

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
ФАКУЛТЕТ „ЕЛЕКТРОТЕХНИКА И ЕЛЕКТРОНИКА”

Приета с решение на ФС
Протокол № 6/27.07.2020 г.
/ проф. д-р инж. З.Ненова /

Утвърдил
Декан:

У Ч Е Б Н А П Р О Г Р А М А

По дисциплината: **ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ**
включена в учебния план за специалност:

СОФТУЕРНО И КОМПЮТЪРНО ИНЖЕНЕРСТВО

Образователно-квалификационна степен: **БАКАЛАВЪР**

Област на висше образование: **ТЕХНИЧЕСКИ НАУКИ**

Професионално направление:

КОМУНИКАЦИОННА И КОМПЮТЪРНА ТЕХНИКА /шифър 5.3/

Професионална квалификация:

„КОМПЮТЪРЕН ИНЖЕНЕР”

Форма на обучение: **РЕДОВНА И ЗАДОЧНА**

Катедра: **„КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ”**

I. ИЗВАДКА ОТ УЧЕБНИЯ ПЛАН

Учебна дисциплина	Семестър	Хорариум Л+СУ+ЛУ	Курсова работа (проект)	Форма на контрол
Големи софтуерни проекти РО	VIII	30+0+20	-	И
Големи софтуерни проекти ЗО	VIII	15+0+10	-	И

РО – редовна форма на обучение

ЗО – задочна форма на обучение

II. АНОТАЦИЯ

Дисциплината „Големи Софтуерни Проекти” (ГСП) има за цел да даде основни знания на студентите от специалност „Софтуерно и Компютърно Инженерство”, свързани с проектиране, разработка и тестване на големи софтуерни проекти. Студентите изучават:

1. Жизнен цикъл на големите програмни системи. Основни етапи от създаване на програмни проекти. Финансови, програмни и социални изисквания и ограничения при реализация на големи софтуерни проекти.
2. Управление на големи софтуерни проекти. Основни функции на мениджера на проекта и софтуерните архитекти. Модели и методологии за реализация на жизнения цикъл на софтуерните проекти. Управление на големи софтуерни проекти чрез Scrum и бързо прототипизиране.
3. Реализация на големи софтуерни проекти. Основни шаблони за архитектурен дизайн. Основни стандарти, използвани при реализация на софтуерни проекти. Системи за управление и контрол на програмни грешки (JIRA). Методи за тестване на големи софтуерни проекти. Програмни инструменти за автоматизиране на процеса на тестване. Стандарти за качество и тестване на програмни системи.
4. Документиране на големи софтуерни проекти.

Изучаването на дисциплината се основава на познанията на студентите по дисциплините:

- Обектно-ориентирано програмиране.
- Програмиране в Интернет среда.
- Проектиране на програмни интерфейси.
- Нерелационни бази от данни.
- Дизайн на софтуерни проекти.
- Програмиране за мобилни устройства.
- Управление на софтуерни проекти.
- Документиране и презентирание на софтуерни проекти.
- Тестване на качеството на софтуер.

Получените знания и умения са необходими за следните дисциплини:

- Дипломно проектиране.

III. СЪДЪРЖАНИЕ НА УЧЕБНАТА ПРОГРАМА

№	Теми на лекциите и упражненията	Вид на обучението, часа	
		РО	ЗО
1	2	3	4
	МОДУЛ I: ЖИЗНЕН ЦИКЪЛ НА ГОЛЕМИ ПРОГРАМНИТЕ СИСТЕМИ - 16ч. (12+0+4) Лекции – 12 (6) часа, Лабораторни упражнения - 4 (2) часа		
	А. ЛЕКЦИИ	12	6
1.1	Планиране. Анализ на техническите, икономически и организационни възможности за създаване на системата. Дефиниране на целта и основните задачи. Планиране на колектива и времето, необходимо за разработване на системата. Анализ на използваните стандарти. Анализ на риска.	2	1
1.2	Анализ. Бизнес анализ. Анализ на таргет групата, която ще използва системата, нейната приложимост и необходимост на базата на интервюта и анкети. Дефиниране на изискванията към системата. Анализ на базата на конкретни случаи на използване (use cases). Моделиране на потока от данни.	2	1
1.3	Дизайн. Избор на стратегия за дизайн. Избор на архитектура на системата. Дизайн на потребителския интерфейс. Програмен дизайн. Дизайн на необходимата база данни.	4	2
1.4	Реализация. Реализация на програмния код. Тестване на логиката програмния код. Тестване на производителността и сигурността на програмния код. Документиране. План за обучение за използване на системата.	4	2
	В. ЛАБОРАТОРНИ УПРАЖНЕНИЯ	4	2
1.1	Планиране и анализ на големи софтуерни проекти.	2	1
1.2	Дизайн и реализация на големи софтуерни проекти.	2	1
	МОДУЛ II: УПРАВЛЕНИЕ НА ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ- 16ч. (10+0+6) Лекции – 10 (5) часа, Лабораторни упражнения - 6 (3) часа		
	А. ЛЕКЦИИ	10	5
2.1	Основни роли на мениджъра на проекта. Анализ на изискванията към програмната система. Видове изисквания към един програмен проект (оперативни изисквания, изисквания към производителност и сигурност. Бизнес анализ.	2	1
2.2	Основни роли на програмния архитект. Системен и инфраструктурен анализ. Избор на програмна архитектура, използвани стандарти, технологии и интерфейси. Разпределение на задачите между членовете на колектива.	4	2
2.3	Избор на модел за реализация на жизнения цикъл на системата. Rapid Application Development (RAD) и модели, фокусирани на програмния код (Agile модел), използване на програмни прототипи (системно, бързо и еволюционно прототипиране). Критерии за избор на метод и методологии за реализация на големи програмни системи.	2	1
2.4	Управление на големи софтуерни проекти чрез Scrum и бързо прототипиране.	2	1
	В. ЛАБОРАТОРНИ УПРАЖНЕНИЯ	6	3
2.1	Анализ на изискванията към програмния проект чрез таргетирано анкетиране и интервюта.	2	1
2.2	Критерии за избор на архитектура на програмния проект. Използване на UML диаграми на програмните класове и интерфейси.	2	1
2.3	Управление на големи софтуерни проекти чрез Scrum.	2	1

	МОДУЛ III : РЕАЛИЗАЦИЯ НА ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ - 18ч. (8+0+10) Лекции- 8 (4) часа, Лабораторни Упражнения - 10 (5) часа.		
	A. ЛЕКЦИИ	8	4
3.1	Архитектурен дизайн. Архитектура клиент-сървър. Клиент-базирани и сървър-базирани архитектури. Full-stack архитектури. Облачно-базирани архитектури. Еднослойни срещу многослойни архитектури.	2	1
3.2	Избор на стандарти при реализация на големи програмни проекти. Видове стандарти (стандарти за документиране; стандарти за писане на качествен програмен код; стандарти, свързани с програмната спецификация; стандарти за потребителския интерфейс; стандарти за тестване на програмни системи).	2	1
3.3	Системи за управление на програмни грешки. JIRA.	2	1
3.4	Тестване на големи програмни проекти. Планиране на тестването. Методи за тестване (black box, gray box, white box). Ръчно и автоматизирано тестване. Статично и динамично тестване. Функционално тестване. Критерии за приключване на процеса на тестване. Програмни инструменти за автоматизиране на процеса на тестване. Стандарти за качество и тестване на програмни системи: ISO 9001:2000, ISO 9004:2000, ISO/IEC/IEEE 29119.	2	1
	B. ЛАБОРАТОРНИ УПРАЖНЕНИЯ	10	5
3.1	Разработване на проект с архитектура клиент-сървър (Java).	4	2
3.2	Разработване на проект с full-stack архитектура (JavaScript).	4	2
3.3	Управление на програмни грешки чрез JIRA.	2	1
	Лекции	Общо:	30
	Лабораторни упражнения	Общо:	15
			20
			10

IV. ФОРМИ НА КОНТРОЛ НА ЗНАНИЯТА

1. През семестъра: оценката на знанията през семестъра се извършва чрез две контролни работи, както и две задачи за самостоятелната работа. Студентите решават поставени проблеми по тематиката на дисциплината и представят своите решения в края на семестъра. Въз основа на контролните и самостоятелните задачи се формира средна оценка от текущ контрол. Ако тази оценка е по-висока или равна на 5.00 студентът може да бъде освободен от изпит.

2. Изпит: Семестриалният изпит е писмен. По време на изпита студентите решават тест от отворен тип. Тестът включва въпроси от материалите, разглеждани на лекции, семинарни и практически занятия. Регламентиран е броя точки за всяка оценка по шестобалната система. На базата на резултатите се поставя оценка.

3. Крайната оценка по дисциплината се формира съгласно приетата в катедра КСТ методика.

ЛИТЕРАТУРА

А.Основна:

1. Sungdeok Cha, Richard N. Taylor, Kyochul Kang, Handbook of Software Engineering, Springer, ISBN 978-3-030-00261-9, 2019.
2. Alan dennis, Barbara Haley, Roberta M. Roth, SYSTEM ANALYSIS AND DESIGN (Fifth Edition), John Wiley & Sons, Inc., 2012

Б. Допълнителна:

1. John Dooley, Software Development and Professional Practice, ISBN 978-1-4302-3801-0, Apress, 2011
2. Capers Jones, Software Engineering Best Practices, McGraw-Hill, ISBN: 978-0-07-162162-5, 2010

Съставил:

/ доц.д-р инж. Р. Иванов /

Програмата е приета от КС на катедра КСТ с Протокол № 8/07.07.2020г.

Ръководител катедра:

/ доц. д-р инж. В. Кукенска /

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ГАБРОВО
ФАКУЛТЕТ „ЕЛЕКТРОТЕХНИКА И ЕЛЕКТРОНИКА”**

Приета с решение на ФС
Протокол № 6/27.07.2020 г.

Утвърдил

Декан:

/проф. д-р инж. З.Ненова /

**ХАРАКТЕРИСТИКА
НА ДИСЦИПЛИНАТА „ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ”
ЗА СПЕЦИАЛНОСТ „СОФТУЕРНО И КОМПЮТЪРНО ИНЖЕНЕРСТВО”
РЕДОВНО ОБУЧЕНИЕ**

Обучаваща катедра: „ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ ”

Образователно-квалиф. степен: Бакалавър	Вид на дисциплината: Задължителна	№ по учебен план 41.1	Година: 4
Семестър: VIII	Брой кредити: 5	Водещ преподавател: доц. д-р инж. Росен Стефанов Иванов	
Цел на курса: Дисциплината „Големи софтуерни проекти” има за цел да научи студентите да могат да създават и внедряват големи софтуерни проекти на базата на систематизиране на техните знания на в областта на проектиране, разработка и тестване на софтуерни системи.			
Необходими условия: Лекционна зала, лабораторна зала с компютърни системи с достъп до Интернет, проектор, фирмена литература.			
Съдържание на курса: Основни модули от курса на обучение са: ЖИЗНЕН ЦИКЪЛ НА ГОЛЕМИ ПРОГРАМНИТЕ СИСТЕМИ, УПРАВЛЕНИЕ НА ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ, РЕАЛИЗАЦИЯ НА ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ.			
Препоръчителна литература: <ol style="list-style-type: none"> 1. Sungdeok Cha, Richard N. Taylor, Kyochul Kang, Handbook of Software Engineering, Springer, ISBN 978-3-030-00261-9, 2019. 2. Alan dennis, Barbara Haley, Roberta M. Roth, SYSTEM ANALYSIS AND DESIGN (Fifth Edition), John Wiley & Sons, Inc., 2012 3. John Dooley, Software Development and Professional Practice, ISBN 978-1-4302-3801-0, Appress, 2011 4. Capers Jones, Software Engineering Best Practices, McGraw-Hill, ISBN: 978-0-07-162162-5, 2010. 			
Методи на преподаване: Лекции, лабораторни упражнения, протоколи, решаване на индивидуални задачи, електронни фирмени каталози, програмни среди за тестване и симулиране.			
Методи на оценяване: Семестриалният изпит е писмен. По време на изпита студентите решават отворен тест. Тестът включва въпроси от материалите разглеждани на лекции, семинарни и лабораторни упражнения. Регламентиран е броя точки за всяка оценка по шестобалната система. Окончателната оценка се формира на база оценките от теста и текущия контрол.			
Кредити по видове дейност: Аудиторна заетост: (30л /20лу, общо 50 часа): 2.0кредита Извънаудиторна заетост: (75 часа): 3.0кредита: Д.2. Посещение на библиотека: 0.2 к.; Д.3.Задания за извънаудиторно решаване на задачи 0.7 к.; Д.6.Обучение чрез електронни версии на курсове (E-learning) 0.3 к.; Д.7.Подготовка за изпит 1.2 к.;Д.15.Домашни работи от различен тип0.3 к.;Д.23. Консултации с преподавател 0.3 к.			
Език, на който се преподава: български			

Характеристиката е приета от КС на катедраКСТ с Протокол № 8/07.07.2020г.

Ръководител катедра:

/доц.д-р инж. В.Кукенска/

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ГАБРОВО
ФАКУЛТЕТ „ЕЛЕКТРОТЕХНИКА И ЕЛЕКТРОНИКА”

Приета с решение на ФС
Протокол № 6/27.07.2020 г.
/проф. д-р инж. З.Ненова /

Утвърдил
Декаан:

ХАРАКТЕРИСТИКА
НА ДИСЦИПЛИНАТА „ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ”
ЗА СПЕЦИАЛНОСТ „СОФТУЕРНО И КОМПЮТЪРНО ИНЖЕНЕРСТВО”
ЗАДОЧНО ОБУЧЕНИЕ

Обучаваща катедра: „ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ ”

Образователно-квалиф. степен: Бакалавър	Вид на дисциплината: Задължителна	№ по учебен план 41.1	Година: 4
Семестър: VIII	Брой кредити: 5	Водещ преподавател: доц. д-р инж. Росен Стефанов Иванов	
Цел на курса: Дисциплината „Големи софтуерни проекти” има за цел да научи студентите да могат да създават и внедряват големи софтуерни проекти на базата на систематизиране на техните знания на в областта на проектиране, разработка и тестване на софтуерни системи.			
Необходими условия: Лекционна зала, лабораторна зала с компютърни системи с достъп до Интернет, проектор, фирмена литература.			
Съдържание на курса: Основни модули от курса на обучение са: ЖИЗНЕН ЦИКЪЛ НА ГОЛЕМИ ПРОГРАМНИТЕ СИСТЕМИ, УПРАВЛЕНИЕ НА ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ, РЕАЛИЗАЦИЯ НА ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ.			
Препоръчителна литература: 1. Sungdeok Cha, Richard N. Taylor, Kyochul Kang, Handbook of Software Engineering, Springer, ISBN 978-3-030-00261-9, 2019. 2. Alan dennis, Barbara Haley, Roberta M. Roth, SYSTEM ANALYSIS AND DESIGN (Fifth Edition), John Wiley & Sons, Inc., 2012 3. John Dooley, Software Development and Professional Practice, ISBN 978-1-4302-3801-0, Appress, 2011. 4. Capers Jones, Software Engineering Best Practices, McGraw-Hill, ISBN: 978-0-07-162162-5, 2010.			
Методи на преподаване: Лекции, лабораторни упражнения, протоколи, решаване на индивидуални задачи, електронни фирмени каталози, програмни среди за тестване и симулиране.			
Методи на оценяване: Семестриалният изпит е писмен. По време на изпита студентите решават отворен тест. Тестът включва въпроси от материалите разглеждани на лекции, семинарни и лабораторни упражнения. Регламентиран е броя точки за всяка оценка по шестобалната система. Окончателната оценка се формира на база оценките от теста и текущия контрол.			
Кредити по видове дейност: Аудиторна заетост: (15 часа л+10 часа лу, общо 25 часа): 1,0 кредита Извънаудиторна заетост: (100 часа): 4,0 кредита: Д.2. Посещение на библиотека: 0.2 к.; Д.3. Задания за извънаудиторно решаване на задачи 0.7 к.; Д.6. Обучение чрез електронни версии на курсове (E-learning) 0,3 к.; Д.7. Подготовка за изпит 1.5 к.; Д.8. Подготовка за текущо проверяване и оценяване на постиженията 0.5 к.; Д.15. Домашни работи от различен тип 0.5 к.; Д.23. Консултации с преподавател 0.3 к.			
Език, на който се преподава: български			

Характеристиката е приета от КС на катедра КСТ с Протокол № 8/07.07.2020г.

Ръководител катедра:

/доц.д-р инж. В.Кукенска/

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО
КАТЕДРА „ОСНОВИ НА ЕЛЕКТРОТЕХНИКАТА И ЕЛЕКТРОЕНЕРГЕТИКАТА”

ВЪПРОСНИК

по „ГОЛЕМИ СОФТУЕРНИ ПРОЕКТИ”

за студентите от специалност: "Софтуерно и Компютърно Инженерство" - редовна и задочна форма на обучение за образователно-квалификационна степен „Бакалавър”

1. Жизнен цикъл на големи софтуерни проекти - планиране. Анализ на техническите, икономически и организационни възможности за създаване на системата. Дефиниране на целта и основните задачи. Планиране на колектива и времето, необходимо за разработване на системата. Анализ на използваните стандарти. Анализ на риска.
2. Жизнен цикъл на големи софтуерни проекти - анализ. Бизнес анализ. Анализ на таргет групата, която ще използва системата. Анализ на приложимостта и необходимостта на системата на базата на интервюта и анкети. Дефиниране на изискванията към системата. Анализ на базата на конкретни случаи на използване (use cases). Моделиране на потока от данни.
3. Жизнен цикъл на големи софтуерни проекти - дизайн. Избор на стратегия за дизайн. Избор на архитектура на системата. Дизайн на потребителския интерфейс. Програмен дизайн. Дизайн на необходимата база данни.
4. Жизнен цикъл на големи софтуерни проекти - реализация. Реализация на програмния код. Тестване на логиката програмния код. Тестване на производителността и сигурността на програмния код. Документиране. План за обучение за използване на системата.
5. Основни роли на мениджъра на проекта. Анализ на изискванията към програмната система. Видове изисквания към един програмен проект (оперативни изисквания, изисквания към производителност и сигурност; културни и политически изисквания). Бизнес анализ.
6. Основни роли на програмния архитект. Системен и инфраструктурен анализ. Избор на програмна архитектура, използвани стандарти, технологии и интерфейси. Разпределение на задачите между членовете на колектива.
7. Избор на модел за реализация на жизнения цикъл на системата. Модел Rapid Application Development (RAD) и модели, фокусирани на програмния код (Agile модел), използване на програмни прототипи (системно, бързо и еволюционно прототипиране). Критерии за избор на метод и методологии за реализация на големи програмни системи.
8. Архитектурен дизайн. Архитектура клиент-сървър. Клиент-базирани и сървър-базирани архитектури. Full-stack архитектури. Облачно-базирани архитектури. Еднослойни срещу многослойни архитектури.
9. Избор на стандарти при реализация на големи програмни проекти. Видове стандарти (стандарти за документиране; стандарти за писане на качествен програмен код; стандарти, свързани с програмната спецификация; стандарти за потребителския интерфейс; стандарти за тестване на програмни системи).
10. Системи за управление на програмни грешки. JIRA.
11. Тестване на големи програмни проекти. Планиране на тестването. Методи за тестване (black box, gray box, white box). Ръчно и автоматизирано тестване. Статично и динамично тестване. Функционално тестване. Критерии за приключване на процеса на тестване. Програмни инструменти за автоматизиране на процеса на тестване. Стандарти за качество и тестване на програмни системи: ISO 9001:2000, ISO 9004:2000, ISO/IEC/IEEE 29119. Система за управление на софтуерното тестване HP Quality Center.

ЛИТЕРАТУРА

А. Основна:

1. SungdeokCha, RichardN. Taylor, KyochulKang, HandbookofSoftwareEngineering, Springer, ISBN 978-3-030-00261-9, 2019.
2. Alandennis, BarbaraHaley, RobertaM. Roth, SYSTEMANALYSISANDDESIGN (FifthEdition), JohnWiley&Sons, Inc., 2012.

Б. Допълнителна:

1. John Dooley, Software Development and Professional Practice, ISBN 978-1-4302-3801-0, Appress, 2011.
2. Capers Jones, Software Engineering Best Practices, McGraw-Hill, ISBN: 978-0-07-162162-5, 2010.

Съставил:

/ доц.д-р инж. Р. Иванов /