

ЛЕКЦИЯ 4. МЕТОДОЛОГИИ РАЗРАБОТКИ ПО

Введение

- **Методология разработки ПО** – это совокупность принципов, идей, понятий, методов, способов и средств, определяющих стиль разработки ПО.

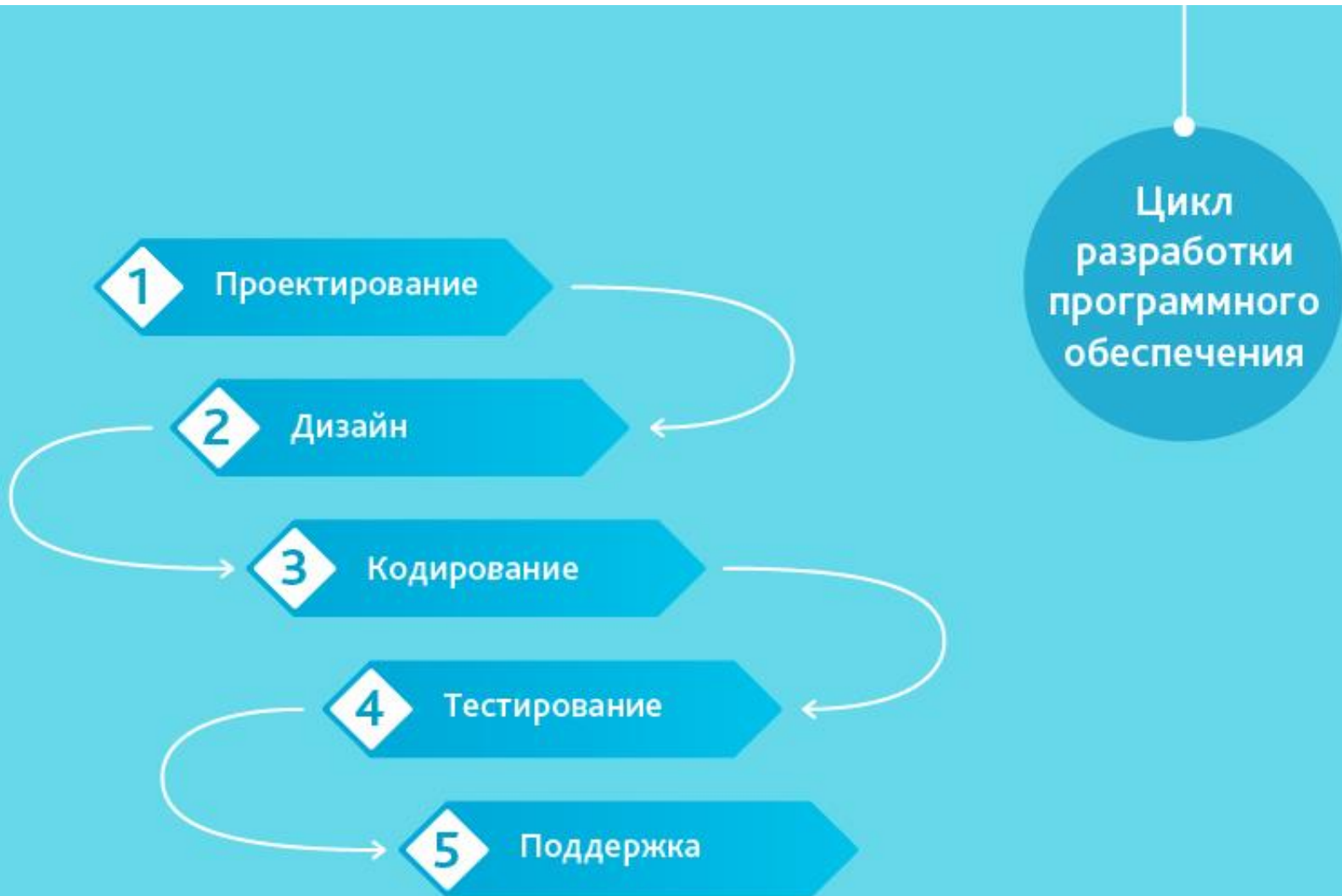
- **Выбор методологии зависит от:**
 - ✓ размера команды;
 - ✓ специфики и сложности проекта;
 - ✓ стабильности процессов в компании
 - ✓ личных качеств сотрудников.

«Waterfall Model»

(каскадная модель или «водопад»)

- Последовательное выполнение всех этапов проекта в строго фиксированном порядке.
- Переход на следующий этап означает полное завершение работ на предыдущем этапе.
- Требования строго документируются в виде ТЗ и фиксируются на все время разработки проекта.
- Каждая стадия завершается выпуском полного комплекта документации.

«Waterfall Model» (каскадная модель или «водопад»)



Преимущества каскадной модели

- широкая известность и простота модели;
- упорядоченность преодоления сложностей и хорошо срабатывает для тех проектов, которые достаточно понятны, но все же трудно разрешимы;
- отличается стабильностью требований;
- удобна, когда требования к качеству доминируют над требованиями к затратам и графику выполнения проекта;
- способствует осуществлению строгого контроля менеджмента проекта;
- позволяет участникам проекта, завершившим действия на выполняемой ими фазе, принять участие в реализации других проектов;
- определяет процедуры по контролю за качеством;
- стадии модели довольно хорошо определены и понятны, легко отслеживаются с помощью временной шкалы или графика Гантта.

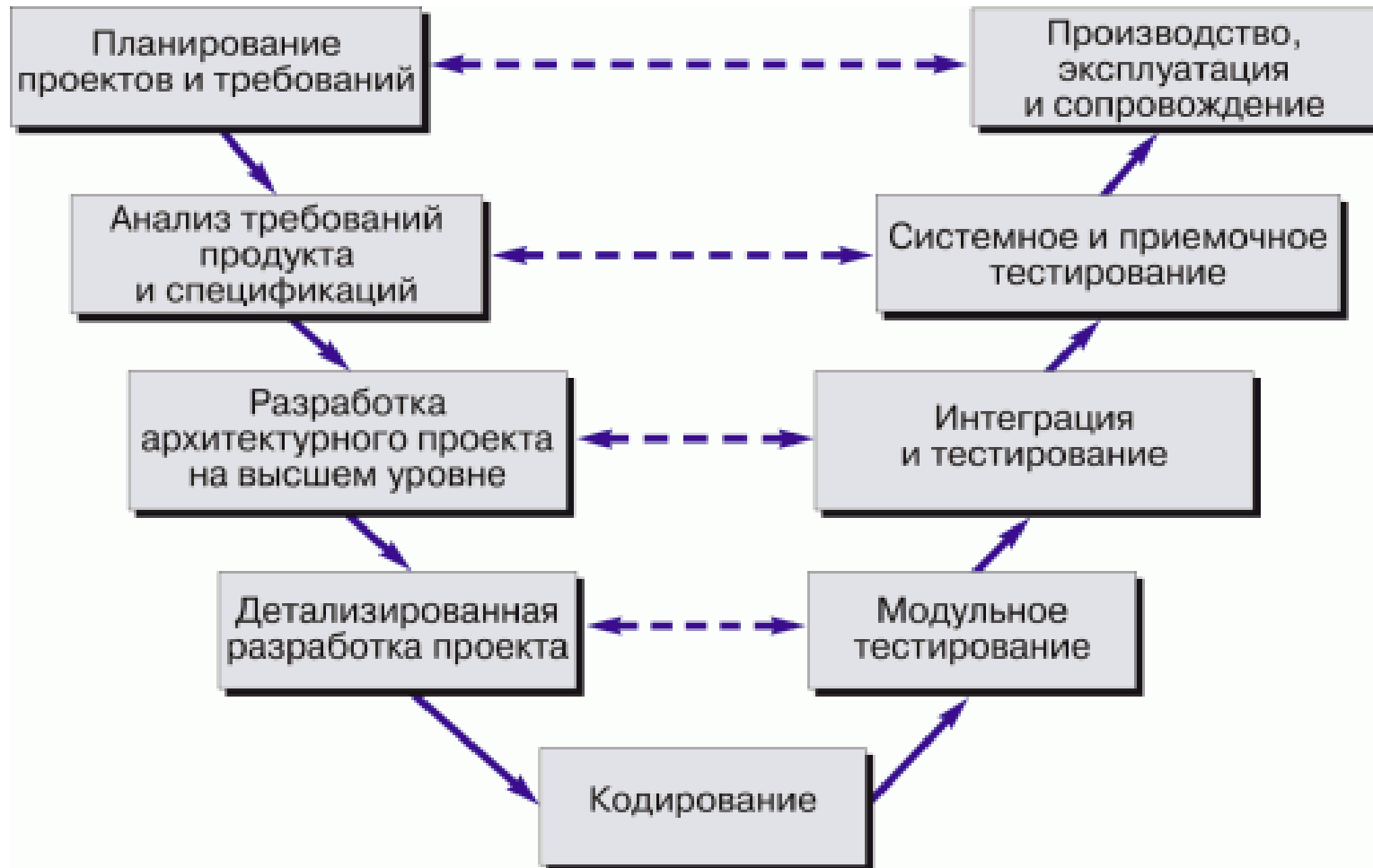
Недостатки каскадной модели

- ❑ в основе модели лежит последовательная линейная структура;
- ❑ невозможность предотвращения возникновения итераций между фазами;
- ❑ она может создать ошибочное впечатление о работе над проектом;
- ❑ интеграция всех полученных результатов происходит внезапно в завершающей стадии работы модели и у клиента практически нет возможности ознакомиться с системой заранее;
- ❑ каждая фаза является предпосылкой для выполнения последующих действий и ее результат считается замороженным;
- ❑ все требования должны быть известны в начале жизненного цикла;
- ❑ необходимость в жестком управлении и контроле;
- ❑ модель основана на документации;
- ❑ весь программный продукт разрабатывается за один раз;
- ❑ отсутствует возможность учесть переделку и итерации за рамками проекта.

Каскадная модель – когда использовать?

- Только тогда, когда требования известны, понятны и зафиксированы.
- Нет проблем с доступностью программистов нужной квалификации.
- В небольших проектах.
- В приложениях, где критична безопасность – банковская сфера, медицина и т.д.

V-образная модель



V-образная модель

- планирование проекта и требований – определяются системные требования, а также то, каким образом будут распределены ресурсы организации с целью их соответствия поставленным требованиям;
- анализ требований к продукту и его спецификации – анализ существующей на данный момент проблемы с ПО, завершается полной спецификацией ожидаемой внешней линии поведения создаваемой программной системы;
- архитектура или проектирование на высшем уровне – определяет, каким образом функции ПО должны применяться при реализации проекта;
- детализированная разработка проекта – определяет и документально обосновывает алгоритмы для каждого компонента, который был определен на фазе построения архитектуры;

Преимущества V-образной модели

- в модели особое значение придается планированию, направленному на верификацию и аттестацию разрабатываемого продукта на ранних стадиях его разработки;
- в модели предусмотрены аттестация и верификация всех внешних и внутренних полученных данных, а не только самого программного продукта;
- 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;
- благодаря модели менеджеры проекта может отслеживать ход процесса разработки, так как в данном случае вполне возможно воспользоваться временной шкалой, а завершение каждой фазы является контрольной точкой;
- 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;
- тестирование требований в жизненном цикле происходит слишком поздно, вследствие чего невозможно внести изменения, не повлияв при этом на график выполнения проекта;
- в модель не входят действия, направленные на анализ рисков.

Область применения V-образной модели

V-образная модель лучше всего срабатывает тогда, когда вся информация о требованиях доступна заранее.

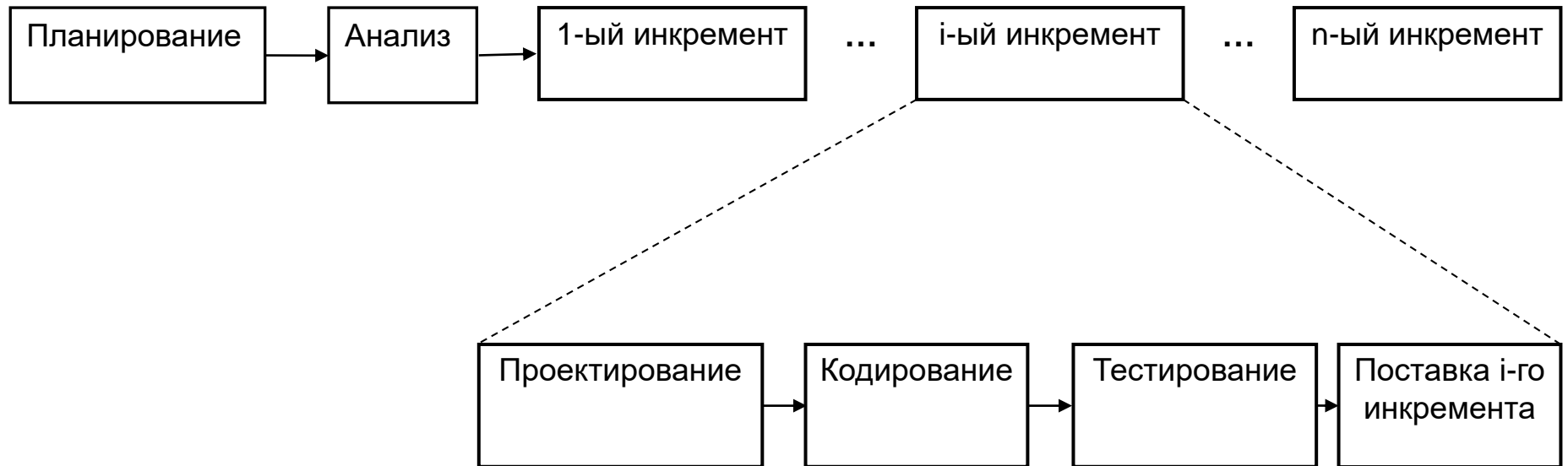
Использование модели эффективно в том случае, когда доступными являются информация о методе реализации решения и технология, а персонал владеет необходимыми умениями и опытом в работе с данной технологией.

V-образная модель – это отличный выбор для систем, в которых требуется высокая надежность.

«Incremental Model» (инкрементная модель)



Инкрементная модель ЖЦ



Преимущества инкрементной

модели

- в результате выполнения каждого инкремента получается функциональный продукт;
- заказчик располагает возможностью высказаться по поводу каждой разработанной версии системы;
- правило по принципу "разделяй и властвуй" позволяет разбить возникшую проблему на управляемые части;
- заказчики могут распознавать самые важные и полезные функциональные возможности продукта на более ранних этапах разработки;
- требования стабилизируются на момент создания определенного инкремента;
- инкременты функциональных возможностей несут больше пользы и проще при тестировании
- 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

- определение полной функциональной системы должно осуществляться в начале ЖЦ, чтобы обеспечить определение инкрементов;
- для модели необходимы хорошее планирование и проектирование;
- использование на этапе анализа общих целей, вместо полностью сформулированных требований, может оказаться неудобным для руководства;
- поскольку создание некоторых модулей будет завершено значительно раньше других, возникает необходимость в четко определенных интерфейсах;
- заказчик должен осознавать, что общие затраты на выполнение проекта не будут снижены;
- в модели не предусмотрены итерации в рамках каждого инкремента;
- формальный критический анализ и проверку намного труднее выполнить для инкрементов, чем для системы в целом;
- может возникнуть тенденция к оттягиванию решений трудных проблем на будущее с целью продемонстрировать руководству успех, достигнутый на ранних этапах разработки.

Критерии применения инкрементной модели

- если большинство требований можно сформулировать заранее, но их появление ожидается через определенный период времени;
- если рыночное окно слишком "узкое" и существует потребность быстро поставить на рынок продукт, имеющий функциональные базовые свойства;
- при разработке программ, связанных с низкой или средней степенью риска;
- при выполнении проекта с применением новой технологии, что позволяет пользователю адаптироваться к системе путем выполнения более мелких инкрементных шагов, без резкого перехода к применению основного нового продукта;
- когда однократная разработка системы связана с большой степенью риска.

Инкрементная модель – когда использовать?

- Когда основные требования к системе четко определены и понятны. Некоторые детали могут дорабатываться с течением времени.
- Требуется ранний вывод продукта на рынок.

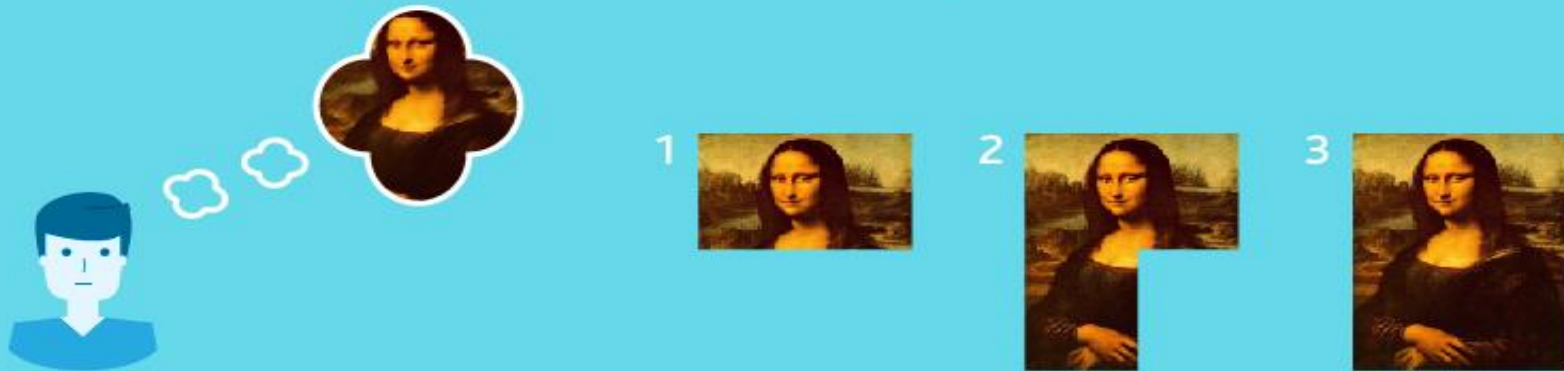
«Iterative Model»

(итеративная или итерационная модель)

- Не требует в начале полной спецификации требований.
- Создание начинается с реализации части функционала, становящейся базой для определения дальнейших требований. Этот процесс повторяется.
- Версия может быть неидеальна, главное, чтобы она работала.

Сравнение инкрементной и итеративной моделей

Инкрементная модель



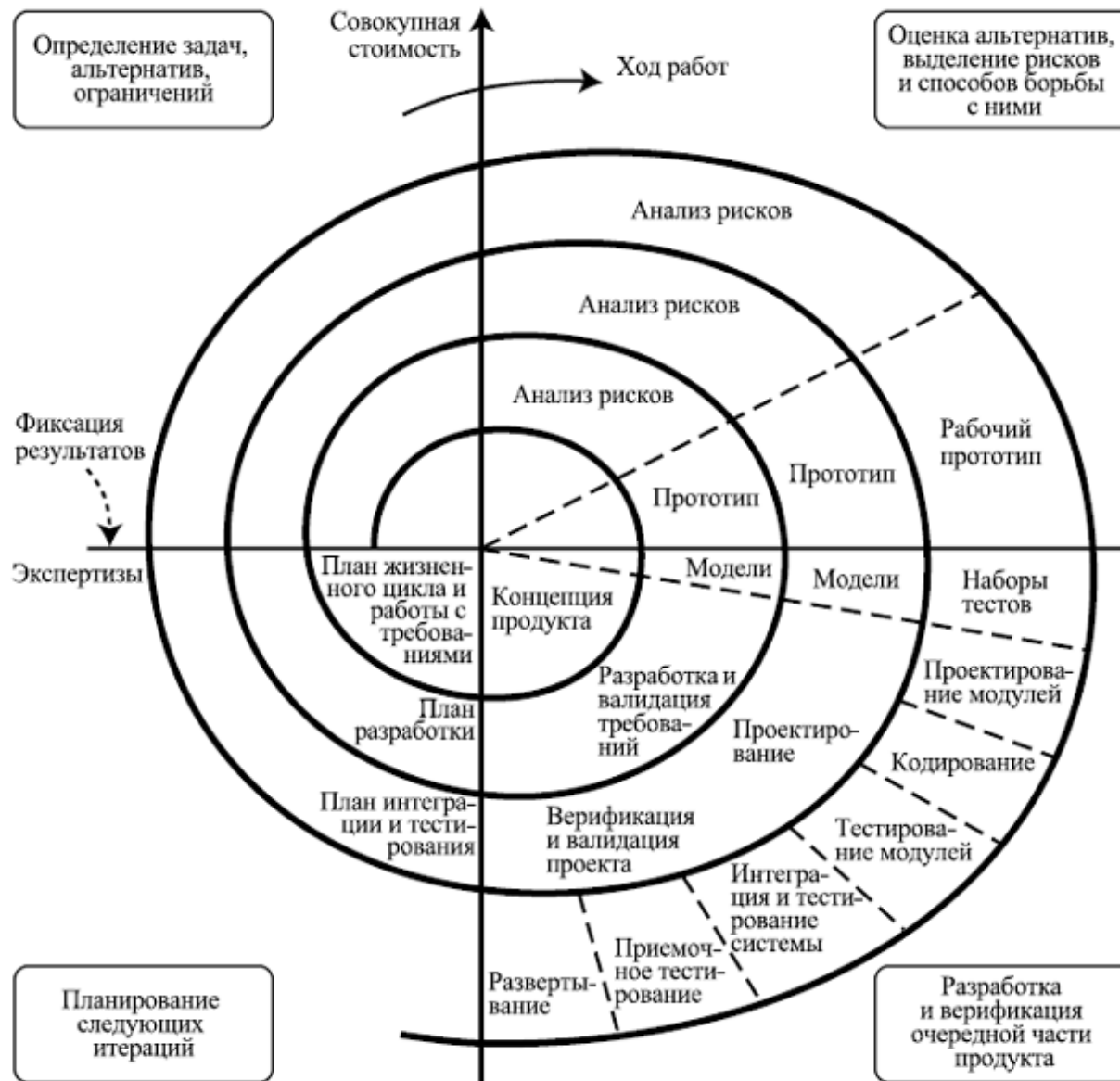
Итеративная модель



Итеративная модель – когда использовать?

- Требования к конечной системе заранее четко определены и понятны.
- Проект большой или очень большой.
- Основная задача должна быть определена, но детали реализации могут эволюционировать с течением времени.

Спиральная модель



Спиральная модель

Определение задач, альтернатив, ограничений:

- Выполняется определение целей (рабочая характеристика, выполняемые функции, возможность внесения изменений, решающих факторов достижения успеха и аппаратного/программного интерфейса).
- Определяются альтернативные способы реализации этой части продукта (конструирование, повторное использование, покупка, субдоговор, и т.п.). Определяются ограничения, налагаемые на применение альтернативных вариантов (затраты, график выполнения, интерфейс, ограничения, относящиеся к среде и др.).
- Создается документация, подтверждающая риски, связанные с недостатком опыта в данной сфере, применением новой технологии, жесткими графиками, плохо организованными процессами и т.д.;

Спиральная модель

Оценка альтернатив, выделение рисков и способов борьбы с ними:

- Выполняется оценка альтернативных вариантов, относящихся к целям и ограничениям.
- Выполняется определение и разрешение рисков.
- Принятие решения о прекращении/продолжении работ над проектом.

Разработка и верификация очередной версии продукта:

- создание проекта;
- критический анализ проекта;
- разработка кода;
- проверка кода;
- тестирование и компоновка продукта.

Спиральная модель

Планирование следующих итераций:

- разработка плана проекта,
- разработка плана менеджмента конфигурацией,
- разработка плана тестирования;
- разработка плана установки программного продукта.

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;
- реализованы преимущества инкрементной модели (выпуск инкрементов, сокращение графика посредством перекрывания инкрементов);
- быстрая обратная связь по направлению от пользователей к разработчикам;
- обеспечивается определение непреодолимых рисков без особых дополнительных затрат;
- эта модель разрешает пользователям активно принимать участие при планировании, анализе рисков, разработке, а также при выполнении оценочных действий;
- при использовании спиральной модели не нужно распределять заранее все необходимые для выполнения проекта финансовые ресурсы.

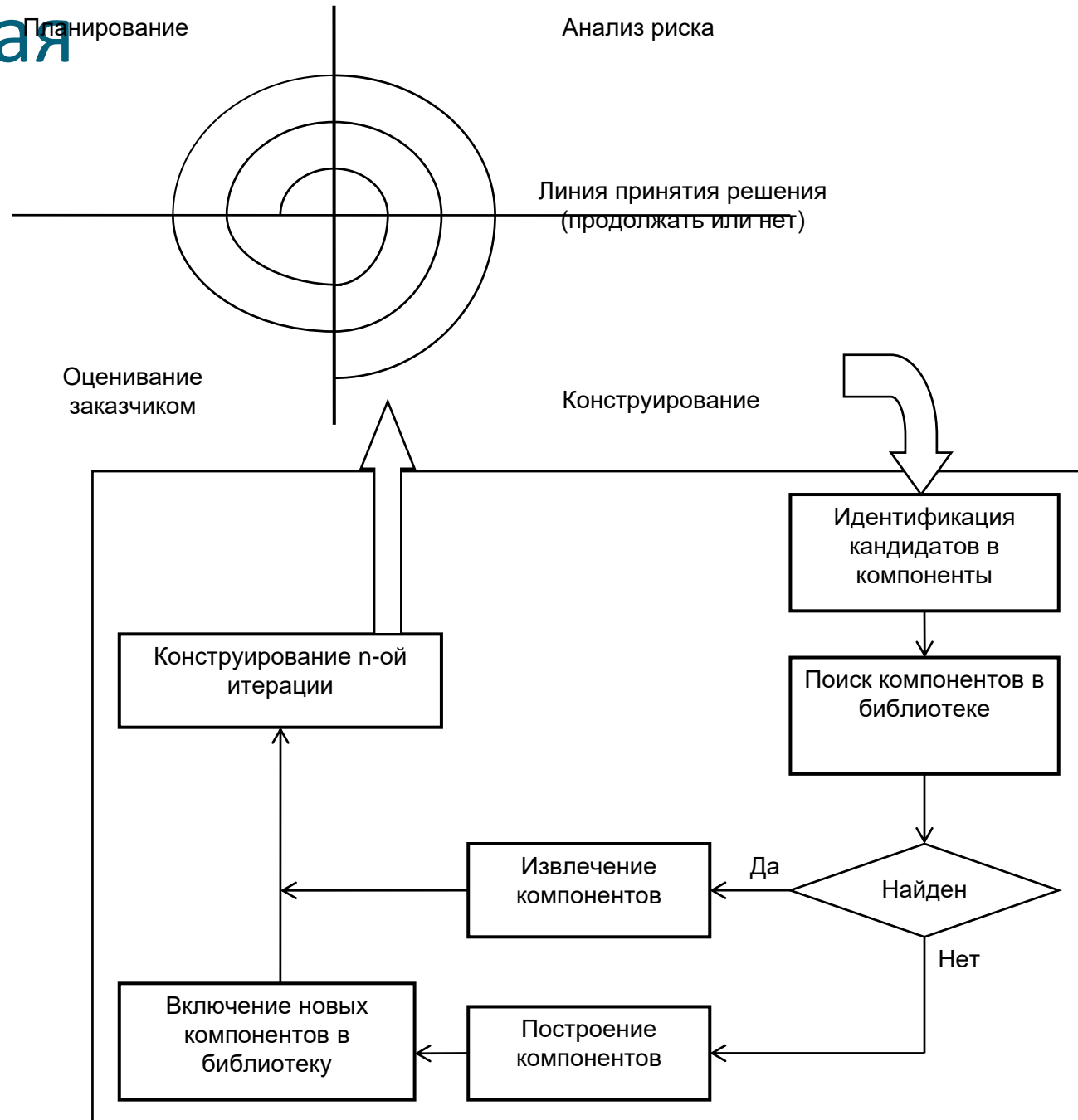
Недостатки спиральной модели

- ❑ модель имеет усложненную структуру, поэтому может быть затруднено ее применение разработчиками, менеджерами и заказчиками;
- ❑ спираль может продолжаться до бесконечности, поскольку каждая ответная реакция заказчика на созданную версию может породить новый цикл;
- ❑ могут возникнуть затруднения при определении целей и стадий, указывающих на готовность продолжать процесс разработки на следующей итерации;
- ❑ большое количество промежуточных стадий может привести к необходимости в обработке внутренней дополнительной и внешней документации;
- ❑ если проект имеет низкую степень риска или небольшие размеры, модель может оказаться дорогостоящей;
- ❑ отсутствие хорошего средства или метода прототипирования может сделать использование модели неудобным.

Критерии применения спиральной модели

- когда создание прототипа представляет собой подходящий тип разработки продукта;
- когда пользователи не уверены в своих потребностях или требования слишком сложные;
- когда речь идет о применении новой технологии и когда необходимо протестировать базовые концепции;
- при разработке новой функции или новой серии продуктов;
- для проектов, выполнение которых сопряжено со средней и высокой степенью риска;
- для организаций, которые не могут себе позволить выделить заранее все необходимые для выполнения проекта денежные средства, и когда в процессе разработки отсутствует финансовая поддержка;
- когда нет смысла браться за выполнение долгосрочного проекта из-за потенциальных изменений, которые могут произойти в экономических приоритетах;
- с целью демонстрации качества и достижения целей за короткий период времени.

Компонентная модель



Модель быстрой разработки приложений

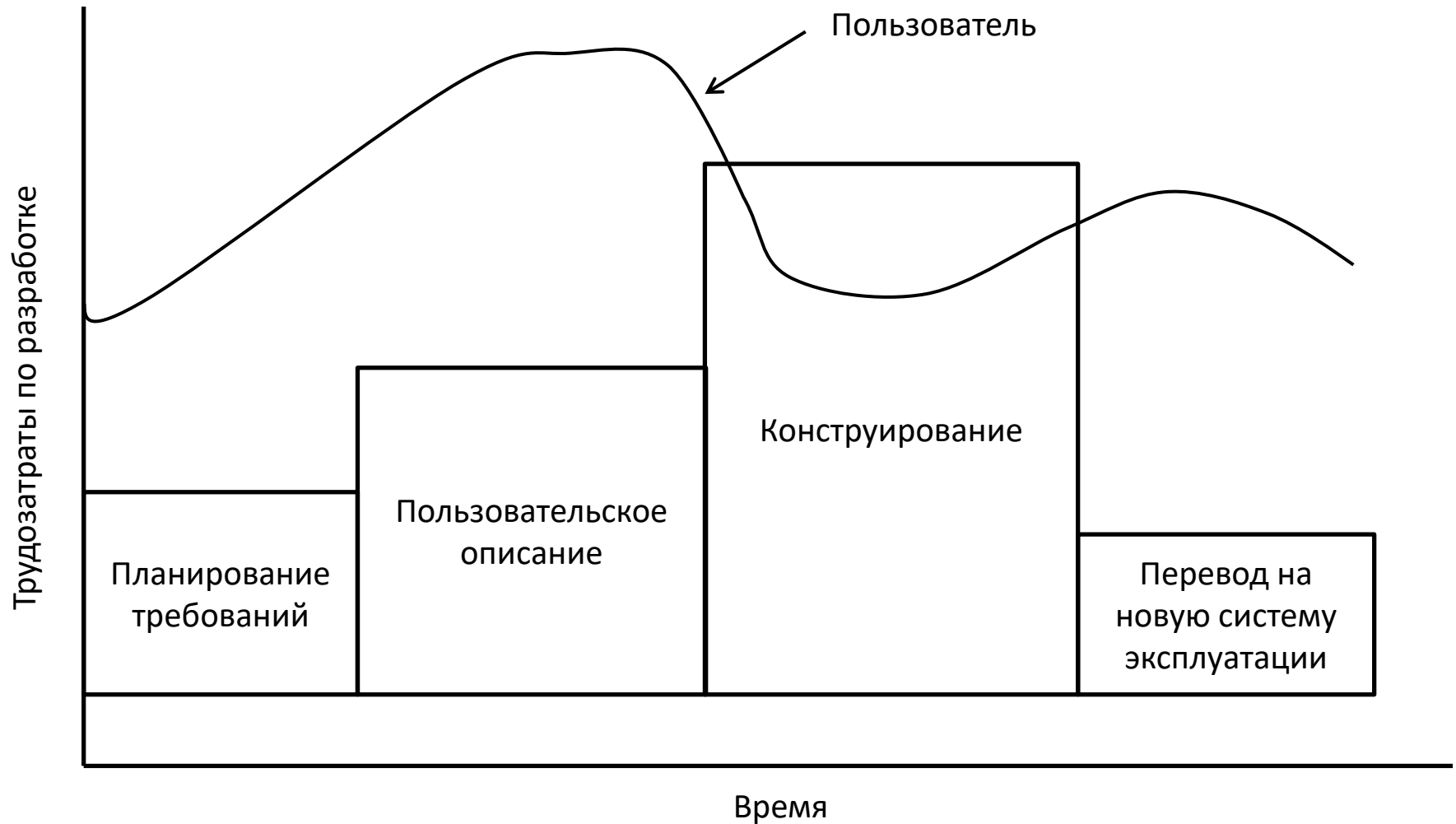
- Благодаря методу RAD (Rapid Application Development) пользователь задействован на всех фазах ЖЦ разработки проекта – не только при определении требований, но и при проектировании, разработке, тестировании, а также конечной поставке программного продукта.
- Характерной чертой RAD является короткое время перехода от определения требований до создания полной системы. Метод основывается на последовательности итераций эволюционной системы или прототипов, критический анализ которых обсуждается с заказчиком. В процессе такого анализа формируются требования к продукту.
- Разработка каждого интегрированного продукта ограничивается четко определенным периодом времени, который, как правило, составляет 60 дней и называется временным блоком.

Модель быстрой разработки приложений

Фазы RAD:

- **этап планирования требований** — сбор требований выполняется при использовании рабочего метода, называемого совместным планированием требований (Joint requirements planning, JRP), который представляет собой структурный анализ и обсуждение имеющихся коммерческих задач;
- **пользовательское описание** — совместное проектирование приложения (Joint application design, JAD) используется с целью привлечения пользователей;
- **фаза конструирования** — эта фаза объединяет в себе детализированное проектирование, построение (кодирование и тестирование), а также поставку программного продукта заказчику за определенное время;
- **перевод на новую систему эксплуатации** — эта фаза включает проведение пользователями приемочных испытаний, установку системы и обучение пользователей.

Модель быстрой разработки



Преимущества RAD

- время цикла разработки сокращается благодаря использованию мощных инструментальных средств;
- требуется меньшее количество специалистов;
- существует возможность произвести быстрый изначальный просмотр продукта;
- уменьшаются затраты;
- благодаря принципу временного блока уменьшаются затраты и риск, связанный с соблюдением графика;
- обеспечивается эффективное использование имеющихся в наличии средств и структур;
- постоянное присутствие заказчика сводит до минимума риск неудовлетворения продуктом и гарантирует соответствие системы коммерческим потребностям и надёжности программного продукта в эксплуатации;

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);
- в модели используются следующие принципы и инструментальные средства моделирования:
 - ✓ деловое моделирование (методы передачи информации, место генерирования информационных потоков, кем и куда направляется, каким образом обрабатывается);
 - ✓ моделирование данных (происходит идентификация объектов данных и атрибутов, а также взаимосвязей);
 - ✓ моделирование процесса (выполняется преобразование объектов данных);
 - ✓ генерирование приложения (методы четвертого поколения);
- повторное использование компонент уже существующих программ.

Недостатки RAD

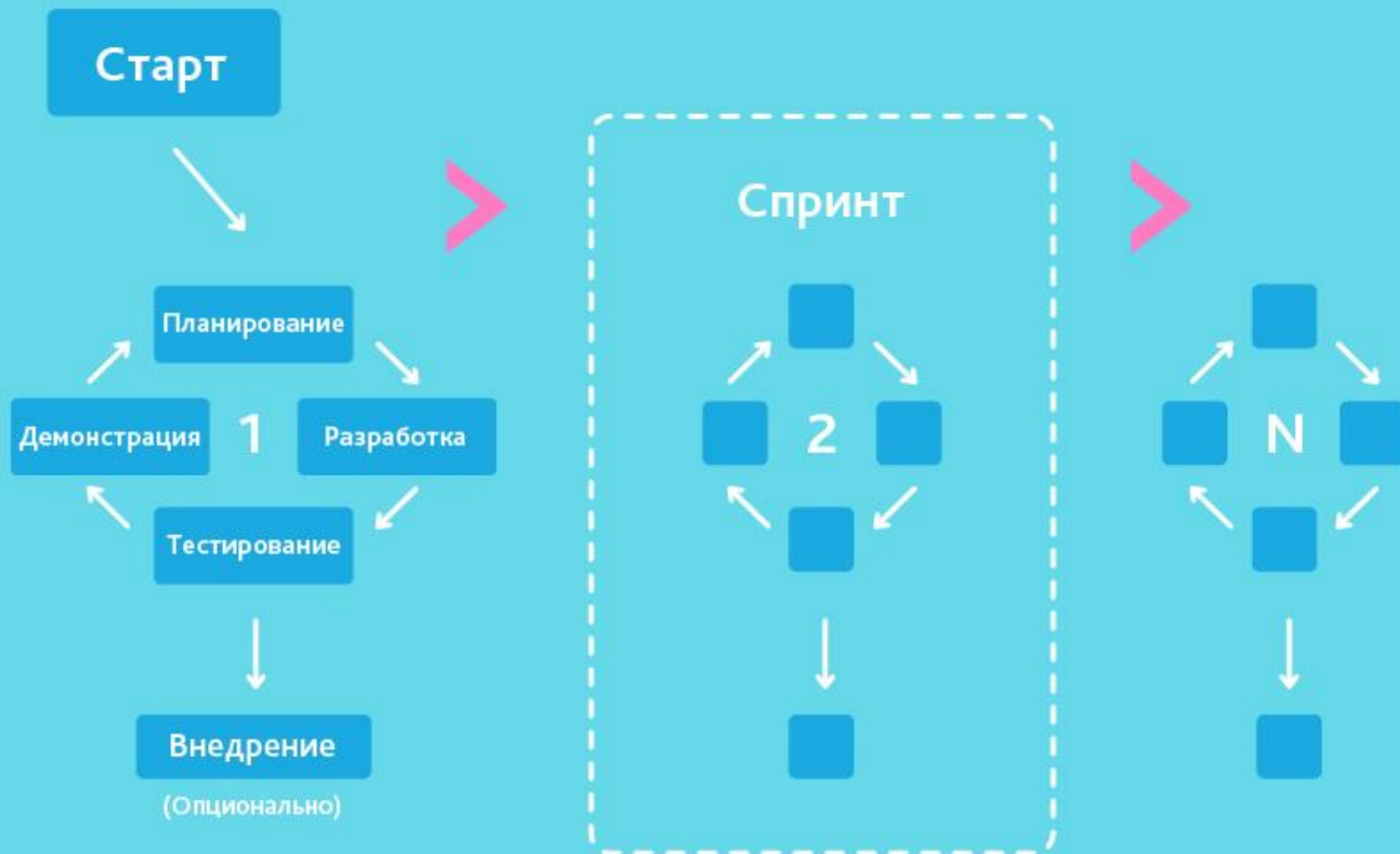
- ❑ непостоянное участие пользователя может негативно сказаться на конечном продукте;
- ❑ для реализации модели требуются разработчики и заказчики, которые готовы к быстрому выполнению действий ввиду жестких временных ограничений;
- ❑ при использовании этой модели необходимо достаточное количество высококвалифицированных разработчиков;
- ❑ использование модели может оказаться неудачным в случае отсутствия пригодных для повторного использования компонент;
- ❑ 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.

Критерии применения RAD

- ❑ в системах, которые поддаются моделированию, а также в масштабируемых системах;
- ❑ в системах, требования для которых в достаточной мере хорошо известны;
- ❑ в информационных системах;
- ❑ в случаях, когда конечный пользователь может и хочет принимать участие в процессе разработки на протяжении всего ЖЦ;
- ❑ при невысокой степени технических рисков;
- ❑ при выполнении проектов, разработка которых должна быть выполнена в сокращенные сроки (как правило, не более, чем за 60 дней);
- ❑ в системах, которые предназначены для концептуальной проверки, являются некритическими или имеют небольшой размер;
- ❑ когда затраты и соблюдение графика не являются самым важным вопросом (например при разработке внутренних инструментальных средств).

«Agile Model»

(гибкая методология разработки)



«Agile Model»

(гибкая методология разработки)

Преимущества:

- подходит для больших или нацеленных на длительный жизненный цикл проектов, постоянно адаптируемых к условиям рынка;
- лучше всего подходит для руководителей, которым свойственно генерировать, выдавать и опробовать новые идеи еженедельно или даже ежедневно;
- после каждой итерации заказчик может наблюдать результат и понимать, удовлетворяет он его или нет.

Недостатки:

- из-за отсутствия конкретных формулировок результатов сложно оценить трудозатраты и стоимость, требуемые на разработку.

Гибкая методология - когда использовать?

- Когда потребности пользователей постоянно меняются в динамическом бизнесе.
- Изменения на Agile реализуются за меньшую цену из-за частых инкрементов.
- В отличие от модели водопада, в гибкой модели для старта проекта достаточно лишь небольшого планирования.

«Spiral Model» (спиральная модель)



«Spiral Model»

(спиральная модель)

Преимущества:

- хорошо подходит для решения критически важных бизнес-задач, когда неудача несовместима с деятельностью компании;
- применяется при необходимости научных исследований и практической апробации.

Недостатки:

- не подходит для малых проектов, она резонна для сложных и дорогих.

Сравнение каскадной и гибкой моделей

