


Устройство и функционирование информационных систем



Теоретические основы проектирования ИС

Жизненный цикл ИС

Жизненный цикл ИС

Понятие *жизненного цикла* является одним из базовых понятий методологии проектирования информационных систем.

Жизненный цикл ИС

Жизненный цикл ИС представляет собой **непрерывный процесс**, начинающийся с момента принятия решения о создании системы и заканчивается в момент полного изъятия её из эксплуатации.

Жизненный цикл ИС

Международный стандарт, регламентирующий жизненный цикл ИС называется ISO/IEC 12207.

- *International Organization of Standardization - Международная организация по стандартизации.*
- *International Electrotechnical Commission - Международная комиссия по электротехнике.*

Стандарт ISO/IEC 12207

Стандарт определяет структуру ЖЦ, состоящую из трёх групп:

1. Основные процессы жизненного цикла (приобретение, поставка, разработка, эксплуатация, сопровождение).

Стандарт ISO/IEC 12207

2. Вспомогательные процедуры, обеспечивающие выполнение основных процессов:
документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит.

Стандарт ISO/IEC 12207

3. Организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого жизненного цикла, обучение).

Российский стандарт

Российский стандарт - ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств», идентичный международному стандарту ISO/IEC 12207:2008 «System and software engineering — Software life cycle processes».

Принят Федеральным агентством по техническому регулированию и метрологии РФ 01.03.2012 г.

Структура жизненного цикла информационной системы

В жизненном цикле ИС можно выделить следующие этапы:

- стратегическое планирование,
- анализ,
- проектирование,
- реализация,
- внедрение,
- эксплуатация ИС,
- вывод из эксплуатации.

По версии разработчиков ИС Rational Software ЖЦ ИС содержит:

- Начало;
- Уточнение;
- Конструирование;
- Переход (передача) в эксплуатацию.

Начальная стадия

На начальной стадии идентифицируются все функциональные возможности системы и производится описание наиболее существенных из них, оценивается риск, определяются ресурсы, необходимые для разработки, составляется примерный план с указанием сроков завершения основных этапов.

Стадия конструирования

На стадии конструирования разрабатывается законченное изделие, готовое к передаче пользователю, по окончании этой стадии определяется работоспособность разработанного ПО.

Стадия перехода (передача в эксплуатацию)

На стадии перехода производится передача разработанного программного обеспечения пользователям.

В конце стадии перехода необходимо определить, достигнуты цели разработки или нет.

Модели жизненного цикла информационных систем

Модели ЖЦ ИС

Моделью ЖЦ ИС называется структура, определяющая последовательность осуществления процессов, действий и задач, выполняемых на протяжении жизненного цикла ИС.

- ISO/IEC 12207 Standard for Information Technology — Software Life Cycle Processes (процессы жизненного цикла ПО, есть его российский аналог ГОСТР-1999).
- Определяет общую структуру жизненного цикла ПО в виде 3-х ступенчатой модели, состоящей из процессов, видов деятельности и задач.

Модели ЖЦ ИС

В стандарте ISO/IEC12207 не конкретизируются в деталях методы реализации и выполнения действий и задач, входящих процессы жизненного цикла ИС, а лишь описываются структуры этих процессов.

Модели ЖЦ ИС

Модель ЖЦ зависит от специфики ИС и условий, в которых она создается и функционирует.

Модели ЖЦ ИС

Наибольшее распространение получили модели:

1. Каскадная модель, иногда называемая моделью «водопад» (waterfall);
2. Спиральная модель.

Стандарт ISO/IEC 15288

Модели ЖЦ ИС

ISO/IEC 15288 Standard for Systems Engineering — System Life Cycle Processes (процессы жизненного цикла систем).

Отличается от предыдущего нацеленностью на рассмотрение программно-аппаратных систем в целом.

Модели ЖЦ ИС

В данный момент продолжается работа по приведению этого стандарта в соответствие с предыдущим.

Модели ЖЦ ИС

ISO/IEC 15288 предлагает похожую схему рассмотрения жизненного цикла системы в виде набора процессов.

Каждый процесс описывается набором его результатов (outcomes), которые достигаются при помощи различных видов деятельности.

ISO/IEC 15288

«Системная инженерия - процессы жизненного цикла систем»

- ISO/IEC 15288 – это «конкретизированный ISO 9000 для сверхбольших проектов 21 века», процессный стандарт.
- Хотим внедрить в практику проектирования и строительства сверхбольших систем ряд ключевых идей системной инженерии:
 - системного подхода
 - жизненного цикла
 - инжиниринга требований
 - архитектурного дизайна
 - процессного подхода
 - проектного подхода

V – модель



V-Model (или VEE модель) является моделью разработки информационных систем (ИС), направленной на упрощение понимания сложностей, связанных с разработкой систем.

Используется для определения единой процедуры разработки программных продуктов, аппаратного обеспечения и человеко-машинных интерфейсов.

Каскадная модель

Каскадная модель

Каскадная модель предусматривает последовательную организацию работ.

Основной особенностью является разбиение всей разработки на этапы, переход с этапа на этап происходит только тогда, когда завершены все работы на предыдущем.

Каскадная модель

Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

Каскадная модель

В процессе применения каскадной модели на практике выработались устойчивые этапы разработки не зависящие от предметной области:

- Анализ требований заказчика.
- Проектирование.
- Разработка.
- Тестирование и опытная эксплуатация.
- Сдача готовой продукции.



Основные достоинства каскадной модели:

1. На каждом этапе формируется законченный набор проектной документации, отвечающая критериям полноты и согласованности.

Основные достоинства каскадной модели:

2. На заключительном этапе разрабатывается пользовательская документация, соответствующая стандартам (т.е. все виды обеспечения ИС: организация методов, информационное, программное, аппаратное).

Основные достоинства каскадной модели:

3. Выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения и соответствующие затраты.

Каскадная модель

4. Применение каскадной модели для проектирования ИС предпочтительно для тех ИС, в самом начале разработки которых можно достаточно точно и полно сформулировать все требования, оставляя разработчикам некоторую свободу выбора реализации, наилучшей с технической точки зрения.

К таким ИС относятся сложные расчетные системы, системы реального времени.

Каскадная модель - недостатки

Недостатком такой модели является то, что реальный процесс создания ИС обычно полностью не укладывается в такую жесткую схему.

Каскадная модель - недостатки

Практически постоянно возникает потребность возвращаться к предыдущим этапам, уточнять или пересматривать ранее принятые решения.

Каскадная модель - недостатки

В результате затягиваются сроки получения результатов, а пользователи могут вносить замечания лишь по завершению всех работ с системой.

При этом модели автоматизируемого объекта могут устареть к моменту их утверждения.

Каскадная модель - недостатки

Т.о. проекты, разрабатываемые по каскадной схеме имеют повышенный уровень риска.

Каскадная модель - недостатки

Исследования американской фирмы The Standish Group показали, что в США более 31% проектов заканчиваются неуспехом, 53% допускают перерасход бюджета почти в 2 раза – 189%, и только 16,2% проектов укладываются в срок и в бюджет.

Каскадная модель - недостатки

Существует еще один отрицательный момент, присущий каскадной модели - это влияние человеческого фактора.

Каскадная модель - недостатки

Руководителю проекта часто вместо того, чтобы правильно организовать работу приходится следить за дисциплиной исполнителей, а она чем строже, тем менее творческой становится атмосфера в коллективе.

Каскадная модель - недостатки

Это может привести и к тому, что наиболее одаренные кадры со временем покинут коллектив.

Доделывать и переделывать чужие разработки считается более трудоёмким процессом, чем построение новых.

Поэтапная модель (поэтапный принцип)

Поэтапная модель (поэтапный принцип)

Поэтапная модель обычно включает промежуточный контроль на любом этапе и межэтапные корректировки.

Обеспечивает меньшую трудоемкость по сравнению с каскадной моделью, но время жизни каждого этапа становится равным всему жизненному циклу.

Поэтапная модель

- Межэтапные корректировки позволяют уменьшить трудоемкость процесса разработки по сравнению с каскадной моделью.
- Другое название модели – модель *жизненного цикла с промежуточным контролем.*



Спиральная модель жизненного цикла ИС

Спиральная модель жизненного цикла

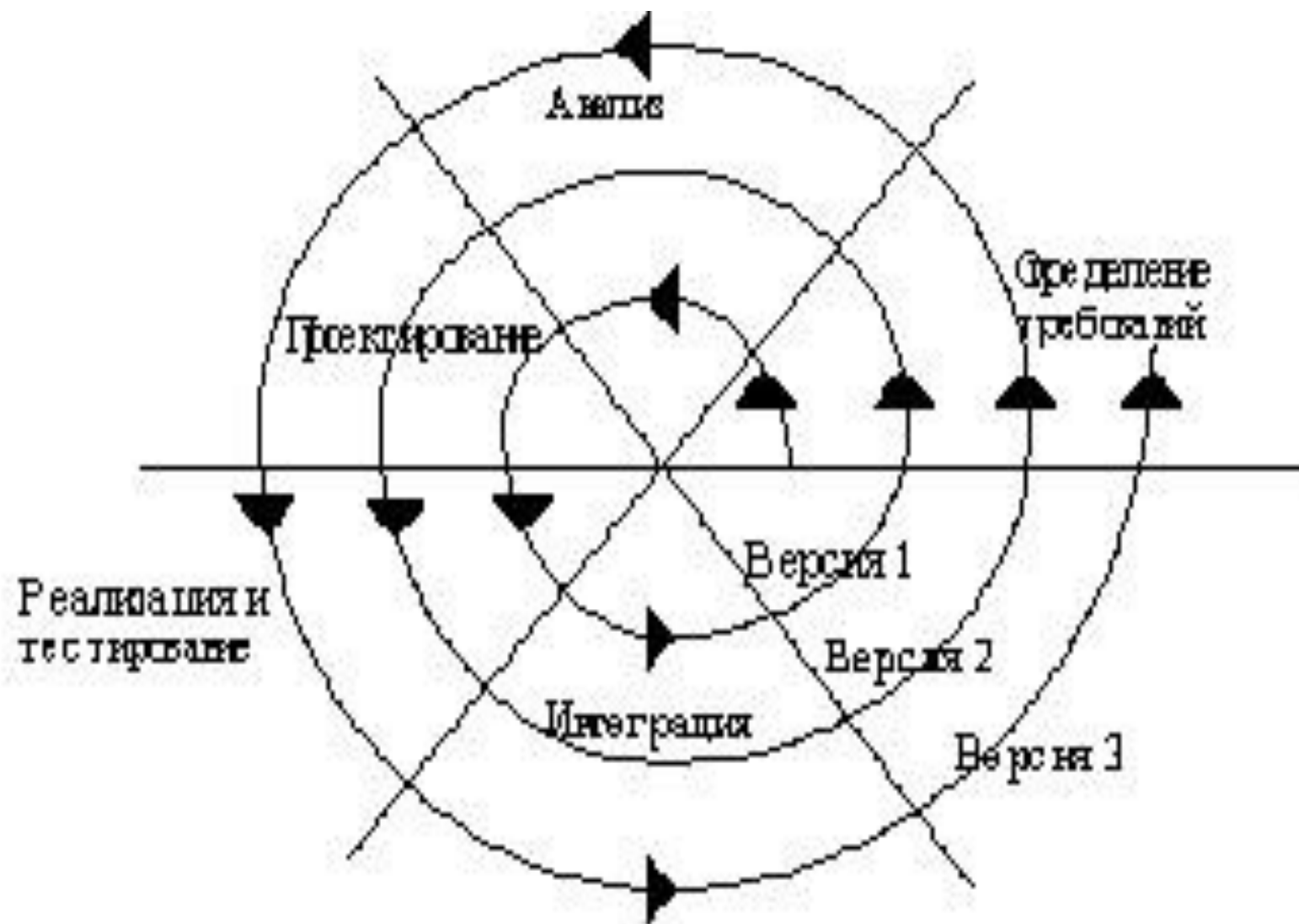
Спиральная модель характеризуется тем, что на начальных этапах ЖЦ осуществляются выработка стратегии, анализ требований и предварительное детальное проектирование.

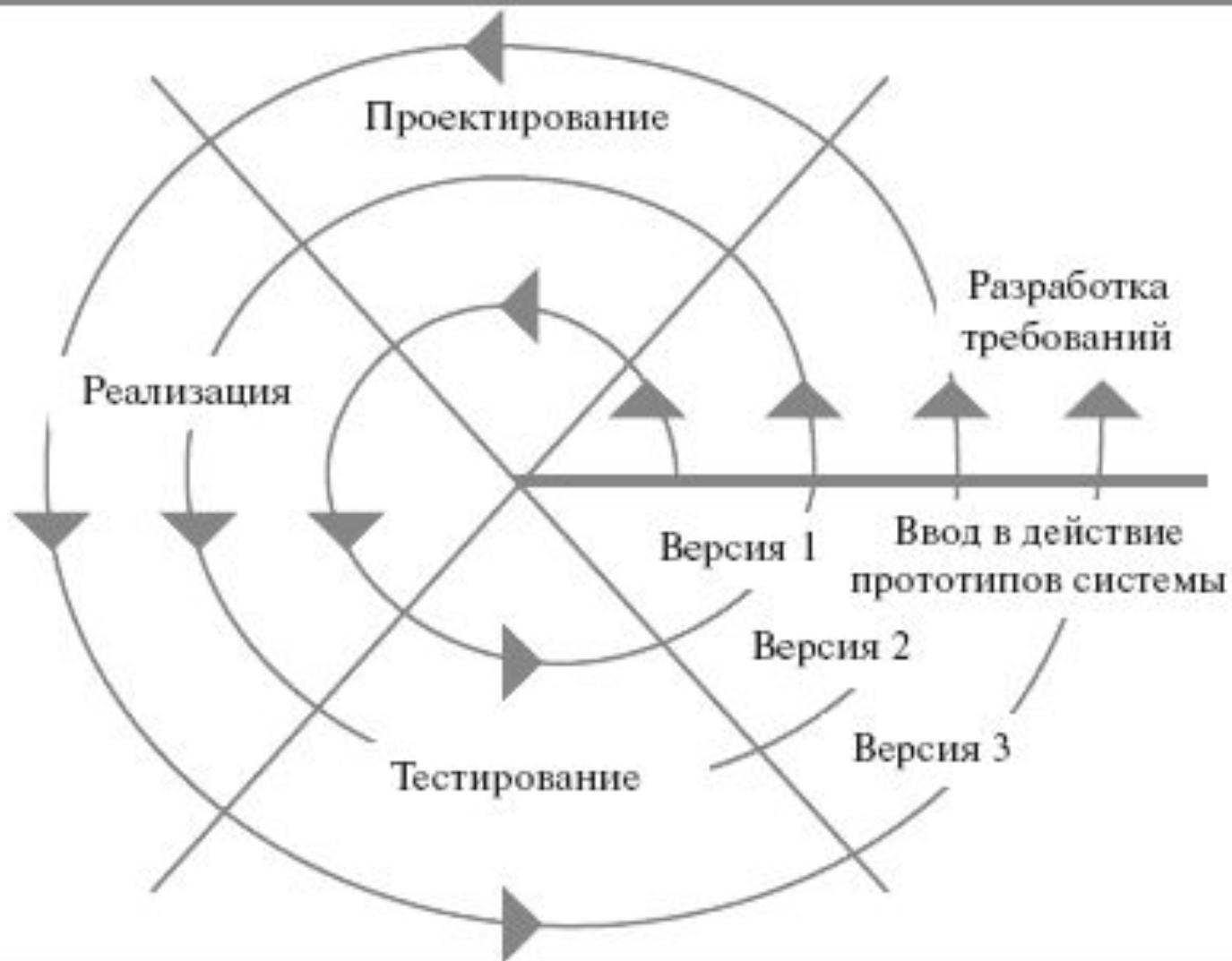
Спиральная модель жизненного цикла

При этом создаются прототипы (макеты), позволяющие проверить и обосновать реализуемость технических решений.

Спиральная модель жизненного цикла

Каждый виток спирали соответствует поэтапной модели создания фрагмента или версии изделия.





Спиральная модель жизненного цикла

Другое название спиральной модели – *итерационная*.

Спиральная модель жизненного цикла

По завершению этапа (витка) уточняются цели и характеристики проекта, определяется его качество, и планируются работы следующего витка спирали.

В результате выбирается обоснованный вариант, который и реализуется.

Преимущества спиральной модели

Преимущества спиральной модели

- Итерационная разработка упрощает внесение изменений в проект.
- При использовании спиральной модели отдельные элементы ИС интегрируются в единое целое постепенно.

Преимущества спиральной модели

- Уменьшается уровень рисков.
- Уровень рисков максимален в начале разработки проекта.

Преимущества спиральной модели

- Итерационная разработка обеспечивает большую гибкость в управлении проектом.

Преимущества спиральной модели

- Например, можно сократить сроки разработки за счет уменьшения функциональности системы или использовать в качестве составных частей системы продукцию сторонних фирм вместо собственных разработок.

Это актуально в условиях конкурентной борьбы, когда надо противостоять продвижению изделия конкурентов.

Недостатки спиральной модели

Недостатки спиральной модели

- Основная проблема спирального цикла — определение момента перехода на следующий этап.

Недостатки спиральной модели

- Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла.

Иначе процесс разработки может превратиться в бесконечное совершенствование уже сделанного.

Недостатки спиральной модели

- При итерационном подходе полезно следовать принципу «лучшее — враг хорошего».

Поэтому завершение итерации должно производиться строго в соответствии с планом, даже если не вся запланированная работа закончена.

Недостатки спиральной модели

- Планирование работ обычно проводится на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

Зависимость рисков от времени разработки



- Технологии CASE

- Модели жизненного цикла ИС объединяются в методологии средств разработки **CASE.**



CASE (Computer Aided Software Engineering) – методология разработки ПО (ИС), основанная на комплексном использовании компьютеров не только для разработки исходного кода, но и для анализа и моделирования соответствующей предметной области.

Парадигма CASE-технологий



CASE-средства (CASE-tools) – ПО, которое предназначено для разработки визуальных моделей программных систем и генерации исходного кода или схемы базы данных на некотором языке.

Computer Aided Software Engineering.

Первоначальное значение термина CASE - автоматизация разработки ПО, в настоящее время охватывает процесс разработки сложных ИС в целом.



- В настоящее время под термином **CASE-средства** понимаются программные средства (оболочки), поддерживающие процессы создания и сопровождения информационных систем, включая анализ и формулировку требований, проектирование прикладного ПО (приложений) и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.



- CASE-средства вместе с системным ПО и техническими средствами образуют полную (интегрированную) среду разработки информационных систем (ИС).

CASE



Появлению CASE технологий и средств предшествовали исследования в области методологии программирования

CASE

Программирование обрело черты *системного подхода* с разработкой и внедрением *языков высокого уровня, методов структурного и модульного программирования, языков проектирования и средств их поддержки, формальных и неформальных языков описаний системных требований и спецификаций* и т.д.

CASE

Также появлению CASE способствовали факторы:

- подготовка аналитиков, программистов и экономистов, обученных концепциям модульного и структурного программирования;

CASE

- широкое внедрение и постоянный рост производительности компьютеров, позволившие использовать графические средства и автоматизировать большинство этапов проектирования.

CASE

- CASE-инструменты классифицируются по типам и категориям :
 - построение и анализа модели предметной области;
 - проектирования баз данных;
 - разработки приложений;
 - реинжиниринга процессов;
 - планирования и управления проектом;
 - тестирования;
 - документирования.

CASE

- Типичные CASE-инструменты:
 - управления конфигурацией;
 - моделирования данных;
 - анализа и проектирования;
 - преобразования моделей;
 - редактирования программного кода;
 - рефакторинга кода;
 - генераторы кода;
 - построения UML-диаграмм.

Методология RAD- Rapid Application Development

RAD

Методология разработки ИС, основанная на использовании средств быстрой разработки приложений - RAD.

RAD

RAD- это комплекс специальных инструментальных средств быстрой разработки прикладных ИС, позволяющих оперировать с определенным набором графических объектов, функционально отображающих отдельные информационные компоненты приложений.

RAD

Процесс быстрой разработки приложений основан на 3 основных элементах:

1. небольшая команда программистов (от 2 до 10);
2. тщательно разработанный производственный график работ, рассчитанный на сравнительный короткий срок разработки (от 2 до 6 мес);

RAD

Итерационный цикл



3. итерационная (спиральная) модель разработки, основанная на тесном взаимодействии с заказчиком – повторяющийся цикл.

Основные принципы RAD:

RAD (Rapid Application Development — быстрая разработка приложений)



Использование спиральной модели разработки предполагает:

1. Полное завершение работ на каждом этапе жизненного цикла не обязательно.
2. В процессе разработки ИС обязателен тесный контакт с заказчиком и будущими пользователями.

Основные принципы и методологии RAD:

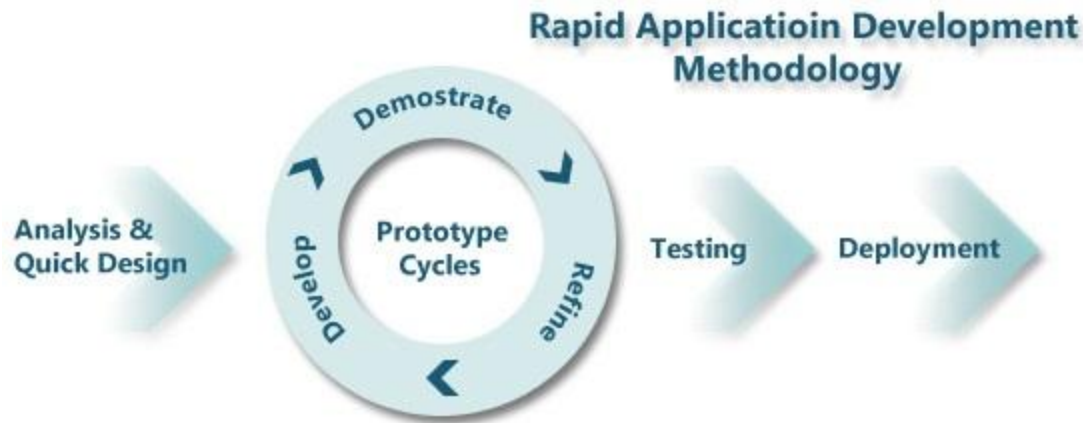
3. Необходимость применения CASE-средств и средств быстрой разработки приложений.
4. Применение средств управления конфигурацией, позволяющие вносить изменения.



Принципы RAD технологии направлены на обеспечение трёх основных её преимуществ — высокой скорости разработки, низкой стоимости и высокого качества.

RAD

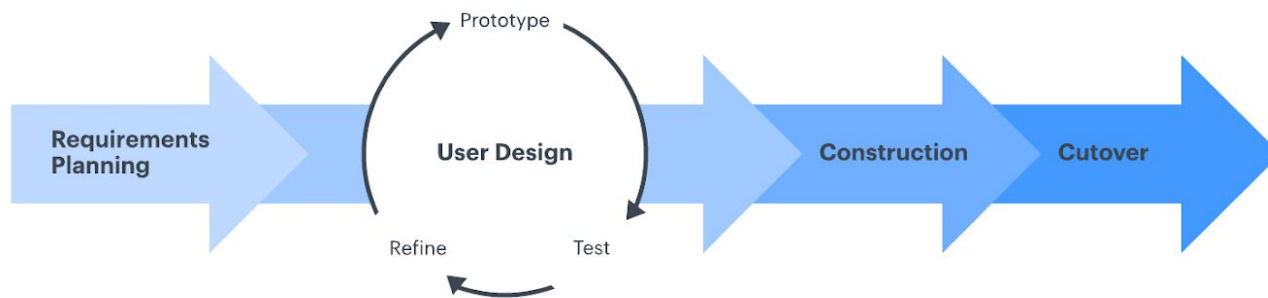
- Для этого:
- Инструментарий должен быть нацелен на минимизацию времени разработки.
- Создание *прототипа* для уточнения требований заказчика.
- Цикличность разработки: каждая новая версия продукта основывается на оценке результата работы предыдущей версии заказчиком.



RAD

- Минимизация времени разработки версии, за счёт переноса уже готовых модулей и добавления функциональности в новую версию.
- Команда разработчиков должна тесно сотрудничать, каждый участник должен быть готов выполнять несколько обязанностей.
- Управление проектом должно минимизировать длительность цикла разработки.

Rapid Application Development (RAD)



RAD

RAD-технология не является универсальной, её применение целесообразно не всегда.

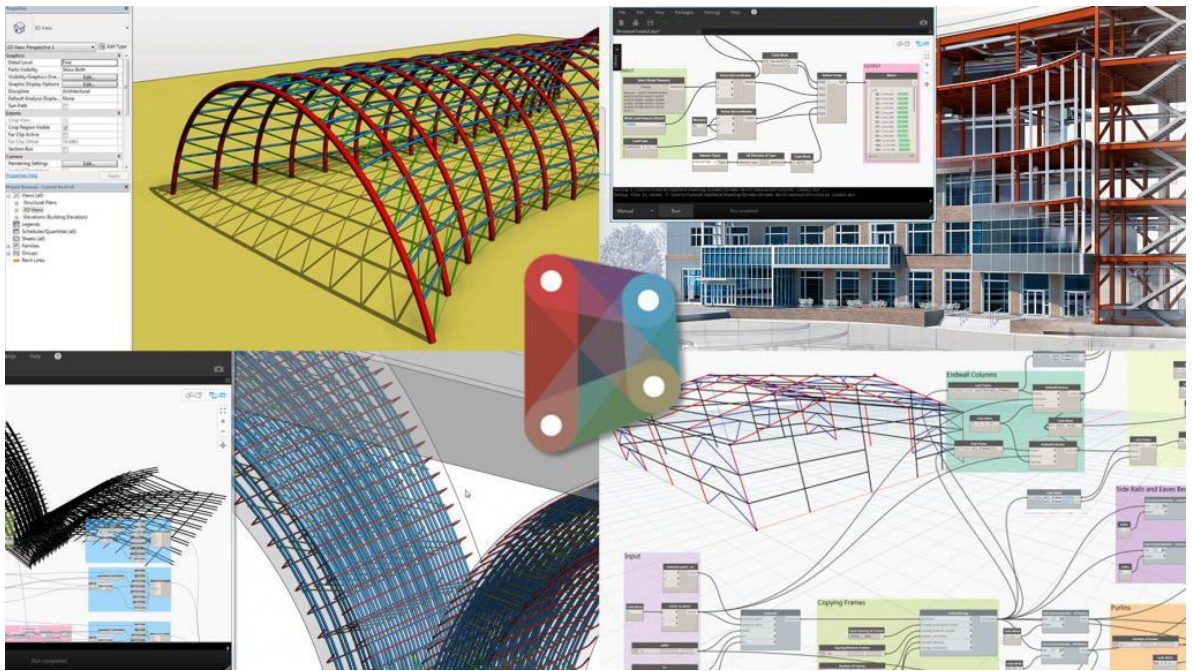
Например, в проектах, где требования к ПО четко определены и не должны меняться, вовлечение заказчика в процесс разработки не требуется и более эффективной может быть иерархическая разработка (каскадный метод).

RAD

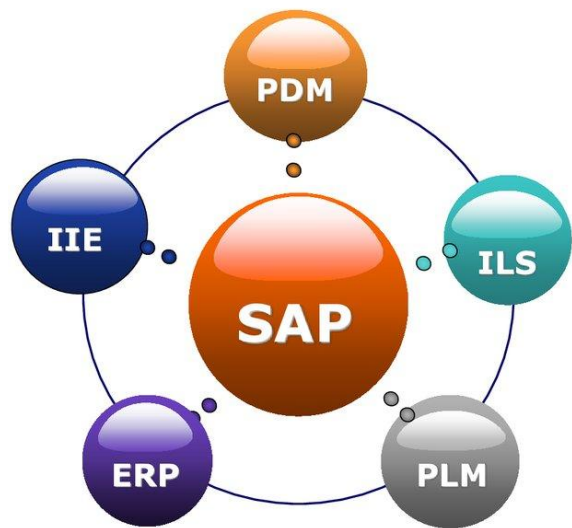
RAD-технология не в состоянии обеспечивать разработку сложных продуктов, содержащих много фрагментов, программирование которых занимает более двух недель.

Эта технология ориентирована на разработку достаточно простого заказного ПО, чем на индустриальное проектирование ИС.





Cals - ТЕХНОЛОГИИ



CALS-технологии (*Continuous Acquisition and Life cycle Support* — непрерывная информационная поддержка поставок и жизненного цикла изделий), или **ИПИ** (информационная поддержка процессов жизненного цикла изделий) — подход к проектированию и производству высокотехнологичной и наукоёмкой продукции, заключающийся в использовании компьютерной техники и информационных технологий на всех стадиях жизненного цикла изделия.



Интенсивное развитие информационных технологий в областях коммуникации и связи, проектирования сложных систем и разработки новых ИС создало предпосылки для CALS-технологии.

На сегодняшний день все виды человеческой деятельности используют ИС.

Медицина, промышленность, игровая индустрия, научная и инновационная деятельность - различное ПО облегчает процессы, и являются их критически необходимым элементом.

Суть Cals заключается в создании системы управления, которая автоматизирует не отдельные виды работ, функций и задач, а позволяет формализовать все процессы деятельности предприятия - проектирование, производство, снабжение, сбыт, послепродажный сервис.

Базовые принципы CALS

ЕДИНОЕ ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО

Маркетинг

Проектирование

Производство

Поставки

Эксплуатация

1

2

3

4

5



Базовые технологии
управления данными

Базовые технологии
управления процессами

Основные принципы CALS-технологий базируются на контроле и организации этапов существования продукции. К ним относятся:

- обеспечение системного управления (использование специальных информационных пространств);
- минимизацию затрат на всех стадиях;
- использование стандартных механизмов описания управляемых объектов (интеграция информационных потоков);

- дифференциацию программных элементов на основе использования общих стандартов (данных и интерфейсов доступа) и применение стандартных платформ;
- представление информации на безбумажной основе;
- сопутствующий инжиниринг все процессов;
- непрерывное корректирование и усовершенствование с целью создания оптимальной модели управления.



Cals технологии

- Большинство ИС сегодня могут функционировать автономно и это активно используется.
- Но результаты применения CALS –технологии будут заметно больше. Все данные, которые генерируются в единой системе, могут быть доступны и другим системам. Все применяемые в них решения станут намного более обоснованными и понятными.

- Для того, чтобы выйти на требуемый уровень ИС, необходимо сформировать общее информационное пространство. Обеспечивается такое пространство при помощи унификации содержания и формы информации о продукции во всех производственных процессах её создания.



CALS-технологии дают возможность существенно уменьшить себестоимость всех выпускаемых предприятием изделий.

При этом качество увеличивается, а эксплуатация становится более удобной.

История создания CALS

В 1985 г. мин. обороны США объявило планы создания глобальной автоматизированной системы электронного описания всех этапов проектирования, производства и эксплуатации продуктов военного назначения.

Для координации работ по инициативе CALS был создан управляющий совет НАТО по CALS — NATO CALS Management Board, существовавший до 2001 года.

За прошедшие годы CALS получила широкое развитие в оборонной промышленности и военно-технической инфраструктуре МО США.

Это позволило ускорить выполнение НИОКР на 30—40 %, уменьшить затраты на закупку военной продукции на 30 %, сократить сроки закупки ЗИП на 22 %, а также в 9 раз сократить время на корректировку проектов.

CALS-технологии в России используются на многих отечественных предприятиях, как гражданского, так и военного сектора.

Используется электронная документация для многих изделий.

К примеру, в авиации для самолетов, вертолетов, авиационных двигателей и комплектующих; разработки систем навигации, телефонной и радио связи, управления; при проектировании и разработке автотракторной техники и др.



Элементы Cals
используются на
Воронежском
механическом заводе,
в государственной
корпорации
"Росатом", НПП
"Аэросила", ОАО
"Российские
железные дороги" и
др.

Тема 2.2. Основные понятия технологии проектирования информационных систем.

Общие сведения об управлении проектами

Понятие проекта

Понятие проекта

Любая Информационная Система (ИС) разрабатывается как некоторый проект.

Понятие проекта

Проект – это ограниченная по времени и ресурсам целенаправленная деятельность:

- С четко определенными целями,
- С установленными требованиями:
 - к срокам,
 - результатам,
 - риску,
 - рамкам расходования средств
 - ресурсов
 - к организационной структуре.

Понятие проекта

- Проекты характеризуются:
- изменчивостью (целенаправленный перевод системы из существующего в некоторое желаемое состояние);
- ограниченностью конечной цели;
- ограниченностью продолжительности;
- ограниченностью бюджета;

Понятие проекта

- ограниченностью требуемых ресурсов;
- новизной для предприятия, для которого реализуется проект;
- комплексностью (наличие большого числа факторов, прямо или косвенно влияющих на прогресс и результаты проекта);
- правовым и организационным обеспечением.

Характеристики проекта

Характеристики проекта

- Объем работ;
- Сроки выполнения;
- Себестоимость;
- Экономическая эффективность, обеспечиваемая реализацией проекта;
- Социальная и общественная значимость проекта и др.

Классификация проектов

Классификация проектов

По составу и структуре проекта различают:

- Монопроект (отдельный проект, который может быть любого типа, вида и масштаба);
- Мультипроект (комплексный проект, состоящий из монопроектов и требующий применения многопроектного управления);

Классификация проектов

По типу:

- технический
- организационный
- экономический
- социальный
- смежный
- и пр.

ИС как проект

Разработка ИС относится к техническим проектам и имеют особенности:

- главная цель проекта четко определена, но отдельные цели должны уточняться по мере достижения частных результатов;

ИС как проект

- срок завершения и продолжительность проекта определены заранее, однако они также могут корректироваться в зависимости от полученных промежуточных результатов и общего прогресса проекта.

Классификация проектов

Масштаб проекта:

- мелкие проекты;
- малые проекты;
- средние проекты;
- крупные проекты.

Основные фазы проектирования

Основные фазы проектирования

Выделяют следующие фазы развития проекта ИС:

1. Формирования концепции;
2. Разработка технического задания;
3. Проектирование;
4. Изготовление;
5. Ввод системы в эксплуатацию.

Основные фазы проектирования

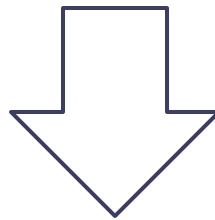
Вторую и частично третью фазу принято называть фазами системного проектирования, а последние две – фазами реализации.

Формирования концепции

Концептуальная фаза проекта

Содержание работ:

1. Формирование идеи, постановку целей;
2. Формирования ключевой команды проекта;
3. Изучения мотивации и требования заказчика и других участников;



Концептуальная фаза проекта

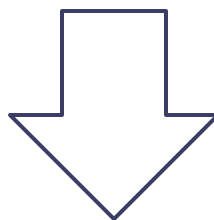
4. Сбор исходных данных и анализ существующего состояния;
5. Определения основных требований и ограничений, материальных, финансовых и трудовых ресурсов.
6. Сравнительную оценку альтернатив.
7. Представления предложений, их экспертизу и утверждение.

Разработка технического предложения (задания)

Разработка ТЗ

Перечень работ этой фазы :

1. Разработка основного содержания проекта, базовой структуры проекта.
2. Разработка и утверждение документа - технического задания (ТЗ).



Разработка ТЗ

3. Планирование, декомпозиция базовой структурной модели проекта.
4. Составление сметы и бюджета проекта, определение потребности в ресурсах.
5. Разработка календарных планов укрупненных графиков работы.
6. Подписание контракта с заказчиком.
7. Ввод в действие средств коммуникации участников проекта и контроля за ходом работ.

Проектирование

Проектирование

На этой фазе определяются подсистемы, их взаимосвязи, выбираются наиболее эффективные способы выполнения проекта и использование ресурсов.

Проектирование

Работы:

1. Выполнение базовых проектных работ;
2. Разработка частных технических заданий.
3. Выполнение концептуального проектирования.
4. Составление технических спецификаций и инструкций.
5. Представление проектной разработки, экспертизы и утверждение.

Разработка

Разработка

Производится координация и оперативный контроль работ по проекту, осуществляется изготовление подсистем, их объединение и тестирование:

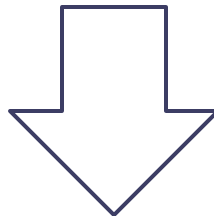
1. Разработка программного обеспечения (ПО).
2. Выполнение подготовки по внедрению системы.
3. Контроль и регулирование основных показателей проекта.

Ввод системы в эксплуатацию

Ввод системы в эксплуатацию

Виды работ:

1. Комплексные испытания.
2. Подготовка кадров для эксплуатации создаваемой системы.
3. Подготовка рабочей документации, сдача системы заказчику и ввод ее в эксплуатацию.



Ввод системы в эксплуатацию

4. Сопровождение, поддержка, сервисное обслуживание.
5. Оценка результатов проекта и подготовка итоговых документов.
6. Разрешение конфликтных ситуаций и закрытие работ по проекту.
7. Накопление опытных данных для последующих проектов, анализ опыта, состояние, определение направлений развития.

В управлении проектами ИС существуют дополнительные этапы, например, закрытие контрактов проекта, сопровождение, постаудит и др.,

*подробнее об этом на предмете
«Управление проектами» 4курса!*

