

ТЕМА 3.
Технологии
проектирования ИС.

Лекция 7.
Современные технологии
проектирования ИС.

Современные технологии проектирования

Название	Сокращение	Разработчик
Rational Unified Process	RUP	IBM (Rational Software)
Custom Development Method	CDM	Oracle
Microsoft Solutions Framework	MSF	Microsoft

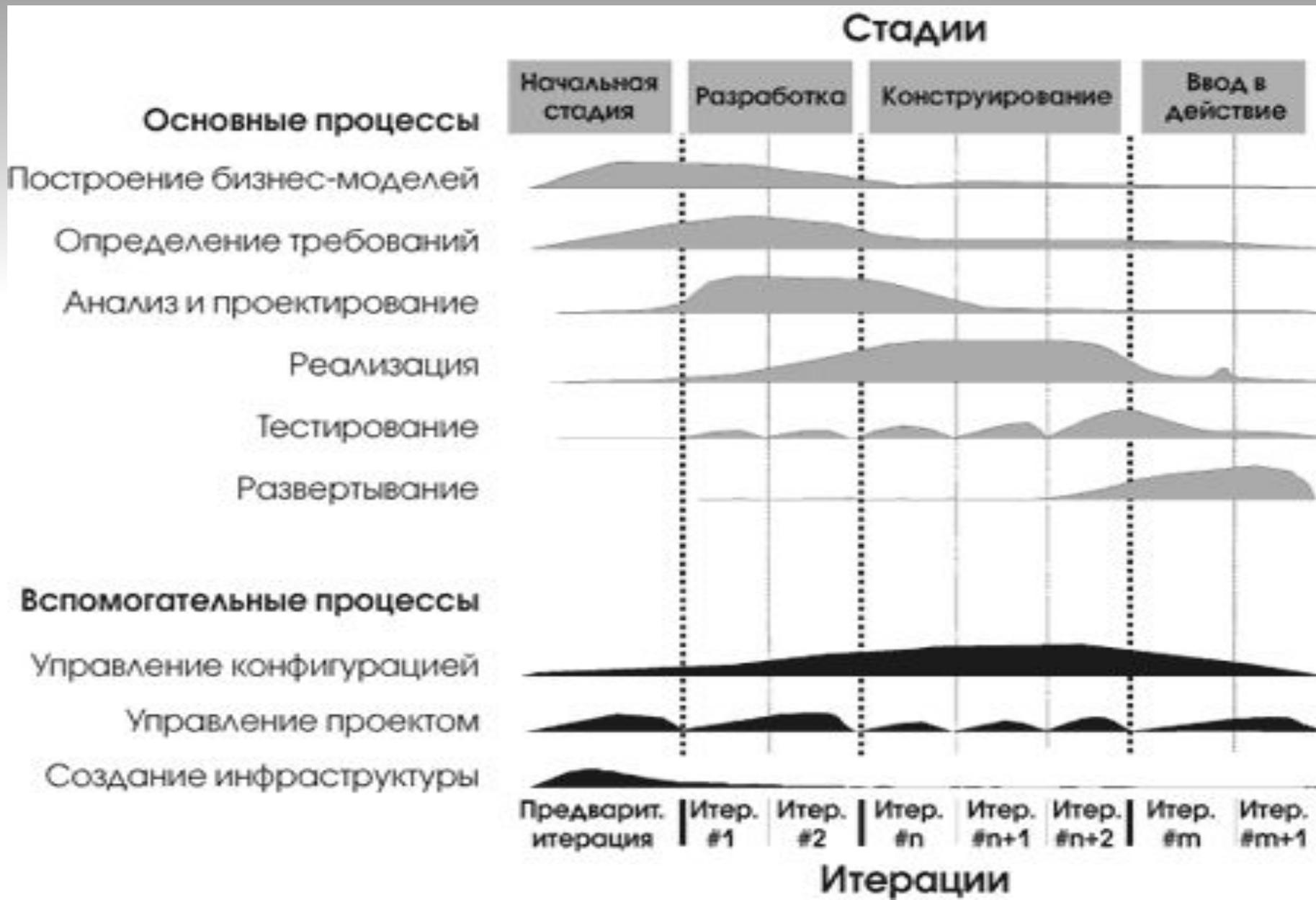
Технология Rational Unified Process (RUP)

RUP соответствует стандартам и нормативным документам, связанным с процессами ЖЦ ПО и оценкой технологической зрелости организаций-разработчиков (ISO 12207, ISO 9000, CMM и др.).

Ее основными принципами являются:

- Итерационный и инкрементный (наращиваемый) подход к созданию ПО.
- Планирование и управление проектом на основе функциональных требований к системе - вариантов использования.
- Построение системы на базе архитектуры ПО.

Общее представление RUP



Начальная стадия RUP

Результаты:

- общее описание системы: основные требования к проекту, его характеристики и ограничения;
- начальная модель вариантов использования (степень готовности – 10-20%);
- начальный проектный глоссарий (словарь терминов);
- начальный бизнес-план;
- план проекта, отражающий стадии и итерации;
- один или несколько прототипов.

Стадия разработки RUP

Результаты:

- модель вариантов использования (завершенная на 80%), определяющая функциональные требования к системе;
- перечень дополнительных требований;
- описание базовой архитектуры будущей системы:
 - модель предметной области;
 - технологическая платформа;
- работающий прототип;
- уточненный бизнес-план;
- план разработки всего проекта, отражающий итерации и критерии оценки для каждой итерации.

Стадия конструирования RUP

- Стадия конструирования заключается в определении последовательности итераций конструирования вариантов использования, реализуемых на каждой итерации.
- **Результатом** стадии является продукт, готовый к передаче конечным пользователям:
 - ПО, интегрированное на требуемых платформах;
 - руководства пользователя;
 - описание текущей реализации.

Стадия ввода в действие

- Стадия ввода в действие предназначена для передачи готового продукта в распоряжение пользователей.
- Данная стадия включает:
 - бета-тестирование, позволяющее убедиться, что новая система соответствует ожиданиям пользователей;
 - параллельное функционирование с существующей системой, которая подлежит постепенной замене;
 - конвертирование баз данных;
 - оптимизацию производительности;
 - обучение пользователей и специалистов службы сопровождения.

Статический аспект RUP

1. **Роль** (role) – определяет поведение и ответственность личности или группы личностей, составляющих проектную команду.
2. **Вид деятельности** (activity) – единица выполняемой работы, соответствует понятию технологической операции. Каждый вид деятельности сопровождается набором руководств (guidelines), представляющих собой методики выполнения технологических операций.
3. **Рабочий продукт** (artifact) – модель, элемент модели, документ, исходный код или план, являющиеся результатом вида деятельности.
4. **Дисциплина** (discipline) – последовательность действий, приводящую к получению значимого результата, соответствует понятию технологического процесса.

Дисциплины RUP

Основные дисциплины:

- 1) построение бизнес-моделей;
- 2) определение требований;
- 3) анализ и проектирование;
- 4) реализация;
- 5) тестирование;
- 6) развертывание.

Вспомогательные дисциплины:

- 1) управление конфигурацией и изменениями;
- 2) управление проектом;
- 3) создание инфраструктуры.

Компоненты RUP

- Описание всех элементов динамического и статического аспекта RUP;
- навигатор по всем элементам RUP, глоссарий и средство быстрого обучения технологии;
- руководства для всех участников проектной команды, охватывающие весь жизненный цикл ПО;
- рекомендации по использованию инструментальных средств, входящих в состав Rational Suite;
- примеры и шаблоны проектных решений для Rational Rose;
- шаблоны проектной документации для SoDa;
- шаблоны в формате Microsoft Word, предназначенные для поддержки документации по всем процессам и действиям жизненного цикла ПО;
- планы в формате Microsoft Project, отражающие итерационный характер разработки ПО.

Инструментальные средства для поддержки RUP

RUP опирается на интегрированный комплекс инструментальных средств *Rational Suite*. Он существует в следующих вариантах:

- *Rational Suite AnalystStudio* – предназначен для определения и управления полным набором требований к разрабатываемой системе;
- *Rational Suite DevelopmentStudio* – предназначен для проектирования и реализации ПО;
- *Rational Suite TestStudio* – представляет собой набор продуктов, предназначенных для автоматического тестирования приложений;
- *Rational Suite Enterprise* – обеспечивает поддержку полного жизненного цикла ПО и предназначен как для менеджеров проекта, так и отдельных разработчиков, выполняющих несколько функциональных ролей в команде разработчиков.

Состав IBM Rational Suite

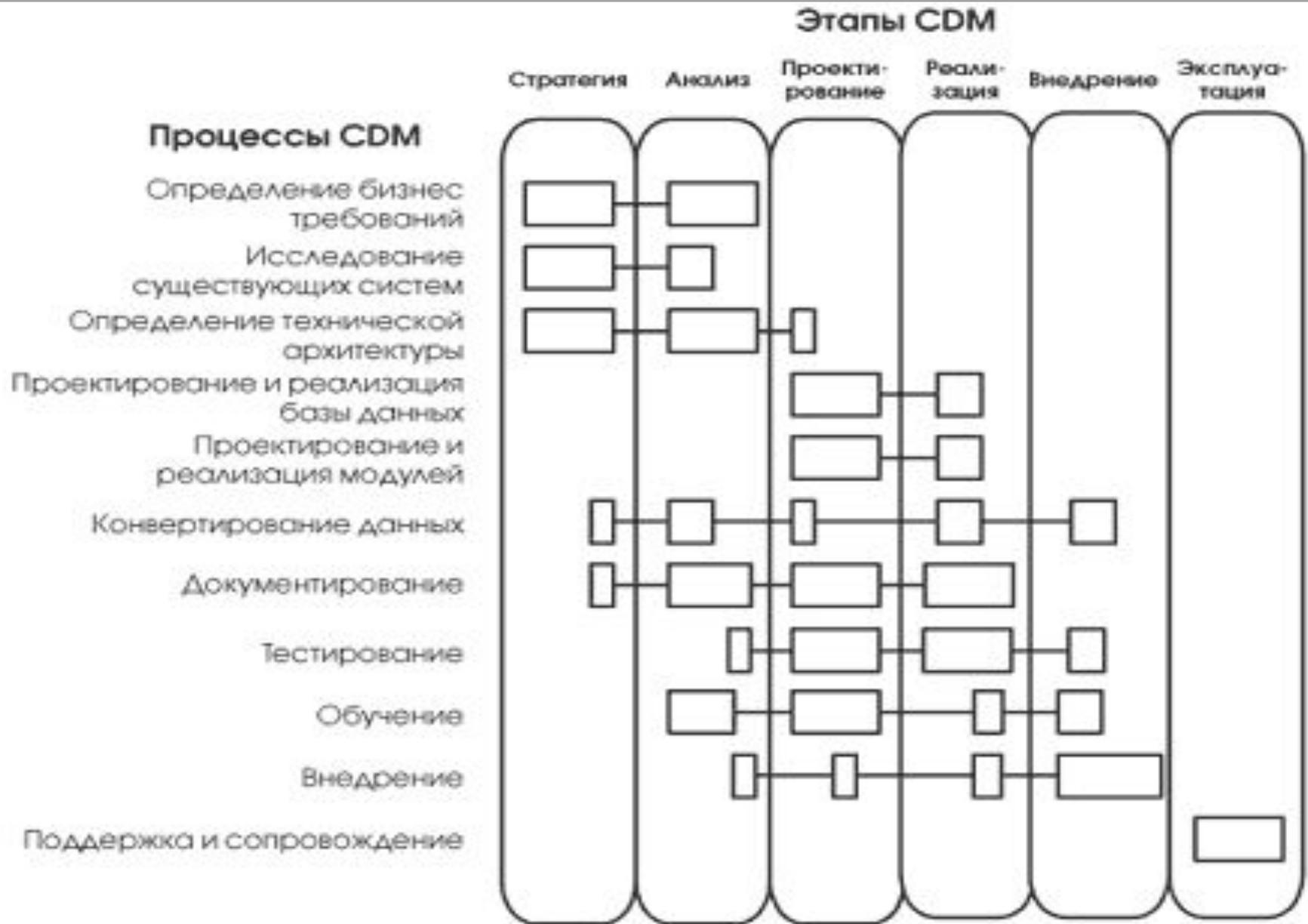
- *IBM Rational RequisitePro* – средство управления требованиями;
- *IBM Rational Rose* – средство визуального моделирования;
- *IBM Rational XDE* – средство генерации объектного кода;
- *IBM Rational RapidDeveloper* – средство разработки;
- *IBM Rational ClearCase* – средство конфигурационного управления;
- *IBM Rational ClearQuest* – средство управления изменениями;
- *IBM Rational SoDA* – средство автоматизированного документирования;
- *IBM Rational Quantify* – средство количественного определения узких мест, влияющих на общую эффективность работы программы;
- *IBM Rational TestManager* – средство планирования функционального и нагрузочного тестирования;
- *IBM Rational Robot* – средство записи и воспроизведения тестовых сценариев;
- *IBM Rational TestFactory* – средство тестирования надежности;
- *IBM Rational Quality Architect* – средство генерации кода для тестирования.

Технология Oracle

- Методическая основа технологии создания ПО корпорации Oracle – комплекс методов, охватывающий большинство процессов ЖЦ ПО.
- В состав комплекса входят:
 - **CDM** (Custom Development Method) – разработка прикладного ПО;
 - **PJM** (Project Management Method) – управление проектом;
 - **AIM** (Application Implementation Method) – внедрение прикладного ПО;
 - **BPR** (Business Process Reengineering) – реинжиниринг бизнес-процессов;
 - **OCM** (Organizational Change Management) – управление изменениями.

Стадии	Предназначение
Стратегия	Определение целей создания системы, приоритетов и ограничений, разработка системной архитектуры и формирование плана разработки.
Анализ	Построение модели информационных потребностей, диаграмм функциональной иерархии, матрицы перекрестных ссылок и диаграмм потоков данных.
Проектирование	Разработка подробной архитектуры системы, схемы реляционной БД и программных модулей, установление перекрестных ссылок между компонентами системы для анализа их взаимного влияния и контроля за изменениями.
Реализация	Создание БД, разработка и тестирование прикладных систем, проверка их качества и соответствия требованиям пользователей, разработка системной документации, материалов для обучения и руководства пользователей.
Внедрение и эксплуатация	Анализ производительности и целостности системы, ее поддержка и модификация.

Этапы и процессы CDM



Критерии выбора метода разработки по CDM

При определении подхода к разработке оценивается:

- масштаб, степень сложности и критичность будущей системы;
- стабильность требований пользователей;
- сложность и количество бизнес-правил;
- количество автоматически выполняемых функций;
- разнообразие и количество пользователей;
- степень взаимодействия с другими системами.

Характеристики	Классический подход	Подход быстрой разработки
Количество этапов	5	4
Характеристики проекта	<ul style="list-style-type: none"> ■ Высокая сложность ■ Большой масштаб ■ Нечетко определенная задача 	<ul style="list-style-type: none"> ■ Несложная архитектура системы ■ Небольшие и средние по масштабу проекты ■ Четкая постановка задачи
Характеристики исполнителей	Невысокая квалификация исполнителей, неподготовленные пользователи	Высококвалифицированные универсальные исполнители, хорошо подготовленные пользователи
Продолжительность проекта	8 – 36 месяцев	4 – 16 месяцев

Процессы РЖМ для разработки ПО в СДМ

1. Управление проектом и предоставление отчетности (*Control and Reporting*).
2. Управление работой (*Work Management*).
3. Управление ресурсами (*Resource Management*).
4. Управление качеством (*Quality Management*).
5. Управление конфигурацией (*Configuration Management*).

Задачи РЖМ и уровни их решения

- Задачи:
 - Задачи планирования
 - Задачи управления
 - Задачи завершения
- Уровни:
 - Уровень всего проекта
 - Уровень отдельных этапов

Комплекс Oracle Developer Suite для быстрой разработки

- *Oracle Designer* - средство моделирования и генерации приложений;
- *Oracle Forms* - средство быстрой разработки приложений;
- *Oracle Reports* - визуальное средство разработки отчетов;
- *Oracle JDeveloper* - средство визуального программирования на языке Java;
- *Oracle Discoverer* - средство для разработки аналитических приложений;
- *Oracle Warehouse Builder* - система для построения хранилищ данных;
- *Oracle Portal* - средство разработки информационного портала организации.

Технология Microsoft Solution Framework

Microsoft Solutions Framework представляет собой согласованный набор концепций, моделей и правил.

Состав MSF:

- Модель процессов;
- Модель проектной группы;
- Дисциплина управления проектами;
- Дисциплина управления рисками;
- Дисциплина управления подготовкой.

Модель проектной группы

- **Модель проектной группы MSF (*MSF Team Model*)** описывает подход *Microsoft* к организации работающего над проектом персонала и его деятельности в целях максимизации успешности проекта.
- **Модель проектной группы основана на:**
 - 6 принципах
 - 6 концепциях
 - 6 ролевых кластерах

Основные принципы модели проектной группы

1. Распределение ответственности при фиксации отчетности
2. Наделение членов команды полномочиями
3. Концентрация на бизнес-приоритетах
4. Единое видение проекта
5. Готовность к переменам
6. Свободное общение членов группы

Ключевые концепции модели проектной группы

1. Проектная группа – команда соратников
2. Сфокусированность на нуждах заказчика
3. Нацеленность на конечный результат
4. Установка на отсутствие дефектов
5. Стремление к самосовершенствованию
6. Заинтересованные команды работают эффективно

Ролевые кластеры

1. **Управление продуктом (product manager)** — бизнес-приоритеты, маркетинг, представительство интересов заказчика.
2. **Управление программой (program manager)** — разработка архитектуры решения, административные службы
3. **Разработка (developer)** — разработка приложений и инфраструктуры, технологические консультации
4. **Тестирование** — планирование, разработка тестов и отчетности по тестам
5. **Управление выпуском (release manager)** — инфраструктура, сопровождение, бизнес-процессы, выпуск готового продукта
6. **Удовлетворение заказчика (user experience)** — обучение, эргономика, графический дизайн, техническая поддержка

	<i>Менеджер продукта</i>	<i>Менеджер програм- мы</i>	<i>Разрабо тчик</i>	<i>Тестиро вщик</i>	<i>Менеджер по выпуску</i>	<i>Спец. по удобству использо- вания</i>
<i>Менеджер продукта</i>		—	—	+	—/+	—/+
<i>Менеджер программы</i>	—		—	—/+	+	—/+
<i>Разработ- чик</i>	—	—		—	—	—
<i>Тестиров- щик</i>	+	—/+	—		+	+
<i>Менеджер по выпуску</i>	—/+	+	—	+		—/+
<i>Спец. по удобству использова- ния</i>	—/+	—/+	—	+	—/+	

Этапы и контрольные точки модели процессов MSF



Создание общей картины приложения

- Определение состава команды;
- определение структуры проекта;
- определение бизнес-целей;
- оценка существующей ситуации;
- создание документа общей картины и области действия проекта;
- определение требований и профилей пользователей;
- разработка концепции решения;
- оценка риска;
- закрытие этапа.

Планирование

- **На этапе концептуального проектирования** задача рассматривается с точки зрения пользовательских и бизнес-требований и заканчивается определением набора сценариев использования системы.
- **На этапе логического проектирования** задача рассматривается с точки зрения проектной команды, решение представляется в виде набора сервисов.
- **На этапе физического проектирования** задача рассматривается с точки зрения программистов, уточняются используемые технологии и программные интерфейсы.

Контрольные точки этапа планирования

- Функциональная спецификация;
- план управления рисками;
- определение среды разработки и тестирования;
- генеральный план и календарный график проекта.

Этап разработки

■ Задачи:

- создание прототипа приложения;
- разработка программных компонентов приложения;
- создание решения (последовательность ежедневных или более частых сборок приложения);
- закрытие разработки (реализация всех функций, поставка кода и документации).

■ Результаты:

- исходный текст кода и исполняемые файлы;
- сценарии установки и конфигурации для развертывания;
- окончательная функциональная спецификация;
- элементы поддержки решения;
- спецификации и сценарии тестирования.

Стабилизация

■ Задачи:

- тестирование компонентов;
- тестирование баз данных;
- тестирование инфраструктуры;
- тестирование защиты;
- тестирование интеграции;
- анализ удобства работы с продуктом;
- нагрузочное тестирование (включая анализ ресурсоемкости и производительности);
- регрессивное тестирование;
- ведение отчетности по тестированию.

■ Результат:

- подтверждение готовности продукта к выпуску и полноценному развертыванию в промышленной среде.

Развертывание

■ Задачи:

- установка решения и необходимых компонентов окружения;
- проведение стабилизации продукта в промышленных условиях;
- передача проекта в руки группы сопровождения;
- анализ проекта в целом на предмет уровня удовлетворенности заказчика.

Открытое управление жизненным циклом приложений (ALM)

ALM – Application Lifecycle Management – включает следующие этапы:

- Defining – определение требований;
- Designing – анализ и проектирование;
- Developing – разработка;
- Testing – тестирование;
- Deploying – развертывание.

Подходы к созданию ИС

1. Разработка (самостоятельно или силами другой компании)
2. Покупка готового решения, его адаптация и настройка под специфику предприятия
3. Покупка ядра ИС и ее модификация
4. Прототипирование
5. Аренда ИС у ASP провайдера (*Application Service Provider*).

Собственная разработка ИС

Достоинства	Недостатки
<ul style="list-style-type: none">■ возможность разработки АИС для конкретных целей предприятия;■ отсутствие функциональных, информационных и других ограничений, присущих готовым АИС;■ повышение степени совместимости АИС с уже использующимися на предприятии системами.	<ul style="list-style-type: none">■ большие затраты ресурсов;■ сложность в определении пользователем своих потребностей;■ необходимость в жестком планировании и контроле над разработкой;■ необходимость адекватной оценки возможностей;■ отсутствие необходимой квалификации у сотрудников.

Приобретение готового решения ИС

Достоинства	Недостатки
<ul style="list-style-type: none">■ минимальные задержки и затраты до внедрения ИС;■ возможность выбора пакета модулей, наиболее соответствующих требованиям организации;■ возможность наглядной оценки функциональных возможностей готового продукта;■ наличие полного пакета документации на ИС.	<ul style="list-style-type: none">■ наличие вероятности того, что разработчик прекратит свое существование или обслуживание ИС;■ отсутствие полного соответствия между возможностями готовых IT-продуктов и потребностями организации;■ выбор и оценка готовых решений требуют дополнительных ресурсов.

Приобретение ядра ИС с последующей модификацией

Достоинства	Недостатки
<ul style="list-style-type: none">■ уменьшение затрат ресурсов организации по сравнению с самостоятельной разработкой;■ преодоление функциональных ограничений;■ повышение степени удовлетворения потребностей организации.	<ul style="list-style-type: none">■ возможность возникновения трудностей при модификации, что порождает новые ошибки и проблемы контроля внедрения;■ усложнение процесса ведения документации по внесенным изменениям;■ возможность отказа со стороны разработчика в обслуживании модифицированных решений.

Прототипирование

- *Прототипирование* – это подход к разработке ИС, при котором создается ее упрощенная действующая модель (прототип).
- **Условия использования:**
 - небольшая команда проектировщиков-универсалов (от 2 до 10 человек);
 - короткий, но тщательно проработанный производственный график (от 2 до 6 мес.);
 - использовании спиральной модели ЖЦ ИС;
 - тесное взаимодействие с заказчиком.

Прототипирование

Достоинства	Недостатки
<ul style="list-style-type: none">■ лучшее определение потребностей пользователей;■ большая вовлеченность пользователей в разработку;■ ускорение времени разработки;■ обнаружение многих ошибок при экспериментах;■ простота внесения изменений;■ меньшая стоимость	<ul style="list-style-type: none">■ большой расход времени пользователей;■ иллюзия готовности ИС;■ низкое качество проектной и эксплуатационной документации;■ сосредоточенность на интерфейсе пользователя в ущерб проработке системных функций;■ дублирование модулей;■ несогласованность данных.

Границы применимости прототипирования

- Объем проекта и требования бизнеса четко определены, не изменяются, а сам проект невелик;
- проект не зависит от других средств автоматизации бизнеса, количество внешних интерфейсов ограничено;
- система ориентирована на экранные формы, обработка данных и системные функции составляют незначительную часть, удобство экранных форм является важнейшим фактором успеха проекта;
- пользователи имеют высокую квалификацию и изначально положительно оценивают идею создания новой системы.

Аренда ИС у ASP провайдера

- **Application Service Providing** – это технология, позволяющая создавать решения по предоставлению в аренду пользователю необходимого набора телекоммуникационных служб и приложений, на основе удаленного доступа к информационному комплексу, на котором установлено специальное программное обеспечение.

Задачи, решаемые с помощью ASP

- хостинг web - сайтов, почтовых служб;
- предоставление в аренду виртуальных торговых площадок для осуществления продаж/покупок через Интернет;
- обеспечение гибко настраиваемого доступа пользователей к различным функциям приложений;
- предоставление защищенного доступа к корпоративным данным;
- поддержка процессов электронного обмена данными;
- предварительная настройка компонентов ERP - систем на типовые задачи, что позволяет максимально сократить время внедрения таких систем в эксплуатацию;
- эксплуатация сложных ERP-систем

Типы ASP-решений

- Офисные и персональные приложения (Microsoft Office, игры, обучающие программы);
- Коммуникационные средства – электронная почта, проведение голосовых и видеоконференций, форум и т.д.;
- Приложения для электронной коммерции – электронные магазины, системы оплаты платежей;
- ERP-системы и отдельные приложения, например, CRM;
- Аналитические приложения – исследования и прогнозирование спроса, рисков и т.д.;
- Группы отраслевых приложений, представляющие собой специфические решения для определенных отраслей промышленности.

Аренда ИС у ASP провайдера

Достоинства	Недостатки
<ul style="list-style-type: none">■ Более низкая стоимость за счет распределения стоимости ASP-решения на нескольких арендаторов;■ гарантия фиксированной оплаты услуг;■ круглосуточная техническая поддержка;■ быстрое обновление оборудования.	<ul style="list-style-type: none">■ обеспечение информационной безопасности;■ обеспечение качественной бесперебойной связи;■ ответственность провайдера услуг при остановке или сбоях в работе сервера за бизнес своих клиентов