



РЕПУБЛИКА БЪЛГАРИЯ
Заместник-министър на образованието и науката

ЗАПОВЕД

№ РД 09 – 2488/18.09.2020 г.

На основание чл. 13д, ал. 2, т. 1 от Закона за професионалното образование и обучение, при спазване на изискванията на чл. 66, ал. 1 и 2 от Административнопроцесуалния кодекс и във връзка с осигуряването на обучението по учебен предмет и Заповед № РД 09-3708/23.08.2017 г. на министъра на образованието и науката

УТВЪРЖДАВАМ

Учебна програма за специфична професионална подготовка по учебен предмет **софтуерно инженерство** за специалност код **4810201 „Системно програмиране“** от професия код **481020 „Системен програмист“** и специалност код **4810301 „Приложно програмиране“** от професия код **481030 „Приложен програмист“** от професионално направление код **481 „Компютърни науки“**.

Учебната програма влиза в сила от учебната 2020/2021 година.

X

Таня Михайлова
Зам.-министър на образованието и науката

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

У Ч Е Б Н А П Р О Г Р А М А

ЗА СПЕЦИФИЧНА ПРОФЕСИОНАЛНА ПОДГОТОВКА

ПО

СОФТУЕРНО ИНЖЕНЕРСТВО

УТВЪРДЕНА СЪС ЗАПОВЕД № РД 09 – 2488/18.09.2020 г.

ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ: 481 „КОМПЮТЪРНИ НАУКИ“

ПРОФЕСИИ: 481020 „СИСТЕМЕН ПРОГРАМИСТ”

481030 „ПРИЛОЖЕН ПРОГРАМИСТ”

СПЕЦИАЛНОСТИ: 4810201 „СИСТЕМНО ПРОГРАМИРАНЕ“

4810301 „ПРИЛОЖНО ПРОГРАМИРАНЕ“

София, 2020 година

I. ОБЩО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА

Учебната програма по **софтуерно инженерство** е предназначена за специалности код 4810201 „Системно програмиране” и код 4810301 „Приложно програмиране”, за които в типовите учебни планове е предвидено изучаването на учебния предмет **софтуерно инженерство** в 58 часа.

Програмата е разработена в съответствие с Държавния образователен стандарт за придобиване на квалификация по професията.

Учебното съдържание в програмата е структурирано в четири раздела, които дават възможност на учениците да получат знания и умения, свързани с работния процес, методологиите за разработка на софтуер, моделирането на софтуер, работата с чужд код, работата в екип и софтуерното тестване.

Обучението по предмета се извършва във взаимна връзка с предмети от отрасловата и специфичната професионална подготовка - „Разработка на софтуер”, „Бази данни” и „Интернет програмиране”.

II. ЦЕЛИ НА ОБУЧЕНИЕТО ПО ПРЕДМЕТА

Обучението по предмета има за цел учениците да придобият знания и умения за цялостния работен процес по изграждането и поддържането на софтуер. За постигане на основната цел на обучението по **софтуерно инженерство** е необходимо изпълнението на следните подцели:

- придобиване на знания и умения, свързани с основните процеси на софтуерното тестване;
- Разбиране и прилагане на основните методологии за разработка на софтуер;
- Придобиване на знания и умения за работа със и създаване на техническа документация;
- Затвърждаване на уменията за работа със системи за контрол на версията в рамките на екип;
- Разбиране на процеса на управление на задачите в софтуерен екип;
- Затвърждаване на уменията за работа с чужд код.

III. УЧЕБНО СЪДЪРЖАНИЕ

1. Учебното съдържание е структурирано в раздели и теми. За всеки раздел в програмата

е определен минимален брой учебни часове. Учителят разпределя броя учебни часове за нови знания, упражнения и оценяване, при спазване изискванията за минимален брой часове по раздели.

2. Разликата между броя на учебните часове в учебния план и общия минимален брой, предвиден в учебната програма, определя резерва часове. Те се разпределят по теми в началото на учебната година от учителя.

3. Раздели и теми

№	Наименование на разделите	Минимален брой часове
1.	Работен процес в софтуерната индустрия	18
2.	Методологии за разработка на софтуер. Работа с чужд код	16
3.	Софтуерно тестване	14
4.	Софтуерна документация	6
	Общ минимален брой часове	54
	Резерв часове	4
	Общ брой часове	58

Раздел 1. Работен процес в софтуерната индустрия

- a. Основи на процесът на разработка на софтуер
- b. Етапи в разработката на софтуер
- c. Методологии за разработка на софтуер
- d. Софтуерни изисквания
- e. Случаи на употреба (use cases) и Потребителски истории (user stories)
- f. Спецификация на изискванията
- g. Прототипиране на потребителски интерфейс

Раздел 2. Методологии за разработка. Работа с чужд код

- a. Методологии за разработка на софтуер - Waterfall и Agile
- b. Scrum. Scrum артефакти, екипи, роли и събития
- c. Инструменти за управление на екип
- d. Работа със системи за контрол на версиите
- e. Работа с чужд код
- f. Преглед на чужд код (code review)

Раздел 3. Софтуерно тестване

- a. Софтуерно тестване
- b. Компонентно тестване
- c. Регресивно тестване
- d. Mocking при тестване
- e. Интеграционно тестване
- f. Инструменти за непрекъсната интеграция

Раздел 4. Софтуерна документация

- a. Софтуерна документация
- b. Документация на код
- c. Документация за крайния потребител

IV. ОЧАКВАНИ РЕЗУЛТАТИ ОТ УЧЕНЕТО – ЗНАНИЯ, УМЕНИЯ И КОМПЕТЕНТНОСТИ

В края на обучението по учебния предмет учениците трябва да:

- Познават основните етапи в разработката на софтуер
- Познават и различават различни методологии за разработка на софтуер
- Познават Agile методологията за разработка на софтуер
- Познават SCRUM работната рамка, както и артефактите, екипите, ролите, събитията в рамките на SCRUM.
- Правят разлика между use case и user story
- Познават начини за създаване на прототипи на потребителски интерфейс на софтуерни приложения
- Умеят да работят с инструменти за създаване на документация
- Познават различни системи за контрол на версиите
- Познават начините за работа с основните системи за контрол на версиите
- Разбират понятията commit, branch, merge, push и pull в контекста на source/version

control системата, която използва

- Разбират понятието конфликт в контекста на програмирането и знае методи за преодоляването на конфликти
- Разбират необходимостта от управление на софтуерната разработка и основни методологии за това
- Разбират идеята зад това да има „задачи“ в разработката на един софтуер
- Знаят нужните реквизити за една задача (описание, приоритет, свързани задачи, поемане на отговорност за задача)
- Разбират основните състояния на една задача (Backlogged, Pending/ToDo, Doing, For Review, Done)
- Разбират концепцията за "блокиране" на задача от друга задача
- Умеят да четат тестове като документация на чужд код
- Умеят да работят с чуждо API
- Ориентират се в документацията на чужд код

V. ОБЯСНИТЕЛНИ БЕЛЕЖКИ

За целите на обучението се препоръчва учениците да използват инструменти като Jira и Trello за управление на екипите, в които участват, следвайки принципите на SCRUM.

VI. ЛИТЕРАТУРА

1. Martin C. R., Clean Code: A Handbook of Agile Software Craftsmanship, Pearson, 2008, ISBN: 9780132350884
2. McDowell G., Cracking the Coding Interview: 189 Programming Questions and Solutions, CarrerCup, 2015, ISBN: 0984782850
3. R. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 2000, ISBN: 0073655783
4. Feathers M., Working Effectively with Legacy Code, Pearson, 2004, ISBN: 9780131177055
5. Loeliger J. & Mccullough M., Version Control with Git: Powerful Tools And Techniques For Collaborative Software Development Second Edition, O'Reilly Media, 2012, ISBN: 9781449316389
6. <https://www.scrumguides.org/scrum-guide.html>

VII. АВТОРСКИ КОЛЕКТИВ

Програмата е разработена, обсъдена и оформена от експертна група към Национална програма „Обучение за ИТ кариера“ към МОН в състав:

1. доц. д-р Димитър Минчев, Бургаски свободен университет, Бургас
2. доц. д-р Ивайло Старибратов - ПУ „Паисий Хилендарски“
3. Петър Петров – ПГЕЕ „Константин Фотинов“, Бургас
4. Росен Вълчев, МГ „Акад. Кирил Попов“, Пловдив
5. инж. Хриси Плачкова, МГ „Акад. Кирил Попов“, Пловдив