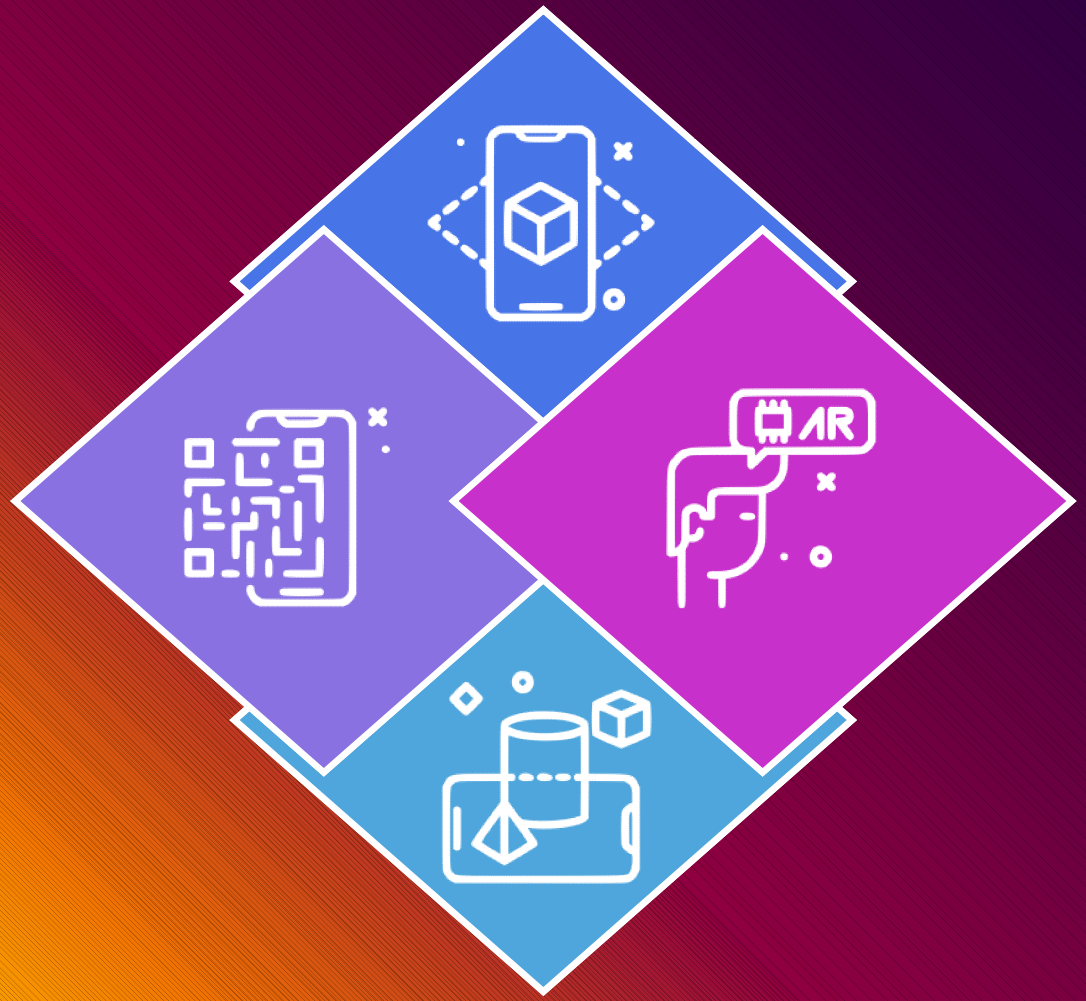
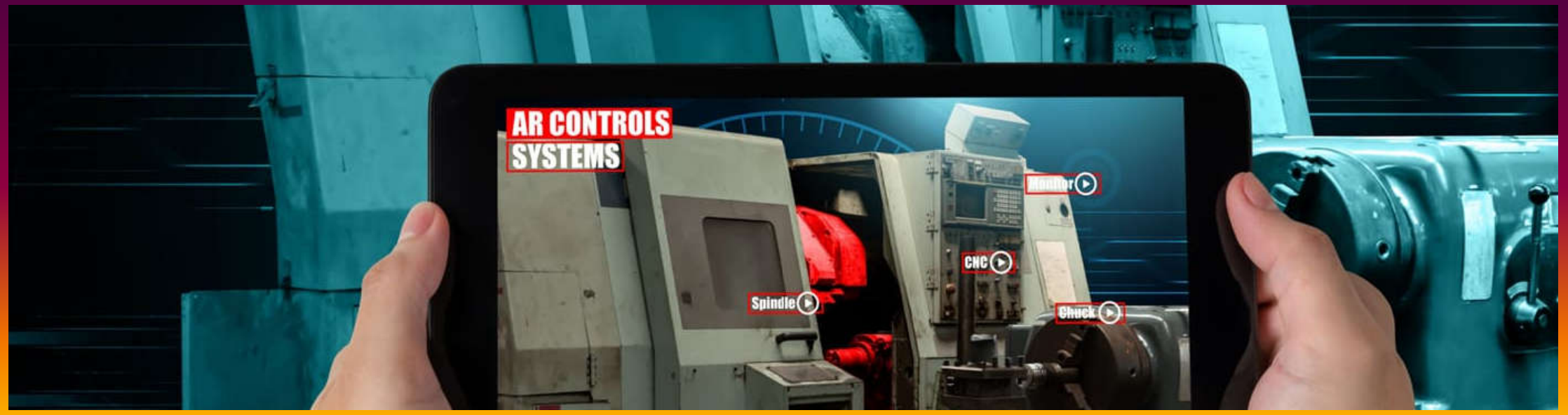
ЖИЗНЕННЫЙ ЦИКЛ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Выполнил: Григорьев Глеб 20П-3



- Понятие жизненного цикла
- Основные процессы жизненного цикла
- Модели жизненного цикла
- Дополнительные модели жизненного цикла

Жизненный цикл (software life cycle) – это совокупность процессов (software process), которая связана с последовательным изменением состояния программного обеспечения от формирования исходных требований к нему до полного изъятия его из эксплуатации.



Основные процессы жизненного цикла

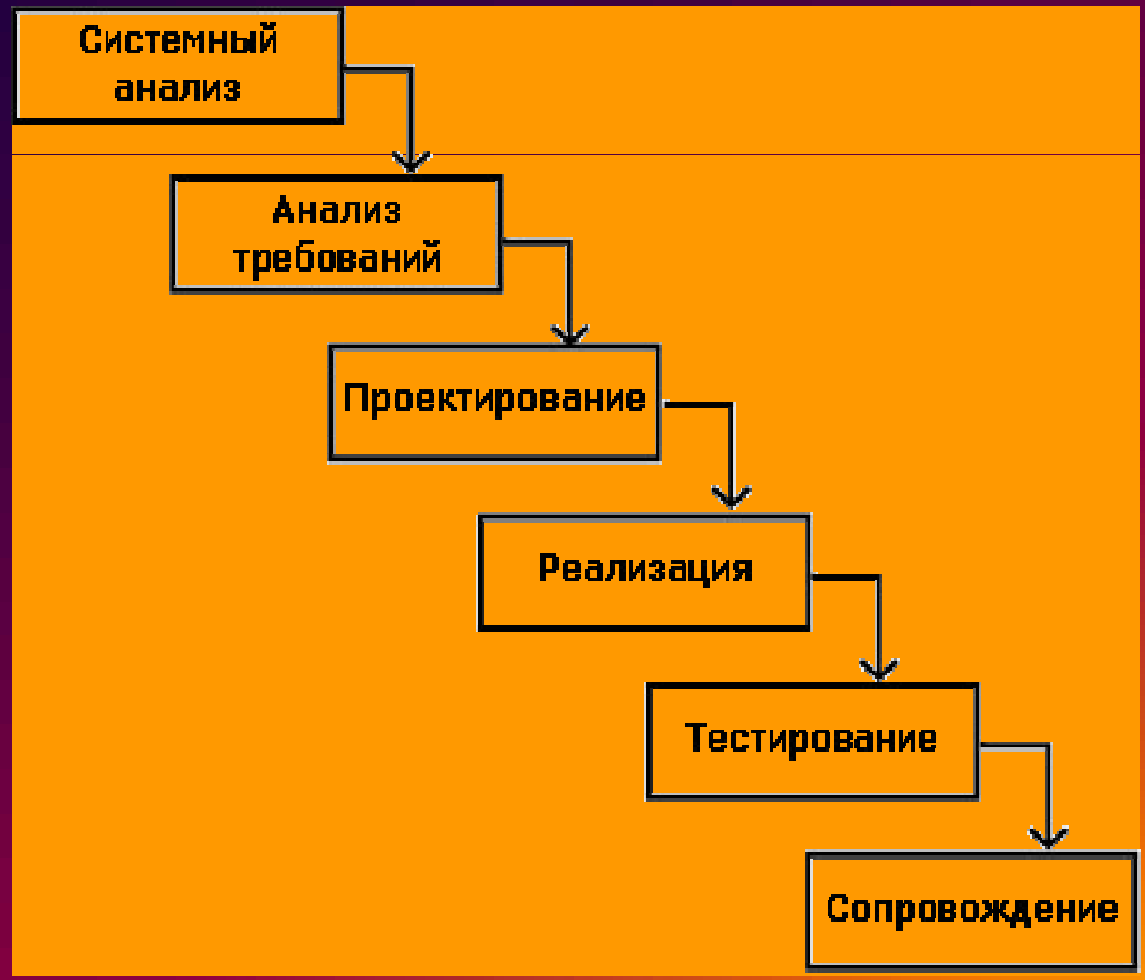
- *Процесс приобретения*. Определяет действия предприятия-покупателя, которое приобретает программный продукт.
- *Процесс поставки*. Определяет действия предприятия-поставщика, которое снабжает покупателя программным продуктом.
- *Процесс разработки*. Определяет действия предприятия-разработчика, которое разрабатывает принцип построения программного обеспечения и программный продукт.
- *Процесс эксплуатации*. Определяет действия персонала эксплуатации, который обеспечивает обслуживание вычислительной системы в процессе еѐ функционирования в интересах пользователей.
- *Процесс сопровождения*. Определяет действия персонала сопровождения, который обеспечивает инсталляцию и удаление программного продукта, его сопровождение, что представляет собой поддержку текущего состояния и функциональную пригодность, а также действия по управлению модификациями.

Модели жизненного цикла

- *Модель жизненного цикла* – формализованная упрощенная структура, которая определяет последовательность выполнения практических этапов и их взаимосвязи на протяжении жизненного цикла программного средства.
- *Этап* представляет собой логически завершенную часть жизненного цикла.

Водопадная модель жизненного цикла

- Водопадная модель *(waterfall model)* является классической моделью жизненного цикла. Эта модель предполагает, что переход к следующему этапу осуществляется только после полного завершения работ предыдущего этапа.



- *Системный анализ* определяет назначение создаваемого программного продукта, персонал, программную и аппаратную части, чтобы оценить требуемые трудозатраты, составить план проектных работ и определить риски.
- *Анализ требований* определяет функции программного обеспечения для планирования проекта.
- *Проектирование* создает архитектуру программного средства с учетом требуемого качества.
- *Реализация* состоит в переводе результатов проектирования в тексты выбранных языков программирования и баз данных с последующей реализацией при помощи соответствующих инструментальных средств.
- *Тестирование* определяет дефекты в реализации с целью их исправления, чтобы повысить качество программного продукта.
- *Сопровождение* заключается в исправлении ошибок эксплуатации, адаптации продукта к изменениям внешней среды и, возможно, его совершенствованию по требованиям заказчика или пользователей. Этот этап имеет отношение к существующей системе, но не к разработке новой.

Преимущества применения водопадной модели:

– на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;

– выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Недостатки водопадной модели:

– неприспособленность к изменениям требований к проекту;

– существенное запаздывание с получением результата;

– длительный период создания системы, который может привести к тому, что реализация проекта морально устареет одновременно с утверждением.

«Spiral Model» (спиральная модель)

- Спиральная модель *(spiral model)* жизненного цикла, которая предполагает повторение одинаковой последовательности действий более одного раза. Каждая итерация (виток спирали) завершается выпуском новой версии выполняемого программного обеспечения.

- *Планирование* заключается в определении целей и ограничений на разработку.
- *Анализ риска* состоит в распознавании и оценивании рисков. Если выявленные на этом этапе риски слишком велики, возможен отказ от создания программного обеспечения.
- *Конструирование* представляет собой разработку программного средства следующего уровня. На этом этапе используется классическая (водопадная) модель жизненного цикла, согласно которой проводится анализ требований, проектирование, реализация и тестирование.
- *Переходный период* служит для оценки текущих результатов конструирования

Спиральная модель предполагает 4 этапа для каждого витка:

Основные преимущества спиральной модели:

– накопление и повторное использование программных средств, моделей и прототипов;

– ориентация на развитие и модификацию программного обеспечения в процессе проектирования;

– анализ издержек в процессе проектирования;

– приспособленность к изменениям требований к проекту.

Главная проблема при использовании спиральной модели :

заключается в определении момента перехода на следующий этап. Для решения этой проблемы обычно вводятся временнЫе ограничения на каждый из этапов. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. План, как правило, составляется на основе личного опыта разработчиков.

«RAD Model» (rapid application development model или быстрая разработка приложений)

- RA RAD-модель — разновидность инкрементной модели. В RAD-модели компоненты или функции разрабатываются несколькими высококвалифицированными командами параллельно. Временные рамки одного цикла жестко ограничены. Созданные модули затем интегрируются в один рабочий прототип.

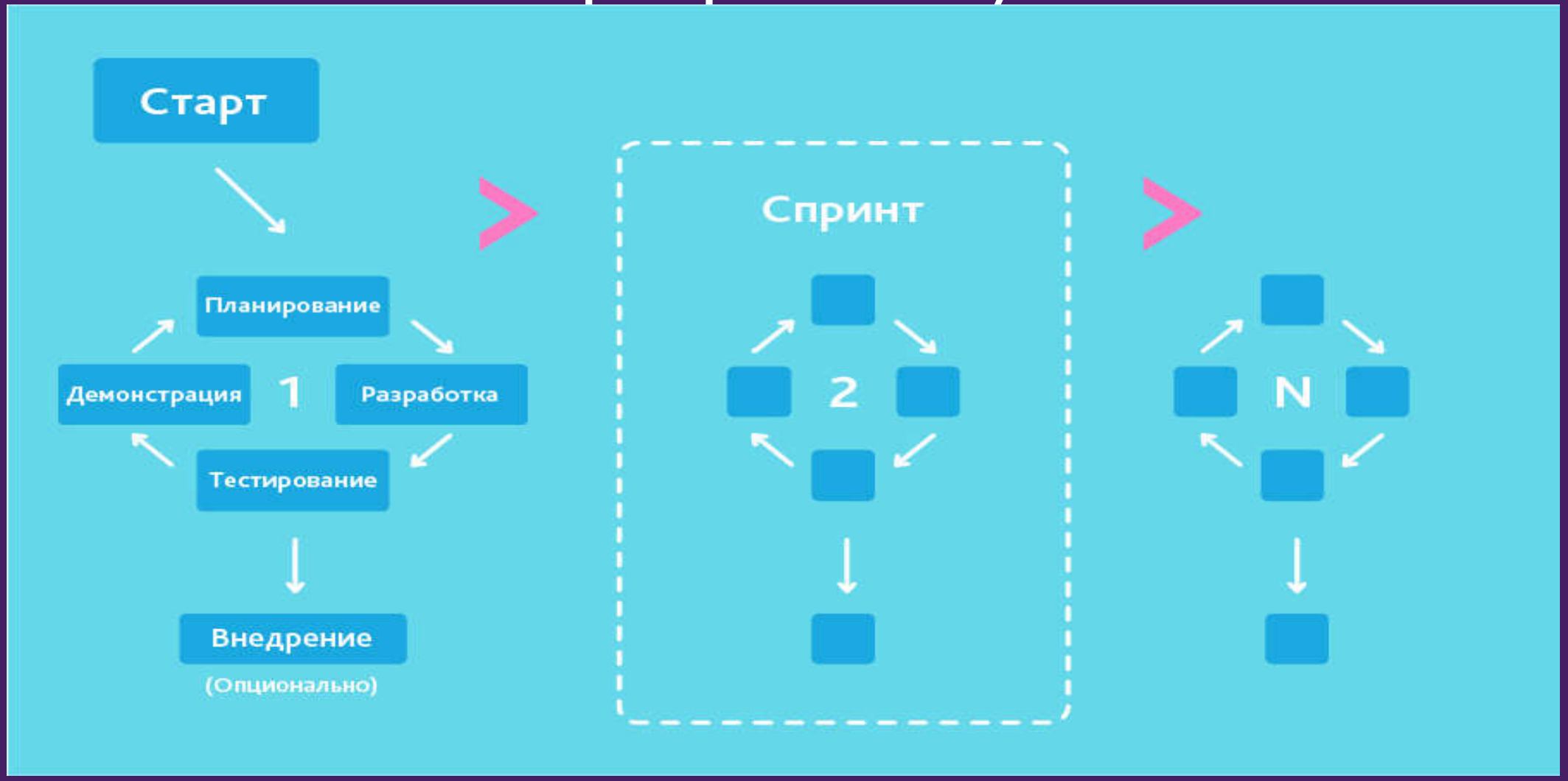
Модель быстрой разработки приложений включает следующие фазы:

- *Бизнес-моделирование*: определение списка информационных потоков между различными подразделениями.
- *Моделирование данных:* информация, собранная на предыдущем этапе, используется для определения объектов и иных сущностей, необходимых для циркуляции информации.
- *Моделирование процесса*: информационные потоки связывают объекты для достижения целей разработки.
- *Сборка приложения:* используются средства автоматической сборки для преобразования моделей системы автоматического проектирования в код.
- *Тестирование*: тестируются новые компоненты и интерфейсы.

Когда используется RAD-модель?

Может использоваться только при наличии высококвалифицированных и узкоспециализированных архитекторов. Бюджет проекта большой, чтобы оплатить этих специалистов вместе со стоимостью готовых инструментов автоматизированной сборки. RAD-модель может быть выбрана при уверенном знании целевого бизнеса и необходимости срочного производства системы в течение 2-3 месяцев.

«Agile Model» (гибкая методология разработки)



В «гибкой» методологии разработки после каждой итерации заказчик может наблюдать результат и понимать, удовлетворяет он его или нет. Это одно из преимуществ гибкой модели. К ее недостаткам относят то, что из-за отсутствия конкретных формулировок результатов сложно оценить трудозатраты и стоимость, требуемые на разработку. Экстремальное программирование (XP) является одним из наиболее известных применений гибкой модели на практике.

В основе такого типа — непродолжительные ежедневные встречи — «Scrum» и регулярно повторяющиеся собрания (раз в неделю, раз в две недели или раз в месяц), которые называются «Sprint». На ежедневных совещаниях участники команды обсуждают:

- отчёт о проделанной работе с момента последнего Scrum’а;
- список задач, которые сотрудник должен выполнить до следующего собрания;
- затруднения, возникшие в ходе работы.

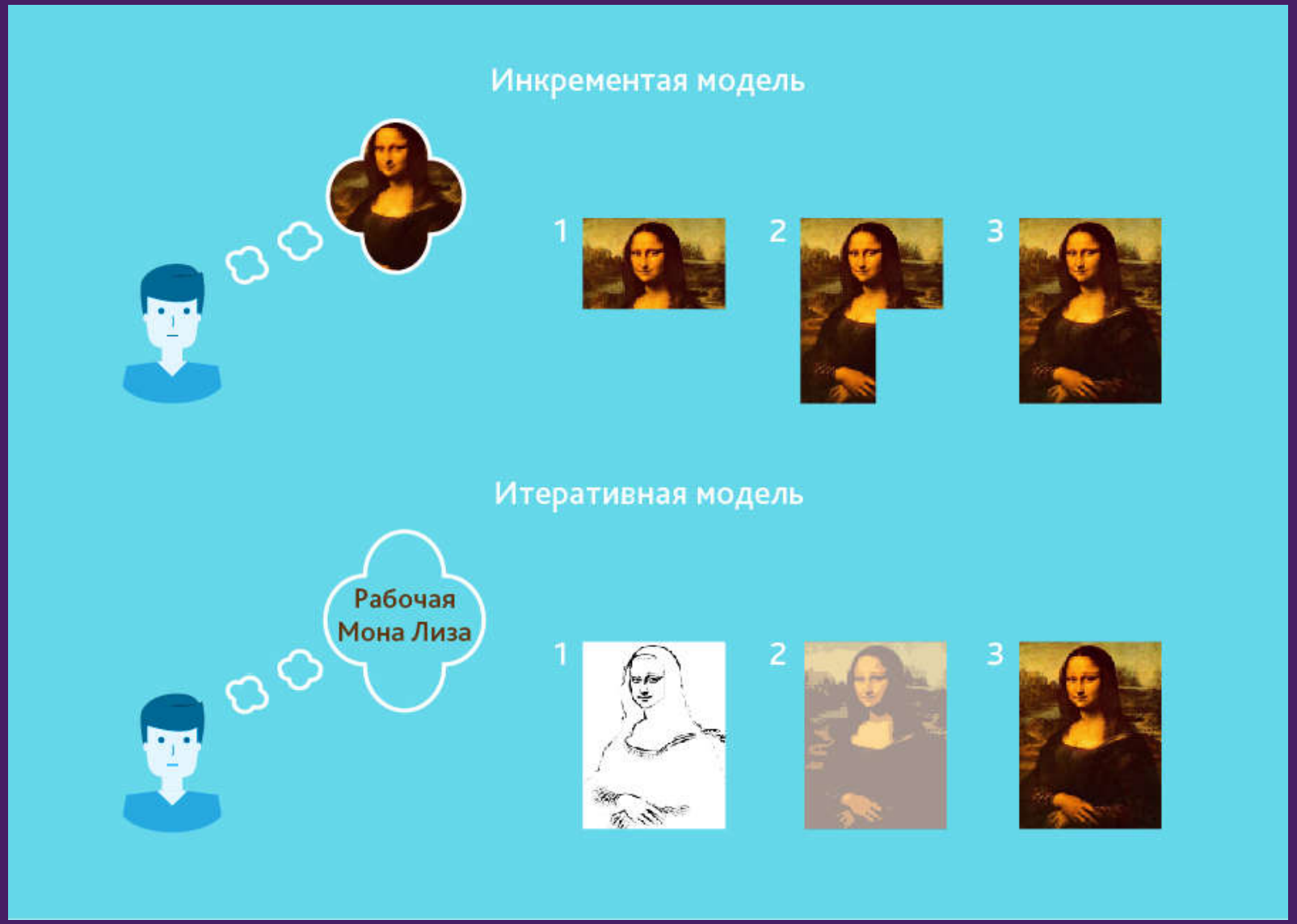
Когда использовать Agile?

- 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.

«Iterative Model» (итеративная или итерационная модель)

Итерационная модель жизненного цикла не требует для начала полной спецификации требований. Вместо этого, создание начинается с реализации части функционала, становящейся базой для определения дальнейших требований. Этот процесс повторяется. Версия может быть неидеальна, главное, чтобы она работала. Понимая конечную цель, мы стремимся к ней так, чтобы каждый шаг был результативен, а каждая версия — работоспособна.

На диаграмме показана итерационная «разработка» Мона Лизы. Как видно, в первой итерации есть лишь набросок Джоконды, во второй — появляются цвета, а третья итерация добавляет деталей, насыщенности и завершает процесс. В инкрементной же модели функционал продукта наращивается по кусочкам, продукт составляется из частей. В отличие от итерационной модели, каждый кусочек представляет собой целостный элемент.



Когда оптимально использовать итеративную модель?

- 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.