

Лекция 2

по дисциплине: «Компьютерные технологии в науке и производстве в области информационных систем»

Тема: Жизненный цикл программного обеспечения ИС.

Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде **жизненного цикла (ЖЦ)** ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Жизненный цикл ИС можно представить как ряд событий, происходящих с системой в процессе ее создания и использования.

Модель жизненного цикла отражает различные состояния системы, начиная с момента возникновения необходимости в данной ИС и заканчивая моментом ее полного выхода из употребления. **Модель ЖЦ** - структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.

Модели жизненного цикла

Каскадная модель предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.



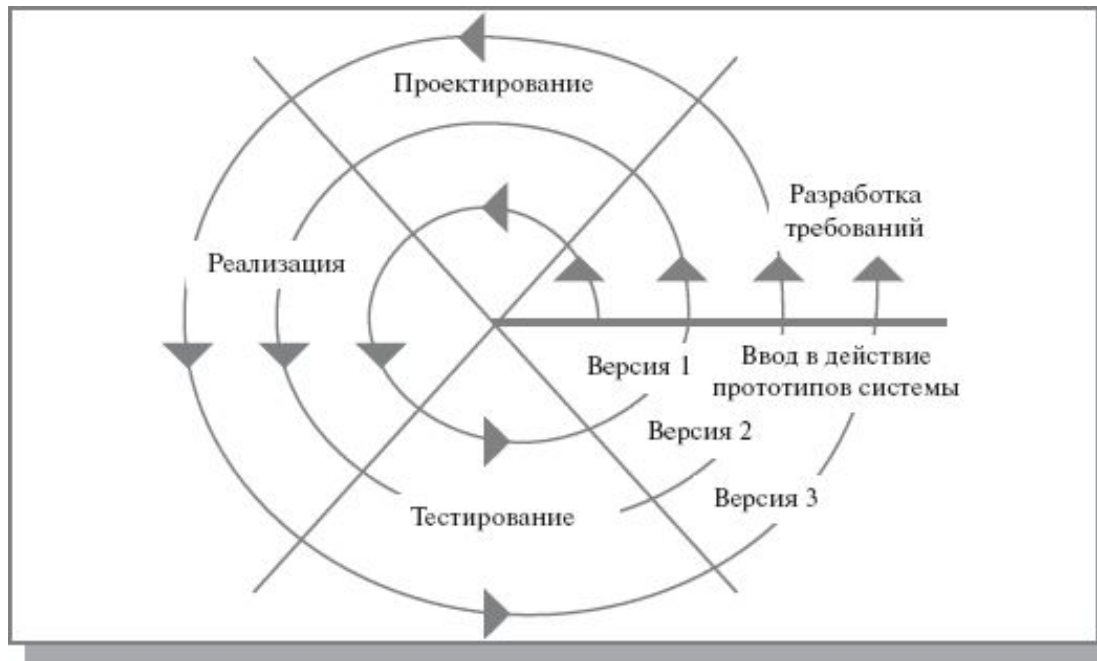
Модели жизненного цикла

Поэтапная модель с промежуточным контролем. Разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.



Модели жизненного цикла

Спиральная модель. На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки - анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).



На практике наибольшее распространение получили **две основные модели жизненного цикла**:

- *каскадная модель* (характерна для периода 1970-1985 гг.);
- *спиральная модель* (характерна для периода после 1986 г.).

В ранних проектах достаточно простых ИС каждое приложение представляло собой единый, функционально и информационно независимый блок. Для разработки такого типа приложений эффективным оказался каскадный способ. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

Можно выделить следующие **положительные стороны применения каскадного подхода**:

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты

Каскадный подход хорошо зарекомендовал себя при построении относительно **простых ИС**, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к системе. **Основным недостатком** этого подхода является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания ИС оказывается соответствующим ***поэтапной модели с промежуточным контролем***.

Однако и эта схема не позволяет оперативно учитывать возникающие изменения и уточнения требований к системе. Согласование результатов разработки с пользователями производится только в точках, планируемых после завершения каждого этапа работ, а общие требования к ИС зафиксированы в виде технического задания на все время ее создания. Таким образом, пользователи зачастую получают систему, не удовлетворяющую их реальным потребностям.

Спиральная модель ЖЦ была предложена для преодоления перечисленных проблем. На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы. Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.

Итеративная разработка отражает объективно существующий спиральный цикл создания сложных систем. Она позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем и решить главную задачу - как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

Основная проблема спирального цикла - определение момента перехода на следующий этап. Для ее решения вводятся временные ограничения на каждый из этапов *жизненного цикла*, и переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. Планирование производится на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

Несмотря на настойчивые рекомендации компаний - вендоров и экспертов в области проектирования и разработки ИС, многие компании продолжают использовать *каскадную модель* вместо какого-либо варианта итерационной модели. **Основные причины**, по которым *каскадная модель* сохраняет свою популярность, следующие:

- Привычка
- Иллюзия снижения рисков участников проекта
- Проблемы внедрения при использовании итерационной модели

Каждая из стадий создания системы предусматривает выполнение определенного объема работ, которые представляются в виде **процессов ЖЦ**.

Процесс определяется как совокупность взаимосвязанных действий, преобразующих входные данные в выходные. Описание каждого процесса включает в себя перечень решаемых задач, исходных данных и результатов.

Существует **целый ряд стандартов**, регламентирующих ЖЦ ПО, а в некоторых случаях и процессы разработки.

Значительный вклад в теорию проектирования и разработки информационных систем внесла компания IBM, предложив еще в середине 1970-х годов **методологию BSP (Business System Planning - методология организационного планирования)**.

Метод структурирования информации с использованием матриц пересечения бизнес-процессов, функциональных подразделений, функций систем обработки данных (информационных систем), информационных объектов, документов и баз данных, предложенный в BSP, используется сегодня не только в ИТ-проектах, но и проектах по реинжинирингу бизнес-процессов, изменению организационной структуры.

Важнейшие шаги процесса BSP, их последовательность (получить поддержку высшего руководства, определить процессы предприятия, определить классы данных, провести интервью, обработать и организовать данные интервью) можно встретить практически во всех формальных методиках, а также в проектах, реализуемых на практике.

Среди наиболее **известных стандартов** можно выделить следующие:

- **ГОСТ 34.601-90** - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания.
- **ISO/IEC 12207:1995** - стандарт на процессы и организацию *жизненного цикла*.
- **Custom Development Method** (методика Oracle) по разработке прикладных информационных систем.
- **Rational Unified Process (RUP)** предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Суть работы в рамках RUP - это создание и сопровождение моделей на базе UML.
- **Microsoft Solution Framework (MSF)** сходна с RUP, но в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.
- **Extreme Programming (XP)**. Экстремальное программирование. В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС, а разработка ведется с использованием последовательно дорабатываемых прототипов.

В соответствии с базовым международным стандартом **ISO/IEC 12207** все *процессы ЖЦ ПО* делятся на три группы:

1. Основные процессы:

- приобретение;
- поставка;
- разработка;
- эксплуатация;
- сопровождение.

2. Вспомогательные процессы:

- документирование;
- управление конфигурацией;
- обеспечение качества;
- разрешение проблем;
- аудит;
- аттестация;
- совместная оценка;
- верификация.

3. Организационные процессы:

- создание инфраструктуры;
- управление;
- обучение;
- усовершенствование.