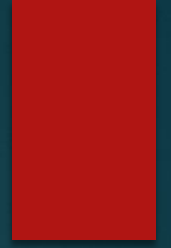


# МОДЕЛИ ЖЦ ПО

ВЫПОЛНИЛИ: СМИРНОВ МАКСИМ И ПРОХОРОВ ОЛЕГ, СТУДЕНТЫ  
ГРУППЫ 2-1П11

# ВИДЫ МОДЕЛЕЙ ЖЦ ПО



1. Каскадная модель
2. V-образная модель
3. Процесс макетирования ПО
4. Инкрементная модель
5. Спиральная модель
6. Компонентная модель
7. Модель быстрой разработки приложений

# КАСКАДНАЯ МОДЕЛЬ



# КАСКАДНАЯ МОДЕЛЬ

Водопадная или каскадная модель разработки программного обеспечения (waterfall, водопад) — это процесс разработки, в котором последовательно проходят фазы сбора и анализа требований, проектирования и прототипирования, реализации, тестирования, интеграции и поддержки.

# ПРИЕМУЩЕСТВА КАСКАДНОЙ МОДЕЛИ

- ▶ широкая известность и простота модели;
- ▶ отличается стабильностью требований;
- ▶ определяет процедуры по контролю за качеством;
- ▶ способствует осуществлению строгого контроля менеджмента проекта.

# НЕДОСТАТКИ КАСКАДНОЙ МОДЕЛИ

- ▶ в основе модели лежит последовательная линейная структура;
- ▶ она может создать ошибочное впечатление о работе над проектом;
- ▶ необходимость в жестком управлении и контроле;
- ▶ модель основана на документации;
- ▶ отсутствует возможность учесть переделку и итерации за рамками проекта.

# V-ОБРАЗНАЯ МОДЕЛЬ



# V-ОБРАЗНАЯ МОДЕЛЬ

V-образная модель была создана с целью помочь работающей над проектом команде в планировании с обеспечением дальнейшей возможности тестирования системы. В этой модели особое значение придается действиям, направленным на верификацию и аттестацию продукта. Она демонстрирует, что тестирование продукта обсуждается, проектируется и планируется на ранних этапах жизненного цикла разработки.



# ПРЕИМУЩЕСТВА V-ОБРАЗНОЙ МОДЕЛИ

- ▶ в V-образной модели определение требований выполняется перед разработкой проекта системы, а проектирование ПО – перед разработкой компонентов;
- ▶ модель определяет продукты, которые должны быть получены в результате процесса разработки;
- ▶ модель проста в использовании.

# НЕДОСТАТКИ V-ОБРАЗНОЙ МОДЕЛИ

- ▶ с ее помощью непросто справиться с параллельными событиями;
- ▶ в ней не учтены итерации между фазами;
- ▶ в модели не предусмотрено внесение требования динамических изменений на разных этапах жизненного цикла;
- ▶ в модель не входят действия, направленные на анализ рисков.

# ПРОЦЕСС МАКЕТИРОВАНИЯ ПО

Макетирование (прототипирование) – это процесс создания модели разрабатываемого программного продукта.



# ПРЕИМУЩЕСТВА МАКЕТИРОВАНИЯ

- ▶ конечный пользователь может "увидеть" системные требования в процессе их сбора командой разработчиков;
- ▶ возможность внесения новых или неожиданных требований пользователя;
- ▶ модель позволяет выполнять гибкое проектирование и разработку;
- ▶ образуются постоянные, видимые признаки прогресса в выполнении проекта;

# НЕДОСТАТКИ МАКЕТИРОВАНИЯ

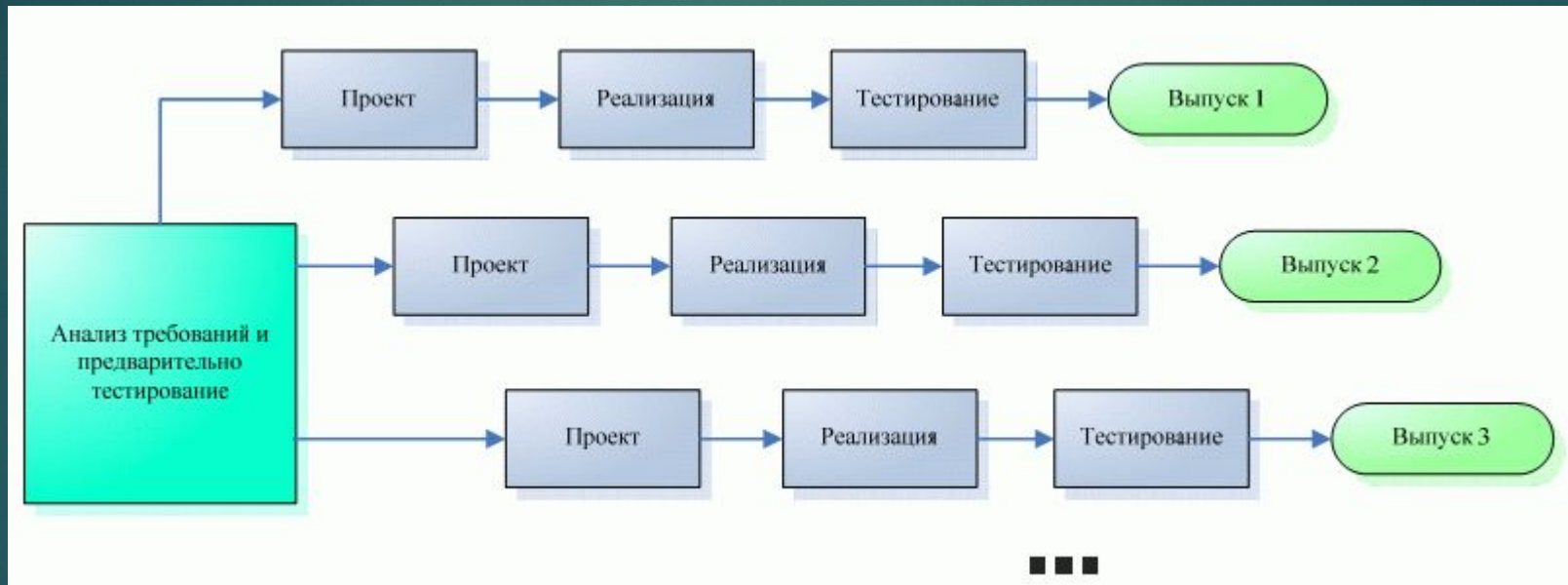
- ▶ требуется активное участие заказчика;
- ▶ прототипирование вызывает зависимость и может продолжаться слишком долго;
- ▶ при использовании модели решение трудных проблем может отодвигаться на будущее;
- ▶ заказчик может предпочесть получить прототип, вместо того, чтобы ждать появления полной, хорошо продуманной версии.

# ИНКРЕМЕНТНАЯ МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА

Инкрементная разработка представляет собой процесс частичной реализации всей системы и медленного наращивания функциональных возможностей.

Инкрементная модель действует по принципу каскадной модели с перекрытиями.

# ИНКРЕМЕНТНАЯ МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА



# ПРЕИМУЩЕСТВА ИНКРЕМЕНТНОЙ МОДЕЛИ

- ▶ в результате выполнения каждого инкремента получается функциональный продукт;
- ▶ снижается риск неудачи и изменения требований;
- ▶ риск распределяется на несколько меньших по размеру инкрементов;
- ▶ существует возможность пересмотреть риски, связанные с затратами и соблюдением установленного графика;
- ▶ заказчик может привыкать к новой технологии постепенно.

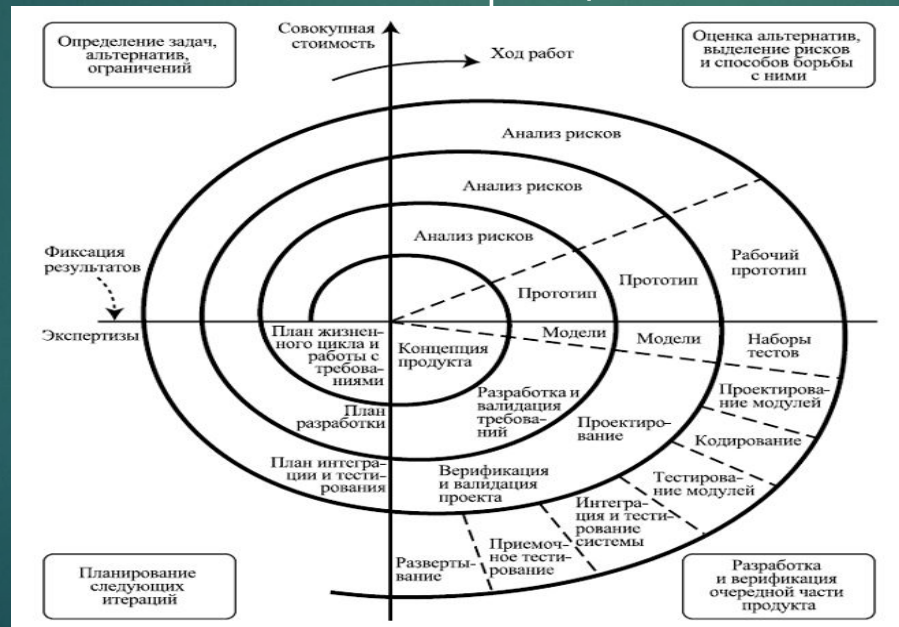


# НЕДОСТАТКИ ИНКРЕМЕНТНОЙ МОДЕЛИ

- ▶ Недостатки инкрементной модели
- ▶ для модели необходимы хорошее планирование и проектирование;
- ▶ использование на этапе анализа общих целей, вместо полностью сформулированных требований, может оказаться неудобным для руководства;
- ▶ в модели не предусмотрены итерации в рамках каждого инкремента;
- ▶ формальный критический анализ и проверку намного труднее выполнить для инкрементов, чем для системы в целом.

# СПИРАЛЬНАЯ МОДЕЛЬ

Спиральная модель отображает базовую концепцию, которая заключается в том, что каждый цикл представляет собой набор операций, которому соответствует такое же количество стадий, как и в модели каскадного процесса.



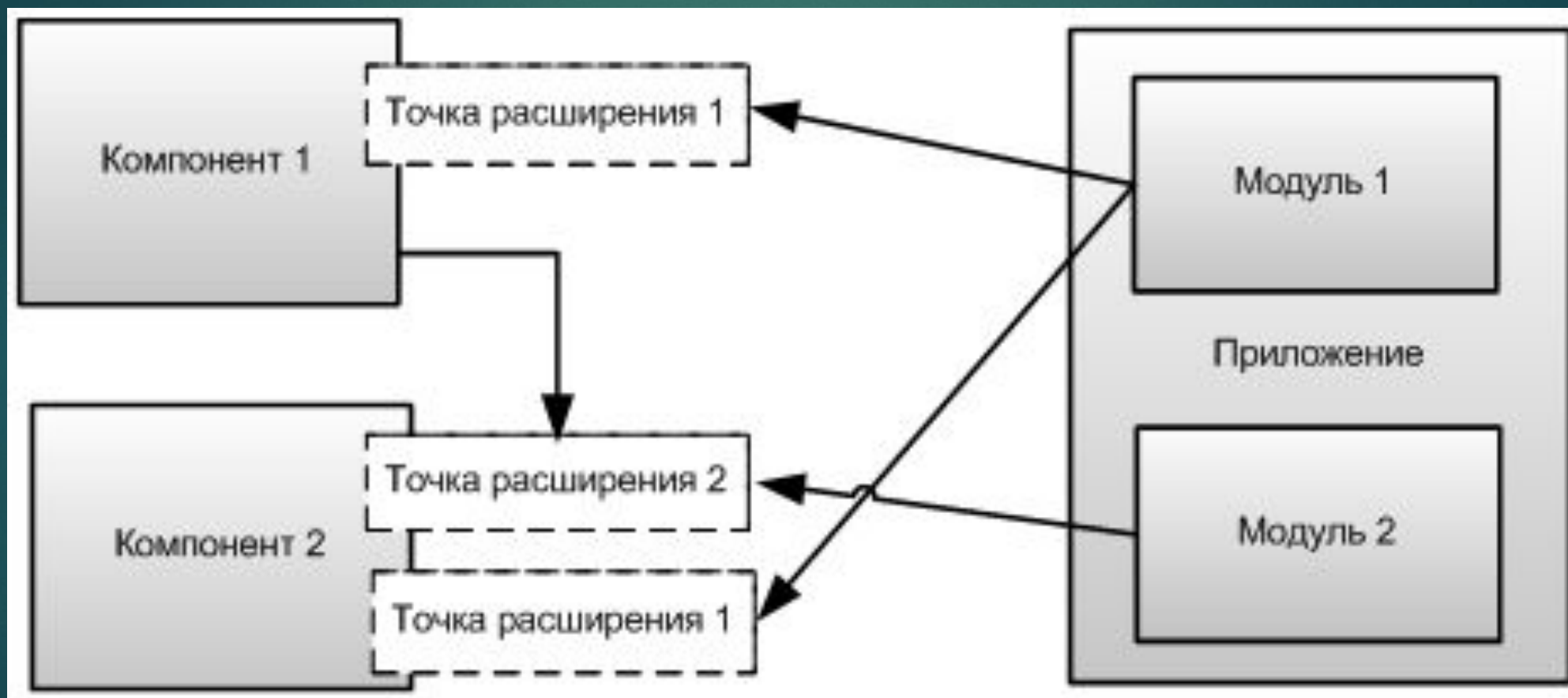
# ПРЕИМУЩЕСТВА СПИРАЛЬНОЙ МОДЕЛИ

- позволяет пользователям "увидеть" систему на ранних этапах;
- она обеспечивает разбиение большого потенциального объема работы по разработке продукта на небольшие части;
- в модели предусмотрена возможность гибкого проектирования;
- при использовании спиральной модели не нужно распределять заранее все необходимые для выполнения проекта финансовые ресурсы.

# НЕДОСТАТКИ СПИРАЛЬНОЙ МОДЕЛИ

- ▶ модель имеет усложненную структуру, поэтому может быть затруднено ее применение разработчиками, менеджерами и заказчиками;
- ▶ если проект имеет низкую степень риска или небольшие размеры, модель может оказаться дорогостоящей;
- ▶ отсутствие хорошего средства или метода прототипирования может сделать использование модели неудобным.

# КОМПОНЕНТНАЯ МОДЕЛЬ



# ПРЕИМУЩЕСТВА RAD

- основное внимание переносится с документации на код, причем при этом справедлив принцип "получаете то, что видите" (What you see is what you get, WYSIWYG);
- повторное использование компонент уже существующих программ.

# НЕДОСТАТКИ RAD

- ▶ непостоянное участие пользователя может негативно сказаться на конечном продукте;
- ▶ при использовании модели "вслепую" на затраты и дату завершения работы над проектом ограничения не накладываются;
- ▶ искусственное «затягивание» разработки ПО;
- ▶ существует риск, что работа над проектом никогда не будет завершена.