

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра компьютерных сетей

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 18 » мая 2020 г.

Рабочая программа дисциплины
«Основы современной промышленной разработки программного обеспечения»

Направление подготовки
09.03.03 Прикладная информатика

Профиль
«Прикладная информатика в экономике»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 16 апреля 2020 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 7 от
17 мая 2020 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Основы современной промышленной разработки программного обеспечения» являются изучение принципов и методологии промышленной разработки программного обеспечения (ПО), изучение особенностей управления разработкой ПО, освоение инструментов разработки ПО.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Основы современной промышленной разработки программного обеспечения» относится к факультативной части ОП бакалавриата.

Она базируется на знаниях и навыках, полученных студентами при изучении как общепрофессиональных дисциплин компьютерного цикла, в частности дисциплин «Основы программирования», «Языки и методы программирования», «Базы данных», так и дисциплин вариативной части, обеспечивающих получение профессиональных компетенций.

Дисциплина «Основы современной промышленной разработки программного обеспечения» обобщает профессиональные знания и навыки студента и готовит его к полноценной профессиональной деятельности.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Профессиональные компетенции		
ПК-1 Способность проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе	ПК-1.1. Демонстрирует способность проводить обследование организаций, выявлять информационные потребности пользователей ПК-1.2. Демонстрирует способность формировать требования к информационной системе ПК-1.3 Демонстрируют понимание влияния этических норм на разработку ПО, умение просчитывать риски и ответственность ИС	Знать: – роль требований к программному обеспечению в промышленной разработке; – инструменты разработки программного обеспечения; – методологию разработки программного обеспечения; – инструменты разработки программного обеспечения; Уметь: – реализовывать этапы разработки программного обеспечения; – управлять организацией этапов разработки

		<p>программного обеспечения;</p> <ul style="list-style-type: none"> – оценивать этапы разработки программного обеспечения; – применять корпоративные стандарты разработки программного обеспечения <p>Владеть навыками:</p> <ul style="list-style-type: none"> – работы с инструментами проектирования и разработки программного обеспечения; – работы с инструментами для управления разработкой программного обеспечения.
--	--	---

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 1 зач. ед., 36 академ. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)	
			Контактная работа							
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа		
1.	Извлечение требований к ПО.	7		2				2		
2.	Управление требованиями к ПО.	7		1				2	Контрольная работа	

3.	Проектирование пользовательских интерфейсов.	7		1	1			2	
4.	Гибкая методология разработки ПО.	7		1				1	Контрольная работа
5.	Планирование выпусков ПО.	7		1				1	
6.	Использование трекера задач.	7		1	1			1	
7.	Использование систем управления версиями	7		1				1	Контрольная работа
8.	Инструменты автоматической сборки и непрерывной интеграции	7		1	1			1	
9.	Инструменты профилирования и статического анализа кода	7		1	1			1	
10.	Проблемы управления конфигурацией и инструменты для их решения.	7		1				1	Контрольная работа
11.	Читаемость и сопровождаемость программного кода.	7		1				1	
12.	Управление знаниями в процессе промышленной разработки и задачи документирования программного обеспечения.	7		1				1	
13.	Организация и автоматизация технической поддержки программных продуктов.	7		1				0,7	Тест
Всего за 7 семестр				14	4			0,3	Зачет
Всего				14	4			18	

Содержание разделов дисциплины:

1. Извлечение требований к ПО. Понятие требования. Роль требований в процессе промышленной разработки. Источники и пути выявления требований. Свойства качественных требований. Анализ и тестирование требований.
2. Управление требованиями к ПО. Характеристика процесса управления требованиями и его роль в промышленной разработке. Жизненный цикл требований. Основные задачи управления требованиями. Инструментальные средства управления требованиями.
3. Проектирование пользовательских интерфейсов. Общая характеристика задачи проектирования пользовательских интерфейсов и её место в процессе разработки ПО. Понятие user experience (UX). Этапы разработки пользовательского интерфейса. Методы и инструментальные средства проектирования интерфейсов. Методы и инструментальные средства тестирования интерфейсов.

4. Гибкая методология разработки ПО. Основные положения гибкой методологии. Отличия гибкой методологии разработки от традиционной. Возможности и область применимости гибкой методологии. Отдельные методики в рамках гибкой методологии: XP, Scrum, Kanban, Lean.
5. Планирование выпусков ПО в рамках гибкой методологии разработки. Долгосрочное и краткосрочное планирование выпусков. Методики оценки трудозатрат на выполнение конкретных задач: покер планирования, определение размера футболки, оценка относительных масс, оценка на стене.
6. Использование трекера задач в промышленной разработке. Задачи, решаемые трекером задач в рамках процесса разработки. Жизненный цикл задачи в трекере. Трекер задач как средство регламентации деятельности разработчика.
7. Использование систем управления версиями в промышленной разработке. Виды систем контроля версий: централизованные и распределённые. Коллективная разработка с использованием систем контроля версий. Интеграция систем контроля версий в проектную инфраструктуру. Сценарии решения типичных задач промышленной разработки с использованием систем контроля версий.
8. Инструменты автоматической сборки и непрерывной интеграции в промышленной разработке. Задачи систем автоматической сборки в промышленной разработке. Обзор возможностей наиболее распространённых систем автоматической сборки. Понятие непрерывной интеграции, её преимущества и недостатки, область применения. Инструменты непрерывной интеграции и сценарии их использования в промышленной разработке.
9. Инструменты профилирования и статического анализа кода в промышленной разработке. Назначение процедуры профилирования кода и её область применения. Принципы работы профайлеров. Анализ профиля и техники оптимизации, основанные на нём. Назначение процедуры статического анализа кода и её область применения. Возможности существующих статических анализаторов. Роль статического анализа в промышленной разработке.
10. Проблемы управления конфигурацией и инструменты для их решения. Задача управления конфигурацией и её актуальность в промышленной разработке. Проблема воспроизводимости процесса и подходы к её решению. Роль управления конфигурации в управлении проектом. Инструменты управления конфигурацией и основные сценарии их применения.
11. Читаемость и сопровождаемость программного кода. Жизненный цикл программного кода в промышленной разработке. Связь читаемости кода и его сопровождаемости. Роль корпоративных стандартов на код в промышленной разработке. Методы и приёмы обеспечения читаемости и сопровождаемости кода. Инструментальные средства проверки кода на соответствие стандартам.
12. Управление знаниями в процессе промышленной разработки и задачи документирования программного обеспечения. Актуальность задачи управления знаниями в процессе разработки. Инструментальные средства управления знаниями (howto, wiki и т. п.). Актуальность автоматической генерации документации. Требования предъявляемые к документации. Инструменты для автоматической генерации документации.
13. Организация и автоматизация технической поддержки программных продуктов. Организация взаимодействия с пользователями и ресурсы, необходимые для неё. Техническая поддержка по электронной почте и через call-центры. Инструменты автоматизации технической поддержки.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами

OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Инженерия программных систем и комплексов на основе гибкой методологии разработки : учебно-методическое пособие / И. В. Парамонов, А. М. Васильев ; Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2015. — 47 с.
2. Парамонов И. В. Инженерия программных систем и комплексов на основе гибкой методологии разработки: учеб.-метод. пособие для студентов, обучающихся по направлению Прикладная математика и информатика. [Электронный ресурс]/ И. В. Парамонов; Яросл. гос. ун-т им. П. Г. Демидова, Науч.-метод. совет ун-та - Ярославль: ЯрГУ, 2015. - 47 с. <http://www.lib.uni Yar.ac.ru/edocs/iuni/20150402.pdf>

б) дополнительная:

1. Макконнелл, С., Совершенный код. Мастер-класс : пер. с англ / С. Макконнелл, М., Русская редакция, 2013, 868с
2. Мацяшек Л. А. Практическая программная инженерия на основе учебного примера. / Л. А. Мацяшек, Б. Л. Лионг; пер. с англ. А. М. Епанешникова, В. А. Епанешникова - М.: Бином. Лаборатория знаний, 2013. - 956 с.: ил.

в) ресурсы сети «Интернет»

1. <http://parallels.nsu.ru/~fat/subversion.ppt> - Лекции по инструментам управления конфигурацией, Иртегов Д.В
2. <http://zzet.org/git/learning/undev/coursify/2014/02/09/lecture-1-git-course-undev.html> - Лекции о системе управления версиями Git
3. http://lib.custis.ru/Continuous_Integration - Непрерывная интеграция

4. http://www.redmine.org/projects/redmine/wiki/RusUser_Guide - Руководство пользователя системы Redmine
5. <http://svnbook.red-bean.com/index.ru.html> - Управление версиями в Subversion
6. <http://git-scm.com/book/ru/v1> - Использование Git

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы) :

Доцент кафедры КС И. В. Пармонов

**Приложение №1 к рабочей программе дисциплины
«Основы современной промышленной разработки
программного обеспечения»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки
знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы
формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей
аттестации**

Типовые варианты контрольных работ

Контрольная работа по теме «Управление требованиями к ПО»

1. Что такое выполнение оценки программного проекта?
2. Охарактеризуйте рекомендуемое правило распределения затрат проекта.
3. В чем состоит планирование управления риском?

Контрольная работа по теме «Гибкая методология разработки ПО»

1. Основные положения гибкой методологии.
2. Отличия гибкой методологии разработки от традиционной.
3. Возможности и область применимости гибкой методологии.

Контрольная работа по теме «Использование систем управления версиями»

1. Из каких подсистем состоит система контроля версий?
2. Чем отличаются централизованная и распределённая системы контроля версий?
3. Перечислите сценарии решения типичных задач промышленной разработки с использованием систем контроля версий.

Контрольная работа по теме «Проблемы управления конфигурацией и инструменты
для их решения»

1. Дайте определение модели данных, модели архитектуры и модели подсистем.
2. Поясните понятия модуля и модульности. Зачем используют модули?
3. Роль управления конфигурации в управлении проектом.

Первые две контрольные работы проверяют уровень сформированности компетенции ПК-11, последние две контрольные работы проверяют уровень сформированности компетенции ПК-21.

Критерии оценивания:

Оценка	Критерии
Отлично Уровень формирования компетенций: высокий	Демонстрирует полное знание материалы по всем трем вопросам При ответе на вопрос рассуждает уверенно, аргументированно, приводит примеры.

Хорошо Уровень формирования компетенций: продвинутый	Демонстрирует полное знание материалы по крайней мере в двух вопросах, ответ на третий может быть не полным, но без грубых ошибок или присутствуют неточности при ответе на вопросы При ответе на вопрос рассуждает уверенно, но не всегда аргументированно, примеры приводит не всегда.
Удовлетворительно Уровень формирования компетенций: пороговый	Демонстрирует полное знание материалы по крайней мере в одном вопросе, ответы на другие вопросы не полные, но без грубых ошибок или присутствуют ошибки при ответе на все вопросы, но основной материал описан правильно. При ответе на вопрос рассуждает неуверенно, не всегда аргументированно, злоупотребляет воспроизведением теоретического материала, слабо фокусируясь на поставленном вопросе.
Неудовлетворительно	Не знает ответы на два из трех вопросов или допускает грубые ошибки в ответах на все вопросы. Не понимает или плохо понимает суть поставленного вопроса. При ответе на вопрос демонстрирует неспособность логично рассуждать, злоупотребляет воспроизведением теоретического материала, не фокусируясь на поставленном вопросе.

Список вопросов к зачету

1. Понятие требования к ПО.
2. Роль требований в процессе промышленной разработки.
3. Источники и пути выявления требований.
4. Свойства качественных требований.
5. Анализ и тестирование требований.
6. Характеристика процесса управления требованиями и его роль в промышленной разработке.
7. Жизненный цикл требований.
8. Основные задачи управления требованиями.
9. Инструментальные средства управления требованиями.
10. Задача проектирования пользовательских интерфейсов и её место в процессе разработки ПО.
11. Понятие user experience (UX).
12. Этапы разработки пользовательского интерфейса.
13. Методы и инструментальные средства проектирования интерфейсов.
14. Методы и инструментальные средства тестирования интерфейсов.
15. Основные положения гибкой методологии.
16. Отличия гибкой методологии разработки от традиционной.
17. Возможности и область применимости гибкой методологии.
18. Отдельные методики в рамках гибкой методологии: XP, Scrum, Kanban, Lean.
19. Планирование выпусков ПО в рамках гибкой методологии разработки.
20. Долгосрочное и краткосрочное планирование выпусков.
21. Методики оценки трудозатрат на выполнение конкретных задач: покер планирования, определение размера футболки, оценка относительных масс, оценка на стене.
22. Задачи, решаемые трекером задач в рамках процесса разработки.
23. Жизненный цикл задачи в трекере.
24. Трекер задач как средство регламентации деятельности разработчика.
25. Виды систем контроля версий: централизованные и распределённые.

26. Коллективная разработка с использованием систем контроля версий.
27. Интеграция систем контроля версий в проектную инфраструктуру.
28. Сценарии решения типичных задач промышленной разработки с использованием систем контроля версий.
29. Задачи систем автоматической сборки в промышленной разработке.
30. Возможности систем автоматической сборки.
31. Понятие непрерывной интеграции, её преимущества и недостатки, область применения.
32. Инструменты непрерывной интеграции и сценарии их использования в промышленной разработке.
33. Назначение процедуры профилирования кода и её область применения.
34. Принципы работы профайлеров.
35. Анализ профиля и техники оптимизации, основанные на нём.
36. Назначение процедуры статического анализа кода и её область применения.
37. Возможности существующих статических анализаторов.
38. Роль статического анализа в промышленной разработке.
39. Задача управления конфигурацией и её актуальность в промышленной разработке.
40. Проблема воспроизводимости процесса и подходы к её решению.
41. Роль управления конфигурации в управлении проектом.
42. Инструменты управления конфигурацией и основные сценарии их применения.
43. Жизненный цикл программного кода в промышленной разработке.
44. Связь читаемости кода и его сопровождаемости.
45. Роль корпоративных стандартов на код в промышленной разработке.
46. Методы и приёмы обеспечения читаемости и сопровождаемости кода.
47. Инструментальные средства проверки кода на соответствие стандартам.
48. Задачи документирования программного обеспечения.
49. Инструментальные средства управления знаниями (howto, wiki и т. п.).
50. Требования предъявляемые к документации.
51. Инструменты для автоматической генерации документации.
52. Организация взаимодействия с пользователями и ресурсы, необходимые для неё.
53. Техническая поддержка по электронной почте и через call-центры.
54. Инструменты автоматизации технической поддержки.

Вопросы 1-27 проверяют уровень сформированности компетенции ПК-11, вопросы 28-54 проверяют уровень сформированности компетенции ПК-21.

Пример зачетного теста:

1. Выберите из перечисленного характеристики требований к программному обеспечению:

- **атомарность**
- **проверяемость**
- сопровождаемость
- интероперабельность
- функциональность

2. User experience (XP) — это

- процесс разработки интерфейса, на каждом этапе которого существенное внимание уделяется потребностям и ограничениям конечных пользователей продукта
- **желаемый, ожидаемый или фактический опыт пользователя, взаимодействующего с продуктом**
- способ взаимодействия пользователя и программной системы

- черновой прототип пользовательского интерфейса

3. Выберите из перечисленного методологии разработки программного обеспечения

- волновая методология
- методология «большого взрыва»
- **каскадная методология**
- **итеративная методология**

4. В методологии Scrum выделяются следующие формальные события инспекции и адаптации:

- **обзор спринта**
- заполнение трекера задач
- перемена ролей разработчиков
- **ежедневное совещание**

5. Бэклог спринта включает в себя

- все решённые за время спринта задачи
- **все задачи, которые надо решить за время спринта**
- все задачи, не были решены за время спринта
- отложенные задачи

6. Трекер задач — это приложение, предназначенное для

- хранения исходных текстов программ
- **слежения за деятельностью за разработчиком в реальном времени**
- управления и контроля за задачами и их выполнением
- разработки модульных и интеграционных тестов приложения

7. Выберите из перечисленного распределённые системы управления версиями.

- cvs
- subversion
- **mercurial**
- **git**
- nslookup

8. Скрипты автоматизации сборки могут быть предназначены для

- выявления требований пользователей
- **компиляции исходного кода в бинарный код**
- генерации отчётов о времени, затраченном разработчиками на решение задачами
- **выполнения тестов**

9. Выберите из перечисленного средства автоматизации сборки с функциями управления зависимостей

- make
- ant
- **maven**
- **gradle**

10. Благодаря использованию профилировщика можно решать следующие задачи:

- авторизация пользователей в приложении
- **определение времени выполнения каждого метода**

- **выявление дубликатов SQL-запросов**
- автоматическое выполнение тестов

11. Выделяются следующие типы сопровождения:

- **совершенствующее**
- **адаптирующее**
- исследовательское
- **неразрушающее**

12. Отличительной особенностью непрерывной интеграции является

- откладывание тестирования до самого последнего этапа проекта
- отправка отчётов о тестировании по электронной почте
- вывод программного кода на экран в процессе его генерации
- **максимально частое выполнение автоматизированной сборки проекта**

Правильные ответы выделены жирным шрифтом.

Критерии оценивания:

За каждый выбранный правильный ответ начисляется количество баллов равное единице, делённое на общее количество правильных ответов на вопрос. За каждый выбранный неправильный ответ у студента снимается количество баллов, равное 0.5 балла. Если суммарный балл студента за вопрос оказывается отрицательным, вместо него засчитывается 0.

Баллы	Оценка	Степень сформированности ПК-11	Степень сформированности ПК-21
0-3	Неудовлетворительно	не сформирована	не сформирована
4-6	Удовлетворительно	пороговый уровень	пороговый уровень
7-9	Хорошо	продвинутый уровень	продвинутый уровень
10-12	Отлично	высокий уровень	высокий уровень

Методические указания по выставлению зачета

Зачет выставляется по результатам выполнения всех контрольных работ и зачетного теста на оценку не ниже удовлетворительно.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ПК-11	Контрольная работа. Тест.	1-13	<p>Знать:</p> <ul style="list-style-type: none"> – роль требований к программному обеспечению в промышленной разработке; – инструменты разработки программного обеспечения; <p>Уметь:</p> <ul style="list-style-type: none"> – реализовывать этапы разработки программного обеспечения; – управлять организацией этапов разработки программного обеспечения; <p>Владеть навыками:</p> <ul style="list-style-type: none"> – работы с инструментами проектирования и 	<p>1. Понимать роль требований в процессе промышленной разработки.</p> <p>2. Выполнять проектирование пользовательских интерфейсов. Знать этапы и методы разработки пользовательского интерфейса. Использовать инструментальные средства проектирования интерфейсов.</p> <p>3. Знать и использовать инструменты автоматической сборки и непрерывной</p>	<p>1. Уметь определять требования к ПО. Понимать роль требований в процессе промышленной разработки. Работать с различными источниками и путями выявления требований. Использовать инструментальные средства управления требованиями.</p> <p>2. Выполнять проектирование пользовательских интерфейсов. Знать этапы и методы разработки пользовательского интерфейса. Использовать инструментальные средства проектирования интерфейсов, инструментальные средства тестирования интерфейсов.</p> <p>3. Знать и использовать инструменты автоматической</p>	<p>1. Уметь определять требования к ПО. Понимать роль требований в процессе промышленной разработки. Работать с различными источниками и путями выявления требований. Осуществлять анализ и тестирование требований. Использовать инструментальные средства управления требованиями.</p> <p>2. Выполнять проектирование пользовательских интерфейсов. Знать этапы и методы разработки пользовательского интерфейса. Использовать инструментальные средства проектирования интерфейсов, инструментальные средства тестирования интерфейсов.</p> <p>3. Знать и использовать инструменты автоматической</p>

			разработки программного обеспечения.	интеграции в промышленной разработке. 4. Понимать связь читаемости кода и его сопровождаемости. 5. Уметь организовывать взаимодействие с пользователями.	проектирования интерфейсов. 3. Знать и использовать инструменты автоматической сборки и непрерывной интеграции в промышленной разработке. 4. Понимать связь читаемости кода и его сопровождаемости. Знать роль корпоративных стандартов на код в промышленной разработке. 5. Уметь организовывать взаимодействие с пользователями.	сборки и непрерывной интеграции в промышленной разработке и инструменты профилирования и статического анализа кода в промышленной разработке. 4. Понимать связь читаемости кода и его сопровождаемости. Знать роль корпоративных стандартов на код в промышленной разработке. Использовать приёмы обеспечения читаемости и сопровождаемости кода и инструментальные средства проверки кода на соответствие стандартам. 5. Уметь организовывать взаимодействие с пользователями. Знать инструменты автоматизации технической поддержки.
ПК-21	Контрольная работа. Тест.	1-13	Знать: – методологию разработки программного обеспечения; – инструменты разработки программного обеспечения; Уметь:	1. Знать возможности и область применимости гибкой методологии. 2. Знать и применять методики оценки трудозатрат на выполнение конкретных задач 3. Использовать систему управления	1. Знать возможности и область применимости гибкой методологии. 2. Уметь планировать выпуск ПО. Знать и применять методики оценки трудозатрат на выполнение конкретных задач 3. Использовать систему управления	1. Знать возможности и область применимости гибкой методологии. Использовать отдельные ее методики в рамках гибкой методологии: XP, Scrum, Kanban, Lean. 2. Уметь планировать выпуск ПО. Знать и применять методики оценки трудозатрат

			<p>оценивать этапы разработки программного обеспечения;</p> <p>применять корпоративные стандарты разработки программного обеспечения</p> <p>Владеть навыками:</p> <p>1. работы с инструментами для управления разработкой программного обеспечения.</p>	<p>версиями в промышленной разработке.</p> <p>4. Знать и использовать инструменты автоматической сборки и непрерывной интеграции в промышленной разработке.</p> <p>6. Понимать связь читаемости кода и его сопровождаемости. Знать роль корпоративных стандартов на код в промышленной разработке.</p> <p>7. Уметь организовывать взаимодействие с пользователями.</p>	<p>версиями в промышленной разработке. Знать сценарии решения типичных задач промышленной разработки с использованием систем контроля версий.</p> <p>4. Знать и использовать инструменты автоматической сборки и непрерывной интеграции в промышленной разработке.</p> <p>5. Понимать связь читаемости кода и его сопровождаемости. Знать роль корпоративных стандартов на код в промышленной разработке.</p> <p>6. Уметь организовывать взаимодействие с пользователями.</p>	<p>на выполнение конкретных задач</p> <p>3. Использовать разные системы управления версиями в промышленной разработке. Знать сценарии решения типичных задач промышленной разработки с использованием систем контроля версий.</p> <p>4. Знать и использовать инструменты автоматической сборки и непрерывной интеграции в промышленной разработке и инструменты профилирования и статического анализа кода в промышленной разработке.</p> <p>5. Понимать связь читаемости кода и его сопровождаемости. Знать роль корпоративных стандартов на код в промышленной разработке. Использовать приёмы обеспечения читаемости и сопровождаемости кода и инструментальные средства проверки кода на соответствие стандартам.</p> <p>6. Уметь организовывать взаимодействие с пользователями. Знать</p>
--	--	--	---	--	---	---

						инструменты автоматизации технической поддержки.
--	--	--	--	--	--	--

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено», «незачтено».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за контрольные работы и тест.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за контрольные работы и тест.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за контрольные работы и тест.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Основы современной промышленной разработки программного обеспечения»

Методические указания для студентов по освоению дисциплины

Основными видами учебных занятий по курсу «Основы современной промышленной разработки программного обеспечения» являются лекции и семинары.

В рамках лекций предполагается максимально уйти от репродуктивного стиля обучения и широко применять интерактивные элементы. В частности, предполагается использовать технику проблемных вопросов и диалогичность, позволяя студентам самостоятельно воссоздавать новое знание, а не пассивно воспринимать уже подготовленную информацию. Данный подход выглядит для рассматриваемого курса особенно оправданным в силу специфики предмета, выраженной в множестве возможных точек зрения на объекты его предметной области и необходимости постоянного нахождения компромиссов в ходе практической деятельности участников процесса разработки.

В рамках семинаров предполагается рассмотрение проблемных ситуаций, специально разработанных в рамках проекта на основе анализа процессов индустрии программного обеспечения. Другим полезным в рамках разрабатываемого курса подходом может быть мозговой штурм в командах с последующим обсуждением результатов. Такие методы позволят не только сориентировать студентов на восприятие нового материала, но также помогут им в построении логических цепочек между изучаемыми техниками и процессами.

Для закрепления знаний предполагается активно использовать практику выполнения контрольных работ оценкой студентами работ других участников по заранее подготовленному списку критериев. Также возможно финальное обсуждение предложенных решений для обсуждения границ их применимости и типичных ошибок.

Особенностью предлагаемого курса является использование интерактивных форм обучения, которые подразумевают активное участие со стороны студентов. На основании проявленного ими интереса и степени их вовлечённости в рассматриваемые ситуации можно сделать выводы о доступности предлагаемого материала: его сложности и понимании контекста студентами. Помимо внешней оценки, проводимой преподавателям, студентам также будет дополнительно предложено самостоятельно провести оценку, дать советы по улучшению по улучшению материала, например, в рамках обсуждения результатов выполнения заданий.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ

(http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека «[Книгообеспеченность](#)»

(http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «[Книгообеспеченность](#)» доступна в сети университета и через Личный кабинет.