

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
Ярославский государственный университет им. П.Г.Демидова

УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ

Промышленная разработка

Направление подготовки (специальность):
09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Образовательная программа
Искусственный интеллект в корпоративных информационных системах

очная форма обучения

Составитель:
Парамонов И.В., к.ф.-м.н., доцент

г. Ярославль

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

Основная литература:

1. 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. / И. В. Парамонов; Яросл. гос. ун-т им. П. Г. Демидова, Науч.-метод. совет ун-та - 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.uniyar.ac.ru/edocs/iuni/20150402.pdf>

Дополнительная литература:

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.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

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

**Учебно-методические указания и рекомендации
к изучению тем лекционных и практических занятий, самостоятельной
работе студентов**

Очная форма обучения

Лекции

№	Тема лекции	Количество часов
1.	Извлечение требований к ПО.	1
2.	Управление требованиями к ПО.	1
3.	Проектирование пользовательских интерфейсов.	1
4.	Гибкая методология разработки ПО.	1
5.	Планирование выпусков ПО.	1
6.	Использование трекера задач.	1
7.	Использование систем управления версиями	1
8.	Инструменты автоматической сборки и непрерывной интеграции	1
9.	Инструменты профилирования и статического анализа кода	1
10.	Проблемы управления конфигурацией и инструменты для их решения.	1
11.	Читаемость и сопровождаемость программного кода.	1
12.	Управление знаниями в процессе промышленной разработки и задачи документирования программного обеспечения.	1
	Итого:	12

Практические занятия

№	Тема практического занятия	Количество часов
1	Извлечение требований к ПО.	2
2	Управление требованиями к ПО.	2
3	Проектирование пользовательских интерфейсов.	2
4	Гибкая методология разработки ПО.	2

№	Тема практического занятия	Количество часов
5	Планирование выпусков ПО.	2
6	Использование трекера задач.	2
7	Использование систем управления версиями	2
8	Инструменты автоматической сборки и непрерывной интеграции	2
9	Инструменты профилирования и статического анализа кода	2
10	Проблемы управления конфигурацией и инструменты для их решения.	2
11	Читаемость и сопровождаемость программного кода.	2
12	Управление знаниями в процессе промышленной разработки и задачи документирования программного обеспечения.	2
	Итого:	24

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.

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.

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

Методические указания по выставлению зачета

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