

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Институт компьютерных систем и информационных технологий
Кафедра компьютерных систем и сетей



Кочевский А. А.

20 25 года

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

«Технология разработки программного обеспечения»
09.04.01 Информатика и вычислительная техника
«Сети ЭВМ и телекоммуникации»

Разработчик:
ст. преп. М. В. Гарагуля Гарагуля Н.В.
(подпись)

ФОС рассмотрен и одобрен на заседании кафедры
компьютерных систем и сетей

от « 10 » 03 20 25 г., протокол № 8

Заведующий кафедрой С. В. Попов Попов С. В.
(подпись)

Луганск 2025 г.

**Комплект оценочных материалов по дисциплине
«Технология разработки программного обеспечения»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

Выберите один правильный ответ

1. Что означает термин 'рефакторинг' в разработке программного обеспечения?

- А) Добавление новых функций
- Б) Повышение производительности системы
- В) Улучшение структуры кода без изменения его поведения
- Г) Исправление критических ошибок

Правильный ответ: Б

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

2. Какой из методов используется для оценки трудозатрат при разработке ПО?

- А) Scrum
- Б) Function Point Analysis
- В) Kanban
- Г) CI/CD

Правильный ответ: Б

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

3. Что такое SRS-документ?

- А) Документ, описывающий тестовые сценарии
- Б) Исходный код приложения
- В) Спецификация требований к программному обеспечению
- Г) Отчёт о проверке качества

Правильный ответ: В

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

4. Какой шаблон проектирования относится к порождающим?

- А) Команда (Command)
- Б) Посредник (Mediator)

В) Строитель (Builder)

Г) Цепочка обязанностей (Chain of Responsibility)

Правильный ответ: В

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

Задания закрытого типа на установление соответствия

1. Установите соответствие между принципами SOLID и их описанием. Каждому элементу левого столбца соответствует только один элемент правого столбца.

SOLID		Описание	
1) S - Single Responsibility Principle	A)	Подтипы должны быть взаимозаменяемы с базовыми типами	
2) O - Open/Closed Principle	Б)	Класс должен иметь одну причину для изменения.	
3) L - Liskov Substitution Principle	В)	Подтипы должны быть взаимозаменяемы с базовыми типами	
4) I - Interface Segregation Principle	Г)	Классы должны быть открыты для расширения, но закрыты для модификации	

Правильный ответ:

1	2	3	4
Б	Г	А	В

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

2. Установите соответствие между этапами тестирования и их назначением. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Этапы тестирования		Назначения	
1) Unit Test	А)	Проверка отдельных функций/методов	
2) Integration Test	Б)	Проверка всей системы как единого целого	
3) System Test	В)	Проверка взаимодействия между модулями	
4) Acceptance Test	Г)	Проверка соответствия требованиям заказчика	

Правильный ответ:

1	2	3	4
A	B	B	Г

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

3. Установите соответствие между типами документации и их назначением. Каждому элементу левого столбца соответствует только один элемент правого столбца.

тип		Название	
1)	Пользовательская	A)	Описывает общую структуру системы
2)	Архитектурная	B)	Представляет реализацию логики приложения
3)	Исходный код	B)	Содержит формальные требования к системе
4)	SRS	Г)	Объясняет, как использовать программу

Правильный ответ:

1	2	3	4
Г	A	B	B

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

Задания закрытого типа на установление правильной последовательности

1. Упорядочите этапы процесса анализа требований:

A. Документирование

B. Утверждение

B. Сбор требований

Г. Анализ требований

Правильный порядок: B, Г, A, B

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

2. Расположите шаги процесса разработки программы:

A. Внедрение

B. Кодирование

B. Тестирование

Г. Проектирование

Правильный порядок: Г, B, B, A

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

3. Упорядочите этапы работы с системой баг-трекинга:

А. Проверка и закрытие

Б. Создание задачи

В. Исправление ошибки

Г. Назначение исполнителя

Правильный порядок: Б, Г, В, А

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

4. Расположите фазы тест-дизайна:

А. Анализ требований

Б. Разработка тест-кейсов

В. Определение условий тестирования

Г. Подготовка тестовых данных

Правильный порядок: А, В, Б, Г

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

Задания открытого типа

Задания открытого типа на дополнение

1. Напишите пропущенное слово (словосочетание).

Модель разработки ПО, в которой каждый этап выполняется последовательно, называется _____.

Ответ: каскадная модель

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

2. Основной документ, описывающий функциональные и нефункциональные требования к системе, называется _____.

Ответ: спецификация требований (SRS)

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

3. Напишите пропущенное слово (словосочетание).

Методология гибкой разработки программного обеспечения называется _____.

Ответ: Agile

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

4. Напишите пропущенное слово (словосочетание).

Процесс улучшения внутренней структуры кода без изменения его внешнего поведения называется _____.

Ответ: рефакторинг

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

5. Напишите пропущенное слово (словосочетание).

UML-диаграмма, отображающая поведение объектов во времени, называется _____.

Ответ: диаграмма последовательностей

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

6. Напишите пропущенное слово (словосочетание).

Основной принцип DevOps — это автоматизация и _____ между командами.

Ответ: внедрение

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

Задания открытого типа с кратким свободным ответом

1. Модель, в которой разработка программного обеспечения выполняется итерациями, называется моделью _____.

Варианты ответа: итеративная / инкрементная / итеративно-инкрементная.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

2. Процесс выявления, документирования и анализа требований называется _____.

Варианты ответа: анализ требований / сбор требований / requirements analysis.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

3. Система управления версиями, наиболее популярная на сегодняшний день — это _____.

Варианты ответа: Git, / git.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

4. Методология, в которой ключевыми принципами являются гибкость, взаимодействие и быстрая поставка продукта — это _____.

Варианты ответа: Agile / Аджайл / Аджайл.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

5. Какой язык используется для визуализации систем?

Варианты ответа: UML / Диаграмма.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

6. Какой элемент UML описывает поведение системы?

Варианты ответа: Кейс/ Сценарий.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

Задания открытого типа с развернутым ответом

1. Опишите процесс управления требованиями в разработке программного обеспечения. Укажите ключевые этапы, участников, возможные трудности и способы их преодоления. Объясните, почему эффективное управление требованиями критически важно для успеха проекта.

Время выполнения – 50 мин.

Ожидаемый результат:

Управление требованиями — это систематический процесс выявления, документирования, анализа, отслеживания, приоритизации и контроля изменений требований на протяжении всего жизненного цикла программного обеспечения. Этот процесс является фундаментом успешного проекта, так как требования определяют, что именно должно быть реализовано в продукте. Процесс начинается с сбора требований, когда аналитики, заказчики, пользователи и другие заинтересованные стороны взаимодействуют для формулировки потребностей и ожиданий. На этом этапе применяются методы интервью, опросов, мозговых штурмов, анализа существующих систем. Важно не только понять, что нужно сделать, но и почему это нужно. Следующий этап — анализ и спецификация требований. Аналитики структурируют собранную информацию, устраняют противоречия, классифицируют требования (функциональные, нефункциональные, системные, пользовательские и т.д.) и подготавливают документ спецификации требований (SRS). Этот документ должен быть понятным, однозначным и проверяемым.

Затем идет верификация и утверждение требований: заказчик и другие заинтересованные стороны проверяют требования на полноту и точность. Это критически важный этап, так как после него начинается проектирование и разработка.

Кроме этого, важно организовать отслеживаемость требований, то есть возможность связать каждое требование с этапами проектирования, реализации, тестирования и внедрения. Это позволяет убедиться, что ни одно требование не потеряно и реализовано должным образом.

Управление требованиями обязательно включает управление изменениями. В ходе проекта могут появляться новые требования или меняться

старые. Для этого создаются процессы анализа влияния изменений, процедуры согласования и обновления документации.

Участниками процесса управления требованиями обычно являются:

- Бизнес-аналитики — главные посредники между заказчиком и командой.
- Разработчики — нуждаются в чётком понимании требований для написания кода.
- Тестировщики — используют требования как основу для написания тест-кейсов.
- Заказчики и конечные пользователи — формулируют ожидания от продукта.
- Менеджеры проекта — следят за соблюдением сроков и бюджета, опираясь на требования.

Основные трудности при управлении требованиями:- Неясные или неполные требования

- Частые изменения и "ползучесть" функционала (scope creep)
 - Недостаточное взаимодействие между заказчиком и командой
 - Противоречия между требованиями разных заинтересованных сторон
- Для преодоления этих проблем применяются:

- Использование шаблонов и стандартов описания требований (например, IEEE 830);
- Активное участие заказчика на всех этапах;
- Создание прототипов и пользовательских историй;
- Ведение единого хранилища требований с системой версионирования;
- Применение инструментов (Jira, Confluence, IBM DOORS и др.).

Эффективное управление требованиями снижает риски неудачи проекта, позволяет точно оценить трудозатраты и сроки, улучшает качество конечного продукта и повышает удовлетворенность заказчика. Без чёткого процесса управления требованиями команда может реализовать "не тот продукт" или пропустить важные детали, что приведёт к переработке и увеличению затрат.

Критерии оценивания:

- Управление требованиями — основа успешной разработки ПО.
- Этапы: сбор, анализ, спецификация, верификация, отслеживание, управление изменениями.
- Участники: бизнес-аналитик, заказчик, разработчик, тестировщик, менеджер проекта.
- Трудности: неясные/противоречивые требования, частые изменения, слабое взаимодействие.
- Решения: прототипирование, регулярные встречи, единый репозиторий требований, инструменты (Jira и др.).
- Без правильного управления требованиями — рост сроков, стоимости и снижение качества продукта.

Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

2. Опишите принципы и практики гибких методологий разработки программного обеспечения (Agile). Приведите примеры их применения и объясните, как Agile помогает повысить гибкость и эффективность командной работы.

Время выполнения – 40 мин.

Ожидаемый результат:

Agile — это совокупность гибких методологий, основанных на Манифесте Agile, который появился в 2001 году. В его основе — 4 ценности и 12 принципов, ориентированных на быстрое реагирование на изменения, тесное взаимодействие с заказчиком и частые поставки работающего программного обеспечения.

Основные ценности Agile:

1. Люди и взаимодействие важнее процессов и инструментов.
2. Работающее программное обеспечение важнее исчерпывающей документации.
3. Сотрудничество с заказчиком важнее согласования условий контракта.
4. Реагирование на изменения важнее следования первоначальному плану.

На практике эти ценности реализуются через методологии, такие как Scrum, Kanban, Extreme Programming (XP), Lean и другие. Самая популярная среди них — Scrum.

В Scrum команда делится на роли: Scrum-мастер, владелец продукта (Product Owner) и разработчики. Работа делится на короткие итерации — спринты (обычно 1–4 недели). В начале каждого спринта проводится планирование, где выбираются задачи из бэклога продукта, формируется бэклог спринта, и команда определяет цели. Каждый день проходит daily scrum — короткое совещание, где участники делятся статусом и препятствиями.

По окончании спринта проходит демонстрация (review) готового инкремента и ретроспектива, на которой команда анализирует, что можно улучшить в процессе. Такой цикл повторяется до завершения проекта.

Agile предполагает непрерывную поставку ценности. Заказчик получает рабочую версию продукта каждые несколько недель, а не через несколько месяцев, как в классических моделях. Это даёт возможность быстро вносить изменения, ориентируясь на реальную обратную связь.

Agile усиливает гибкость командной работы:

- Все члены команды активно участвуют в планировании и обсуждении задач.
- Каждый имеет представление о текущем статусе проекта.
- Коммуникация прозрачна и регулярна.
- Ошибки выявляются быстро и устраняются по мере возникновения.

Преимущества Agile:

- Быстрая реакция на изменения требований
- Повышенная вовлеченность заказчика
- Частые релизы, позволяющие быстрее получать обратную связь
- Улучшенное качество благодаря постоянному тестированию и интеграции
- Увеличенная мотивация команды за счёт автономии и самоорганизации

Примеры применения Agile:

- В стартапах, где бизнес-цели часто меняются
- В мобильной и веб-разработке с частыми обновлениями
- В разработке пользовательского ПО, где важна обратная связь от пользователей.

Однако Agile требует высокой дисциплины, хорошей коммуникации и доверия между участниками. Без этих условий методология может привести к хаосу. Также Agile не всегда подходит для проектов с жёсткими регламентами или фиксированными требованиями, например, в оборонной или медицинской сфере.

В целом, Agile — это не просто набор методик, а мышление, ориентированное на результат, ценность и постоянное улучшение. Именно это делает его мощным инструментом современной разработки ПО.

Критерии оценивания:

- Agile — гибкий подход, основанный на ценностях и принципах Agile-манифеста.
 - Важны: взаимодействие, быстрая поставка, готовность к изменениям.
 - Методики: Scrum, Kanban, XP. Самая популярная — Scrum.
 - Scrum включает роли (РО, Scrum-мастер, команда), спринты, daily meetings, review, ретроспективы.
 - Повышается прозрачность, скорость и качество.
 - Применим в стартапах, веб-разработке, проектах с быстро меняющимися требованиями.
 - Требуется вовлеченности, самоорганизации и доверия.
 - Agile — это культура и мышление, а не только набор техник.
- Компетенции (индикаторы): УК-2, ОПК-5, ОПК-8, ПК-5

Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по дисциплине «Технология разработки программного обеспечения» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки: 09.04.01 Информатика и вычислительная техника.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии института компьютерных
систем и информационных технологий



Ветрова Н.Н.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)