|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Каскадная | Водоворот | Спиральная | V-модель | RUP | Scrum | Kanban |
| Фазы проекта выполняются в строгой последовательности, одна за другой. Нужно внимательно выполнять каждую стадию, так как при наличии ошибок на начальных этапах их уже невозможно будет исправить при переходе на последующие без высоких затрат ресурсов. Тестирование начинается только после того, как разработка завершена или почти завершена. По сравнению с итеративными моделями роль тестировщика несколько менее значима, так как работа тестировщика осуществляется только на стадии тестирования. Однако цена ошибки в этой модели выше, так как пропустив её, к ней уже будет не вернуться, не допустив серьёзных отклонений в графике и бюджете. Для команды эта модель хороша тем, что не предполагает каких-то изменений, т.е. можно один раз понять цели, сроки и затраты и впоследствии ориентироваться только на них. А плоха тем, что очень велика цена допущенной и сразу не замеченной ошибки.  **Преимущества:**  Большая прозрачность схемы проекта, как следствие, с ней просто и понятно разобраться и легко ей управлять. Благодаря жёсткой схеме выполнения проекта, разработка проходит быстро, стоимость и срок заранее определены и фиксированы **Недостатки:**  1. Высокая трудо- и затратоёмкость возможности пофиксить баги/исправить ошибки, возникшие на ранних этапах, на более поздних. 2. Излишняя формализация модели, т.е. план и структура довлеют над актуальностью, рациональностью и трудозатратами 3. Невозможность серьёзных корректировок и актуализации в ходе проекта, так как для  инициализации приходится ждать завершения всего проекта.  Подходит для ситуаций, когда требования известны, понятны и зафиксированы. Противоречивых требований не имеется. Государственные заказы (армия – военная техника), медицинская сфера | Практически идентична каскадной, за исключением  того **преимущества**, что имеет обратные связи с каждым этапом жизненного цикла, при этом порождает очень весомый **недостаток**: 10-ти кратное увеличение затрат на разработку. | Представляет собой процесс разработки программного обеспечения, сочетающий в себе как проектирование, так и постадийное прототипирование с целью сочетания преимуществ восходящей и нисходящей концепции.  **Преимущества**: Быстрое получение результата Повышение конкурентоспособности Изменяющиеся требования — не проблема **Недостатки**: Отсутствие регламентации стадий | Улучшенная версия классической каскадной модели. Здесь на каждом этапе происходит контроль текущего процесса, для того чтобы убедится в возможности перехода на следующий уровень. В этой модели тестирование начинается еще со стадии написания требований, причем для каждого последующего этапа предусмотрен свой уровень тестового покрытия. Для каждого уровня тестирования разрабатывается отдельный тест-план, то есть во время тестирования текущего уровня, мы также занимаемся разработкой стратегии тестирования следующего. Создавая тест-планы, мы также определяем ожидаемые результаты тестирования и указываем критерии входа и выхода для каждого этапа. В V-модели каждому этапу проектирования и разработки системы соответствует отдельный уровень тестирования. Здесь процесс разработки представлен нисходящей последовательностью в левой части условной буквы V, а стадии тестирования – на ее правом ребре. Соответствие этапов разработки и тестирования показано горизонтальными линиями. **Преимущества** V-модели: строгая этапизация; планирование тестирования и верификация системы производятся на ранних этапах; улучшенный, по сравнению с каскадной моделью, тайм-менеджмент; промежуточное тестирование. **Недостатки:** — недостаточная гибкость модели; — собственно создание программы происходит на этапе написания кода, то есть уже в середине процесса разработки; — недостаточный анализ рисков; — нет работы с параллельными событиями и возможности динамического внесения изменений. Когда использовать V-модель: – В проектах, в которых существуют временные и финансовые ограничения; – Для задач, которые предполагают более широкое, по сравнению с каскадной моделью, тестовое покрытие. | Раннее обнаружение и постоянное устранение рисков. Ожидание изменений в деталях проекта (адаптивность). Обеспечение качества на всех уровнях развития проекта. Отказ от формальностей в пользу работающего продукта Оптимальная численность – около 9 человек (с тестерами и разработчиками)  Возрастает от начала итерации к концу. На начальной стадии роль ниже, увеличивается на стадии уточнения, достигает пика на конструировании и постепенно уменьшается на стадии внедрения Данная модель сложнее для команды тем, что требует постоянных корректировок и завязана на том, чтобы выполнять работу в сжатые сроки. Однако вследствие небольших сроков и объёмов работ команде гораздо проще определить свою скорость работы и внести корректировки.  **Преимущества:**  организация эффективной обратной связи проектной команды с потребителем (заказчиками) более равномерная загрузка участников проекта Более раннее, чем у каскадной модели, обнаружение ошибок Концентрация на наиболее важных и критических направлениях проекта Большая мобильность и адаптивность к изменяющимся условиям – т.е. гибкость Более прогнозируемый результат, как для команды, так и для заказчика **Недостатки:** Сложнее влиться в проект с «нуля», так как времени на изучение материала «раскачку» очень мало  Отсутствие стабильного объёма и характера работы, так как в ходе daily meeting возможно резкое изменение этих параметров из-за установки новых приоритетов  Подходит для ситуаций, когда потребности пользователей постоянно меняются в динамическом бизнесе.  Динамичные и высокотехнологичные сферы бизнеса, искусственный интеллект, нанотехнологии, промышленная инженерия | Во главе угла стоит итеративность  Данная модель «завязана» на общении: ежедневные митинги, на которых обсуждаются текущие проблемы и пути их решения  Scrum-подход делит рабочий процесс на равные спринты –(от недели до месяца). Перед спринтом формулируются задачи на данный спринт, в конце – обсуждаются  результаты, а команда начинает новый спринт. Спринты очень удобно сравнивать между собой, что позволяет управлять эффективностью работы.  Отказ от формальностей в пользу работающего продукта  Оптимальная численность – около 9 человек (с тестерами и разработчиками)  Поскольку численность команды не столь велика, роль тестировщика высока. Он должен быть готов как к высокому темпу работы, так и к тому, что некоторые операции придётся проделывать неоднократно.  Данная модель сложнее для команды тем, что требует постоянных корректировок и завязана на том, чтобы выполнять работу в сжатые сроки. Однако вследствие небольших сроков и объёмов работ команде гораздо проще определить свою скорость работы и внести корректировки.  **Преимущества:** организация эффективной обратной связи проектной команды с потребителем (заказчиками)  более равномерная загрузка участников проекта  Более раннее, чем у каскадной модели, обнаружение ошибок  Концентрация на наиболее важных и критических направлениях проекта  Большая мобильность и адаптивность к изменяющимся условиям – т.е. гибкость  Можно адекватно сравнить участки работы между собой на предмет эффективности  Более прогнозируемый результат, как для команды, так и для заказчика  **Недостатки:**  Сложнее влиться в проект с «нуля», так как времени на изучение материала «раскачку» очень мало  Отсутствие стабильного объёма и характера работы, так как в ходе daily meeting возможно резкое изменение этих параметров из-за установки новых приоритетов  Подходит для ситуаций, когда потребности пользователей постоянно меняются в динамическом бизнесе.  Динамичные и высокотехнологичные сферы бизнеса, искусственный интеллект, нанотехнологии, промышленная инженерия | Имеет много общего со scrum моделью. Основные отличия scrum от kanban:  Методика scrum требует фиксированных ролей, тогда как у kanban нет необходимых ролей. Scrum основана на итерациях, объединяющих планирование, оптимизацию процессов и выпуск. В kanban это можно делать регулярно или каждый раз, когда вам нужно. Команда scrum требует оценки своей работы, тогда как команде kanban это не нужно.  Согласно методологии Kanban для каждого этапа разработки еще на стадии проектирования выбирается максимальное число задач, которые могут выполняться одновременно. Четкого руководства по выбору того или иного числа задач для определенного этапа нет и оптимальное значение чаще всего определяется по завершении нескольких итераций.  Scrum, в свою очередь, не ограничивает вас количеством задач, которые могут выполняться на определенном этапе. Можно говорить только об определенном количестве задач, запланированном для спринта в целом.  **Преимущества** у kanban-модели те же, что у и scrum плюс перечисленные особенности. **Недостатки** – те же, что у и scrum модели, а также ограничение на количество задач, которые могут выполняться на определённом этапе. |

Примеры применения:

<https://vc.ru/20942-agile-victims>

<https://dev.by/lenta/main/kogda-agile-ne-rabotaet-neskolko-primerov>

(применение Agile в Кремниевой долине, Google, Uber, Facebook, мобильное приложение для интернет-банкинга от «Альфа-Банка» РФ)

<https://dou.ua/forums/topic/14015/>

(применение методологии Kanban на заводах компании «Тойота»)

<https://habr.com/company/edison/blog/269789/>

(применение Agile-методики на сайте <https://www.kvzrm.ru>, также применение спиральной модели на проекте по разработке СЭД для ОДУ Сибири СО ЕЭС )

<https://xbsoftware.ru/blog/waterfall-scrum-kanban-primer-projekta/>

(пример применения Waterfall методологии при разработке Excel add-in для приложения.)