Практическая работа №1

№1

# **MSF**

**Назначение**: Эта методология подходит для работы в проектной группе или организации любого масштаба.

**Вес**: Тяжелая

**Модель процесса MSF**: выработка концепции (Visioning), планирование (Planning), разработка (Developing), стабилизация (Stabilizing), внедрение (Deploying).

**Фазы и стадии методологии MSF**:

## Выработка концепции

Команда вырабатывает единое видение проекта или его части. Совместными усилиями коллеги решают, какая именно функциональность будет разрабатываться в ходе итерации, определяют основные концепции, которые лягут в основу разработки. Этап завершается вехой «Концепция утверждена».

## Планирование

Задачи, которые необходимо выполнить в ходе итерации, разбиваются на подзадачи, определяется сложность их реализации, устанавливаются сроки и назначаются ответственные. В планы закладывается время на тестирование и исправление дефектов, а также предварительно намечается, как именно будет проходить тестирование.

## Разработка

На данном этапе MSF создается программный код новой функциональности в соответствии с концепцией и утвержденными планами.

## Стабилизация

К делу подключаются тестировщики. После тестирования выявленные баги и недочеты возвращаются разработчикам для исправления.

## Внедрение

Очередной релиз программного продукта передается заказчику и устанавливается на клиентских компьютерах.

# **RUP**

**Назначение**: Это процесс разработки программного обеспечения для объектно-ориентированных моделей.

**Вес**: Тяжелая

**Модель процесса MSF**: начальная стадия (Inception), уточнение (Elaboration), Построение (Construction), Внедрение (Transition).

**Фазы и стадии методологии MSF**:

## Начальная стадия

Цель этой фазы заключается в определении общей концепции проекта, анализе основных рисков, а также в создании бизнес-кейса. В результате этой фазы команда должна иметь четкое понимание основных требований и рамок проекта.

## Уточнение

В этой фазе команда уточняет требования, разрабатывает основную архитектуру системы и разрабатывает план выпуска продукта. Здесь основное внимание уделяется анализу рисков и проработке ключевых аспектов системы.

## Построение

Здесь основное внимание уделяется разработке, кодированию и тестированию. К концу этой фазы команда должна иметь рабочую версию продукта, готовую к выпуску.

## Внедрение

В этой фазе происходит переход продукта к пользователю. Это включает в себя устранение ошибок, тренировку пользователей и развертывание системы в продуктивной среде.

# **DSDM**

**Назначение**: это метод создания и выпуска проектов по Agile. DSDM создали, чтобы облегчить разработку программного обеспечения и создание кода.

**Вес**: Легкая

**Модель процесса MSF**: предпроектная, стадия жизненного цикла проекта и постпроектная стадия.

**Фазы и стадии методологии MSF**:

## Предпроектная

На этой стадии определяются вероятные проекты, происходит выделение средств и определение проектной команды. Решение задач на этой стадии поможет избежать проблем на более поздних стадиях проекта

## Жизненный цикл проекта

Структура DSDM состоит из пяти последовательных этапов: технико-экономическое обоснование, бизнес-исследование, итерация функциональной модели, итерация проектирования и построения, а также реализация. На этапах технико-экономического обоснования и бизнес-анализа оценивается жизнеспособность проекта и его соответствие целям организации. После этих начальных этапов платформа вступает в итеративные производственные циклы, в ходе которых функциональная модель, а также процессы проектирования и сборки уточняются в сотрудничестве с заинтересованными сторонами. Заключительный этап реализации включает развертывание, передачу и обслуживание программного обеспечения, а также закрытие проекта.

## Постпроектная

На этой стадии обеспечивается эффективная работа системы. Это достигается за счёт поддержания проекта, его улучшения и исправления ошибок согласно принципам DSDM. Поддержка проекта осуществляется продолжением разработки, основанной на итеративной и инкрементной природе DSDM. Вместо того, чтобы закончить проект за один цикл обычно возвращаются к предыдущим стадиям или этапам, чтобы улучшить продукт.

# **SCRUM**

**Назначение**: это методология гибкой разработки, относящаяся к семейству Agile. В отличие от классического каскадного (последовательного) метода предлагает совершенно иную и более эффективную структуру работы над продуктом.

**Вес**: Легкая

**Модель процесса MSF**: организация бэклога, планирование спринта, спринт, ежедневное Scrum-совещание, обзор итогов спринта, ретроспектива спринта

**Фазы и стадии методологии MSF**:

## Организация бэклога

Это командная встреча с владельцем продукта, на которой он доводит до сведения участников список приоритетных задач по реализации проекта. Первоначальная презентация бэклога-продукта, согласованного с пожеланиями заказчика, может преображаться от спринта к спринту из-за изменений на рынке либо в следствие корректив после тестирования продукта пользователем. За поддержание приоритетов требований и актуальность списка задач отвечает владелец продукта.

Для этого он постоянно держит руку на пульсе и чутко реагирует на изменения пожеланий заказчика, своевременно осведомляя о них команду Scrum.

## Планирование спринта

Sprint Planning Meeting – самое длительное собрание команды Scrum, на котором присутствуют все разработчики, скрам-мастер и владелец продукта.

Целью этого собрания является разработка стратегии спринта. Идет обсуждение с владельцем, какие пункты бэклога-продукта будут реализовываться в текущем спринте, разработчики, согласно уровню своей производительности, объективно оценивают возможность выполнения поставленных в плане задач, команда определяет критерии тестирования и возможную скорость изготовления артефакта, а скрам-мастер помогает распределить функционал между участниками проекта.

По итогу собрания вырабатывается четкая стратегия на спринт с учетом пожеланий заказчика, расстановки приоритетов от владельца продукта и согласованности действий разработчиков. Все это способствует реализации основной задачи Scrum-проекта – повышению ценности продукта в рамках текущего цикла.

## Спринт

Спринт – это непрерывная коммуникация членов команды в течение определенного срока. По практике средняя оптимальная длительность спринта – две недели. В это время каждый участник вносит свой вклад в общее дело, поэтому, если при планировании структуры спринта команда считает нужным сократить или увеличить время, то это не возбраняется.

Но у каждой команды есть право распределять время самостоятельно, на то она и самоуправляемая. Чтобы у скрам-команды выработался опыт определения оптимальной длительности спринта под ту или иную задачу, необходимо разработать от начала до конца несколько продуктов. Тогда можно будет проанализировать ошибки и взять на вооружение достоинства предыдущих проектов в части эффективного определения времени спринта.

## Ежедневное Scrum-совещание

Такое собрание команда может запланировать на любое время дня, главное, чтоб оно проводилось каждый день в одном и том же месте и длилось не слишком долго. Достаточно 15 минут, поэтому собрание проводится стоя. У каждого участника к моменту собрания готов небольшой отчет по итогам проделанной за вчерашний день работы и сформулированы задачи на сегодня.

## Обзор итогов спринта

По окончании срока спринта Scrum-команда предъявляет владельцу продукта демоверсию конечного продукта – инкремент. Для более точной и всеобъемлющей оценки итогов работы цикла можно пригласить на собрание разработчиков из других команд, руководителя проектов и конечных пользователей продукта.

Отзывы всех участников обзора помогают подытожить, были ли решены задачи спринта в полном объеме, надо ли что-то изменить в бэклоге и получилось ли выпустить полноценный тестируемый продукт на данном этапе. Эта обратная связь дает возможность более точно определить, что именно нужно потребителю.

Обзор итогов спринта – это не мероприятие строгой отчетности, а конструктивная встреча участников проекта. Длится такое собрание не более 4-х часов для месячного спринта. В итоге владелец проекта не просто принимает работу команды, но и получает более точное представление о формировании следующего бэклога.

## Ретроспектива спринта

После обзора итогов спринта назначается очередной митинг – ретроспектива спринта. Это встреча команды, на которой обсуждается эффективность предыдущего этапа работы. Анализируются все минусы и плюсы продукта, выявленные на обзоре. Каждый может высказаться, почему не получилось в прошлый раз и предложить свои методы преодоления трудностей разработки.

№2

|  |  |  |
| --- | --- | --- |
| Вес модели | Преимущества | Недостатки |
| Тяжелые | Процессы рассчитаны на среднюю квалификацию исполнителей.  Невысокие требования к стабильности команды.  Нет ограничений на распределение участников.  Отсутствуют огра­ничения по объему и сложности выполняемых проектов. Используются для проектов с высоки­ми рисками.  Легче получить высокий уровень сертификации стандартов качества. | Требуют существенной управленческой надстройки.  Более длительные стадии анализа и проекти­рования.  Более формализованные коммуникации.  Большие затраты на документацию длительного процесса сопровождения ПО |
| Легкие | Меньше непроизводительных расходов, связанных с управлением проектом, рисками, изме­нениями, конфигурациями, сопровождением ПО.  Упрощенные стадии анализа и проектиро­вания. · Основной упор на разработку функци­ональности, совмещение ролей. Неформальные коммуникации | Используется для проектов с невысокими рисками и небольшого масштаба и сложности.  Участники проекта должны располагаться ком­пактно.  Плохо подходят для новых проектов.  Требуют более квалифицированной и стабиль­ной команды.  Сложнее получить высокий уро­вень сертификации стандартов качества |

Контрольные вопросы:

1. Модель процессов MSF - модель процесса, то есть последовательность действий, необходимых для построения IT-решения. Модель не предписывает конкретных процедур и не содержит жестких формализованных требований к процессу — при создании MSF компания Microsoft стремилась сделать ее гибкой и адаптируемой к условиям любого проекта. В MSF объединились две концепции разработки: «Водопад» и спиральная модель.
2. Технология XP – это гибкая методология разработки программного обеспечения. Как и у других agile-методологий, у нее есть особенные инструменты, процессы и роли. Хотя автор XP не придумал ничего нового, а взял лучшие практики гибкой разработки и усилил до максимума.
3. Методология RUP – это процесс разработки программного обеспечения, созданный компанией Rational Software, который сосредотачивается на итерационной разработке и объектно-ориентированном анализе и проектировании.
4. Метод DSDM – это главным образом методика разработки программного обеспечения, основанная на концепции быстрой разработки приложений (Rapid Application Development, RAD). DSDM - это итеративный и инкрементный подход, который придаёт особое значение продолжительному участию в процессе пользователя/потребителя.
5. Метод SCRUM - это методика гибкого управления проектами, которая помогает командам структурировать свою работу и управлять ею с помощью набора ценностей, принципов и практик.