**ПРАКТИЧЕСКАЯ РАБОТА № 1**

Тема: Сравнительный анализ методологий проектирования.

№1

**# Модель процессов MSF.**

**Назначение:** MSF предназначена для упрощения и структурирования процесса разработки программного обеспечения в проектах Microsoft. Она помогает командам более эффективно управлять проектами, обеспечивая гибкость и адаптивность.

**«Вес» модели:** Средний. MSF сочетает элементы гибких подходов и традиционных моделей, позволяя командам адаптировать процесс в зависимости от проекта.

**Модель жизненного цикла:** Итеративная и инкрементальная. MSF включает в себя несколько итераций, каждая из которых может включать в себя все стадии разработки.

**Фазы и стадии методологии:**

1) Инициация: Определение целей и требований.   
2) Планирование: Составление плана проекта, определение ресурсов и сроков.

3) Разработка: Итеративная разработка программного обеспечения.

4) Тестирование: Проверка и валидация продукта.

5) Развертывание: Внедрение в рабочую среду.

6) Поддержка: Обслуживание и улучшение продукта.

**# Технология XP.**

**Назначение:** XP сосредоточена на повышении качества программного обеспечения и гибкости в условиях меняющихся требований. Она ориентирована на разработчиков и включает множество практик, направленных на улучшение взаимодействия между членами команды.

**«Вес» модели:** Высокий. XP акцентирует внимание на практиках, таких как парное программирование, тестирование и непрерывная интеграция.

**Модель жизненного цикла:** Итеративная и инкрементальная, где каждое изменение проходит через цикл разработки, тестирования и внедрения.

**Фазы и стадии методологии:**

1) Планирование: Определение требований и создание пользовательских историй.

2) Проектирование: Создание архитектуры и дизайна системы.

3) Кодирование: Написание кода с использованием парного программирования.

4) Тестирование: Непрерывное тестирование и интеграция.

5) Релиз: Выпуск готового продукта.

**# Методология RUP.**

**Назначение:** RUP является гибридной методологией, которая сочетает элементы как традиционных, так и гибких подходов. Она предназначена для обеспечения четкой структуры и предсказуемости в процессе разработки программного обеспечения.

**«Вес» модели:** Средний. RUP предлагает гибкость в адаптации процесса к конкретным проектам.

**Модель жизненного цикла** Итеративная. Разработка проходит через несколько фаз, каждая из которых включает в себя итерации.

**Фазы и стадии методологии:**

1) Инициация: Определение видения и целей проекта.

2) Уточнение: Сбор требований и анализ рисков.

3) Конструирование: Разработка и тестирование системы.

4) Передача: Внедрение системы и обучение пользователей.

5) Поддержка: Обслуживание и развитие продукта.

**# Метод DSDM.**

**Назначение:** DSDM ориентирована на быстрое и гибкое развитие программного обеспечения с акцентом на активное взаимодействие с пользователями и бизнес-требованиями.

**«Вес» модели:** Высокий. DSDM акцентирует внимание на вовлечении пользователей и быстрой доставке ценности.

**Модель жизненного цикла** Итеративная и инкрементальная, с акцентом на активное взаимодействие с заказчиком на всех этапах.

**Фазы и стадии методологии:**

1) Предварительное исследование: Определение бизнес-целей и требований.

2) Функциональное моделирование: Разработка функциональных спецификаций.

3) Проектирование и разработка: Создание системы на основе функциональных требований.

4) Тестирование: Проверка системы на соответствие требованиям.

5) Внедрение: Развертывание системы и обучение пользователей.

6) Поддержка и развитие: Обслуживание и улучшение системы.

**# Метод SCRUM.**

**Назначение:** SCRUM предназначен для управления проектами в условиях быстро меняющихся требований. Он фокусируется на взаимодействии команды и скорой поставке работающего программного обеспечения.

**«Вес» модели:** Высокий. SCRUM подразумевает гибкость и адаптацию в процессе работы.

**Модель жизненного цикла** Итеративная. Процесс делится на спринты, которые обычно длятся 2-4 недели.

**Фазы и стадии методологии:**

1) Планирование спринта: Определение целей и задач на текущий спринт.

2) Исполнение спринта: Команда работает над задачами, проводя ежедневные встречи (Daily Scrum).

3) Обзор спринта: Демонстрация выполненной работы и получение обратной связи.

4) Ретроспектива спринта: Анализ работы команды и выявление областей для улучшения.

№2

|  |  |  |
| --- | --- | --- |
| **Все модели** | **Преимущества** | **Недостатки** |
| **Тяжелые** | 1. Структурированность: Четкая последовательность этапов, что упрощает управление проектом.  2. Документирование: Подробная документация на каждом этапе, что облегчает поддержку и понимание проекта.  3. Управление рисками: позволяет заранее идентифицировать и анализировать риски.  4. Простота в планировании: Легче планировать ресурсы и сроки, так как этапы фиксированы.  5. Устойчивость к изменениям: применяется в проектах с низкой вероятностью изменений требований. | 1. Негибкость: трудно адаптироваться к изменениям в требованиях, особенно на поздних стадиях.  2. Длительное время разработки: Процесс может занять больше времени из-за строгой последовательности этапов. 3. Высокие затраты на документацию: требуется много времени и ресурсов для создания и поддержки документации.  4. Сложность: может быть избыточной для небольших проектов, что приводит к ненужным затратам.  5. Медленная реакция на обратную связь: Пользовательская обратная связь может быть получена только после завершения значительной части разработки. |
| **Легкие** | 1. Гибкость: легко адаптируется к изменяющимся требованиям и приоритетам.  2. Быстрая доставка: частые релизы позволяют получать обратную связь от пользователей быстрее.  3. Акцент на взаимодействии: поддерживает активное взаимодействие между командой и заказчиком.  4. Низкие затраты на документацию: Уменьшение времени, затрачиваемого на документацию, позволяет сосредоточиться на разработке.  5. Улучшение качества: Постоянное тестирование и итеративный подход способствуют повышению качества продукта. | 1. Зависимость от команды: Успех зависит от уровня самодисциплины и организованности команды. 2. Нехватка структуры: может привести к неопределенности и путанице, если команда не организована.  3. Трудность в масштабировании: может быть сложно управлять большими проектами или командами.  4. Риски в управлении: Меньшая документация может усложнить дальнейшую поддержку и понимание проекта.  5. Проблемы с вовлечением пользователей: требует активного участия пользователей, что не всегда возможно. |

**Контрольные вопросы.**

1. Модель процессов MSF – это методология разработки программного обеспечения, разработанная компанией Microsoft, которая предоставляет структурированный подход к созданию программных решений.
2. Технология XP – это методология разработки программного обеспечения, ориентированная на улучшение качества кода и гибкость в условиях изменяющихся требований.
3. Методология RUP – это гибридная методология разработки программного обеспечения, основанная на итеративном и инкрементальном подходе.
4. Метод DSDM – это методология разработки программного обеспечения, ориентированная на быструю и гибкую доставку решений с акцентом на активное взаимодействие с пользователями и бизнес-требованиями. DSDM использует итеративный подход и четко определяет роли и процессы, обеспечивая быструю реакцию на изменения.
5. Метод SCRUM – это гибкая методология управления проектами, основанная на итеративном подходе, которая фокусируется на быстрой доставке работающего программного обеспечения через короткие циклы (спринты) и активное сотрудничество команды. SCRUM включает регулярные встречи для планирования, оценки и улучшения процесса разработки.