Практика №3

Первая часть

Задание 1: Изучить методологии проектирования ИС   
Задание 2: Дать характеристику каждой модели:

1. 1. Microsoft Solutions Framework (MSF)
   2. Эта модель сочетает в себе свойства двух стандартных моделей: каскадной и спиральной. Модель процессов MSF охватывает весь жизненный цикл создания решения, начиная с его отправной точки и заканчивая внедрением. Процесс MSF ориентирован на «вехи» (milestones) – ключевые точки проекта, характеризующие достижение в его рамках какого-либо существенного (промежуточного либо конечного) результата, причем этот результат может быть оценен и проанализирован. Критерии оценки результата должны быть сформулированы еще до начала проекта. Для декомпозиции больших этапов работы может быть введено также большое количество промежуточных вех. Модель процессов MSF учитывает постоянные изменения проектных требований. Она исходит из того, что разработка решения должна состоять из коротких циклов, реализующих поступательное движение от начальных версий системы к ее окончательному виду. В рамках MSF программный код, документация, дизайн, планы и другие рабочие материалы создаются, как правило, итеративными методами.
   3. Фазы и стадии методологии:
      1. MSF следует начинать с разработки решения, с построения, тестирования и внедрения его базовой функциональности.
      2. Далее к решению добавляются все новые и новые возможности, т. Е. с каждой новой версией эволюционирует функциональность решения
      3. Документация должна изменяться по мере реализации проекта вместе с изменениями требований к конечному продукту. В рамках MSF предлагается ряд шаблонов стандартных документов для каждой стадии разработки продукта, которые могут быть использованы для планирования и контроля процесса разработки.
2. 1. Extreme Programming (XP)
   2. Технология XP (англ. Extreme Programming) базируется на спиральной модели жизненного цикла. Для оценки проектов с точки зрения применимости XP применяются два показателя – критичность и масштаб. Критичность определяется последствиями, вызываемыми дефектами ПО, и может иметь один из четырех уровней: 1. C – дефекты вызывают потерю удобства. 2. D – дефекты вызывают потерю возместимых ресурсов. 3. E – дефекты вызывают потерю невозместимых ресурсов. 4. L – дефекты могут создавать угрозу для человеческой жизни. Масштаб определяется количеством разработчиков, участвующих в проекте: от 1 до 5 человек – малый масштаб, от 6 до 20 – средний масштаб, свыше 20 человек – большой масштаб. XP применима в проектах малого и среднего масштаба с низкой критичностью (C или D).   
      Методики, которые используются при разработки:
      1. Test-Driven Development – разработка через тестирование).
      2. Парное программирование
      3. Непрерывная интеграция
      4. Упрощенное программирование
      5. Коллективное владение
   3. Каждая итерация обычно длится две-три недели и выглядит как программный проект в миниатюре, включая все этапы жизненного цикла системы:Разработка в XP ведется небольшими трехнедельными итерациями, в течение которых уточняются и реализуются требования к системе:
      1. Планирование
      2. Анализ требований
      3. Проектирование
      4. Программирование
      5. Тестирование
      6. Документирование
3. 1. Rational Unified Process (RUP)
   2. Методология RUP основана на спиральной модели жизненного цикла ИС. В качестве языка моделирования в RUP используется язык Unified Modelling Language (UML). Наибольшее внимание RUP уделяет начальным стадиям разработки проекта – анализу и моделированию, – что призвано снизить риски проекта за счет раннего обнаружения возможных ошибок. Последовательный выпуск версий организован таким образом, чтобы наиболее существенные риски устранялись в первую очередь. Методология RUP широко использует так называемые прецеденты, или сценарии использования – описание последовательностей действий, которые может осуществлять система, взаимодействуя с внешними действующими факторами. Прецеденты создаются при помощи UML и включают варианты как правильных, так и ошибочных последовательностей (исключений). Прецеденты служат для документирования требований заказчика к проектируемой информационной системе. Прецедент описывает целостный фрагмент поведения системы в виде последовательности сообщений, которыми система обменивается с действующими лицами.  
      Другие существенные черты методологии RUP:
      1. Заранее предусматриваются изменения в требованиях в проектных решениях в течение всего процесса разработки
      2. Постоянное обеспечение качества на всех этапах разработки ИС
      3. Компонентная архитектура качества на всех этапах разработки ИС
      4. Компонентная архитектура используется на всех этапах разработки ИС
   3. RUP использует итеративную модель разработки. В конце каждой итерации (продолжительностью в несколько недель) команда разработчиков должна получить функционирующую версию конечного продукта, позволяющую достичь запланированных на данную итерацию целей. Итеративная разработка позволяет быстро реагировать на меняющиеся требования, обнаруживать и устранять риски на ранних стадиях проекта, а также эффективно контролировать качество создаваемого продукта. Полный жизненный цикл разработки ИС состоит из четырех фаз, описываемых ниже. Каждая фаза может включать в себя одну или несколько итераций процесса создания системы.
      1. Начальная стадия
         1. Формируется единая точка зрения на проект у заказчика и разработчиков, а также определяются границы проекта
         2. Создается экономическое обоснование разработки
         3. Определяются основные требования, ограничения и функциональность системы
         4. Создается базовая версия модели прецедентов
         5. Оцениваются риски
      2. Уточняющая стадия
         1. Анализ требований заказчика, включая детальное описание для большенства прецедентов
         2. Проектирование архетикруты ИС, спецификация функций и пользовательского интерфейса
         3. Планирование работ по проекту и всех необходимых ресурсов
      3. Построение
         1. В фазе «Построение» происходит итеративная реализация требуемых функций ИС. Это основная фаза проектирования и создания программного кода. Фаза завершается выпуском бета- версии системы.
      4. Внедрение
         1. В фазе «Внедрение» финальная версия системы внедряется у ее заказчика. Фаза включает проведение испытаний системы, обучение пользователей, а также оценку качества ИС. В том случае, если качество системы не соответствует требованиям заказчика, фаза «Внедрение» выполняется повторно. Выполнение всех требований и достижение целей проекта означает завершение полного цикла разработки.
4. 1. Dynamic Systems Development Method (DSDM)
   2. Основан на концепции быстрой разработки приложений (RAD). Метод DSDM – это итеративный и инкрементный подход разработки программного обеспечения, который придает особое значение продолжительному участию в процессе заказчика системы. Цель метода DSDM – соблюдение сроков и бюджета проекта при допущении изменений в требованиях к системе во время ее разработки. Как представитель RAD-технологии DSDM фокусируется на проектах информационных систем, характеризующихся сжатыми сроками и бюджетами. DSDM входит в семейство гибкой методологии разработки программного обеспечения, а также может применяться для разработок, не входящих в сферу информационных технологий. Существует возможность включения в DSDM частей других методик, таких как Rational Unified Process (RUP) или XP. Другой гибкий метод, похожий на DSDM по процессу и концепции, – Scrum. DSDM содержит указания на типичные ошибки проектов информационных систем, такие как превышение бюджета, несоблюдение сроков сдачи, недостаточное вовлечение пользователей и менеджеров организации-заказчика в работу над проектом.
   3. Фазы и стадии методологии:
      1. Жизненный цикл проекта:
         1. Предпроектная стадия – на этой стадии определяются риски проекта, происходит выделение средств и определение проектной команды. Решение задач на этой стадии поможет избежать проблем на более поздних стадиях проекта.
         2. Стадия проекта – Это самая детально разработанная стадия DSDM. Она состоит из пяти этапов, которые формируют итеративный, инкрементный подход к разработке информационных систем
            1. Исследование реализуемости.
            2. Исследование экономической целесообразности.
            3. Создание функциональной модели.
            4. Проектирование и разработка.
            5. Этап реализации.
         3. Пост проектная стадия – На этой стадии обеспечивается внедрение и эксплуатация системы. Это достигается за счет поддержания проекта, его улучшения и исправления ошибок согласно принципам DSDM. Поддержка проекта осуществляется как продолжение 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.
5. 1. Scrum
   2. Методология Scrum устанавливает правила управления процессом разработки и позволяет использовать уже существующие практики кодирования, корректируя требования или внося тактические изменения. Использование этой методологии дает возможность выявлять и устранять отклонения от желаемого результата на более ранних этапах разработки программного продукта.
   3. Чем меньше формализма, тем более гибко и эффективно можно работать, – это основной принцип данной методологии. Scrum представляет собой простой каркас, который можно использовать для организации команды и достижения результата более продуктивно и с более высоким качеством за счет анализа сделанной работы и корректировки направления развития между итерациями. Методология позволяет команде выбрать задачи, которые должны быть выполнены, учитывая бизнес-приоритеты и технические возможности, а также решить, как их эффективно реализовать. Это позволяет создать условия, при которых команда работает с удовольствием и максимально продуктивно. Scrum фокусируется на постоянном определении приоритетных задач, основываясь на бизнес-целях, что увеличивает полезность и доходность проекта на его ранних стадиях. Так как при инициации проекта его доходность определить почти невозможно, Scrum предлагает концентрироваться на качестве разработки и к концу каждой итерации иметь промежуточный продукт, который можно использовать, пусть и с минимальными возможностями.
   4. Стадии данной методологии являются цикличными:
      1. Создание требований к продукту
      2. Планирование итерации
      3. Выполнение итерации
      4. Анализ результатов, пересмотр требований

Задание 3: Перечислите преимущества и недостатки разработки ПО по «весу», заполнив таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Все модели | Преимущества | Недостатки |
| Тяжелые | MSF | - Хорошо подходит для крупных проектов с четкими требованиями.  - Обеспечивает высокую степень контроля над процессом разработки. | - Требует большого объема документации и планирования.  - Менее гибок и плохо адаптируется к изменениям. |
| RUP | - Четко структурированная методика, обеспечивающая полный контроль над проектом.  - Подходит для сложных проектов с большим количеством участников. | - Высокая сложность и необходимость значительных ресурсов для внедрения.  - Недостаточная гибкость для быстрого реагирования на изменения. |
| Легкие | XP | - Быстрое получение обратной связи и возможность адаптации к изменениям.  - Улучшение качества кода благодаря непрерывному рефакторингу и парному программированию. | - Может потребовать высокой дисциплины и самоорганизации от команды.  - Меньше подходит для проектов с жесткими требованиями и сроками. |
| DSDM | - Акцент на приоритетах бизнеса и быстром получении результатов.  - Хорошая поддержка изменений и быстрое принятие решений. | - Требуется высокая вовлеченность заказчиков и управление ожиданиями.  - Возможны сложности с масштабируемостью для больших проектов. |
| Scrum | - Простота и гибкость, позволяющие быстро реагировать на изменения.  - Эффективное взаимодействие внутри команды и с заказчиком. | - Необходимость постоянного участия владельца продукта и грамотного скрам-мастера.  - Могут возникать проблемы с долгосрочным планированием и контролем бюджета. |

Вторая часть

Мы выбрали методологию Extreme Programming (XP) для нашего проекта, потому что она обеспечивает гибкость, высокое качество кода, простую реализацию и эффективное сотрудничество в небольшой команде, позволяя быстро получать результаты и адаптироваться к изменениям.

Третья часть

