ГИБКАЯ МЕТОДОЛОГИЯ РАЗРАБОТКИ ПО

ИСТОРИЧЕСКАЯ СПРАВКА

В 1970 году Уинстон Ройс опубликовал статью под названием «Управление разработкой больших программных систем» (Winston Royce, «Managing the Development of Large Software Systems»). В ней он жестко прошелся по традиционной каскадной модели, показав, что при неитерационной разработке качество продукта получается низкое, а цена каждой ошибки начального уровня велика.

В качестве альтернативы он привел гибкую модель разработки, когда продукт разрабатывается одновременно несколькими группами программистов, а объем определяется порядком итерации. Алгоритм такой: сначала планируется, создается и тестируются базовые возможности приложения, оно получает фидбэк от клиентов или руководства, после чего устраняются недостатки, добавляется функциональность, цикл повторяется вновь и так далее до бесконечности. В идеале каждая итерация должна занимать 2–4 недели, занятость программистов на всех этапах создания продукта должна быть равной.

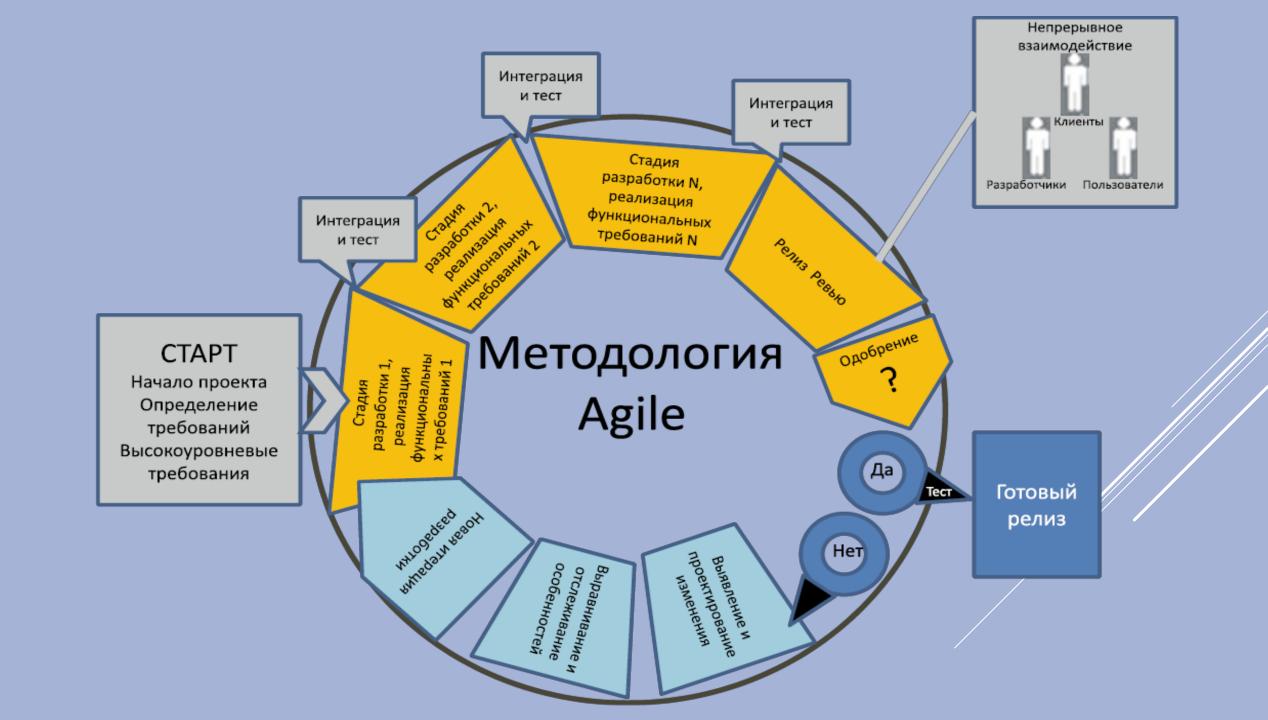


ОСНОВНЫЕ ЭТАПЫ

Важно помнить, что Agile— это общая концепция. Где-то под циклом понимается путь от анализа и планирования до выпуска, где-то это просто этап разработки, связанный с определённым уровнем сложности. Но везде есть 4 стадии:

- ▶ Планирование. Ограниченный по времени этап из-за большого числа итераций и постоянно меняющихся требований.
- ▶ Обратная связь. Она может поступать от клиентов, программистов, тестировщиков и самих разработчиков — любых заинтересованных лиц.
- Реализация. Собственно сама разработка кода и графики.
- ▶ Тестирование. Автоматическое или ручное выявление ошибок и несоответитвий требованиям заказчика.

Описанная концепция универсальна для всех гибких методологий, которые также имеют общие плюсы и минусы.



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.

ПРЕИМУЩЕСТВА

Думаю сначала стоит рассказать о плюсах:

- 1. Максимальная удовлетворённость клиента. У него есть постоянный контакт с разработчиками, его пожелания быстро находят отклик в проекте.
- 2. Благоприятная атмосфера в коллективе. Даже в случае разногласий эффективнее все решать словами.
- 3. Высокие цели. Постоянный фидбек предполагает внимание к мелочам, эргономике, соответствию трендам.
- 4. Безопасность. Множество итераций исключает существование ошибок, допущенных в самом начале.

НЕДОСТАТКИ

Но у нее есть и свои минусы:

- 1. Слабое внимание к документации. В случае возникновения претензий и конфликтных ситуаций аргументов в свою защиту у исполнителя просто не будет.
- 2. Проблемы с реализацией комплексных продуктов. Если в самом начале разработки были заложены ограничения на поздних этапах исправлять их будет сложно.
- 3. Вероятность отказа клиентов в процессе разработки. Если у заказчика нет четкого понимания, что он хочет видеть на выходе выполнять его требования очень сложно.
- 4. Влияние иерархии. Руководители принимают все решения, так как все процессы цикла реализуются параллельно. Важность рядовых программистов минимальна.

ВЫВОД

Данная методология в первую очередь подходит именно для больших или нацеленных на длительный жизненный цикл проектов, постоянно адаптируемых к условиям рынка. Соответственно, в процессе реализации требования изменяются. Стоит вспомнить класс творческих людей, которым свойственно генерировать, выдавать и опробовать новые идеи еженедельно или даже ежедневно. И именно для таких руководителей подходит гибкая разработка лучше всего.