

Popis předmětu

Zkratka předmětu:	KIV/SAR-E	Strana:	1 / 4
Název předmětu:	Architektury softwarových systémů		
Akademický rok:	2023/2024	Tisknuto:	01.06.2024 10:27

Pracoviště / Zkratka	KIV / SAR-E			Akademický rok	2023/2024
Název	Architektury softwarových systémů			Způsob zakončení	Zkouška
Akreditováno/Kredity	Ano, 6 Kred.			Forma zakončení	Kombinovaná
Rozsah hodin	Přednáška 3 [HOD/TYD] Cvičení 2 [HOD/TYD]			Zápočet před zkouškou	ANO
Obs/max	Statut A	Statut B	Statut C	Počítán do průměru	ANO
Letní semestr	0 / -	0 / -	0 / -	Min. (B+C) studentů	20
Zimní semestr	23 / -	0 / -	2 / -	Opakovaný zápis	NE
Rozvrh	Ano			Vyučovaný semestr	Zimní semestr
Vyučovací jazyk	Angličtina			Počet dnů praxe	0
Volně zapisovatelný předmět	Ano			Hodn. stup. zp. před zk.	S/N
Hodnotící stupnice	1 2 3 4				
Počet hodin kontaktní	Ne				
Automat. uzn. záp. před zk.	Ne				
Periodicita	K				
Nahrazovaný předmět	Žádný				
Vyloučené předměty	KIV/SAR				
Podmiňující předměty	Nejsou definovány				
Předměty informativně doporučené	Nejsou definovány				
Předměty, které předmět podmiňuje	KIV/SA, KIV/SASZ, KIV/SSSZ				

Cíle předmětu (anotace):

Cílem předmětu je naučit studenty vybrat a efektivně použít vhodnou architekturu pro projektovaný softwarový systém stejně tak jako analyzovat a porozumět architekturám systémů stávajících.

Požadavky na studenta

Týmový projekt - 60% hodnocení
Aktuální téma z oblasti softwarových architektur - 10% hodnocení
Kombinovaná zkouška - 30% hodnocení

Z důvodu průběžné aktualizace předmětu je pro získání zápočtu při opakovaném zapsání předmětu (viz SZŘ čl. 24 odst. 3) nutné souhlasné vyjádření garanta předmětu.

Upozornění:

Termíny a forma ověřování splnění požadavků mohou být upraveny s ohledem na opatření vyhlášená v souvislosti s vývojem epidemiologické situace v ČR.

Obsah

- Obsah pojmu architektura software, význam architektury, vztah architektury a kvalitativních charakteristik software.
- Výchozí principy: Modularita software, skrývání informace, open/closed, kontrakt, význam rozhraní modulu, koncept softwarových komponent. Realizace v monolitických, modulárních, komponentových jazycích/prostředcích. Vícenásobná použitelnost modulů (reuse), návrh s ohledem na ni.
- Závislosti mezi moduly, principy vyhledání závislostí (dependency lookup) a vkládání závislostí (dependency injection). Příklady realizace v konkrétních technologiích.
- Vazby a komunikace mezi moduly a jejich abstrakce, koncept konektoru. Typy konektorů pro komunikaci a pro integraci modulů; příklady realizace.
- Architektonické styly (n-tier, stream/pipe, blackboard, monolit, component, service-oriented, cloud), příklady aplikací.
- Podrobnosti o vícevrstvých a servisně-orientovaných architekturách. Produktové řady, variabilita software na úrovni

architektury.

7. Způsoby dokumentace architektury -- UML, ADL (Architecture Description Languages), různé pohledy na architekturu, ad-hoc diagramy. Dokumentace API. Standardy v oblasti softwarových architektur.

8. Vztah architektury a disciplín v rámci vývoje software. Získání a analýza požadavků na architekturu, role architektury při návrhu a ověřování kvality software. Význam pro integraci, provoz a údržbu systémů. Přístup k tvorbě architektury v různých modelech vývoje software (sekvenční, řízený riziky, agilní). Role architekta v softwarovém týmu.

9. Postupy tvorby software a nástroje pro řízení vývoje s vazbou na architekturu (konfigurační řízení, správa sestavení). Ověření kvality architektury, výkonové apod. testování, přístup "executable architecture".

10. Analýza architektury software, metriky a kvalitativní charakteristiky architektur (výkonnost, spolehlivost, udržitelnost).

Nástroje pro analýzu, měření a vizualizace architektur software. Výběr architektury vzhledem k účelu software, design tradeoffs.

11. Vývoj software řízený návrhem (MDA, Model Driven Architectures), význam a použití modelů, PIM a PSM úrovně návrhu.

12. Prostor pro zvanou přednášku -- příklady z praxe / technologie pro efektivní návrh software / aktuální výzkum v oboru.

Studijní opory

Garanti a vyučující

- **Garanti:** Doc. Ing. Roman Mouček, Ph.D. (100%)
- **Přednášející:** Doc. Ing. Roman Mouček, Ph.D. (100%)
- **Cvičící:** Doc. Ing. Roman Mouček, Ph.D. (100%)

Literatura

- **Základní:** Taylor; et al. *Software Architecture: Foundations, Theory, and Practice*. Wiley, 2009. ISBN 978-0470167748.
- **Doporučená:** Richards, M., Ford, N. *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly, 2020. ISBN 978-1492043454.
- **Doporučená:** Research and technology sources - journals IEEE TSE, IEEE Software; conferences ICSA, ICSE, etc.; portals InfoQ, DZone.
- **Doporučená:** Bass, Len; et al. *Software Architecture in Practice*. 3rd Edition. Addison-Wesley, 2012. ISBN 978-0321815736.

Časová náročnost

Všechny formy studia

Aktivita	Časová náročnost aktivity [h]
Příprava na zkoušku [10-60]	20
Projekt individuální [40]	20
Projekt týmový [20-60 / počet studentů]	75
Kontaktní výuka	40
Příprava prezentace (referátu) [3-8]	5
Celkem:	160

Hodnotící metody

Odborné znalosti - odborné znalosti dosažené studiem předmětu jsou ověřovány hodnotícími metodami:

- Kombinovaná zkouška,
- Test,
- Skupinová prezentace,

Odborné dovednosti - odborné dovednosti dosažené studiem předmětu jsou ověřovány hodnotícími metodami:

- Demonstrace dovedností (praktická činnost),

Individuální prezentace,
Výstupní projekt,
Skupinová prezentace,

Obecné způsobilosti - obecné způsobilosti dosažené studiem předmětu jsou ověřovány hodnoticími metodami:

Skupinová prezentace,
Demonstrace dovedností (praktická činnost),

Předpoklady

Odborné znalosti - pro úspěšné zvládnutí předmětu se předpokládá, že je student před zahájením výuky schopen:

rozumět principům objektově orientovaného programování
orientovat se v principech návrhu a implementace webových aplikací
rozumět principům architektury softwarových aplikací
orientovat se v základech UML
popsat základní postupy vývoje software

Odborné dovednosti - pro úspěšné zvládnutí předmětu se předpokládá, že student před zahájením výuky dokáže:

vytvářet počítačové programy v objektově orientovaném jazyce (např. C++ / Java / .NET)
vytvářet webové aplikace s využitím webových technologií (např. Spring MVC, .NET WPF)
vytvářet moderní aplikace s využitím základních návrhových vzorů (např. MVC, MVVM)
posuzovat a používat vhodný postup při vývoji software

Obecné způsobilosti - před zahájením studia předmětu je student schopen:

mgr. studium: používají své odborné znalosti, odborné dovednosti a obecné způsobilosti alespoň v jednom cizím jazyce,
mgr. studium: dle rámcového zadání a přidělených zdrojů koordinují činnost týmu, nesou odpovědnost za jeho výsledky,

Vyučovací metody

Odborné znalosti - pro dosažení odborných znalostí jsou užívány vyučovací metody:

Projektová výuka,
Přednáška s aktivizací studentů,
Přednáška s demonstrací,
Přednáška založená na výkladu,
Samostudium,

Odborné dovednosti - pro dosažení odborných dovedností jsou užívány vyučovací metody:

Cvičení (praktické činnosti),
Demonstrace dovedností,
Samostatná práce studentů,
Řešení problémů,

Obecné způsobilosti - pro dosažení obecných způsobilostí jsou užívány vyučovací metody:

Přednáška s aktivizací studentů,

Výsledky učení

Odborné znalosti - po absolvování předmětu prokazuje student znalosti:

popsat různé druhy architektur softwarových systémů a orientovat se v nich
rozumět principům modularity a skrývání informace
popsat a interpretovat architektonické návrhy software grafickými notacemi, zejména s použitím UML
vysvětlit kritéria vhodnosti a kvality návrhu rozsáhlého softwarového systému
vysvětlit jak se lze orientovat se i v rozsáhlém programu a analyzovat jeho vlastnosti

Odborné dovednosti - po absolvování předmětu prokazuje student dovednosti:

porozumět architektuře rozsáhlých softwarových systémů

navrhnout architekturu rozsáhlého softwarového systému s využitím modelů a grafických notací
 posoudit kvalitu a vhodnost architektonického návrhu rozsáhlého softwarového systému
 použít vhodné technologie při vytváření rozsáhlých softwarových celků
 aktivně používat moderní technologie a nástroje pro vývoj rozsáhlých programových systémů

Obecné způsobilosti - po absolvování předmětu je student schopen:

mgr. studium: dle vyvíjejících se souvislostí a dostupných zdrojů vymezí zadání pro odborné činnosti, koordinují je a nesou konečnou odpovědnost za jejich výsledky,
 dokáží zodpovědně pracovat jako členové, případně vedoucí, menšího odborného týmu, a srozumitelně shrnout názory ostatních členů týmu

Předmět je zařazen do studijních programů:

Studijní program	Typ stud.	Forma stud.	Obor	Etap	V.st.pl.	Rok	Blok	Statut	D.roč.	D.sem.
Distribuované výpočetní systémy	Navazující	Prezenční	Distribuované systémy	1	2022 akr	2023	Povinné předměty	A	2	ZS
Distribuované výpočetní systémy	Navazující	Prezenční	Výpočetní technika	1	2022 akr	2023	Povinné předměty	A	2	ZS
Softwarové a informační systémy	Navazující	Prezenční	Softwarové a informační systémy	1	2022 akr	2023	Povinné předměty	A	2	ZS