

РЕПУБЛИКА БЪЛГАРИЯ

Заместник-министър на образованието и науката

**З А П О В Е Д**

**№ РД 09 – 2491/18.09.2020 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

**У Т В Ъ Р Ж Д А В А М**

Учебна програма за специфична професионална подготовка по учебен предмет **учебна практика: софтуерно инженерство** за 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.



*Приложение*

**МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА**

**У Ч Е Б Н А П Р О Г Р А М А**

**ЗА СПЕЦИФИЧНА ПРОФЕСИОНАЛНА ПОДГОТОВКА**

по

**УЧЕБНА ПРАКТИКА: СОФТУЕРНО ИНЖЕНЕРСТВО**

УТВЪРДЕНА СЪС ЗАПОВЕД **№ РД 09 – 2491/18.09.2020 г.**

**ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ: 481 „КОМПЮТЪРНИ НАУКИ“**

**ПРОФЕСИИ: 481030 „ПРИЛОЖЕН ПРОГРАМИСТ”**

**СПЕЦИАЛНОСТИ: 4810301 „ПРИЛОЖНО ПРОГРАМИРАНЕ“**

**София, 2020 година**

1. **ОБЩО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА**

Учебната програма по **учебна практика: софтуерно инженерство** е предназначена за специалност код 4810301 „Приложно програмиране”, за която в типовите учебни планове е предвидено изучаването на учебния предмет **учебна практика: софтуерно инженерство** в 87 часа.

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

Учебното съдържание в програмата е структурирано в пет 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”.

1. **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 **учебна практика: 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.

1. **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. | Подготовка за разработка на екипен проект | 10 |
| 2. | Първи спринт от разработката на проекта | 18 |
| 3. | Втори спринт от разработката на проекта | 18 |
| 4. | Трети спринт от разработката на проекта | 18 |
| 5. | Четвърти спринт от разработката на проекта | 18 |
|  | **Общ минимален брой часове** | **82** |
|  | **Резерв часове** | **5** |
|  | **Общ брой часове** | **87** |

###### Раздел 1. Подготовка за разработка на екипен проект

* 1. Разпределяне на учениците по екипи
  2. Анализ на заданието на проекта
  3. Планиране на проекта
  4. Дизайн на проекта
  5. Подготовка за започване на работата на проекта.

**Раздел 2. Първи спринт от разработката на проекта**

* 1. Планиране на задачите в рамките на спринта.
  2. Изпълнение на задачите от спринта от всеки от участниците.
  3. Провеждане на спринт ревю за всеки екип.
  4. Провеждане на спринт ретроспекция за всеки екип.

**Раздел 3. Втори спринт от разработката на проекта**

* 1. Планиране на задачите в рамките на спринта.
  2. Изпълнение на задачите от спринта от всеки от участниците.
  3. Провеждане на спринт ревю за всеки екип.
  4. Провеждане на спринт ретроспекция за всеки екип.

**Раздел 4. Трети спринт от разработката на проекта**

* 1. Планиране на задачите в рамките на спринта.
  2. Изпълнение на задачите от спринта от всеки от участниците.
  3. Провеждане на спринт ревю за всеки екип.
  4. Провеждане на спринт ретроспекция за всеки екип.

**Раздел 5. Четвърти спринт от разработката на проекта**

* 1. Планиране на задачите в рамките на спринта.
  2. Изпълнение на задачите от спринта от всеки от участниците.
  3. Провеждане на спринт ревю за всеки екип.
  4. Провеждане на спринт ретроспекция за всеки екип.

1. **ОЧАКВАНИ РЕЗУЛТАТИ ОТ УЧЕНЕТО – ЗНАНИЯ, УМЕНИЯ И КОМПЕТЕНТНОСТИ**

В края на обучението по учебния предмет учениците трябва да:

* + Умеят да следват основните етапи в софтуерната разработка
  + Разбират силните и слабите страни на различните методологии и правилно избират подходяща методология
  + Познават Agile методологията за разработка на софтуер
  + Умеят да прилагат SCRUM работната рамка
  + Ориентират се в спецификация на изискванията на софтуер
  + Умеят да създават прототипи на потребителски интерфейс на софтуерни приложения
  + Избират подходящи софтуерни технологии и аргументират избора, според спецификацията на изискванията на софтуер
  + Умеят да използват автоматизирани 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
  + Познават начини за интеграция на source/version control системи в среди за разработка
  + Познават добри практики, свързани с процеса на поддържане на source/version control системи
  + Разбират необходимостта от управление на софтуерната разработка и основни методологии за това
  + Разбират идеята зад това да има „задачи“ в разработката на един софтуер
  + Знае нужните реквизити за една задача (описание, приоритет, свързани задачи, поемане на отговорност за задача)
  + Разбират основните състояния на една задача (Backlocked, Pending/Todo, Doing, For Review, Done)
  + Разбират концепцията за "блокиране" на задача от друга задача
  + Ориентират се в чужд код
  + Умеят да четат тестове като документация на чужд код
  + Умеят да работят с чуждо API
  + Ориентират се в документацията на чужд код
  + Извършват прегледи на кода (code reviews)

1. **ОБЯСНИТЕЛНИ БЕЛЕЖКИ**

Идеята на учебната практика е тя да се използва изцяло за работа по практически проект - уеб приложение, десктоп приложение, мобилно приложение и др. За целта, учениците се разпределят в екипи, а учителят изпълнява ролята на собственик на продукта (product owner) и/или scrum ръководител (scrum master) за всеки от екипите. Предвидените часове се разделят на спринтове, като за всеки спринт се провеждат съответните SCRUM събития - планиране, дневни срещи, ревю срещи, ретроспективни срещи. Дневните срещи могат да се заменят от срещи в началото на всеки блок от часове между членовете на екипа. В контекста на обучението по тази дисциплина срещите могат да имат различна продължителност от тяхната реална продължителност в индустрията. Авторският колектив препоръчва употребата на инструменти като JIRA и Trello за управление на екипите.

Авторският колектив препоръчва учебната практика по тази дисциплина да се разположи, по възможност, след напълно преминат теоретичен модул по софтуерно инженерство.

1. **ЛИТЕРАТУРА**
2. Martin C. R., Clean Code: A Handbook of Agile Software Craftsmanship, Pearson, 2008, ISBN: 9780132350884
3. МcDowell G., Cracking the Coding Interview: 189 Programming Questions and Solutions, CarrerCup, 2015, ISBN: 0984782850
4. R. Pressman, Software Engineering: A Practitioner‘s Approach, McGraw-Hill, 2000, ISBN: 0073655783
5. Feathers M., Working Effectively with Legacy Code, Pearson, 2004, ISBN: 9780131177055
6. Loeliger J. & Mccullough M., Version Control with Git: Powerful Tools And Techniques For Collaborative Software Development Second Edition, O'Reilly Media, 2012, ISBN: 9781449316389
7. <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