

**Комплект оценочных материалов по дисциплине
«Методология программной инженерии»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. Выберите один правильный ответ.

Какой стандарт определяет жизненный цикл разработки программного обеспечения?

- А) ISO 27001
- Б) IEEE 829
- В) ISO/IEC 12207 (Software Life Cycle Processes)
- Г) CMMI (Capability Maturity Model Integration)

Правильный ответ: В

Компетенции (индикаторы): УК-3, ОПК-3

2. Выберите один правильный ответ.

Какой метод оценки стоимости разработки ПО основан на измерении функциональных точек?

- А) COCOMO (Constructive Cost Model)
- Б) PERT (Program Evaluation and Review Technique)
- В) Function Point Analysis (FPA)
- Г) Gantt Chart

Правильный ответ: В

Компетенции (индикаторы): УК-3, ОПК-3

3. Выберите один правильный ответ.

Какой метод используется для управления версиями программного кода в командной разработке?

- А) CI/CD (Continuous Integration / Continuous Deployment)
- Б) Git Flow
- В) UML (Unified Modeling Language)
- Г) SCRUM

Правильный ответ: Б

Компетенции (индикаторы): УК-3, ОПК-3

4. Выберите один правильный ответ.

Какой принцип подразумевает минимизацию изменений в процессе разработки?

- А) Принцип итеративной разработки
- Б) Принцип адаптивного проектирования
- В) Принцип KISS (Keep It Simple, Stupid)
- Г) Принцип Lean

Правильный ответ: В

Компетенции (индикаторы): УК-3, ОПК-3

Задания закрытого типа на установление соответствия

1. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Модели разработки ПО		Характеристики	
1) Waterfall Model (Каскадная модель)	А)	Гибкая методология, использующая итерации с быстрыми релизами	
2) Spiral Model (Спиральная модель)	Б)	Линейная последовательность этапов, строгая фиксация требований	
3) Incremental Model (Инкрементная модель)	В)	Разработка выполняется поэтапно с добавлением новых функций	
4) Agile Model (Agile-модель)	Г)	Ориентирована на управление рисками, сочетает каскадный и итерационный подход	

Правильный ответ:

1	2	3	4
Б	Г	В	А

Компетенции (индикаторы): УК-3, ОПК-3

2. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Методологии управления проектами разработки ПО		Особенности	
1) SCRUM	А)	Управление процессами с фиксированными спринтами и ролями команды	
2) Kanban	Б)	Минимизация потерь, оптимизация ресурсов, непрерывное улучшение	
3) Waterfall	В)	Визуализация задач в виде доски, непрерывный поток задач	
4) Lean	Г)	Жесткая последовательность этапов, документация важнее итераций	

Правильный ответ:

1	2	3	4
А	В	Г	Б

Компетенции (индикаторы): УК-3, ОПК-3

3. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Стандарты программной инженерии

Назначение

- | | | | |
|----|---------------|----|--|
| 1) | ISO/IEC 12207 | А) | Стандарт требований к документации (Software Requirements Specification) |
| 2) | IEEE 830 | Б) | Определяет процессы жизненного цикла разработки ПО |
| 3) | CMMI | В) | Модель зрелости процессов разработки ПО |
| 4) | ISO 9001 | Г) | Общий стандарт управления качеством продукции |

Правильный ответ:

1	2	3	4
Б	А	В	Г

Компетенции (индикаторы): УК-3, ОПК-3

Задания закрытого типа на установление правильной последовательности

1. Расположите этапы планирования проекта разработки ПО в Agile (SCRUM) в правильном порядке:

- А) Обзор спринта (Sprint Review)
- Б) Ежедневные встречи (Daily Stand-up)
- В) Планирование спринта (Sprint Planning)
- Г) Ретроспектива спринта (Sprint Retrospective)
- Д) Выполнение задач спринта (Sprint Execution)

Правильный ответ: В, Б, Д, А, Г

Компетенции (индикаторы): УК-3, ОПК-3

2. Расположите шаги по оценке стоимости разработки ПО методом COSOMO в правильном порядке:

- А) Определение размера проекта в KLOC (тысячах строк кода)
- Б) Выбор модели оценки (Basic, Intermediate, Detailed)
- В) Расчет стоимости разработки
- Г) Анализ факторов сложности проекта

Правильный ответ: А, Б, Г, В

Компетенции (индикаторы): УК-3, ОПК-3

3. Установите правильную последовательность анализа кода в процессе Code Review:

- А) Автоматическая проверка качества кода (Static Code Analysis)
- Б) Ручная проверка кода (Manual Code Review)
- В) Исправление замечаний (Bug Fixing)

Г) Подтверждение изменений и мердж в основную ветку (Merge & Approve)

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): УК-3, ОПК-3

Задания открытого типа

Задания открытого типа на дополнение

1. Напишите пропущенное слово.

_____ – это методология гибкой разработки, предполагающая работу в спринтах

Правильный ответ: SCRUM.

Компетенции (индикаторы): УК-3, ОПК-3

2. Напишите пропущенное слово.

_____ – это процесс создания предварительной версии системы или её компонентов для визуализации идей и получения обратной связи от пользователей

Правильный ответ: Прототипирование

Компетенции (индикаторы): УК-3, ОПК-3

3. Напишите пропущенное слово.

Методология управления требованиями, определяющая приоритет задач с категорией "Must have", "Should have", "Could have", "Won't have", называется _____.

Правильный ответ: MoSCoW

Компетенции (индикаторы): УК-3, ОПК-3

4. Напишите пропущенное слово.

_____ – это алгоритмическая модель для расчета трудозатрат и стоимости разработки программного обеспечения, учитывающая различные параметры проекта, такие как размер кода, сложность, опыт команды и другие, для расчета трудозатрат и стоимости разработки ПО..

Правильный ответ: COCOMO.

Компетенции: УК-3, ОПК-3

5. Напишите пропущенное слово.

Принцип _____ в программной инженерии направлен на оптимизацию процессов и снижение уровня затрат в разработке ПО.

Правильный ответ: Lean.

Компетенции (индикаторы): УК-3, ОПК-3

6. Напишите пропущенное словосочетание.

_____ – это вид анализа, который позволяет определять и минимизировать риски проекта.

Правильный ответ: Анализ рисков.

Компетенции (индикаторы): УК-3, ОПК-3

7. Напишите пропущенное слово (словосочетание).

_____ – это методология, направленная на автоматизацию процессов сборки, настройки и развёртывания программного обеспечения, которая объединяет разработчиков и специалистов по IT-обслуживанию, способствуя тесному взаимодействию и интеграции их процессов для достижения высокого качества программного продукта.

Правильный ответ: DevOps.

Компетенции (индикаторы): УК-3, ОПК-3

8. Напишите пропущенное словосочетание.

_____ – это подход, который помогает в управлении зависимостями между компонентами системы.

Правильный ответ: Управление конфигурациями

Компетенции (индикаторы): УК-3, ОПК-3

Задания открытого типа с кратким свободным ответом

1. Напишите пропущенное слово (словосочетание).

_____ - это процесс управления данными от их создания или приобретения до удаления или архивации, который включает в себя систематическую и эффективную организацию, хранение, обработку, анализ и удаление данных?

Правильный ответ: Управление жизненным циклом данных/ Data Lifecycle Management / DLM .

Компетенции (индикаторы): УК-3, ОПК-3.

2. Напишите пропущенное слово (словосочетание).

_____ - это подход в разработке программного обеспечения, который фокусируется на использовании моделей в качестве основных артефактов на протяжении всего процесса, а также позволяет автоматизировать различные инженерные задачи, повышая продуктивность и уменьшая ошибки.

Правильный ответ: Model-Driven Engineering / MDE / Подход Model-Driven Engineering / Подход MDE .

Компетенции (индикаторы): УК-3, ОПК-3

3. Напишите пропущенное словосочетание.

_____ – это процесс создания абстрактных моделей программной системы, отображающих различные её аспекты

Правильный ответ: Программное моделирование / Моделирование ПО / Моделирование программного обеспечения .

Компетенции (индикаторы): УК-3, ОПК-3

4. Какой термин обозначает анализ требований в контексте пользовательского опыта?

Правильный ответ: UX-анализ / User Experience analysis / Анализ UX

Компетенции (индикаторы): УК-3, ОПК-3

5. Напишите пропущенное слово (словосочетание).

_____ — это технология автоматизации тестирования и доставки новых модулей разрабатываемого проекта заинтересованным сторонам: разработчикам, аналитикам, инженерам качества, конечным пользователям и другим

Правильный ответ: CI/CD / CICD /Continuous Integration, Continuous Delivery

Компетенции (индикаторы): УК-3, ОПК-3

Задания открытого типа с развернутым ответом

1. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: Какие ключевые аспекты Agile и DevOps дополняют друг друга в процессе разработки ПО?

Время выполнения: 35 минут

Ожидаемый результат:

Agile – это гибкая методология управления разработкой ПО, ориентированная на итеративную разработку, адаптацию к изменениям и взаимодействие с заказчиком.

Основные принципы:

- разработка идет небольшими итерациями (Sprints);
- команда регулярно получает обратную связь и вносит изменения;
- основное внимание уделяется ценности, которую получает пользователь.

DevOps – это культура и набор практик, объединяющий процессы разработки (Dev) и эксплуатации (Ops), чтобы ускорить развертывание, автоматизировать тестирование и мониторинг системы.

Основные принципы:

- использование CI/CD (Continuous Integration / Continuous Deployment) для автоматизированного развертывания и тестирования;
- контейнеризация (например, Docker, Kubernetes) для ускорения доставки ПО;

- мониторинг и логирование в реальном времени.

Agile и DevOps взаимодействуют следующим образом:

- гибкость и скорость: Agile позволяет быстро менять требования, а DevOps автоматизирует доставку обновлений через CI/CD;
- короткие циклы разработки: Agile использует спринты (1-2 недели), а DevOps обеспечивает их быструю интеграцию в продакшен;
- непрерывное тестирование: Agile требует постоянной проверки качества, а DevOps внедряет автоматизированные тесты;
- совместная работа: Agile помогает разработчикам и тестировщикам работать слаженно, а DevOps объединяет их с IT-операторами;
- быстрая обратная связь: Agile использует ретроспективы для анализа работы, а DevOps – мониторинг и логи для выявления проблем.

Пример (краткий) организации взаимодействия Agile и DevOps на практике.

Компания использует Scrum для управления задачами и Jenkins для CI/CD. Это позволяет автоматически тестировать и разворачивать новые версии без простоев.

Развернутый пример: Организация взаимодействия Agile и DevOps на практике.

Компания разрабатывает веб-приложение для электронной коммерции, которое должно поддерживать высокую нагрузку, частые обновления и бесперебойную работу. Для управления разработкой используется Scrum (Agile), а для автоматизации тестирования и развертывания – Jenkins (DevOps).

Как Agile и DevOps работают вместе:

1) Планирование и спринты (Agile):

- команда разработки собирает требования клиентов и фиксирует их в беклоге задач;
- разработчики и тестировщики работают в 2-недельных спринтах, добавляя новые функции.

2) Разработка и интеграция кода (DevOps + Agile):

- после завершения задачи разработчик коммитит код в репозиторий (Git);
- DevOps-инженеры настроили Jenkins, который автоматически проверяет код на ошибки и запускает юнит-тесты.

3) Автоматизированное тестирование (DevOps):

- если код проходит юнит-тестирование, запускаются интеграционные тесты в изолированной среде;
- используются Selenium и JMeter для тестирования UI и производительности.

4) Деплой и обратная связь (DevOps + Agile):

- после успешных тестов Jenkins автоматически разворачивает новую версию приложения на тестовом сервере;
- команда QA проводит ручное тестирование новых функций;

- если тесты пройдены, обновление автоматически разворачивается на продакшене через Kubernetes.

5) Мониторинг и ретроспективы (Agile + DevOps):

- DevOps-инженеры используют Prometheus и Grafana для мониторинга производительности приложения;

- Agile-команда анализирует метрики ошибок и отзыв пользователей на ретроспективе;

- если обнаружены проблемы, они фиксируются в беклоге и устраняются в следующем спринте.

Результат: в данном примере применение DevOps позволяет избежать простоев – обновления происходят без перерыва в работе сервиса. Благодаря Agile команда быстро адаптируется к изменяющимся требованиям бизнеса. Сокращается время между созданием кода и его выпуском – от недель до нескольких часов.

Таким образом Agile обеспечивает гибкость в изменении требований, а DevOps – надежную автоматизацию развертывания и тестирования. Их интеграция позволяет выпускать ПО быстрее и с минимальными рисками.

Критерии оценивания:

- определена суть Agile и DevOps

- корректно объяснены различия между ними;

- четко сформулированы 3-5 аспектов взаимодействия Agile и DevOps.

- объяснено, как методологии дополняют друг друга;

- приведен пример из практики;

- примеры могут быть иными, но должны соответствовать сути.

Ответ может варьироваться по деталям, разные формулировки допустимы, но общий смысл должен быть верным. Студент может привести примеры из разных областей, но важно, чтобы ответ включал описание взаимодействия Agile и DevOps в процессе разработки ПО.

Компетенции (индикаторы): УК-3, ОПК-3

2. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: Какие элементы управления требованиями на 3-м уровне СММІ (Defined) помогают минимизировать несоответствие ожиданий заказчика и реального функционала ПО?

Время выполнения: 35 минут

Ожидаемый результат:

На 3-м уровне СММІ (Defined) управление требованиями становится систематизированным, что помогает сократить разрыв между ожиданиями заказчика и итоговым функционалом ПО.

Основные элементы управления требованиями на 3-м уровне СММІ:

1) Формализация требований (Requirements Documentation):

- все требования фиксируются в стандартизированных документах: SRS (Software Requirements Specification) – Спецификация требований к ПО и Vision – Концепция системы, ее ключевые возможности;

- каждый функциональный и нефункциональный параметр согласуется с заказчиком;

- фиксируется не только то, что нужно сделать, но и ограничения, зависимости, критерии приемки.

2) Трассируемость требований (Requirements Traceability):

- связь требований с конкретными задачами разработки (каждое требование отслеживается от постановки до релиза);

- используется матрица трассируемости – таблица, связывающая требования с модулями системы и тест-кейсами;

- если заказчик вносит изменения, можно быстро определить, на какие части ПО это повлияет.

3) Методы моделирования и визуализации требований:

- для предотвращения разночтений требования дополняются: UML-диаграммами (диаграммы классов, вариантов использования, последовательностей), BPMN (моделирование бизнес-процессов), прототипами UI (интерактивные макеты интерфейсов). Эти инструменты позволяют заказчику "увидеть" будущий продукт и скорректировать ожидания.

4) Формальная верификация и согласование требований;

- прежде чем начать разработку, требования проходят формальные проверки: анализируются на консистентность (непротиворечивость), проверяются на полноту (учтены ли все сценарии), оценивается реализуемость – соответствует ли требование возможностям системы;

- требования подписываются заказчиком перед стартом проекта, что снижает риск "разночтений" позже.

5) Управление изменениями требований (Requirements Change Management):

- внедрен процесс рассмотрения изменений, чтобы избежать хаотичных правок;

- любое изменение требований проходит этап: оценки влияния на проект (временные и финансовые затраты), согласования с заказчиком и командой, актуализации трассируемости;

- это помогает минимизировать ситуации, когда заказчик на позднем этапе просит "немного переделать", не осознавая сложность изменений.

На 3-м уровне СММІ несоответствие ожиданий заказчика и реального функционала ПО минимизируется за счет:

- стандартизированной документации требований (SRS, Vision);

- трассируемости (связь требований с задачами и тестами);

- визуализации (UML, BPMN, UI-прототипы);

- формальной верификации требований;

- процесса управления изменениями.

Эти элементы позволяют избежать ситуаций, когда после релиза заказчик сообщает, что получил не то, что ожидал.

Критерии оценивания:

описаны ключевые элементы управления требованиями на 3-м уровне СММІ (формализация, трассируемость, моделирование, верификация, управление изменениями);

разъяснена связь между этими элементами и минимизацией расхождений между ожиданиями заказчика и конечным функционалом ПО;

ответ логичен, структурирован, содержит не менее 3-х ключевых аспектов управления требованиями;

при наличии примера или описания сценария внедрения оценка повышается.

Ответ может варьироваться по деталям, разные формулировки также допустимы, но общий смысл должен сводиться к пояснению как трассируемость, моделирование и управление изменениями помогают избежать разночтений с заказчиком.

Компетенции (индикаторы): УК-3, ОПК-3

3. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: Объясните различие между верификацией и валидизацией требований. Продемонстрируйте различия на примере, связанном с разработкой мобильного приложения для фитнеса, суть которого в следующем: команда разработчиков создает мобильное приложение для фитнеса, которое позволяет пользователям отслеживать свои тренировки, планировать занятия и получать советы по питанию. Одно из требований заключается в том, чтобы приложение могло отслеживать количество шагов пользователя и отображать эту информацию на главном экране.

Время выполнения: 30-35 минут

Ожидаемый результат:

Верификация – проверка соответствия требованиям стандартам и бизнес-целям.

Валидизация – проверка корректности и полноты требований с точки зрения пользователя.

Верификация в заданном примере отвечает на вопрос: "Правильно ли специфицировано требование о том, что приложение должно отслеживать количество шагов, и соответствует ли оно техническим стандартам?"

Валидизация отвечает на вопрос: "Правильно ли мы понимаем, что функция отслеживания шагов необходима пользователям и соответствует ли она их ожиданиям и потребностям?"

Верификация для данного примера будет включать:

1. Проверка формулировок требования: команда проверяет, что требование четко определяет, как именно должно происходить отслеживание

шагов. Например, уточняется ли, что приложение должно использовать датчики движения устройства?

2. Сравнение с проектной документацией: анализ того, соответствует ли данное требование существующим стандартам, техническим условиям и функциональным спецификациям приложения.

3. Тестирование: разработка тестов, которые проверяют, что приложение корректно отслеживает и отображает количество шагов, например, это может включать сравнение данных приложения с показаниями сертифицированного фитнес-трекера.

Валидизация будет включать:

1. Исследование потребностей пользователей: проведение опросов и интервью с потенциальными пользователями, чтобы понять, действительно ли им нужно отслеживать шаги и насколько они ценят эту функцию.

2. Прототипирование: создание интерактивного прототипа приложения, в котором пользователи могут увидеть, как будет выглядеть отображение данных о шагах. Например: обратная связь от пользователей поможет выяснить, нужна ли эта функция на главном экране или лучше разместить ее в другом разделе приложения.

3. Обсуждение с заинтересованными сторонами: проводится встреча с разработчиками, маркетологами и инвесторами, чтобы обсудить, как отслеживание шагов вписывается в общую стратегию приложения и его позиционирование на рынке.

Критерии оценивания:

- четкое определение понятий верификация и валидация;
 - корректное объяснение различий между ними;
 - приведены разъяснения применения для примера из условия задания;
 - осмысленное применение понятий на примере мобильного приложения.
- логичность и последовательность изложения.

Ответ может варьироваться по деталям, разные формулировки допустимы, но общий смысл должен быть верным.

Ответ может содержать дополнительные детали или примеры, но принципиальное различие между верификацией (проверка соответствия требованиям) и валидацией (оценка соответствия ожиданиям пользователей) должно быть ясно изложено. Если студент путает эти термины или описывает только один из процессов, оценка снижается.

Компетенции (индикаторы): УК-3, ОПК-3

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Методология программной инженерии» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.04.04 Программная инженерия.

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

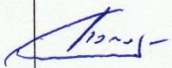
Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

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)
1.	Дополнен комплектом оценочных материалов	протокол заседания кафедры компьютерных систем и сетей № <u>8</u> от <u>10.03.2025</u>	 С.В. Попов