Модели жизненного цикла

Выполнила: Ананьева Алиса
2-1П11

Классификация моделей жизненного цикла

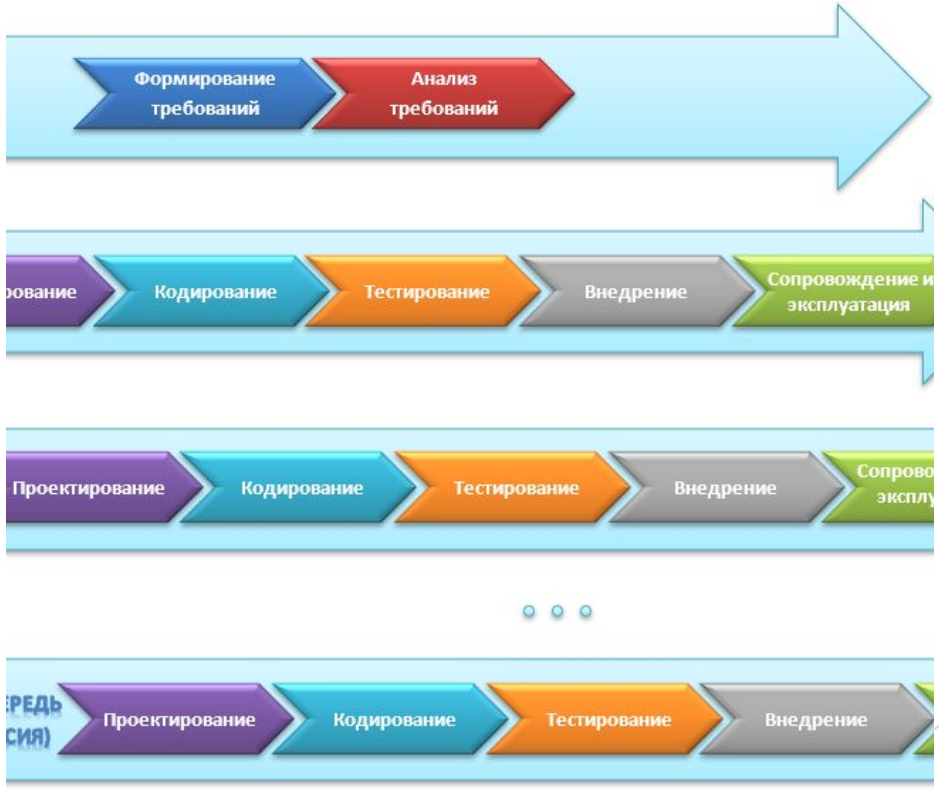
К настоящему времени наибольшее распространение получили следующие модели (стратегии) жизненного цикла [4, 7].

Рис.1. Классификация моделей жизненного цикла

Дальнейшее рассмотрение моделей жизненного цикла ведется с использованием терминологии классического жизненного цикла.

Инкрементная стратегия

- Инкрементная стратегия (англ. increment – увеличение, приращение) подразумевает разработку информационной системы с линейной последовательностью стадий, но в несколько инкрементов (версий), т. е. с запланированным улучшением продукта.



Данная модель жизненного цикла характерна при разработке сложных и комплексных систем, для которых имеется четкое видение (как со стороны заказчика, так и со стороны разработчика) того, что собой должен представлять конечный результат (информационная система). Разработка версиями ведется в силу разного рода причин:

- отсутствия у заказчика возможности сразу профинансировать весь дорогостоящий проект;

- отсутствия у разработчика необходимых ресурсов для реализации сложного проекта в сжатые сроки;

- требований поэтапного внедрения и освоения продукта конечными пользователями. Внедрение всей системы сразу может вызвать у ее пользователей неприятие и только «затормозить» процесс перехода на новые технологии. Образно говоря, они могут просто «не переварить большой кусок, поэтому его надо измельчить и давать по частям».

Достоинства и недостатки

Достоинства и **недостатки** этой стратегии такие же, как и у классической. Но в отличие от классической стратегии заказчик может раньше увидеть результаты. Уже по результатам разработки и внедрения первой версии он может незначительно изменить требования к разработке, отказаться от нее или предложить разработку более совершенного продукта с заключением нового договора

Спиральная стратегия

Спиральная стратегия (эволюционная или итерационная модель, автор Барри Боэм, 1986-88 гг.) [44] подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий.

Данная модель жизненного цикла характерна при разработке новаторских (нетиповых) систем. В начале работы над проектом у заказчика и разработчика нет четкого видения итогового продукта (требования не могут быть четко определены) или стопроцентной уверенности в успешной реализации проекта (риски очень велики). В связи с этим принимается решение разработки системы по частям с возможностью изменения требований или отказа от ее дальнейшего развития. Как видно из рис.3.4, развитие проекта может быть завершено не только после стадии внедрения, но и после стадии анализа риска.

Достоинства модели:

- - позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;
- - допускает изменение требований при разработке информационной системы, что характерно для большинства разработок, в том числе и типовых;
- - обеспечивает большую гибкость в управлении проектом;
- - позволяет получить более надежную и устойчивую систему. По мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации;
- - позволяет совершенствовать процесс разработки – анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации;
- - уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта

Недостатки модели:

- увеличивается неопределенность у разработчика в перспективах развития проекта. Этот недостаток вытекает из предыдущего достоинства модели;

- затруднены операции временного и ресурсного планирования всего проекта в целом. Для решения этой проблемы необходимо ввести временные ограничения на каждую из стадий жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа выполнена. План составляется на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков.

Сравнительный анализ моделей

- Знание различных моделей жизненного цикла и умение их применять на практике необходимы любому руководителю проекта. Правильный выбор модели позволяет грамотно планировать объемы финансирования, сроки и ресурсы, необходимые для выполнения работ, сократить риски как разработчика, так и заказчика. Это способствует повышению авторитета (имиджа) разработчиков в глазах заказчика и в свою очередь оказывает влияние на перспективу дальнейшего сотрудничества с ним и другими заказчиками. Считать, что спиральная модель лучше остальных, неверно. Ведь на каждый проект заключается отдельный договор с определенной стоимостью. Заключать договор на большую сумму с неопределенным итоговым результатом заказчик никогда не будет (если только он не альтруист). В этом случае он предложит вложить вначале небольшую сумму в проект и уже по результатам первой версии (итерации) будет решать вопрос о заключении дополнительного договора на развитие системы.
- Каждая из моделей имеет свои достоинства и недостатки, а также сферы применения в зависимости от специфики разрабатываемой системы, возможностей заказчика и разработчика и т. п. В табл. 3.1 приводится сравнительная характеристика рассмотренных выше моделей, которая должна помочь в выборе стратегии для конкретного проекта.

Характеристика проекта	Модель (стратегия)		
	Каскадная	Инкрементная	Спиральная
Новизна разработки и обеспеченность ресурсами	Типовой. Хорошо проработаны технология и методы решения задачи		Нетиповой (новаторский). Нетрадиционный для разработчика
	Ресурсов заказчика и разработчика хватает для реализации проекта в сжатые сроки	Ресурсов заказчика или разработчика не хватает для реализации проекта в сжатые сроки	
Масштаб проекта	Малые и средние проекты	Средние и крупные проекты	Любые проекты
Сроки выполнения проекта	До года	До нескольких лет. Разработка одной версии может занимать срок от нескольких недель до года	
Заключение отдельных договоров на отдельные версии	Заключается один договор. Версия и есть итоговый результат проекта	На отдельную версию или несколько последовательных версий обычно заключается отдельный договор	
Определение основных требований в начале проекта	Да	Да	Нет
Изменение требований по мере развития проекта	Нет	Незначительное	Да
Разработка итерациями (версиями)	Нет	Да	Да
Распространение промежуточного ПО	Нет	Может быть	

При разработке системы под итоговым продуктом и промежуточным программным обеспечением согласно [12] следует понимать:

- **ревизию** (исправительную или опытную) – любые оперативные изменения программного и информационного обеспечения, а также БД, необязательные в данный момент к передаче на объекты внедрения и связанные с устранением ошибок и усовершенствованием;

- **модификацию** – любые оперативные изменения программного и информационного обеспечения, а также БД, обязательные для передачи на объекты внедрения и обусловливающие изменение эксплуатационных характеристик без изменения функций (предусмотренных ТЗ), а также изменения, связанные с устранением ошибок, усовершенствованием;

- **версию** – любые изменения программного и информационного обеспечения, а также БД, обязательные для передачи на объекты внедрения, позволяющие выполнять заявленные или дополнительные функции, а также обеспечивающие переход на новые операционные системы и информационную среду;

- **развитие *(очередь)*** – плановые изменения информационной системы, связанные с введением новых функций и улучшением эксплуатационных характеристик, переходом на новую информационную среду, внедрением новых комплексов технических средств, новых информационных технологий и пр.