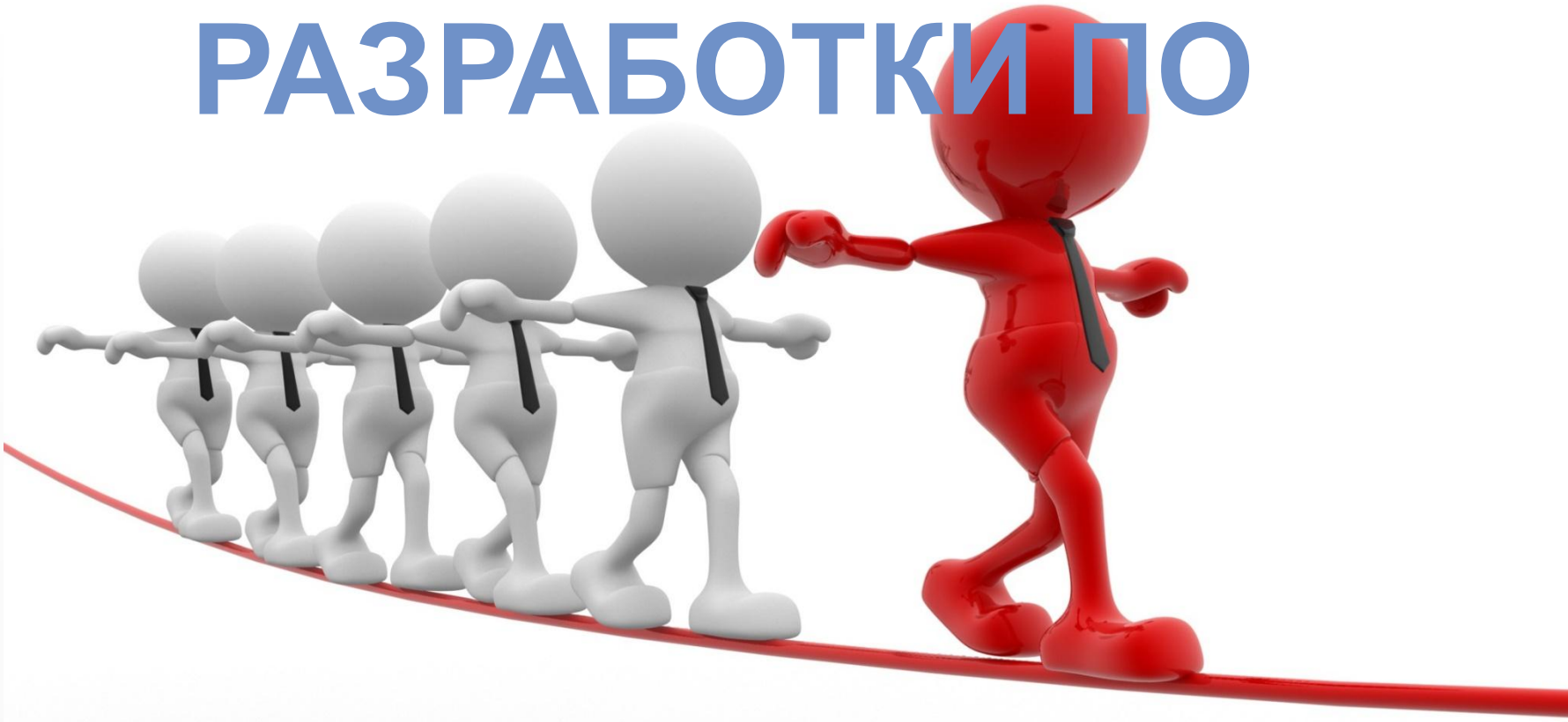
ТЕХНОЛОГИЯ И ПРОЦЕСС РАЗРАБОТКИ ПО



к.п.н., доцент Касаткин Д. А.
e-mail: kasatkinda@cfuv.ru

ТРПО

разработка = анализ + проектирование + программирование (кодирование) + тестирование + отладка

Технология разработки программного обеспечения (ТРПО) – это совокупность процессов и методов создания программного продукта.

Технология программирования (технология разработки ПО) — способ организации **процесса** создания программы, совокупность приемов и способов выполнения определенных видов деятельности.

Программная инженерия (Software engineering)

- Программная инженерия это систематизированный подход к профессиональной разработке, внедрению, сопровождению и изъятию из использования ПО.
- Такой систематизированный подход должен помогать повышать качество и производительность разработки ПО (Q&P).

Качество и производительность разработки ПО зависит от:

- ***квалификации (умения) людей,*** участвующих в разработке ПО;
- ***качества процессов*** организации работы специалистов, которые используют для выполнения различных задач проекта;
- возможностей используемых ***программных средств разработки*** (инструментов).

Типы программных систем

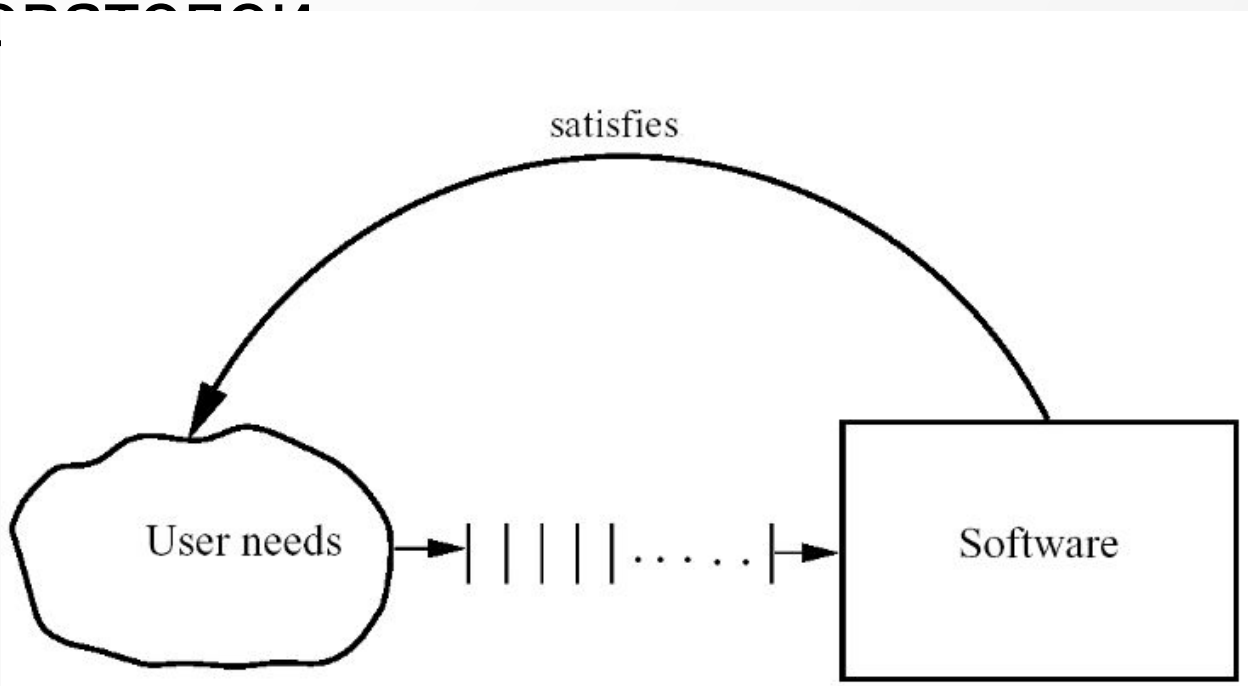
1. системы работающие только с людьми
   – например: информационные системы организаций – используются людьми для ввода, хранения поиска и отображения информации;
2. системы работающие с людьми и внешними техническими устройствами:
   – например: информационные системы соревнований.
3. системы работающие с внешними техническими устройствами
   – встроенные системы реального времени – встраиваются в технические устройства для выполнения управления ими.

Процессы – основа разработки ПО

- В ТРПО основное внимание уделяется на процессы, которые называются систематизированном подходом к разработке ПО.
- Основной задачей ***процессов*** является оказание помощи специалистам достигать высокого качества и производительности путем указания того, какие задачи и как требуется решать.
- Внимание к процессу создания программного продукта и отличает ТРПО от большинства других компьютерных дисциплин.
- Процессы (совместно с инструментами и технологиями для их выполнения) составляют основу ТРПО.

Процесс и проект (process and project)

- Процесс это последовательность шагов, выполняемых для достижения требуемой цели.
- При разработке ПО промышленного уровня целью является создание ПО, удовлетворяющего потребностям пользователей



Процесс

- Для программного проекта ключевую роль играет процесс разработки – в результате выполнение данного процесса достигается цель предоставления заказчику требуемого ПО.
- Однако не достаточно просто разработать требуемое ПО, нужно выполнить проект с ***наименьшими затратами***, за ***наименьшее время*** и с ***требуемым качеством***.
- В связи с этими дополнительными целями роль процессов возрастает.
- Существует много процессов разработки, позволяющих создать требуемое ПО, но ***для достижения высокого качества и производительности требуется некоторый «оптимальный» процесс***.

Спецификация процесса

- Нужно отличать спецификацию (детальное описание) процесса от самого процесса.
- ***Процесс*** является динамической сущностью, включающей выполненные действия.
- ***Спецификация процесса*** является описанием процесса, которому предположительно можно следовать в некотором проекте для достижения цели, для которой данный процесс был спроектирован.
- В проекте спецификация процесса может использоваться, в качестве процесса, который планирует использовать проект.
- Реальный процесс это то, что в действительно выполняется в проекте.
- Он может отличаться планируемого процесса.

Модель процесса

- ***Модель процесса*** описывает (специфицирует) обобщенный процесс*, который является «оптимальным» для* ***некоторого класса проектов***.
- Т.е. в ситуациях, для которых данная модель является применимой, использование данной модели процесса приведет к достижению цели разработки ПО с высокими показателями К&П.
- Модель процесса фактически является объединением лучших практических решений в некоторый «рецепт» успешного выполнения проекта.
- Процесс часто специфицируется (детально описывается) в виде последовательностей высокоуровневых этапов.
- В свою очередь каждый из этапов состоит из более мелких шагов и иногда называется подпроцессом.

Процессы разработки ПО

- Т.к. при разработке ПО требуется достижение разных целей, то требуются разные процессы.
- Многие из них не касаются ТРПО, но влияют на разработку ПО.
- Все процессы можно разделить на две большие группы:
  - Процессы не связанные непосредственно с разработкой ПО
  - Процессы непосредственно связанные с разработкой ПО.

Процессы не связанные непосредственно с разработкой ПО

- ***бизнес-процессы*** –
  - поиска заказчиков,
  - ведение переговоров,
  - убеждение в необходимости заключения договора,
  - оформление договора,
  - договор о цене,
  - оформление оплаты;
- ***социальные процессы*** –
  - организация команд,
  - создание творческой атмосферы,
  - праздничные вечера,
  - коллективные поездки;
- ***процесс обучения*** – лекции, мастер-классы, участие в конференциях.
- ***реклама***
- ***научная работа***
- и пр.

Работаем над задачами проекта

SMART (Specific, Measurable, Achievable, Realistic, Time-limited):

- Для того чтобы проверить, правильно ли вы сформулировали задачи, можно применить метод SMART (Specific, Measurable, Achievable, Realistic, Time-limited):

Программные процессы (software process)

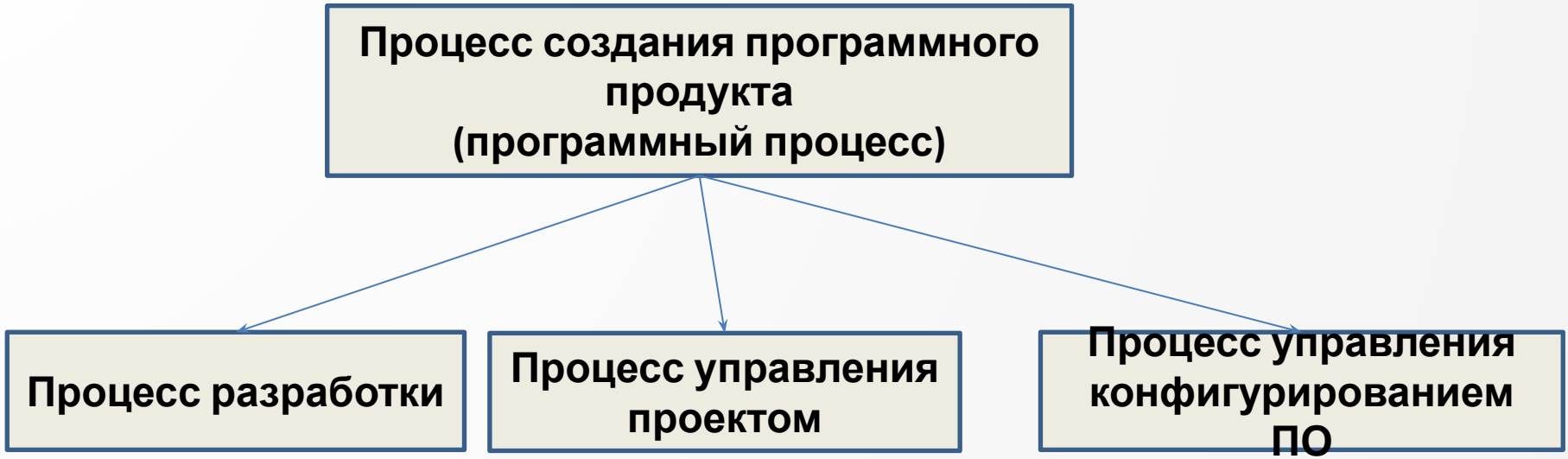
- *Процессы, который непосредственно имеет дело с техническими и управленческими задачами разработки ПО в общем называются* ***программным процессом*** *(**software process**)*.
- Разработка ПО разделена по проектам – в каждом проекте создается ПО для конкретного заказчика.

Программный процесс

- Программный проект (software project) должен
  1. разрабатывать программное обеспечение;
  2. выполнять правильное управление данным проектом.
- В связи с этим в программный процесс состоит из двух основных подпроцессов:
  1. процесс разработки – определяет все требуемые виды инженерной деятельности по созданию ПО;
  2. процесс управления проектом – определяет то, как планируются и управляются эти виды деятельности.
- Процессы эффективной разработки ПО и управления проектом являются основными факторами достижения цели предоставления желаемого ПО, удовлетворяющего требованиям заказчика, при обеспечении высокой производительности и качества.

Программный процесс

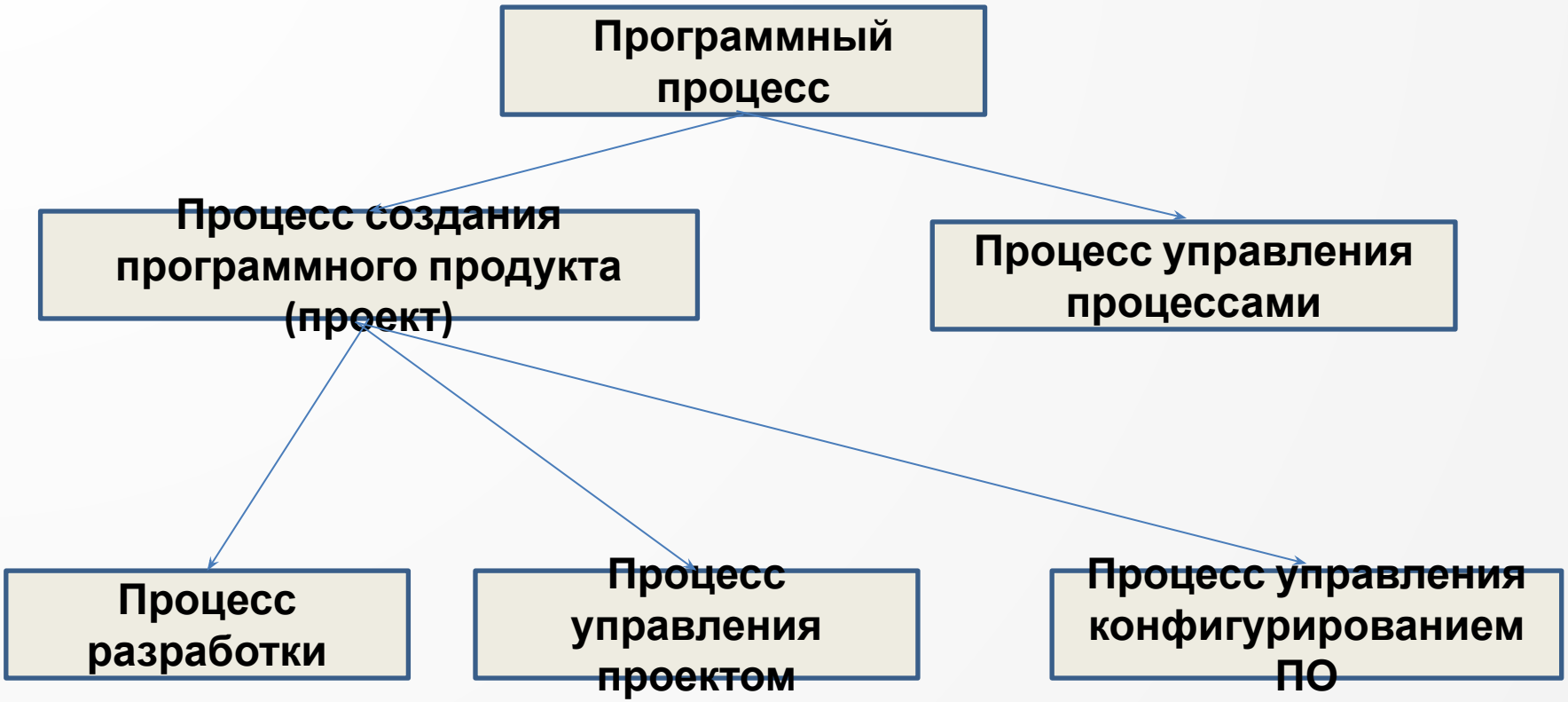
- Подпроцессы программного процесса



SEPG - A Software Engineering Process Group (SEPG) - является координационным центром организации с деятельности по усовершенствованию процессов программного обеспечения. Эти люди выполняют оценку организационных возможностей, разрабатывают планы по осуществлению необходимых улучшений

Программный процесс

- Подпроцессы программного процесса



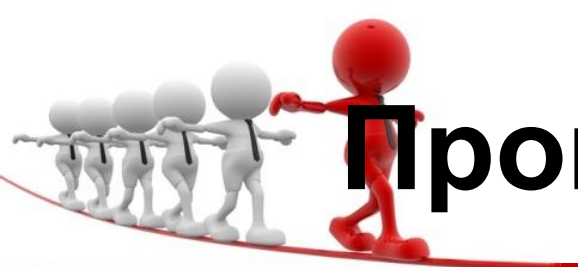
Подпроцесс управления конфигурированием ПО

- В проекте разрабатываются много разных компонент (например, конечный исходный код может состоять из большого числа исходных файлов).
- Эти компоненты развиваются по мере выполнения проекта, создается большое число их версий.
- Для управления развитием и изменениями часто используется подпроцесс управления конфигурированием ПО.
- Данный подпроцесс в основном связан с таким управлением изменениями, чтобы при этом не нарушалась целостность программных продуктов.

Процесс управления процессами разработки ПО

- Сами программные процессы изменяются, развиваются, чтобы приспособиться к улучшению понимания разработки ПО и доступности новых технологий и инструментов
- Основная цель процесс управления процессами разработки ПО является улучшение программного процесса.
- Под улучшением программного процесса понимается их совершенствование для разработки качественных продуктов при низких затратах.

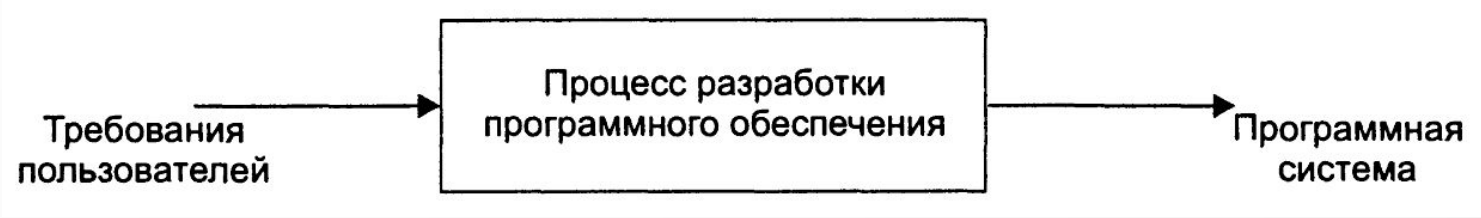
Процесс разработки ПО



Процесс разработки ПО

- Процесса разработки ПО является базовым процессом и его целью является создание высококачественного программного продукта.
- Он включает работы, непосредственно связанные с созданием ПО (такие, как проектирование, кодирование и тестирование).
- Процесс управления принимает решения на основе результатов работы процесса разработки.
- **Процесса разработки ПО определяет то, какие задачи должны решаться в проекте и в каком порядке они должны выполняться.**
- Он ограничивает свободу выполнения проекта, таким образом, что «кратчайшим» путем (или даже наиболее эффективным) перейти от требований потребителя к ПО, которое им удовлетворяет
- Данный процесс направляет проект и существенно влияет на полученные результаты.

- **Процесс разработки программного обеспечения**



Жизненный цикл ПО

- **Жизненный цикл** ПО это период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.
- Этот цикл включает процесс построения и развития ПО.
  1. Выявление и спецификация требований.
  2. Анализ
  3. Проектирование
  4. Кодирование
  5. Тестирование
  6. Передача заказчику
  7. Сопровождение
  8. Развитие

Модели процессов разработки ПО

- Модель процесса (МП) описывает обобщенный процесс, и обычно включает:
  - набор этапов, на которые должен быть разделен процесс;
  - порядок, в котором эти процессы должны выполняться;
  - разные ограничения и условия на выполнение этих этапов.
- Основное предположение для модели процесса заключается в том, что в тех ситуациях, когда данная модель является применимой, ее использование в качестве процесса проекта позволит:
  1. уменьшить расходы,
  2. повысить качество,
  3. уменьшить время разработки,
  4. а также принести другие выгоды.
- Другими словами, модель процесса предоставляет общие методические указания для разработки подходящего процесса выполнения проекта.

Основные модели

Основными моделями процесса разработки ПО являются следующие:

1. Модель водопада
2. Прототипирование
3. Итеративная разработка
4. Rational Unified Process
5. Модель временных ящиков
6. Гибкая разработка.

Модель процесса

- ***Модель процесса*** описывает (специфицирует) общий процесс*, который является «оптимальным» для* ***некоторого класса проектов***.
- Т.е. в ситуациях, для которых данная модель является применимой, использование данной модели процесса приведет к достижению цели разработки ПО с высокими показателями К&П.
- Модель процесса фактически является объединением лучших практических решений в некоторый «рецепт» успешного выполнения проекта.
- Процесс часто специфицируется (детально описывается) в виде последовательностей высокоуровневых этапов.
- В свою очередь каждый из этапов состоит из более мелких шагов и иногда называется подпроцессом.

1. Модель водопада

- Самой простой моделью процесса является модель водопада, в соответствии с которой все этапы организованы в линейном порядке.
- Имеется много вариантов данной модели, включающих разные виды работ и разный поток управления между ними.
- В данной модели проект начинается с ***анализа осуществимости***.
- При успешной демонстрации осуществимости проекта, начинается ***анализ требований и планирование проекта***.
- После завершения анализа требований начинается ***проектирование***, а после него начинается ***составление (кодирование) программы***.
- После успешного завершения программирования, созданный код объединяется и выполняется тестирование.
- После успешного завершения тестирования, система устанавливается у заказчика.
- После этого выполняется постоянное использование созданного ПО и его поддержка.

Модель водопада

1. Реализуемость системы

Отчет о реализуемости

2. Анализ требований и планирование проекта

Спецификация требований и план проекта

3. Системное проектирование

Описание проекта системы

4. Детальное проектирование

Детальный проект системы

5. Кодирование

Программный код

6. Тестирование

План и отчет о тестировании, руководства

7. Установка

Отчет об установке

8. Сопровождение и развитие

Модель водопада

- На данной схеме этап анализа требований назван «анализом и планированием».
- Планирование является критически важной работой для разработки ПО.
- Хороший план основывается на требованиях системы и должен выполняться до того, как начнутся другие этапы проекта.
- Однако на практике, детальные требования не обязательны для планирования, поэтому планирование обычно выполняется одновременно с анализом требований и план подготавливается до того, как начнутся другие этапы проекта.
- Разработанный план является дополнительными исходными данными для всех последующих этапов.

Модель водопада

- Линейный порядок работ имеет некоторые важные последствия.
- Для явного распознавания завершения очередного этапа и начала следующего, требуются использовать некоторый способ подтверждения в конце каждого этапа.
- Это обычно выполняется с помощью некоторой проверки (verification) и утверждения (validation), которые будут гарантировать, что:
  - результат выполнения этапа согласуется с его входными данными (что являлось результатом работы (выходом) предыдущего этапа);
  - результат выполнения данного этапа согласуется с общими требованиями заказчика к системе.
- Следствием потребности в подтверждении является то, что каждый этап должен иметь конкретный результат, который м.б. оценен и утвержден.

Модель водопада

- Результат каждого последующего этапа часто называется продуктом труда и обычно имеет форму некоторого документа (например, описания требований или описание проекта, программный код).
- Хотя набор документов, создаваемых в проекте зависит от его реализации, следующие документы обычно составляют обоснованный набор, который должен быть создан в каждом проекте:
  - спецификация требований;
  - план проекта;
  - документы по проектированию (архитектура системы, описание подсистем, детальный проект);
  - план тестирования и отчет о результатах ;
  - описание конечной программной реализации;
  - руководства по работе с ПО (администратора, пользователя).

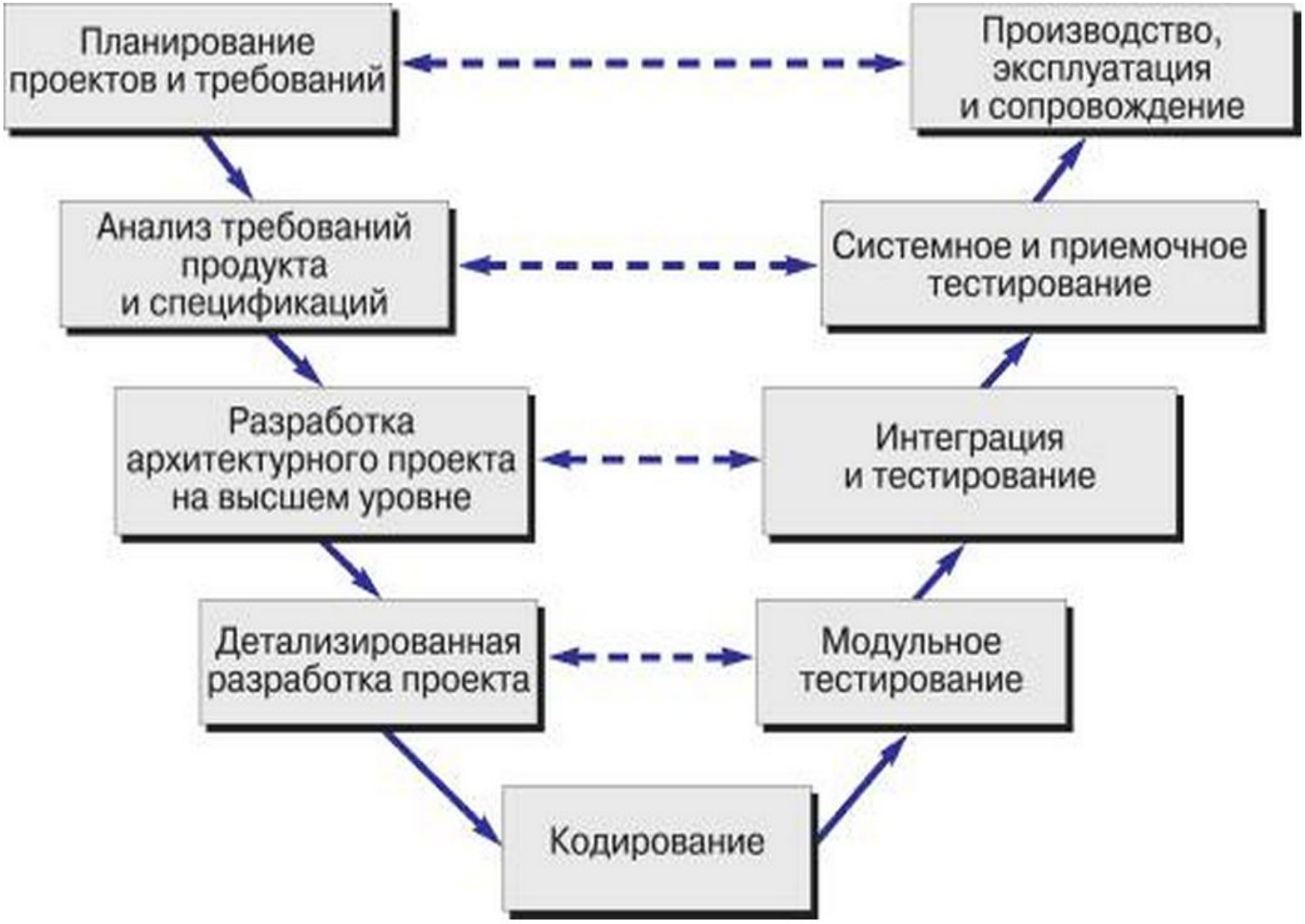
Преимущества модели водопада

- основным преимуществом является простота реализации;
- получение полной и согласованной документации на каждом этапе;
- легкость определения сроков и затрат на проект;
- простота администрирования ходом выполнения проекта (каждый этап завершается и создается его результат, некоторое количество денег передается потребителем разрабатывающей организации)

Недостатки

1. Предполагается, что ***требования к разрабатываемой системе м.б. неизменными (заморожены)*** до начала проектирования.
2. ***Замораживание требований обычно требует выбора технического обеспечения*** (т.к. это составляет часть спецификации требований).
3. Вызывает подход «большого взрыва» - полное ПО предоставляется за один раз, в конце разработки.
4. Поощряется «раздувание требований».
5. Данная модель является управляемой документами, что требует создание формального документа в конце каждого этапа.

V-образная модель



Выводы по модели водопадов

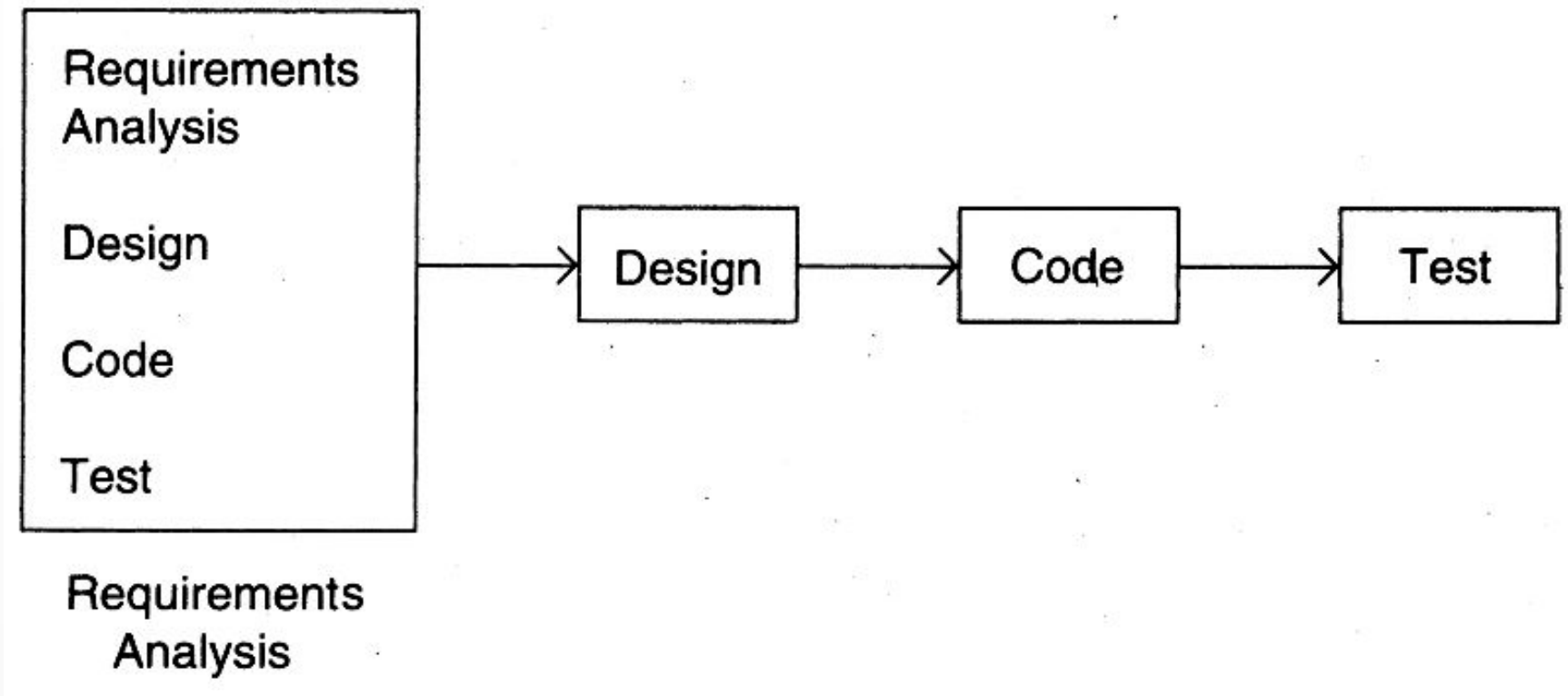
Достоинства	**Недостатки**	**Подходящие типы проектов**
❑Проста. ❑Легкая для выполнения. ❑Интуитивная и логически понятная. ❑Легкая для выполнения договора.	❑Все или ничего – слишком рисковано. ❑Рано замораживаются требования ❑Может быть выбрано устаревшее оборудование/технология. ❑Не позволяются изменения. ❑Нет обратной связи с пользователями ❑Поощряет раздувание требований.	❑Хорошо понятные проблемы. ❑Краткосрочные проекты. ❑Автоматизация существующих ручных систем.

2. Модель прототипирования

- Цель процесса разработки на основе прототипов заключается исправление первого ограничения модели водопадов.
- Основная идея – вместо замораживания требований, создавать, перед началом проектирования и кодирования, одноразовый прототип, который может помочь понять требования заказчика.
- Прототип сильно упрощенный вариант разрабатываемого ПО.
- Он разрабатывается на основе текущих известных требований.
- Очевидно, что разработка прототипа тоже проходит этапы проектирования, кодирования и тестирования, но каждый их этих этапов, но каждый из этих этапов выполняется не формально и не

- Прототипирование является привлекательной идеей для сложных и больших систем, для которых нет ручного процесса или существующей системы, которые могут помочь выявить требования.
- В таких ситуациях, предоставление заказчику возможности «поиграть» с прототипом позволяет выявить неоценимые и неосязаемые результаты, которые могут помочь определить требования к системе.
- Ситуации, в которых требуется пртотипирование:
  - для ***демонстрации осуществимости некоторого нового подхода***;
  - при разработке новых систем, когда не понятно, какие ограничения могут возникнуть, или какие алгоритмы могут быть разработаны для реализации требований.
- В обоих ситуациях риск, связанный с данными проектами м.б. уменьшен, за счет выполнения прототипирования.

Схема модели прототипирования



Процесс использования прототипа

- Процесс разработки с использованием прототипа обычно выполняется следующим образом:

1. Разработка прототипа обычно начинается, когда разработана предварительная версия документа с спецификациями требований.
2. После разработки прототипа, конечные пользователи и заказчику могут использовать и исследовать прототип.
   - Описываются и передаются разработчикам отзывы: что было сделано правильно; чего не хватает; что требуется добавить и т.п.
3. На основе отзывов выполняется доработка прототипа, а затем пользователям и заказчикам опять предоставляется возможность использовать данную ПС.
4. Такой цикл повторяется до тех пор, пока выгода от дальнейшего изменения прототипа превышает стоимость его доработки.
5. На основе такой обратной связи (отзывов), начальные требования модифицируются, чтобы получить окончательную спецификацию требований, которая затем используется разработки ПС производственного уровня.

Стоимость разработки прототипа

- Стоимость разработки прототипа должна быть очень низкой.
- Для этого в прототип включаются только те возможности, которые будут полезными для получения отзыва от пользователей.
- Способы уменьшения затрат на создание прототипа:
  - исключение модулей для реализации тех требований к ПО, которые уже хорошо понятны;
  - использование «быстрого и грязного» подхода, нацеленного на быструю разработку, а не на качество;
  - не используется: обработка исключений, аварийное восстановление, соответствие стандартам и форматам и т.п.;
  - создание только минимальной документации;
  - выполнение только минимально необходимого тестирования.
- С помощью таких способов уменьшения затрат на разработку прототипа, возможно довести стоимость создания прототипа до нескольких процентов общей

Достоинства прототипирования

1. Опыт полученный при разработке прототипа уменьшает стоимость разработки конечного ПО.
2. Получаются более стабильные требования, благодаря обратной связи с пользователями в требованиях будет меньше изменений.
3. Вероятно, что качество конечного ПО будет намного лучше, в связи с тем, что опыт полученный разработчики в ходе создания прототипа позволит улучшить проектирование, составление кода и выполнения тестирования.

Общий вывод

- Обычно прототипирование хорошо подходит для проектов, в которых трудно выявить требования и доверие к заявленным требованиям низкая.
- В тех проектах, у которых требования не совсем понятны в начале работы, использование модели процесса «прототипирование» может быть наиболее методом разработки ПО.
- Данный метод также является хорошим способом уменьшения некоторых рисков, связанных с проектом.

Выводы по прототипированию

Достоинства	**Недостатки**	**Подходящие типы проектов**
❑Помогает выявить требования. ❑Уменьшает риски. ❑Получается более стабильная конечная система.	❑Тяжелое начало. ❑Возможны высокая стоимость и сжатые сроки. ❑Поощряет раздувание требований. ❑Затрудняет поздние изменения требований.	❑Системы с неопытными пользователями; ❑Области с неопределенными требованиями. ❑Системы с большим кол-вом отчетов могут выиграть от прототипа пользовательского интерфейса.

3. Итеративная разработка

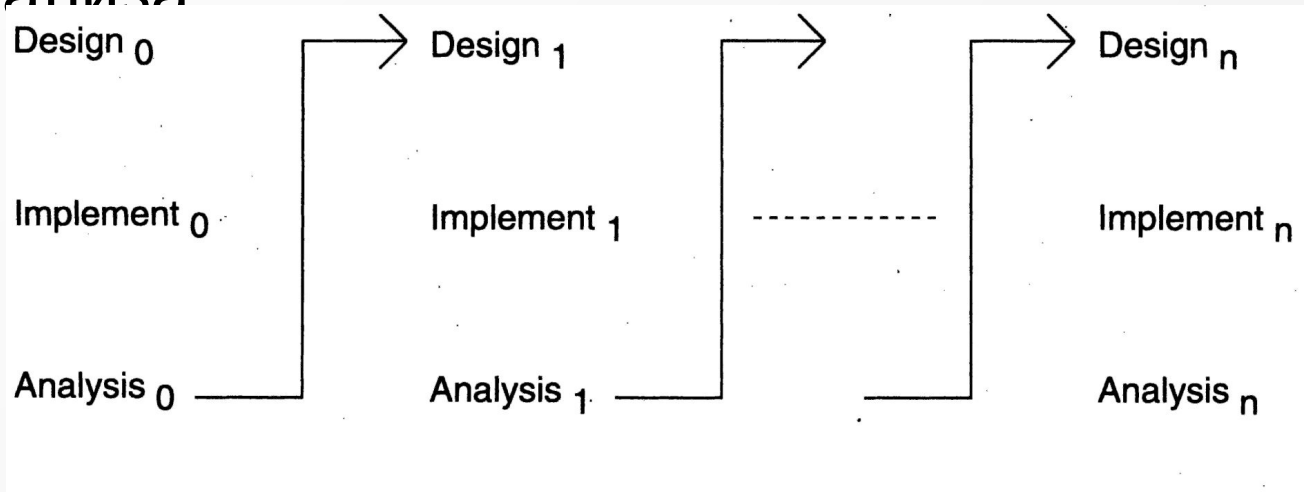
- Основная идея – пошаговая разработка ПО (постепенная, итеративная, инкрементальная).
- На каждом шаге к создаваемой ПС должны добавляться некоторые функциональные возможности, до тех пор, пока система не будет реализована полностью.
- На первом шаге данной модели создается простое начальное приложение, реализующее подмножество задач полного решения проблемы.
- Данное подмножество должно включать некоторые из основных задач рассматриваемой проблемы, которые легко понять и реализовать, и которые формируют полезную и простую в применении систему.

- Создается ***управляющий список***, который содержит упорядоченный набор задач, которые должны быть реализованы для получения конечного решения.
- Данный список проекта дает представление о том, как много этапов должно быть выполнено для получения окончательного варианта ПС.
- Управляющий список проекта руководит последовательностью шагов и содержит список всех задач, которые должны быть выполнены.

- На основе анализа, к задачам данного списка может относиться перепроектирование неправильно разработанных модулей или перепроектирование всей системы в целом.
- Однако перепроектирование системы обычно встречается только на начальных этапах.
- На более поздних этапах, проект должен стабилизироваться и вероятность его изменения должна уменьшаться.

Итеративная модель улучшения ПО

- Каждый шаг итеративной модели состоит из удаления следующей задачи из списка:
  1. проектирование реализации выбранной задачи
  2. кодирование и тестирование созданной реализации
  3. выполнение анализа частично разработанной системы, поученной после данного шага и обновление списка на основе результатов анализа



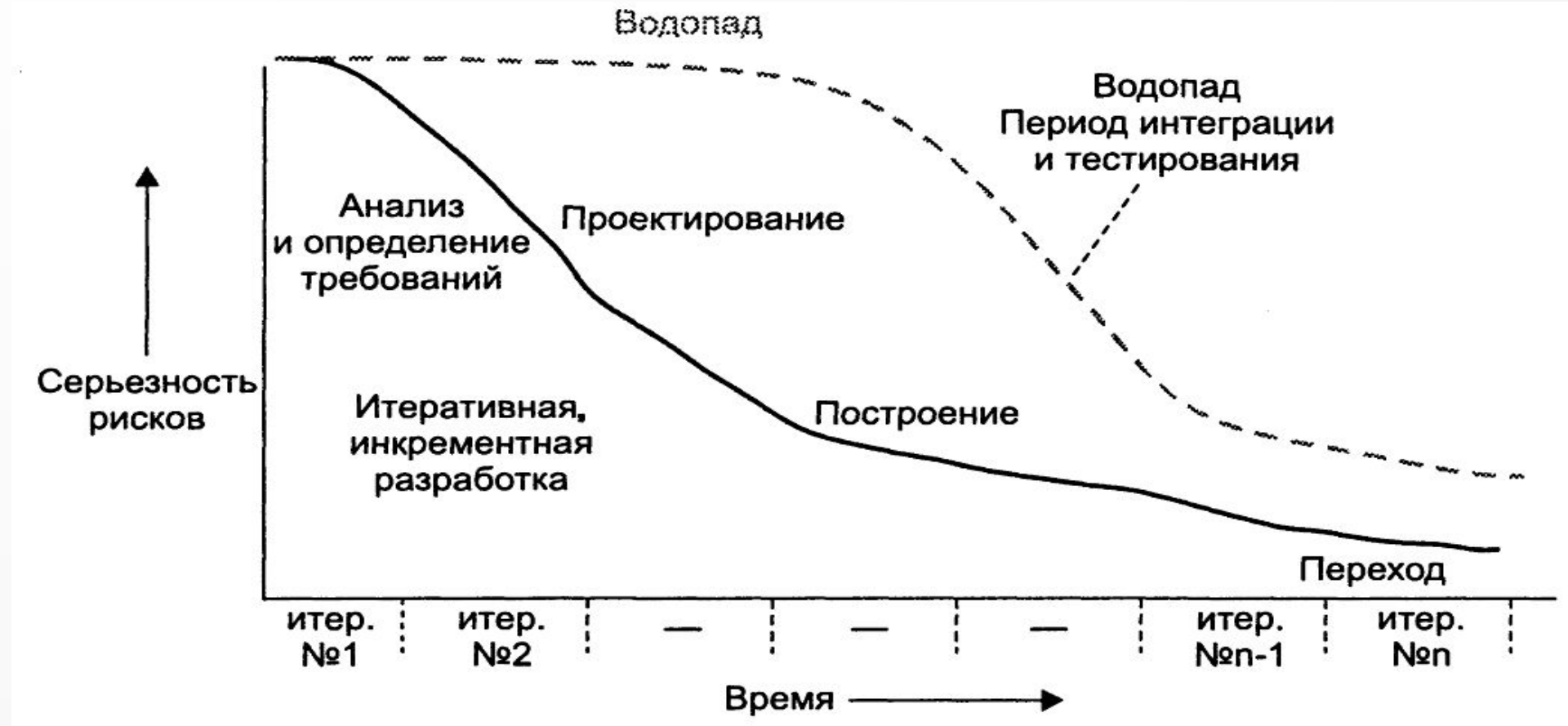
- ***Итеративная разработка в настоящее время является наиболее общим подходом для создания прикладных систем.***
- Данный подход м.б. планово-управляемым, гибким или, более часто, смесью этих подходов.
  - при планово-управляемым подходе требования к системе выявляются заранее.
  - при гибком подходе начальные итерации планируются, но разработка последующих итераций зависит от полученных результатов и приоритетов пользователей.

Достоинства

- Уменьшаются затраты на приспособление к изменившимся требованиям к ПО.
  - Уменьшается объем анализа и кол-во документации, которые должны быть переделаны.
- Легче получить отзыв потребителя на выполненные результаты разработки.
  - Потребитель может комментировать показываемое ему ПО и видеть, насколько много было разработано.
  - Потребителю трудно оценивать ход разработки на основе документов проектирования ПО.
- Возможно ускорить предоставление и развертывание полезного ПО у потребителя, даже если не вся требуемая функциональность в нем имеется.
- Отсутствует риска получить «все или ничего».

Сравнение рисков итеративной и водопадной моделей

- Серьезные риски при итеративной разработке определяются и уменьшаются раньше, чем при водопадной



Недостатки

- Очень долгое время отсутствует целостное понимание возможностей и ограничений проекта.
- При итерациях приходится отбрасывать часть сделанной ранее работы.
- Снижается добросовестность специалистов при выполнении работ, что психологически объяснимо, ведь над ними постоянно довлеет ощущение, что «всё равно всё можно будет переделать и улучшить позже».

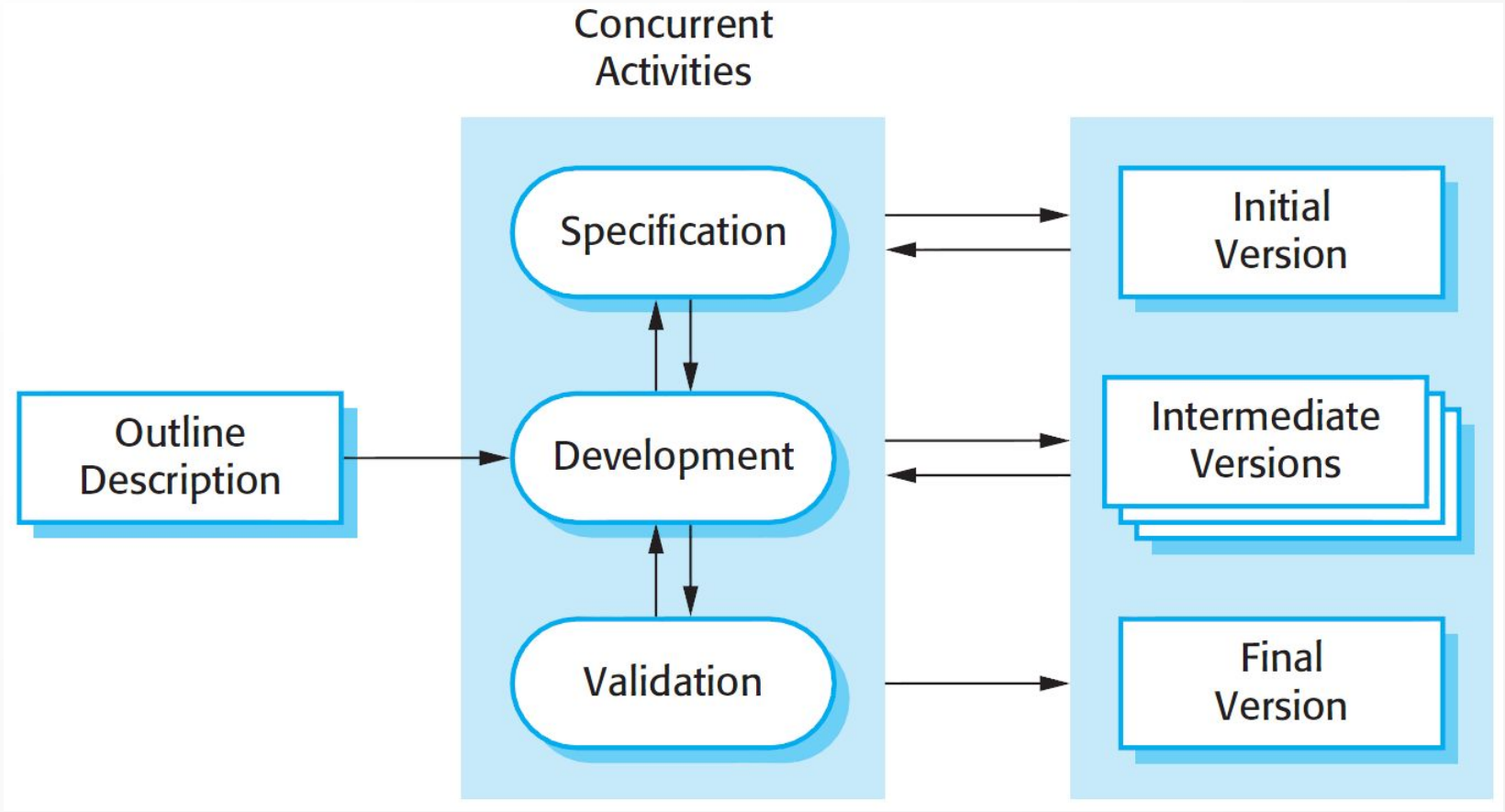
- Инкрементальную или гибкую модель разработки не всегда легко вводить или использовать в больших компаниях со стандартизированными инженерными процессами и в организациях, в которых разработка ПО обычно передается внешним исполнителям (аутсорсинг).
- Если разработка передается внешним исполнителям, то и клиент и исполнитель обычно хотят иметь полное описание того, что уже было сделано.

Основные трудности инкрементальной разработки

1. Проблемы менеджмента
   - Структура управления разработкой ПО в больших организациях создана для работы с такими моделями процесса разработки ПО, которые создают регулярно результаты, по которым можно оценить прогресс разработки.
   - Инкрементальные системы разработки меняются так быстро, что не выгодно по затратам создавать большое количество системной документации.
   - Менеджеры могут посчитать затруднительным использовать существующих сотрудников для выполнения инкрементальных процессов разработки, т.к. они не обладают таким опытом (умением).

Incremental development

- Figure 2.2 Incremental development



Причины популярности итеративной модели

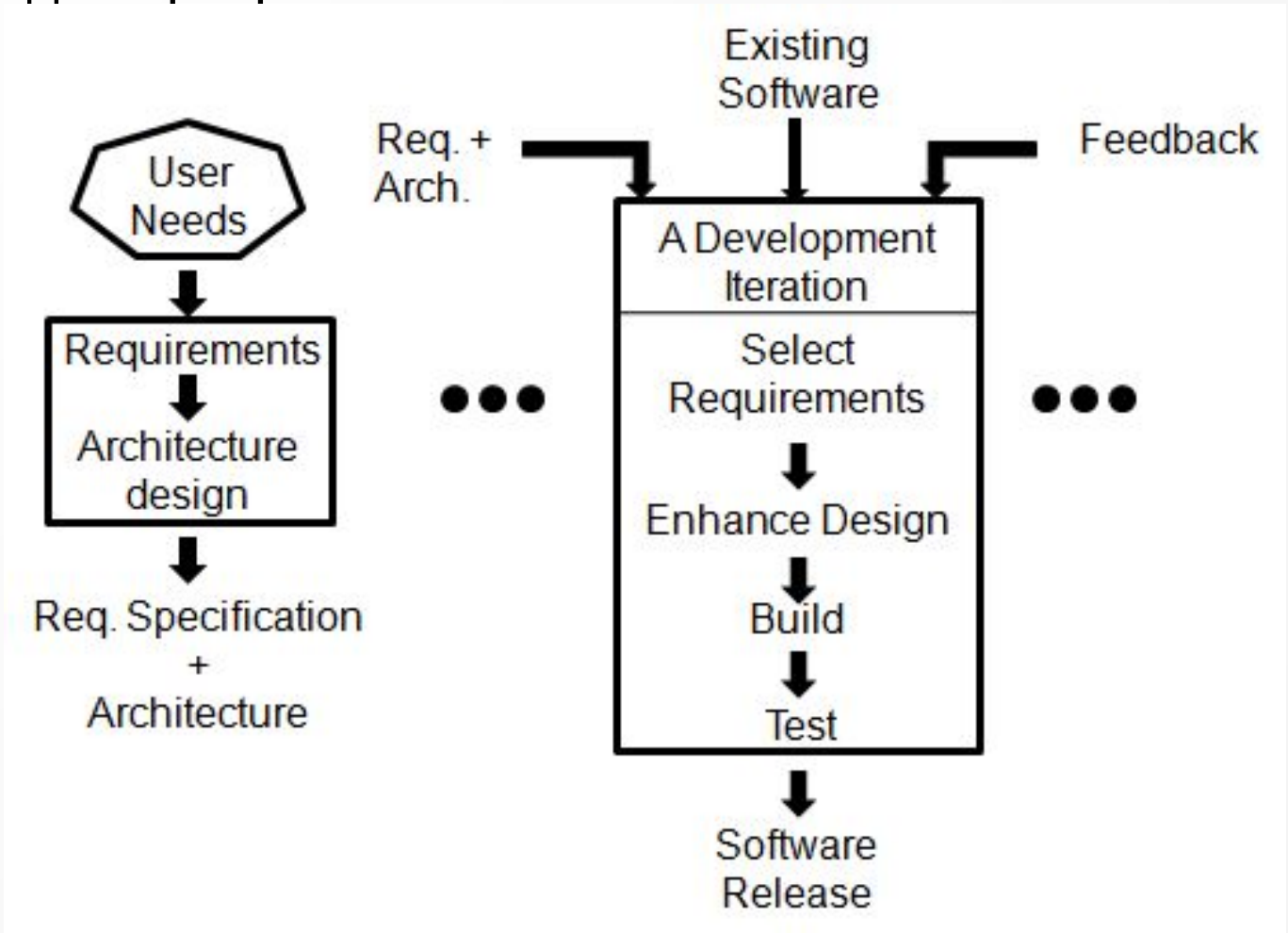
- В современном мире заказчик не хочет вкладывать деньги, если он не видит результаты от их вложения.
- Т.к. бизнес меняется быстро, то они никогда не знают «полностью» требования к ПО и им нужна возможность постоянно добавлять новые возможности к создаваемому ПО, чтобы оно соответствовало новым условиям.
- На каждой итерации создается работающая версия ПО, которая помогает разработать стабильные требования для следующей итерации.

Выводы по итеративному подходу

Достоинства	**Недостатки**	**Подходящие типы проектов**
❑Regular deliveries, leading to biz benefit ❑Can accommodate changes naturally ❑Allows user feedback ❑Avoids req bloating ❑Naturally prioritizes req ❑Allows reasonable exit points ❑Reduces risks	Overhead of planning each iteration Total cost may increase System arch and design may suffer Rework may increase	For businesses where time is imp; risk of long projects cannot be taken; req not known and evolve with time

Вариант итеративной модели

- **Инкрементальная модель** это вариант итеративной модели разработки



Достоинства данного варианта

- В большинстве случаев, требования к ПО заранее не известны.
- Доступно общее представление о ПС и на основе этого м.б разработана подходящая архитектура, которая остается относительно стабильной.
- В связи с этим есть надежда, что количество переделок в итерациях разработки будет уменьшаться.
- Одновременно, конечный результат будет предоставляться пользователю итеративно (постепенно), так что не будет риска получить «все или ничего».
- Т.к. поставка ПО выполняется постепенно, а планирование и выполнение каждой итерации выполняется отдельно, то пожелания пользователей м.б. включены в следующие итерации.
- Даже новые требования, которые были первоначально не выявлены, также могут быть включены в последующие итерации.

Итеративная модель

- *Рис. 4. Итеративная модель предлагает использование итераций на всех этапах жизненного цикла.*

Спиральная модель

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

4. Унифицированный процесс разработки ПО (Rational Unified Process, RUP)

- Унифицированный процесс это вариант итеративного процесса, разработанный компанией Rational Software, (сейчас подразделение IBM).
- общая модель процесса, созданная для ОО разработки с использованием UML.
- Унифицированный процесс (UP) разрабатывается с 1967 года.
  - управляемым рисками и прецедентами (требованиями);
  - архитектуро-центричным;
  - итеративным и инкрементным.
- Это сложившийся открытый процесс разработки ПО от авторов UML.
- Унифицированный процесс компании Rational (RUP) – это коммерческое расширение UP.

- Процесс, направляемый вариантами использования
- Архитектуро-центрированный процесс
- Итеративный и инкрементный процесс

- Унифицированный процесс (UP) разрабатывается с 1967 года.
  - управляемым рисками и прецедентами (требованиями);
  - архитектуро-центричным;
  - итеративным и инкрементным.
- Это сложившийся открытый процесс разработки ПО от авторов UML.
- Унифицированный процесс компании Rational (RUP) – это коммерческое расширение UP.
- Он полностью совместим с UP, но более полный и детализированный.
- UP (и RUP) должны настраиваться под каждый конкретный проект путем добавления внутренних стандартов и др.

Спасибо за внимание

Если ты направился к задуманной цели и постоянно останавливаешься, чтобы швырять камни во всякую лающую на тебя собаку, то никогда не дойдешь до цели.