

Министерство образования и науки, молодежи и
спорта Украины

Харьковский национальный университет

Ревенчук Илона Анатольевна

Программной инженерии

Раздел 2 - Обзор моделей жизненного цикла разработки программного обеспечения

6.050103 - Программная инженерия, 8.05010301 -
Программное обеспечения систем, 8.05010302 -
Инженерия программного обеспечения

Харьков

2005

Содержание

Введение.....	3
Теория.....	4
Обобщенная схема жизненного цикла.....	4
Каскадная модель жизненного цикла SLC (Software life cycle).....	4
V - образная модель ЖЦ.....	6
Модель быстрого прототипирования.....	8
Модель быстрой разработки приложений RAD (Rapid Application Development).....	9
Инкрементная модель.....	11
Спиральная модель.....	13
Адаптированные модели.....	15
Практика.....	17
Вопросы для самопроверки.....	17
Выводы.....	18
Перечень ссылок.....	19

Раздел 2 - Обзор моделей жизненного цикла разработки программного обеспечения

В разделе рассматриваются: модели жизненного цикла их преимущества, недостатки, область применения.

Для разработки продукта в проекте должен применяться процесс. Вместо создания каждого проекта «с нуля» можно выбрать «начальный» жизненный цикл.

Жизненный цикл (ЖЦ) - это «карта-путеводитель» для всех участников проекта. Модель ЖЦ разработки ПО является единственным видом процесса, в котором представлен порядок его осуществления.

Теория

Обобщенная схема жизненного цикла

Обобщенная схема ЖЦ представлена на рисунке 2.1.

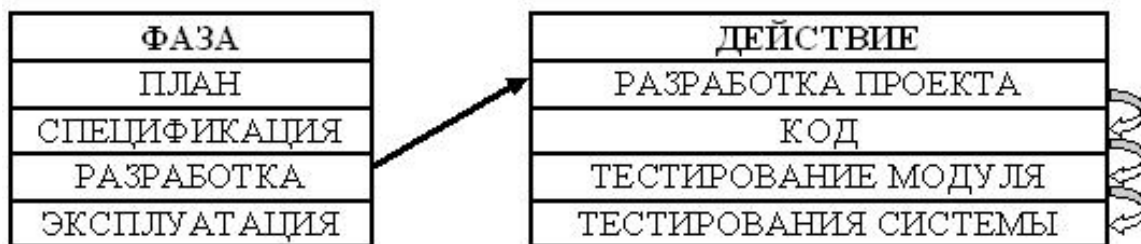


Рисунок 2.1 - Обобщенная схема жизненного цикла

Каскадная модель жизненного цикла SLC (Software life cycle)

Это была первая модель, которая придала особое значение исходным требованиям и проектированию.

Попытки оптимизации данной модели привели к возникновению других циклов разработки ПО. Как правило, все новые модели разрабатывались с целью лучшего отображения устройства реального мира с учетом большой скорости разработок ПО, либо для вовлечения клиентов в процесс разработки с целью улучшения функциональных возможностей создаваемых программ. Для решения этих проблем появилась спиральная модель, быстрая эволюционная модель прототипирования, V-образная модель, а также некоторые другие модели. Большинство практиков вполне согласны с тем, что на сегодняшний день существует настолько много различных типов проектов, что в рамках одномерной модели SLC они вряд ли могут быть описаны. Современная точка зрения заключается в том, что необходимо использовать уникальные модели или комбинации моделей для уникальных проектов.

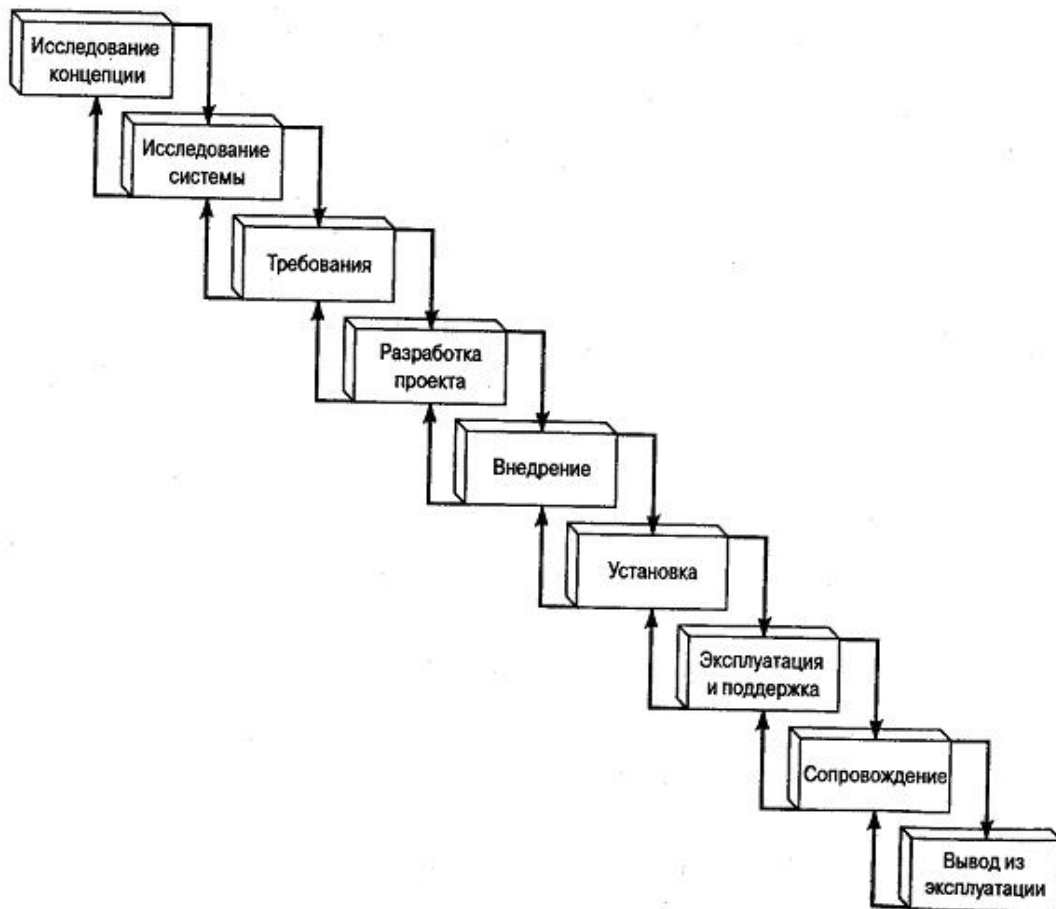


Рисунок 2.2 -Каскадная модель ЖЦ

Преимущества каскадной модели ЖЦ.

- Хорошо известна потребителям.
- Упорядоченно справляется со сложностями.
- Удобна в применении.
- Стабильность требований.
- Дефекты можно обнаружить на ранних этапах.
- Доступна для понимания.
- Хорошо определены стадии модели.
- Легко проследить ход выполнения проекта.

Недостатки каскадной модели ЖЦ.

- В основе - последовательная линейная структура.
- Требования должны быть известны вначале.
- Процесс обучения происходит в конце ЖЦ.
- Замораживание результативных данных по завершению каждой фазы.
- Интеграция полученных результатов происходит на завершающей стадии работы модели.
- Клиент не может ознакомиться с системой заранее.
- Программный продукт разрабатывается за один раз.

Область применения каскадной модели ЖЦ.

- В ситуациях, в которых требования и их реализация четко определены.
- При переносе уже существующего продукта на новую платформу.
- При выполнении больших проектов, в которых задействовано несколько больших команд разработчиков.

V - образная модель ЖЦ

В модели особое значение придается действиям, направленным на верификацию и аттестацию продукта. Эта модель была разработана как разновидность каскадной модели и представлена на рисунке 2.3.

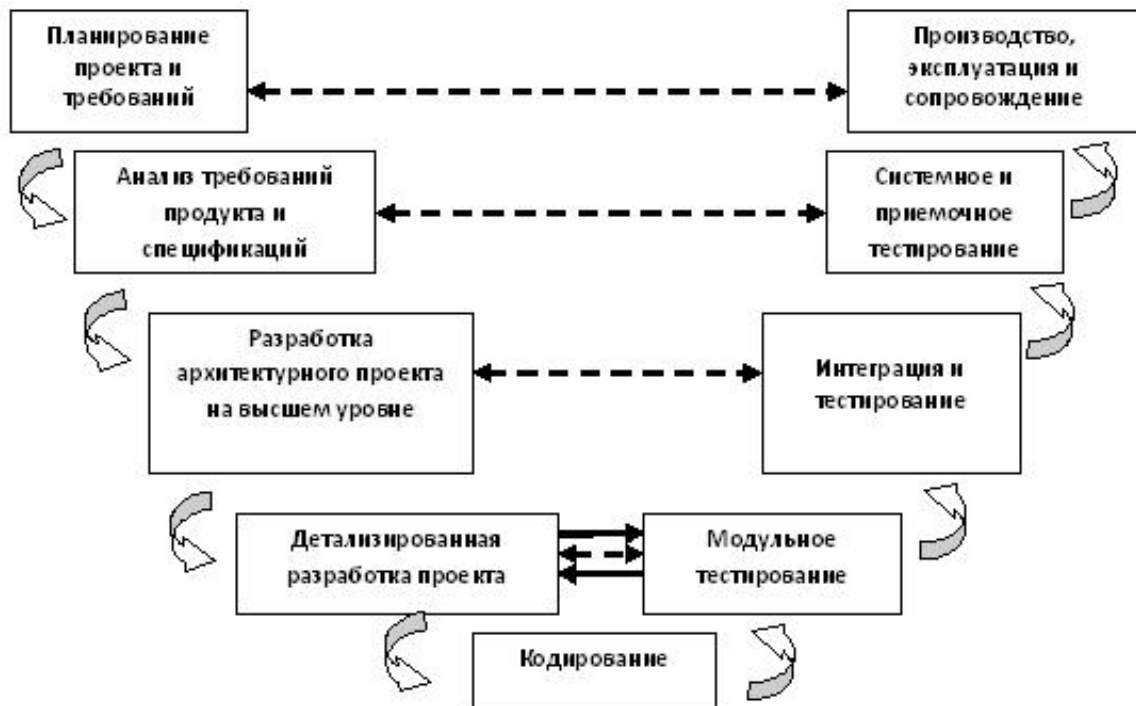


Рисунок 2.3 - V - образная модель ЖЦ

Преимущества V - образной модели.

- Особое значение придается планированию.
- Определяет продукты, которые должны быть получены в результате процесса разработки.
- Предусмотрены аттестация и верификация и внешних полученных данных.
- Определение требований - перед разработкой проекта системы.
- Проектирование ПО - перед разработкой компонентов.
- Можно отслеживать ход процесса разработки.
- Проста в использовании.

Недостатки V - образной модели.

- Плохо справляется с параллельными событиями.
- Не учтены итерации между фазами.

- Поздно происходит тестирование требований.
- Не предусмотрено внесение требования динамических изменений.
- Не содержит действий, направленных на анализ рисков.

Область применения V - образной модели.

- В ситуациях, в которых информация о требованиях доступна заранее.
- В случае, когда доступными являются информация о методе реализации решения и технология.
- В системах, в которых требуется высокая надежность.

Модель быстрого прототипирования

Модель быстрого прототипирования представлена на рисунке 2.4.

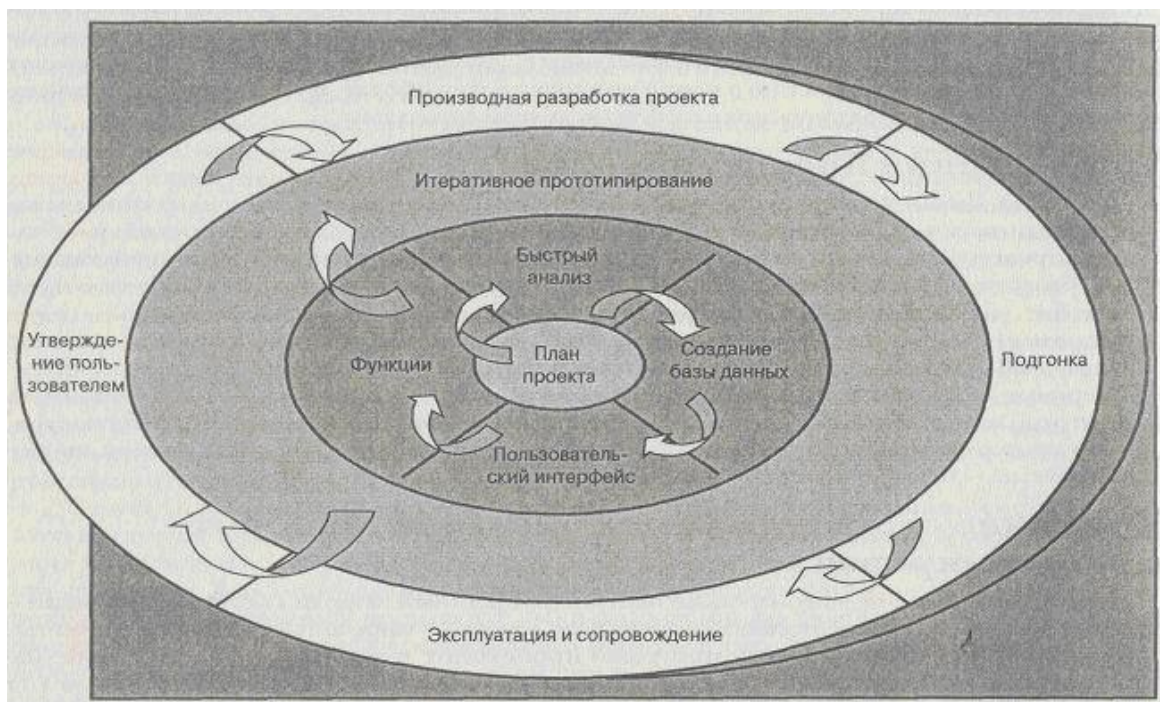


Рисунок 2.4 - Модель быстрого прототипирования

Преимущества модели быстрого прототипирования.

- Взаимодействие заказчика с системой начинается на раннем этапе разработки.
- В процессе разработки можно внести новые требования.
- Можно выявить проблему до привлечения дополнительных ресурсов.
- Позволяет выполнять гибкое проектирование и разработку.
- Позволяет максимально уменьшить количество неточностей в требованиях.
- Небольшой объем доработок.

Недостатки модели быстрого прототипирования.

- Репутация «разработанного на скорую руку» метода.
- Может быть уделено недостаточно внимания качеству ПО или долгосрочной надежности.
- Решение трудных проблем может отодвигаться на будущее.
- При досрочном завершении проекта у пользователя останется только частичная система.
- Прототипирование вызывает зависимость.
- Нет информации о точном числе итераций.

Область применения модели быстрого прототипирования.

- Требования не известны заранее.
- Требования непостоянны.
- Есть потребность в разработке пользовательских интерфейсов.
- Выполняется не имеющая аналогов разработка.
- Осуществляются временные демонстрации.
- При средней и высокой степенях риска.
- Применяется с каскадной моделью.
- Вместе с элементами анализа и проектирования.

Модель быстрой разработки приложений RAD (Rapid Application Development)

Пользователь задействован на всех фазах ЖЦ. Короткое время перехода от определения требований до создания полной системы. Разработка продукта ограничивается 60 днями, называемыми временным блоком с использованием мощных инструментальных средств разработки. Характеризуется высоким уровнем фактора использования.

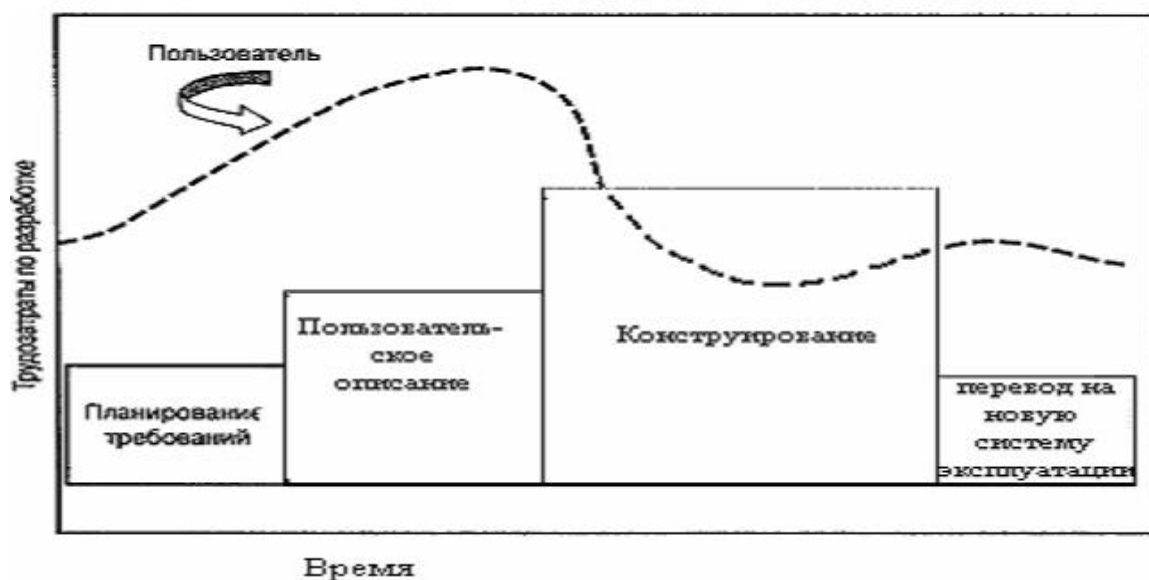


Рисунок 2.5 - Модель быстрой разработки приложений RAD

Преимущества модели быстрой разработки приложений RAD (Rapid Application Development)

- Время цикла разработки проекта сокращается.
- Требуется меньшее количество специалистов.
- Уменьшаются затраты.
- Создается обратная связь.
- Повторно используются компоненты уже существующих программ.
- Возможно произвести быстрый изначальный просмотр продукта.

Недостатки модели быстрой разработки приложений RAD (Rapid Application Development).

- Необходимо достаточное количество высококвалифицированных разработчиков.
- Неудачна при отсутствии компонент для повторного использования.
- Требуются быстрые действия из-за жестких временных ограничений.
- Необходим эффективный ускоренный процесс разработки.
- При использовании "вслепую" затраты не ограничены.

Область применения модели быстрой разработки приложений RAD (Rapid Application Development).

- В моделируемых и масштабируемых системах.
- Требования хорошо известны.
- При невысокой степени технических рисков.
- В информационных системах.
- При выполнении проектов в сокращенные сроки.
- Пригодные к повторному использованию части можно легко получить.
- В системах небольшого размера. Затраты и соблюдение графика не являются самым важным вопросом.

Инкрементная модель

Это процесс частичной реализации всей системы и медленного наращивания функциональных возможностей.

На ранних этапах выполняется конструирование системы в целом

Модель эффективна при использовании как в случае чрезвычайно больших, так и в небольших проектов. Представлена на рисунках 2.6 - 2.7.

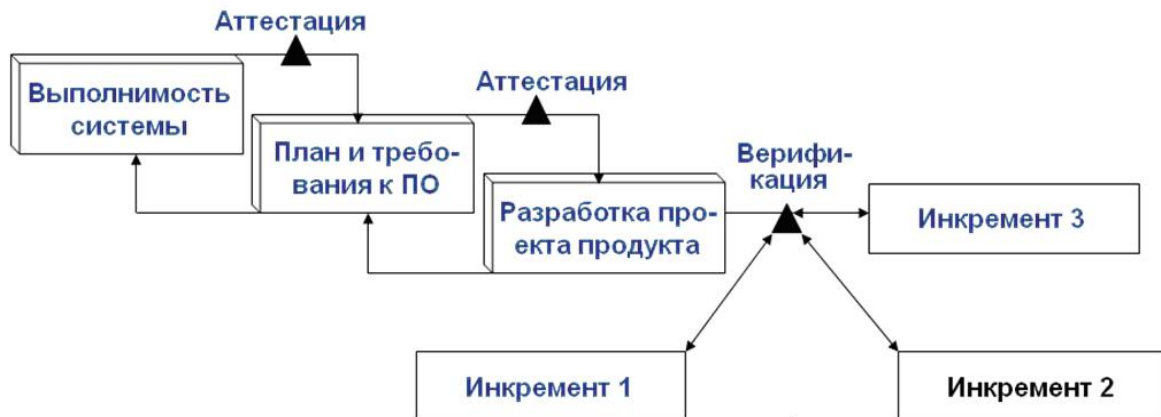


Рисунок 2.6 - Инкрементная модель

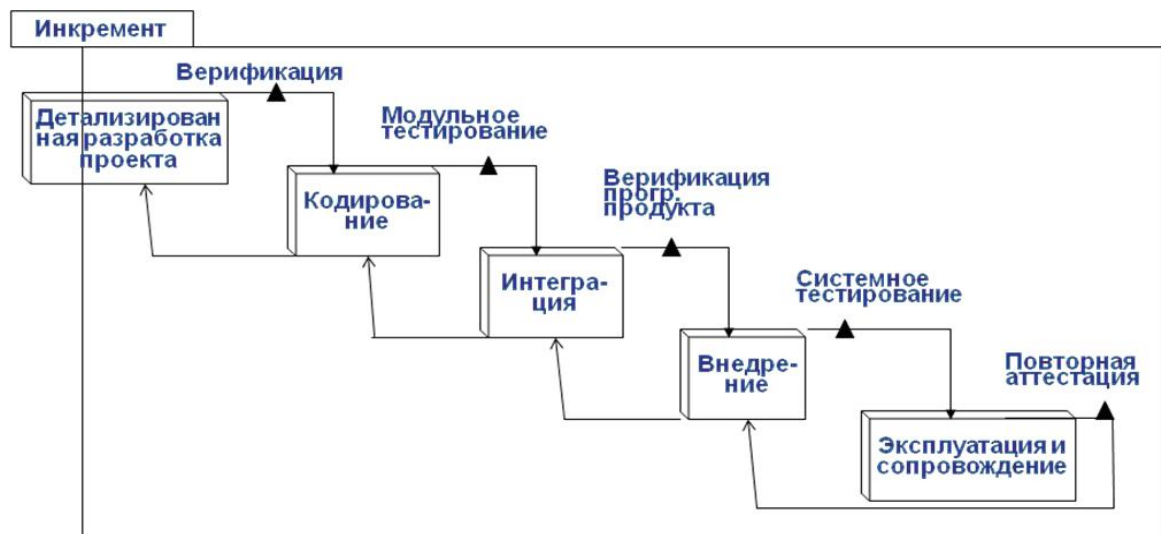


Рисунок 2.7 - Структура инкремента

Преимущества инкрементной модели.

- Не требуется заранее тратить средства, необходимые для разработки всего проекта.
- При выполнении каждого инкремента получается функциональный продукт.
- Заказчик может высказаться по поводу каждой разработанной версии системы.
- Существует возможность поддерживать постоянный прогресс в ходе выполнения проекта.
- Снижаются затраты на первоначальную поставку программного продукта.
- Снижается риск неудачи и изменения требований.

- Риск распределяется на несколько меньших по размеру инкрементов.

Недостатки инкрементной модели.

- Не предусмотрены итерации в рамках каждого инкремента.
- Определение полной функциональной системы должно осуществляться в начале ЖЦ.
- Может возникнуть оттягивание решений трудных проблем на будущее.
- Для инкрементов трудно выполнить анализ и проверку.

Область применения инкрементной модели.

- Требования можно сформулировать заранее.
- Существует потребность быстро поставить на рынок продукт.
- На выполнение проектов предусмотрен большой период времени разработки.
- При равномерном распределении свойств различной степени важности.
- При выполнении проекта с применением новой технологии.

Спиральная модель

Воплощает в себе преимущества каскадной модели.

Включены анализ рисков, управление ими, процессы поддержки и менеджмента.

Разработка продукта с использованием метода прототипирования или быстрой разработки приложения Каждый цикл представляет собой набор операций, которому соответствует такое же количество стадий, как и в модели каскадного процесса.

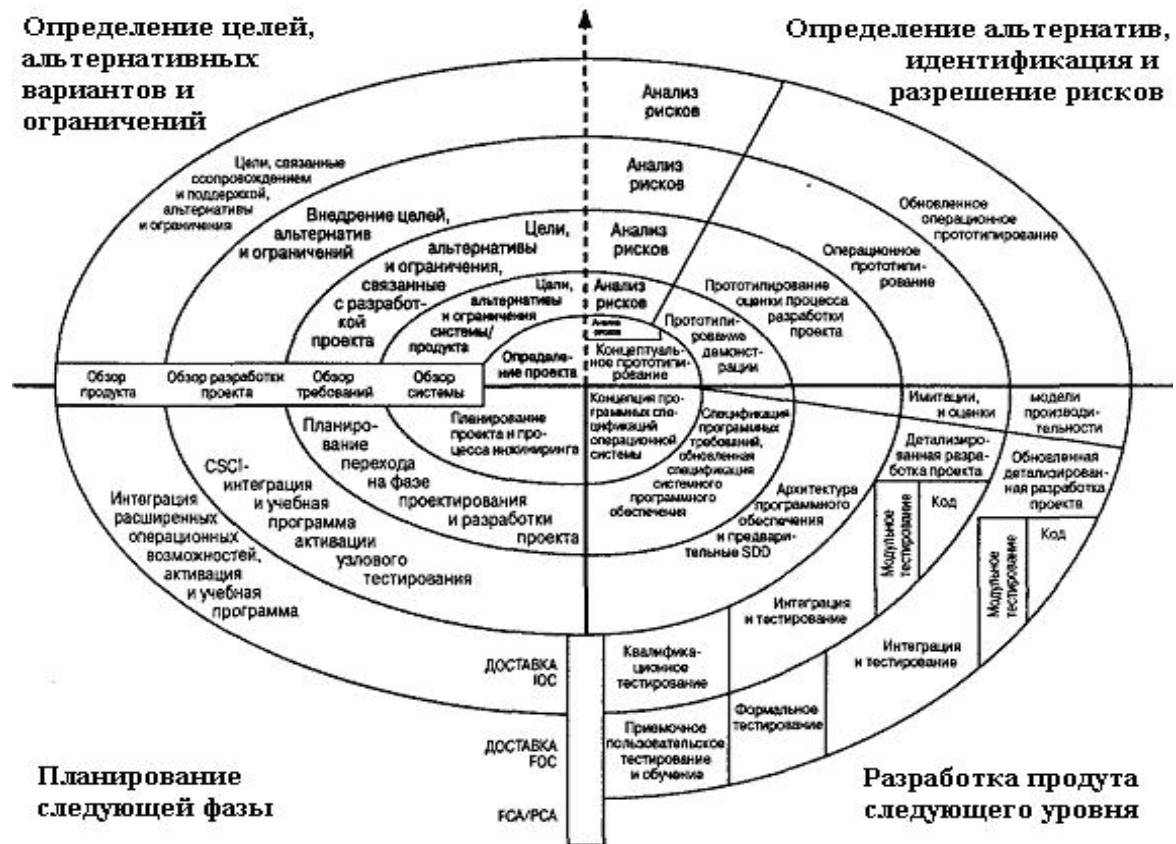


Рисунок 2.8 - Спиральная модель ЖЦ

Преимущества спиральной модели.

- Модель разрешает пользователям "увидеть" систему на ранних этапах.
- Обеспечивается определение непреодолимых рисков.
- Пользователи принимают участие при планировании, анализе рисков, разработке.
- Предусмотрена возможность гибкого проектирования.
- Обеспечивается разбиение большого объема работы по разработке продукта на небольшие части.
- Обратная связь от пользователей к разработчикам выполняется с высокой частотой и на ранних этапах модели.
- Не нужно распределять заранее финансовые ресурсы.

Недостатки спиральной модели.

- При низкой степени риска или небольших размерах, модель может оказаться дорогостоящей.
- Модель имеет усложненную структуру.
- Нужда в высоко профессиональных знаниях для оценки рисков.
- Спираль может продолжаться до бесконечности.

Область применения спиральной модели.

- Для средней или высокой степени риска.
- Когда пользователи не уверены в своих потребностях.
- Когда ожидаются существенные изменения.
- Когда речь идет о применении новой технологии.
- В случае больших проектов.
- При выполнении бизнес-проектов и проектов в области аэрокосмической промышленности, обороны и инжиниринга.

Адаптированные модели

Модель быстрого отслеживания:

- ускоренное прохождение или пропуск фаз жизненного цикла или процессов разработки;
- необходимость в применении возникает в случае критической нехватки времени;
- ЖЦ обычно является менее формальным.

Параллельный инжиниринг:

- создание продуктов более высокого качества за меньший период времени;
- все аспекты ЖЦ проекта учитываются в процессе от проектирования до производства как можно раньше;
- состоит из нескольких действий, которые осуществляются одновременно;
- необходимо оценивать возможные технические риски при использовании метода.

Спиральная модель "Win-Win" (адаптивная модель).

К начальной фазе каждого цикла добавляются так называемые действия Теории W (Теория W – это принцип менеджмента, при реализации которого особое значение придается ключевым организаторам совместного дела, выполняющим разработку системы (пользователь, заказчик, разработчик, наладчик, создатель интерфейсов и т.д.), которые станут "победителями", если проект окажется успешным).

Преимущества:

- Более быстрая разработка ПО.
- Уменьшение стоимости программ.
- Более высокий уровень удовлетворения со стороны участников проекта.
- Более высокое качество ПО.
- Исследование большого количества вариантов построения архитектуры на ранних этапах разработки.

Практика

Вопросы для самопроверки

1. Модели жизненного цикла ПО.
2. Каскадная модель жизненного цикла разработки ПО, ее преимущества и недостатки, и область применения.
3. V -образная модель, ее преимущества и недостатки, и область применения.
5. Модель быстрого прототипирования, ее преимущества и недостатки, и область применения.
6. Модель быстрой разработки приложений RAD, ее преимущества и недостатки, и область применения.
7. Инкрементная модель, ее преимущества и недостатки, и область применения.
8. Спиральная модель, ее преимущества и недостатки, и область применения.

Вывод к разделу 2 - Обзор моделей жизненного цикла разработки программного обеспечения

В разделе рассматривались основные модели жизненного цикла их преимущества, недостатки и область их применения.

Перечень ссылок

Источники, использованные в материалах

Американский национальный стандарт ANSI/PMI 99-001-2004. Руководство к Своду знаний по управлению проектами. Введ. 2004.- Третье издание. (Руководство PMBOOK®). 401с.

Мари Кантор. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения . СПб. Вильямс. 2002. -642с.

Управление программными проектами. Достижение оптимального качества при минимуме затрат. Роберт, Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер / М-СПб-К. Вильямс. 2003. -1118с.

Уокер Ройс. Управление проектами по созданию программного обеспечения . М. Лори. 2002. -450с.

Элейн Маркел. Microsoft Project 2002. Библия пользователя. М. Диалектика. 2003. -880с.

Microsoft Project 2003 course certification materials. Trainer kit. [Электронный ресурс] Режим доступа: <http://www.cheltenhamcourseware.com/>