 hodín, cvičenia: 26 hodín, domáca príprava: 98 hodín. Externé štúdium: kontaktná výučba: 20 hodín, dištančná výučba prostredníctvom LMS a konzultácie: 32 hodín, samostatná príprava: 98 hodín. Podmienky absolvovania: Aktívna účasť na cvičeniach. Práca na vývoji softvérového produktu v tíme, dodržiavanie princípov zvolenej metodiky vývoja, aktívna účasť na jednotlivých šprintoch a stretnutiach tímu, systematické používanie nástrojov vývojára, aktívne zapojenie do tvorby zdrojového kódu, testovania, tvorby špecifikácie softvérového produktu a dokumentácie (60%). Skúška pozostáva z teoretického testu (20%) a spoločnej tímovej prezentácie a obhajoby vytvoreného softvérového produktu (20%). Kredity nebudú udelené študentovi, ktorý získa v celkovom bodovom hodnotení menej ako 70 percent. Hodnotenie: A = 100 % - 95%, B = 94 % - 90 %, C = 89 % - 85 %, D = 84 % -80 %, E = 79 % - 70 %, FX = 69 % - 0 % .	
<b>Výsledky vzdelávania:</b> Znalosti: - architektonické rámce IKT - informačná architektúra - metodiky zabezpečenia kvality - modelovanie podnikových postupov - modely softvérovej architektúry - softvérové metriky - úrovne testovania softvéru - životný cyklus vývoja systémov - zjednotený modelovací jazyk - metodiky riadenia IKT projektov - objektovo orientované modelovanie - softvér integrovaného vývojového prostredia - vývojový softvér Zručnosti:	

- analyzovať softvérové špecifikácie
- definovať softvérovú architektúru
- identifikovať požiadavky zákazníkov
- navrhnuť informačný systém
- navrhnuť podnikovú architektúru
- odstraňovať chyby v softvéri
- plánovať testovanie softvéru
- použiť nástroje softvérového inžinierstva podporované počítačom
- použiť softvérové knižnice
- vykonať analýzu rizík
- vykonať softvérové testy
- vykonávať projektový manažment
- vypracovať štúdiu uskutočniteľnosti
- vytvoriť návrh softvéru
- získať spätnú väzbu zákazníkov o aplikáciách
- dozerať na vývoj softvéru
- poskytnúť používateľskú dokumentáciu
- vytvoriť špecifikácie projektu
- navrhnuť proces
- vyvinúť softvérový prototyp
- rozvoj komunikačných zručností
- rozvoj zručností pre tímovú prácu a lídersstvo

Výsledky vzdelávania:

- Študent vie charakterizovať jednotlivé softvérové procesy. Študent rozumie pokročilým témam softvérového inžinierstva. Študent pozná výhody a nevýhody rôznych modelov životného cyklu vývoja softvérového produktu. Študent ovláda viacero metód a techník špecifikácie požiadaviek, modelovania a vývoja softvérového produktu. Študent rozumie a správne aplikuje diagramy UML v dokumentácii a v tímovej komunikácii. Študent pozná a aplikuje rôzne architektonické vzory. Študent rozumie výhodám opakovaného použitia softvéru a dôležitosti výberu vhodnej softvérovej architektúry. Študent používa pokročilé programátorské techniky, správne programátorské techniky a viaceré programovacie jazyky. Študent ovláda princípy práce v tíme v súlade so zvolenou metodikou. Študent rozvíja technologické i komunikačné schopnosti pre tímovú prácu. Študent preberá zodpovednosť za vývoj softvérového produktu v tíme. Študent vhodne kombinuje viaceré nástroje vývojára. Študent navrhuje rôzne prístupy k testovaniu a validácii softvérového produktu. Študent rozumie dôležitosti vzájomného prepojenia vývoja a prevádzky softvérového produktu a úlohe integrácie nástrojov, automatizácie procesov, ako aj komunikácie a spolupráce v tíme.

#### **Stručná osnova predmetu:**

1. Softvérové procesy, kvalita a vlastnosti softvérového produktu
2. Modely životného cyklu vývoja softvérového produktu
  - domáca príprava: Inštalácia a konfigurácia nástrojov pre prácu a manažment vývojového tímu, výber témy projektu. (5 hod.)
3. Metodiky vývoja softvérového projektu
  - domáca príprava: Špecifikácia požiadaviek v podobe používateľských príbehov, tvorba backlogu, výber architektúry. (5 hod.)
4. Inžinierstvo požiadaviek
  - domáca príprava: Šprint 1 - práca na príbehoch, návrh dátovej vrstvy a GUI (5 hod.)
5. Modelovanie softvérového produktu
  - domáca príprava: Šprint 1 - prezentácia aktuálneho stavu, nastavenia nástrojov vývojára, dokumentácie projektu (5 hod.)

6. Architektonický návrh softvérového produktu - domáca príprava: Šprint 2 - práca na príbehoch (5 hod.)
7. Návrh a implementácia softvérového produktu - domáca príprava: Šprint 2 - prezentácia aktuálneho stavu a funkčnej verzie (5 hod.)
8. Validácia a verifikácia softvérového produktu - domáca príprava: Šprint 3 - práca na príbehoch (5 hod.)
9. Nasadenie, údržba a evolúcia softvéru - domáca príprava: Šprint 3 - prezentácia aktuálneho stavu a funkčnej verzie, dokumentácie projektu (5 hod.)
10. DevOps - Manažment životného cyklu softvérového produktu - domáca príprava: Šprint 4 - práca na príbehoch (5 hod.)
11. Opakované použitie softvérového produktu - domáca príprava: Šprint 4 - prezentácia aktuálneho stavu a funkčnej verzie, dokumentácie projektu (5 hod.)
12. Iné softvérové architektúry - domáca príprava: Šprint 5 - práca na príbehoch, finalizácia projektu (5 hod.)
13. Spoľahlivosť a bezpečnosť softvérového produktu - domáca príprava: Šprint 5 - prezentácia a obhajoba projektu (5 hod.)
14. Príprava a realizácia skúšky - domáca príprava: 38 hod.

#### **Odporúčaná literatúra:**

E-learningový kurz Princípy softvérového inžinierstva (<https://edu.ukf.sk>)  
 Arlow, J., Neustadt, I.: UML a unifikovaný proces vývoje aplikáci. Computer press. Brno. 2008.  
 Bourque, P., Fairley, R.E. eds., Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014; [www.swebok.org](http://www.swebok.org).  
 Dathan, B., Ramnath, S. Object-Oriented Analysis, Design and Implementation. An Integrated Approach. Undergraduate Topics in Computer Science. 2nd ed. Universities Press. Springer. 2015.  
 Drlík, M. Manažment IT projektov. UKF: Edícia Prírodovedec č. 770. Nitra. 2021.  
 Ingeno, J. Software Architecture Handbook. Become a successful software architect by implementing effective architecture concepts. Packt Publishing. 2018.  
 Raj, P., Raman, A., Subramanian, H. Architectural Patterns. Uncover essential patterns in the most indispensable realm of enterprise architecture. Packt Publishing. 2017.  
 Seidl, M., Scholz, M., Huemer, Ch., Kappel, G. UML @ Classroom. An Introduction to Object-Oriented Modeling. Undergraduate Topics in Computer Science. Springer. 2015.  
 Sommerville, I. : Software Engineering. 9th Edition. Addison-Wesley. 2011.  
 Šimko, J., Šimko, M., Bieliková, M. Softvérové inžinierstvo v otázkach a odpovediach. STU. Bratislava. 2017.

#### **Jazyk, ktorého znalosť je potrebná na absolvovanie predmetu:**

slovenský

#### **Poznámky:**

#### **Hodnotenie predmetov**

Celkový počet hodnotených študentov: 250

A	B	C	D	E	FX
13.2	22.8	27.2	14.4	16.8	5.6

<b>Vyučujúci:</b> doc. Mgr. Martin Drlík, PhD., Mgr. Dominik Halvoník, PhD.,
<b>Dátum poslednej zmeny:</b> 06.11.2022
<b>Schválil :</b> prof. RNDr. Michal Munk, PhD. prof. RNDr. Michal Munk, PhD. Dátum schválenia: 26.11.2021