



**МОДЕЛИ ЖИЗНЕННОГО
ЦИКЛА РАЗРАБОТКИ
ПРОГРАММНОГО ПРОДУКТА.**

МОДЕЛЬ ЖЦ ПП -

- **Общая структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых в течение ЖЦ.**
- *Другими словами, модель ЖЦ ПП – это последовательность действий, выполнение которых является необходимым условием для создания ПП, соответствующего заданным требованиям.*



СТРАТЕГИИ РАЗРАБОТКИ

Линейная

Этапы разработки реализуются последовательно друг за другом

Итерационная

Система строится в виде последовательности версий, каждая из которых дополняет предыдущую



ОБЗОР СОВРЕМЕННЫХ МОДЕЛЕЙ:

Линейная стратегия

Каскадная модель

Прямолинейная и простая в использовании.
Разрабатываемое программное обеспечение не доступно для изменений

V-образная модель

Простая в использовании.
Особое значение придается тестированию и сравнению результатов фаз тестирования и проектирования

Итерационная стратегия

Модель прототипирования

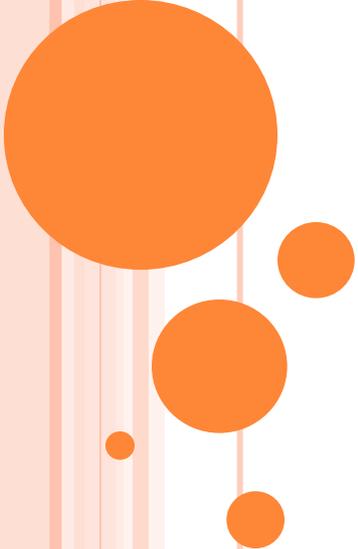
Создается «быстрая» частичная реализация системы до составления окончательных требований.

Модель быстрой разработки приложений

Проектные группы небольшие и составлены из высококвалифицированных специалистов.
Уменьшенное время цикла разработки (до 3 мес) и улучшенная производительность.

Спиральная модель

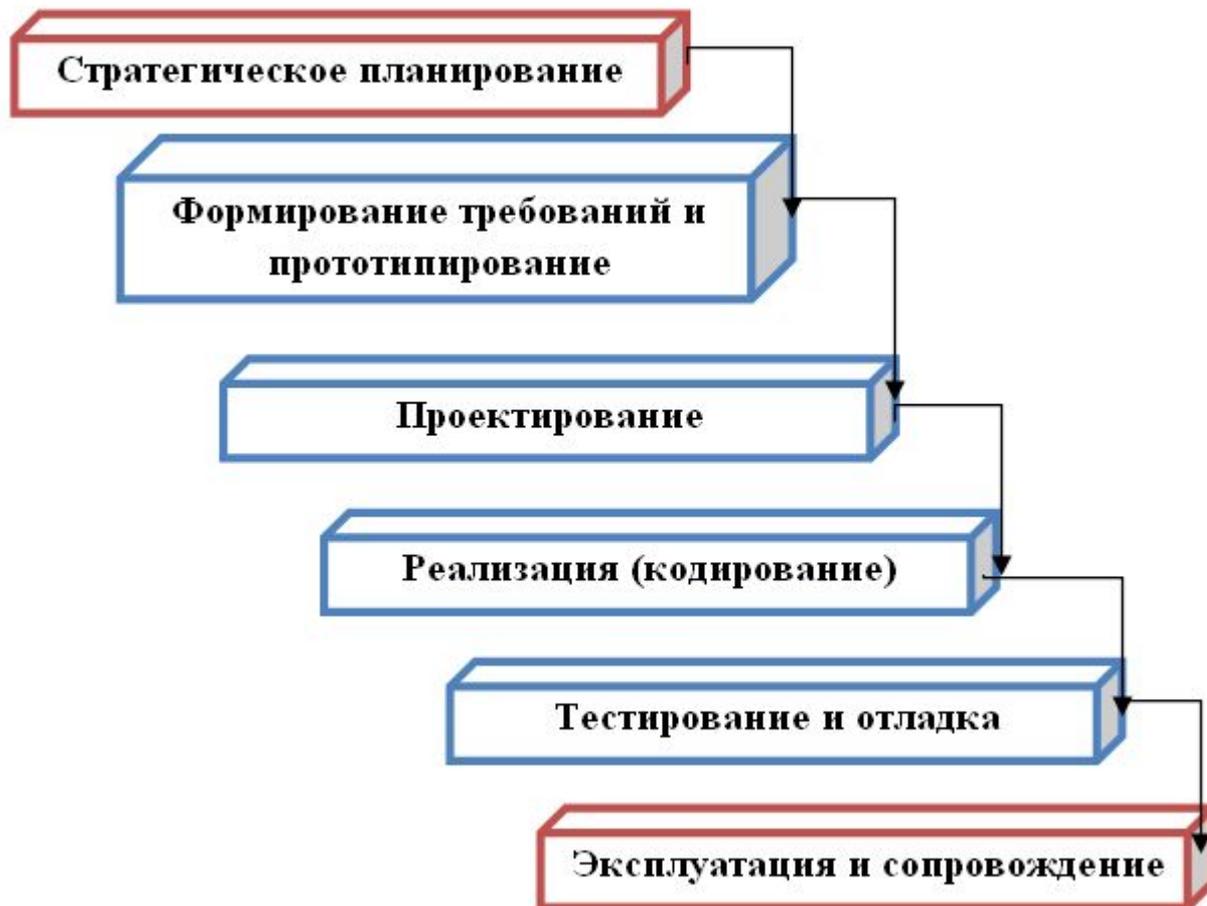
Требования определяются не полностью, дополняются и реализуются на каждом витке спирали



КАСКАДНАЯ МОДЕЛЬ ИЛИ МОДЕЛЬ ВОДОПАДА

(Waterfall model)

Структура каскадной модели имеет вид:



ОСОБЕННОСТИ КАСКАДНОЙ МОДЕЛИ:

1. **Переход на следующий этап осуществляется только после завершения всех работ на предыдущем этапе, и возвратов на пройденные этапы не предусматривается.**
2. **Каждый этап завершается получением некоторых результатов, которые являются исходными для следующего этапа.**
3. **Требования к ПП, определённые на этапе формирования требований, документируются в виде Технического задания (ТЗ) и фиксируются на всё время разработки.**
4. **Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка была продолжена другой командой разработчиков.**
5. **Критерием качества разработки является точность выполнения спецификаций (требований) ТЗ.**



ПРЕИМУЩЕСТВА КАСКАДНОЙ МОДЕЛИ:

1. На каждой стадии формируется законченный набор проектной документации;
2. Выполненные в логичной последовательности стадии работ позволяют планировать сроки завершения всех работ и соответствующие затраты.



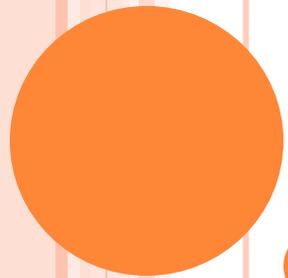
- ❑ Недостатки каскадной модели обусловлены, прежде всего, тем, что реальный процесс разработки ПП никогда не укладывается в такую жёсткую схему.
- ❑ В результате реальный процесс разработки имеет вид:



Недостатки каскадной модели:

1. Позднее тестирование, приводящее к существенным затратам на отладку ПП.
2. Позднее согласование результатов работы с заказчиком, вследствие чего повышается риск его неудовлетворённости ПП.





V – ОБРАЗНАЯ МОДЕЛЬ



V – shaped model



V – ОБРАЗНАЯ МОДЕЛЬ -

- ▣ **разновидность каскадной модели, в которой особое внимание уделяется верификации и аттестации ПП.**

- ▣ *Модель показывает, что тестирование продукта обсуждается, проектируется и планируется, начиная с ранних этапов ЖЦ разработки.*



ПРЕИМУЩЕСТВА V – ОБРАЗНОЙ МОДЕЛИ:

- 1. Большая роль уделяется верификации и аттестации ПП, начиная с ранних стадий его разработки, все действия планируются;**
- 2. Предполагается верификация не только самого ПП, но и всех полученных внутренних и внешних данных;**
- 3. Ход выполнения работ легко отслеживается, т.к. переход на следующий этап осуществляется только после завершения всех работ на предыдущем.**



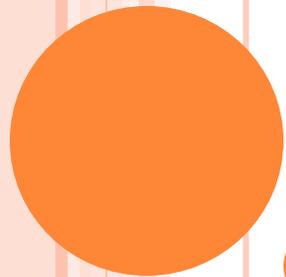
НЕДОСТАТКИ V – ОБРАЗНОЙ МОДЕЛИ:

- 1. Не учитываются итерации между фазами;**
- 2. Нельзя вносить изменения на разных этапах ЖЦ;**
- 3. Тестирование требований происходит слишком поздно, поэтому внесение изменений влияет на выполнение графика работ.**



- ▣ **V – образную модель используют при разработке ПП, главным требованием для которых является высокая надёжность.**





СПИРАЛЬНАЯ МОДЕЛЬ.

Spiral model

Автор – Барри Бозм

- **Спиральная модель — классический пример применения итерационной стратегии конструирования, *когда система строится в виде последовательности версий.***
- **Требования к ПП в начале процесса определяются не полностью и уточняются в результате работы над продуктом.**



СТРУКТУРА СПИРАЛЬНОЙ МОДЕЛИ:



- **Спиральная модель:** 1 — начальный сбор требований и планирование проекта; 2 — та же работа, но на основе рекомендаций заказчика; 3 — анализ риска на основе начальных требований; 4 — анализ риска на основе реакции заказчика; 5 — переход к комплексной системе; 6 — начальный макет системы; 7 — следующий уровень макета; 8 — сконструированная система; 9 — оценивание заказчиком.



ПРИНЦИПИАЛЬНЫЕ ОСОБЕННОСТИ СПИРАЛЬНОЙ МОДЕЛИ:

- ❑ Отказ от фиксации требований и назначение приоритетов пользовательским требованиям;
- ❑ Разработка последовательности прототипов, начиная с требований наивысшего приоритета;
- ❑ Идентификация и анализ риска на каждой итерации;
- ❑ Использование каскадной модели для реализации окончательного прототипа;
- ❑ Оценка результатов по завершении каждой итерации и планирование следующей итерации.



- Таким образом, спиральная модель определяет 4 основных действия (*записаны в 4-х квадрантах спирали*):
1. **Планирование** – определение целей и требований;
 2. **Анализ риска** – анализ вариантов на целесообразность продолжения работы над проектом.
 3. **Конструирование** – разработка ПП следующего уровня;
 4. **Оценивание** – оценка заказчиком текущих результатов конструирования.



ПРИНЦИП РАБОТЫ СПИРАЛЬНОЙ МОДЕЛИ:

- ▣ С каждой итерацией по спирали (продвижением от центра к периферии) строятся всё более полные версии ПП.
- ▣ В первом витке определяются начальные цели и требования. Выполняется анализ риска, на основе которого принимается решение о разработке ПП (*если риск велик, проект может быть остановлен*). Заказчик оценивает промежуточную работу и вносит предложения по модификации.
- ▣ Следующая фаза планирования и анализа риска базируется на предложениях заказчика.
- ▣ Движение по спирали продолжается до тех пор, пока следующая версия ПП полностью не удовлетворит потребности заказчика.



ДОСТОИНСТВА СПИРАЛЬНОЙ МОДЕЛИ:

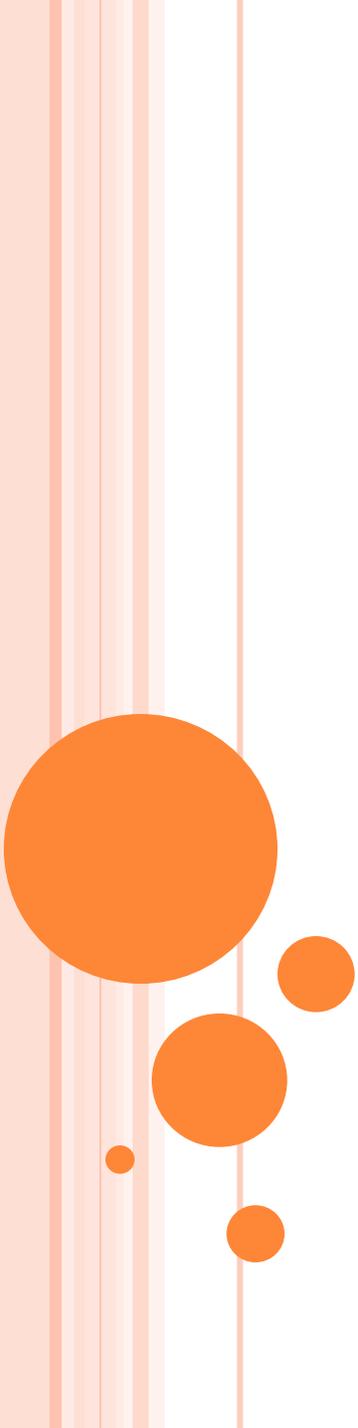
1. Наиболее реально (в виде эволюции) отражает разработку ПП;
2. Ускорение разработки (раннее получение результатов за счёт прототипирования)
3. Постоянное участие заказчика в процессе разработки
4. Разбиение большого объёма работы на небольшие части
5. Учёт риска на каждом витке спирали;



НЕДОСТАТКИ СПИРАЛЬНОЙ МОДЕЛИ:

1. Трудность приведения проекта к конечному результату
2. Сложность планирования (определения количества и длительности итераций, оценки затрат и рисков)
3. Сложность применения модели с точки зрения менеджера и заказчиков (из – за привычки к строгому и детальному планированию)
4. Напряжённый режим работы для разработчиков (при краткосрочных итерациях)
5. Повышенные требования к заказчику;





МОДЕЛЬ БЫСТРОЙ РАЗРАБОТКИ ПРИЛОЖЕНИЙ.

RAD – модель

(RAD – Rapid Application Development);

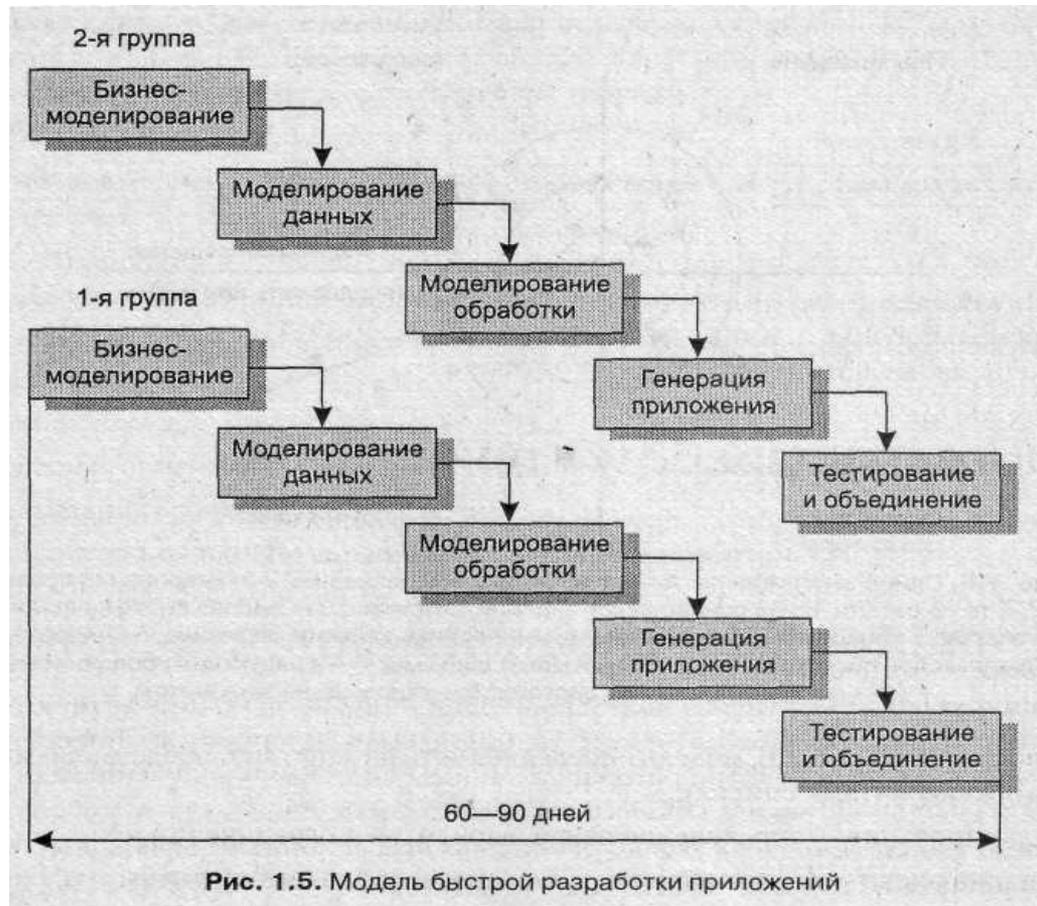
RAD – МОДЕЛЬ -

второй пример применения итерационной стратегии конструирования, основанный на 3-х составляющих:

- Небольшая группа разработчиков (7+-2 человека)
- Короткий, но тщательно разработанный производственный график (до трёх месяцев)
- Повторяющийся цикл, при котором разработчики по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные в результате взаимодействия с заказчиком.



СТРУКТУРА RAD – МОДЕЛИ:



БАЗОВЫЕ ПРИНЦИПЫ RAD:

- ▣ **Разработка приложений итерациями**
- ▣ **Необязательность полного завершения работ на каждой стадии ЖЦ**
- ▣ **Обязательность вовлечения пользователей в процесс разработки**
- ▣ **Применения средств управления конфигурациями и средств автоматизированной разработки, проектирования и документирования приложений (CASE - средств)**
- ▣ **Тестирование и развитие проекта, осуществляемой одновременно с разработкой**
- ▣ **Ведение разработки небольшой, хорошо управляемой командой профессионалов**
- ▣ **Грамотное руководство разработкой системы, чёткое планирование и контроль выполняемых работ.**



УСЛОВИЯ ПРИМЕНЕНИЯ МЕТОДОЛОГИИ RAD:

- применима для относительно небольших проектов, разрабатываемых под конкретного заказчика;
- неприменима для построения сложных расчетных программ, требующих написания большого объема (сотни тысяч строк) уникального кода (*например, операционных систем или программ управления космическими кораблями*);
- неприменима для разработки приложений, от которых зависит безопасность людей (например, управление самолетом или атомной электростанцией), так как итеративный подход предполагает, что первые несколько версий, скорее всего не будут полностью работоспособны, что в данном случае исключается.



ТЕМЫ РЕФЕРАТОВ:

- Экстремальное программирование (**XP - процесс**)
- Методология Scrum
- Методология RUP (**Rational Unified Process**)
- Методология MSF (**Microsoft Solutions Framework**)

