

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Новосибирский государственный технический университет»  
Кафедра вычислительной техники

**Паспорт зачета**

по дисциплине «Технология разработки программного обеспечения», 1 семестр

**1. Методика оценки**

Зачет проводится в письменной форме, по билетам. Билет состоит из 10 тестовых теоретических вопросов по темам дисциплины и формируется по следующему правилу:

- тестовые вопросы выбираются случайным образом из тестовой базы. Тестовые вопросы формируются в соответствии с теоретическими вопросами (темами), на каждый дается письменный ответ – 1-2 абзаца. Вес ответа 0..2 балла, сумма 0..20 баллов.

Таким образом, проверяется уровень сформированности компетенций и соотнесенных с ними индикаторов, закрепленных за дисциплиной.

На зачете преподаватель вправе задавать студенту уточняющие и дополнительные вопросы из общего перечня (п. 4).

**Форма билета для зачета**

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
Факультет АВТФ

Билет № \_\_\_\_\_

к зачету по дисциплине «Технология разработки программного обеспечения»

---

Тест 1. Тема: 18. Метрика проекта. Оценка сроков на основании трудоемкости (по Боэму). Оценка на основе собственного опыта. Метод PERT. Оценка на основе функциональных точек. Оценка по отраслевым данным. Метод COSOMO I.

Вопрос. 9. COSOMO II. Множители трудоемкости и факторы масштаба.

Тест 2. Тема: 15. Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM. Agile UP, ICONIX

Вопрос. 9. Определение и сущность SCRUM

Тест 3. Тема: 5. Фаза развития. Дисциплина «анализ требований». Способы извлечения и фильтрации требований. Бизнес-требования, бизнес-требования, системные требования, функциональные требования, бизнес-правила, атрибуты качества. Разработка и управление требованиями. Документ «спецификация требований к ПО ». Диаграммы прецедентов.

Вопрос. 6. Соотношение требований и других компонент системной аналитики (прецедентов, сценариев).

Тест 4 . Тема: 4. Фаза исследования. Основные дисциплины и артефакты. Дисциплина «анализ предметной области», бизнес-анализ. Функциональные диаграммы. Диаграммы потоков данных, деятельности. Моделирование предметной области.

Вопрос. 10. Пример фрагмента диаграммы деятельности для задания "Продажа билетов в кинотеатре".

Тест 5. Тема: 16. SCRUM как технологический фреймворк.. Терминология. Спринт. Митинг.

---

Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование.  
Вопрос. 7. Определение диаграммы сгорания, целей ее использования.

Тест 6. Тема: 8. Фаза анализа и проектирования. Многоуровневая архитектура клиент-серверных приложений. Тонкие и толстые клиенты. Локальное и сетевое взаимодействие слоев через интерфейсы и протоколы. Совместное использование кода различными типами клиентов

Вопрос. 8. Принципы преобразования локального взаимодействия слоев через интерфейс в сетевое, протокол взаимодействия.

Тест 7. Тема: 22. Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры.

Вопрос. 6. Тестовое покрытие функционального тестирования

Тест 8. Тема: 1. Сущность программной инженерии (ПИ). Связь с computer science. Особенности в сравнении и другими инженерными дисциплинами. Свод знаний и ПИ SWEBOOK

Вопрос. 7. Приведите пример проектной деятельности в других областях, наиболее близкий «по духу» к ПИ. Отметьте общность и различие в проектной деятельности и методологии

Тест 9. Тема: 5. Фаза развития. Дисциплина «анализ требований». Способы извлечения и фильтрации требований. Бизнес-требования, бизнес-требования, системные требования, функциональные требования, бизнес-правила, атрибуты качества. Разработка и управление требованиями. Документ «спецификация требований к ПО ». Диаграммы прецедентов.

Вопрос. 4. Определение, виды и пример бизнес-правила.

Тест 10. Тема: 2. Жизненный цикл (ЖЦ) программного продукта и проекта. «Легкие» и «тяжелые» модели процессов разработки ПО. Этапы и технологические процессы (дисциплины) ЖЦ. Результаты этапов и основные документы. Каскадная, итеративная и спиральная модели

Вопрос. 4. В чем проявляется легковесность/тяжеловесность модели (методологии) - 4 аспекта.

Утверждаю: зав. кафедрой \_\_\_\_\_ должность, ФИО  
(подпись)

(дата)

## 2. Уровни освоения компетенций и критерии оценки

Ответ на билет (тест) для зачета засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок и способен обосновать выбор метода решения. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, сформированы в полном объеме. Оценка составляет *от 18 до 20 баллов*.

Ответ на билет (тест) для зачета засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, дает характеристику процессов, явлений, проводит анализ причин, условий, способен представить количественные и качественные характеристики процессов, не допускает существенных ошибок при решении задачи. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, содержат

несущественные пробелы и сформированы на базовом уровне. Оценка составляет *от 14 до 17 баллов*.

Ответ на билет (тест) для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает непринципиальные ошибки, например, вычислительные. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, содержат пробелы и сформированы на пороговом уровне. Оценка составляет *от 10 до 13 баллов*.

Ответ на билет (тест) для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, не сформированы. Оценка составляет *менее 10 баллов*.

### 3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем заданиям билета составляет от 10 до 20 баллов включительно. Сумма менее 10 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине.

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

### 4. Вопросы к зачету по дисциплине «Технология разработки программного обеспечения»

#### Теоретические вопросы (темы)

1. Сущность программной инженерии (ПИ). Связь с computer science. Особенности в сравнении и другими инженерными дисциплинами. Свод знаний и ПИ SWEBOK
2. Жизненный цикл (ЖЦ) программного продукта и проекта. «Легкие» и «тяжелые» модели процессов разработки ПО. Этапы и технологические процессы (дисциплины) ЖЦ. Результаты этапов и основные документы. Каскадная, итеративная и спиральная модели.
3. Унифицированный процесс UP. Фазы жизненного цикла: исследование, анализ, реализация, внедрение. Содержание и результаты фаз. Итерация и ее рабочие потоки: требования, анализ, проектирование, реализация, тестирование, их содержание.
4. Фаза исследования. Основные дисциплины и артефакты. Дисциплина «анализ предметной области», бизнес-анализ. Функциональные диаграммы. Диаграммы потоков данных, деятельности. Моделирование предметной области.
5. Фаза анализа и проектирования. Дисциплина «анализ требований». Способы извлечения и фильтрации требований. Бизнес-требования, бизнес-требования, системные требования, функциональные требования. Разработка и управление требованиями. Документ «спецификация требований к ПО». Диаграммы прецедентов.
6. Фаза анализа и проектирования. Понятие архитектуры, ее многомерность. Основные методы проектирования и их особенности: структурное, функциональное, объектно-ориентированное, компонентное, проектирование на основе структур данных. Классы анализа. Виды классов: граница, управление, сущность. Диаграммы устойчивости. Архитектурные аспекты технологического процесса проектирования (по SWEBOK)
7. Фаза анализа и проектирования. Дисциплина проектирование (design). Ключевые моменты проектирования по SWEBOK: параллелизм, контроль и обработка событий, распределение компонентов, обработка ошибок и исключительных ситуаций и обеспечение отказоустойчивости, взаимодействие и представление (MVC),

- сохраняемость данных (доступность «долгоживущих» данных).
8. Фаза анализа и проектирования. Многоуровневая архитектура клиент-серверных приложений. Тонкие и толстые клиенты. Локальное и сетевое взаимодействие слоев через интерфейсы и протоколы. Совместное использование кода различными типами клиентов.
  9. Фаза анализа и проектирования Проектирование графического интерфейса (GUI). Основные аспекты. Архитектурное проектирование, основанное на GUI. Факторы, характеризующие GUI: производительность, человеческие ошибки, обучение, субъективное восприятие, запоминание, поиск, визуализация, навигация
  10. Виды моделей. Сущность UML как средства моделирования. Структура UML, статическая и динамическая составляющие модели. Составные элементы: сущности, отношения, диаграммы. Виды сущностей: структурные сущности – класс, интерфейс, кооперация, прецедент, активный класс, компонент, узел; поведенческие сущности – взаимодействия, деятельности, автоматы; группирующая сущность – пакет, аннотационная сущность – примечание.
  11. UML. Виды отношений: зависимость, ассоциация, агрегация, композиция, включение, обобщение, реализация. Отношения. Связи – отношения между объектами. Направленность связи, Сообщения. Диаграммы объектов. Ассоциации – отношения между классами. Свойства ассоциации: имя, кратность, навигация, атрибуты. Рефлексивные ассоциации, деревья и сети. Классы атрибутов ассоциаций (классы-ассоциации). Зависимости. Зависимости использования «use», «call», «parameter», «send» и «instantiate». Зависимости абстракции. Зависимости доступа.
  12. UML. Принятые деления: классификатор-экземпляр, интерфейс-реализация. Расширения: ограничения, стереотипы. Классификация диаграмм. Диаграммы классов (объектов). Диаграммы взаимодействий, коммуникационные диаграммы.
  13. UML. Диаграммы деятельности. Технология сетей Петри. Параллелизм. Поток управления, узел действия, ребро, узел управления, объектный узел, буферизация и в объектном узле. Объектные узлы – параметры, состояния объектных узлов. Контакты. Прерывающие ребра. Контакты исключений. Потоки объектов. Их аналоги в программировании
  14. UML. Диаграммы состояний. Конечные автоматы.
  15. Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM.
  16. SCRUM как технологический фреймворк.. Терминология. Спринт. Митинг. Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование.
  17. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.
  18. Метрика проекта. Оценка сроков на основании трудоемкости (по Боэму). Оценка на основе собственного опыта. Метод PERT. Оценка на основе функциональных точек. Оценка по отраслевым данным. Метод COSOMO II
  19. Управление проектами как инженерная дисциплина. Особенности управления программными проектами. Роль и место УПП в программной инженерии (ПИ). Компоненты организационного (менеджмент) и технологического (исполнение) планирования в УПП.
  20. Определение и характеристики риска. Шкалы оценивания последствий и вероятности. Способы идентификации. Реакция на риски. Наиболее вероятные риски по Боэму и Архипенкову. Качественные оценки рисков. Количественные оценки: анализ чувствительности, дерево решений, имитационное моделирование. Управление, направленное на снижение рисков. Вероятностный характер оценивания, его природа. Последствия «агрессивного» планирования. Исходные данные для оценивания,

характеристики проекта, используемые в оценивании. Оценка сроков на основании трудоемкости (по Боэму).

21. Управление проектом. Функциональная, проектная и матричная структура организации, разрабатывающей ПО. Подбор команды проекта. Функциональная организация команды. Функциональная классификация участников проекта, возможности совмещения функций.