

## **1 слайд:**

Здравствуйте, тема моей презентации “RUP: процессы, фазы, итерации, роли, артефакты, прецеденты”.

## **2 слайд:**

В RUP можно выделить следующие основные роли:

- **Архитектор системы (System Architect):** Ответственен за определение архитектуры системы. Проектирует ключевые элементы системы, интерфейсы и взаимодействия.
- **Бизнес-аналитик (Business Analyst):** Отвечает за выявление бизнес-требований и целей заказчика. Анализирует бизнес-процессы заказчика, определяя, как технологии могут их улучшить.
- **Разработчик (Developer):** Отвечает за написание кода, в соответствии с архитектурой и дизайном системы. Реализует функциональность на основе спецификаций и требований.
- **Тестировщик (Tester):** Осуществляет тестирование системы на соответствие требованиям и отсутствие ошибок. Создает тестовые сценарии и выполняет их, обеспечивая качество продукта.
- **Менеджер проекта (Project Manager):** Ответственен за планирование, мониторинг и контроль выполнения проекта. Координирует деятельность команды и управляет рисками.
- **Интегратор (Integrator):** Собирает все компоненты системы в единое целое. Ответственен за управление версиями и сборку продукта.
- **Конфигурационный менеджер (Configuration Manager):** Отслеживает и контролирует изменения в проекте. Управляет версиями и артефактами проекта.

Роли могут варьироваться в зависимости от конкретного проекта и его требований.

## **3 слайд:**

Одним из основных элементов RUP является итерации. Итерация — это полный цикл разработки фрагмента ПО, в результате которого формируется промежуточная версия, реализующая некоторый функционал. В ходе разработки происходит постепенное усложнение создаваемого ПО, наращивание его функциональных возможностей.

Проект в RUP состоит из последовательности итераций с рекомендованной продолжительностью от 2 до 6 недель.

## **4 слайд:**

Первая фаза это – “Сбор требований”. На этом этапе разработчики должны убедиться, что программный проект осуществим. Для этого они выявляют и анализируют следующие характеристики:

- область применения будущей системы (функциональное назначение и взаимодействие с внешней средой);
- основные требования к архитектуре системы (на базе главных функций программы);
- основные (критические) риски;
- примерную стоимость и план проекта;
- начальный макет системы.

#### **5 слайд:**

Вторая фаза – “Проектирование”. Создание архитектурного базиса системы и подготовка к выполнению третьего этапа — построению. На данном этапе решают следующие задачи:

- создают базовый уровень архитектуры системы;
- определяют существенные риски, методы их отслеживания и устранения или снижения;
- формируют требования к качеству ПО и процесса его разработки, например, стандарты;
- составляют финансовый план проекта;
- проводят анализ большинства прецедентов (до 80 %);
- готовят план итераций для выполнения следующего этапа.

#### **6 слайд:**

Следующая фаза – “Построение”. Создание работоспособного программного продукта. В ходе построения решают следующие вопросы:

- управление ресурсами;
- оптимизация процессов разработки;
- отслеживание существенных и критических рисков;
- оценка качества создаваемого ПО.

#### **7 слайд:**

Последняя фаза – “Внедрение”. Готовое и протестированное ПО устанавливают для эксплуатации конечными пользователями и готовят пользователей для работы с ним. На данном этапе выполняют следующие задачи:

- завершение реализации ПО;
- формирование рекомендаций по установке и эксплуатации ПО;

- подготовка программно-аппаратного обеспечения для работы с ПО (конечными пользователями);
- применение ПО в среде конечного пользователя;
- составление документации и руководств (в соответствии с требованиями заказчика);
- проведение бета-тестов, выявление и устранение ошибок ПО;
- создание версии ПО для приемосдаточных испытаний

## 8 слайд:

Каждая итерация состоит из последовательного выполнения пяти рабочих процессов:

1. Моделирование бизнес-процессов. Применяется для того, чтобы разобраться в структуре исследуемой предметной области, обеспечить единство понимания основных автоматизируемых процессов среди всех участников проекта и определить высокоуровневые требования, которые должны быть реализованы в ходе проекта.
2. Управление требованиями. Позволяет прийти к соглашению с заказчиками и конечными пользователями, определить, что должна уметь делать создаваемая система, предоставить более четкие инструкции участникам проекта о возможностях системы, создать базу для успешного планирования работ в проекте и оценки его статуса в любой момент жизненного цикла.
3. Анализ и проектирование. Служат для последовательного преобразования выявленных требований к системе в спецификации особого вида, которые описывают, как следует конкретно реализовать конечный продукт. Следует при этом делать различия между анализом и проектированием. Основное из них состоит в том, что спецификации анализа не зависят от конкретной платформы и технологии, для которой осуществляется создание ИС. А спецификации проектирования являются точным представлением проектируемой системы, часто позволяя автоматизировать процесс генерации на их основе программного кода.
4. Реализация. Необходима для выявления порядка организации программного кода в терминах отдельных подсистем, преобразования исходного кода в выполняемые компоненты, тестирования созданных компонентов и интеграции отдельных компонентов в подсистемы и системы.
5. Тестирование. Позволяет определять и контролировать качество создаваемых продуктов, следить за тем, насколько качественно осуществлена интеграция компонентов и подсистем, все ли требования

к системе реализованы и все ли выявленные ошибки устранены до того, как система будет развернута на оборудовании конечного пользователя.

Примерное распределение работ по рабочим процессам в зависимости от этапа разработки приведено на слайде.

При каждой итерации осуществляют все рабочие процессы, но удельный вес конкретного рабочего процесса зависит от этапа разработки:

- на первом этапе основное внимание уделяют сбору требований;
- на этапе проектирования — анализу и проектированию;
- на этапе построения — реализации;
- на этапе тестирования — тестированию.

Так же имеются вспомогательные процессы:

- Конфигурационное управление и управление изменениями позволяет организовать эффективную работу с артефактами проекта, контролировать и управлять доступом к ним, вести историю изменений, обеспечить эффективное взаимодействие участников проекта, как в простых командах, так и в распределенных, находящихся на большом удалении друг от друга.
- Управление проектом включает в себя непосредственное формирование условий для эффективного хода всего проекта, определение руководств и руководящих принципов для планирования, формирования команды и мониторинга проекта, выявление и управление рисками, организацию работы участников проекта, формирование бюджета, планирование фаз и итераций.
- Управление средой позволяет осуществить поддержку всех участников проекта. В эту поддержку входят выбор инструментария и его приобретение, настройка и установка, конфигурирование процесса, доработка и адаптация методологии, используемой для ведения проекта, обучение

## **9 слайд:**

Артефакт – подтип рабочего продукта, фрагмент информации, который: 1) производится, изменяется или используется процессом, 2) определяет область ответственности и 3) подлежит контролю версий. Артефакт может быть моделью, элементом модели или документом. Документ может включать другие документы.

Два наиболее важных артефакта планирования итерации в RUP – это План итерации и Оценка итерации. В сочетании они облегчают принятие решений, которые позволяют уменьшить риск, принять изменения и направить проект через каждую итерацию.

После начала итерации команды завершают работу, указанную в Плане итерации. После завершения работы выполняется Оценка итерации, чтобы определить, были ли достигнуты цели итерации и как, используя как можно больше объективных измерений. На основе оценки принимается решение, продолжать проект или нет.

**10 слайд:**

На всех этапах разработки в центре внимания находится прецедент. Прецеденты — это последовательности действий, выполняемых системой для получения наблюдаемого результата. Функциональные возможности системы определяются набором прецедентов, каждый из которых представляет некоторый поток событий. Описание прецедента определяет то, что произойдет в системе, когда прецедент будет выполнен. Каждый прецедент ориентирован на задачу, которую он должен выполнить. Набор прецедентов устанавливает все возможные пути (маршруты) выполнения системы.

**11 слайд:**

В презентации использовались следующие источники.

**12 слайд:**

Спасибо за внимание.