Задание 1. Сравнение ЖЦ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Характеристика | Жизненный цикл | Недостатки и достоинства | Для каких проектов |
| 1 | 2 | 3 | 4 | 5 |
| 1.Каскадная | Однократный проход этапов  разработки. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству или системе в начале процесса разработки. Каждый этап разработки начинается после завершения предыдущего этапа. Возврат к уже выполненным этапам не предусматривается. |  | **Достоинства** 1) стабильность требований в течение ЖЦ разработки;  2)необходимость только одного прохода этапов разработки, что обеспечивает простоту применения стратегии;  3) простота планирования, контроля и управления проектом;  4) доступность для понимания заказчиками.  **Недостатки** 1) сложность полного формулирования требований в начале процесса  разработки и невозможность их динамического изменения на протяжении ЖЦ  2) разрабатываемые ПС или системы обычно слишком велики и сложны, чтобы все работы по их созданию выполнять однократно в результате возврат к предыдущим шагам для решения возникающих проблем приводит к увеличению финансовых затрат и нарушению графика работ  3) непригодность промежуточных продуктов для использования  4) недостаточное участие пользователя в процессе разработки ПС | 1) при разработке проектов с четкими, неизменяемыми в течение ЖЦ  требованиями и понятной реализацией;  2) при разработке проектов невысокой сложности  3) при выполнении больших проектов в качестве составной части моде-  лей ЖЦ |
| 2.**Инкрементная** | Многократный проход этапов разработки с запланированным улучшением результата.  Данная стратегия основана на полном определении всех требований к  разрабатываемому программному средству (системе) в начале процесса разработки. Однако полный набор требований реализуется постепенно в соответствии с планом в последовательных циклах разработки. |  | **Достоинства** 1) возможность получения функционального продукта после реализации  каждого инкремента;  2) короткая продолжительность создания инкремента; это приводит к сокращению сроков начальной поставки, позволяет снизить затраты на поставки программного продукта;  3)стабильность требований во время создания определенного инкремента; возможность учета изменившихся требований;  4) снижение рисков по сравнению с каскадной стратегией; **Недостатки** 1) необходимость полного функционального определения системы или программного средства в начале ЖЦ для обеспечения планирования инкрементов и управления проектом;  2) возможность текущего изменения требований к системе или программному средству, которые уже реализованы в предыдущих инкрементах;  3) сложность планирования и распределения работ;  4) проявление человеческого фактора, связанного с тенденцией к оттягиванию решения трудных проблем на поздние инкременты, что может нарушить график работ или снизить качество программного продукта. | 1) при разработке проектов, в которых большинство требований можно  сформулировать заранее, но часть из них могут быть уточнены через определенный период времени;  2) при разработке сложных проектов с заранее сформулированными требованиями; для них разработка системы или программного средства за один  цикл связана с большими трудностями;  3) при необходимости быстро поставить на рынок продукт, имеющий базовые функциональные свойства;  4) при разработке проектов с низкой или средней степенью рисков;  5) при выполнении проекта с применением новых технологий. |
| 3.V-образная модель. | Модель представляет собой разновидность каскадной модели.  Данная модель поддерживает каскадную стратегию однократного выполнения  этапов процесса разработки ПС или систем и базируется на предварительном  полном формировании требований. В классической V-образной модели каждый шаг начинается после завершения предыдущего шага.  Отличием V образной модели от каскадной является то, что в ней выделены связи между шагами, предшествующими программированию, и соответствующими видами тестирования и испытаний. |  | **Достоинства** 1) планирование тестирования и испытаний на ранних стадиях разработ-  ки системы и программного средства;  2) упрощение аттестации и верификации промежуточных результатов  разработки;  3) упрощение управления и контроля хода процесса разработки. **Недостатки** 1) поздние сроки тестирования требований в жизненном цикле, что оказывает существенное влияние на график выполнения проекта при необходимости изменения требований;  2) отсутствие, как и в остальных каскадных моделях, действий, направленных на анализ рисков. | 1)Если требуется тщательное тестирование продукта, то V-модель оправдает заложенную в себя идею: validation and verification. 2)Для малых и средних проектов, где требования четко определены и фиксированы. 3)В условиях доступности инженеров необходимой квалификации, особенно тестировщиков. |
| 4.Эволюционная | представляет собой многократный проход этапов разработки. Данная стратегия основана на частичном определении требований к разрабатываемому программному средству или системе в начале процесса разработки. Требования постепенно уточняются в последовательных  циклах разработки. |  | **Достоинства** 1) возможность уточнения и внесения новых требований в процессе раз-  работки;  2) пригодность промежуточного продукта для использования;  3) возможность управления рисками;  4) обеспечение широкого участия пользователя в проекте, начиная с ранних этапов, что минимизирует возможность разногласий между заказчиками и  разработчиками  5) реализация преимуществ каскадной и инкрементной стратегий. **Недостатки** 1) неизвестность точного количества необходимых итераций и сложность определения критериев для продолжения процесса разработки на следующей итерации; это может вызвать задержку реализации конечной версии системы.  2) сложность планирования и управления проектом;  3) необходимость активного участия пользователей в проекте, что реально не всегда осуществимо;  4) необходимость в мощных инструментальных средствах и методах  прототипирования;  5) возможность отодвигания решения трудных проблем на последующие  циклы, что может привести к несоответствию полученных продуктов требованиям заказчиков. | 1) при разработке проектов, для которых требования слишком сложны,  неизвестны заранее, непостоянны или требуют уточнения;  2) при разработке сложных проектов, в том числе:  1.больших долгосрочных проектов;  2.проектов по созданию новых, не имеющих аналогов ПС или систем;  3.проектов со средней и высокой степенью рисков;  4.проектов, для которых нужна проверка концепции, демонстрация  технической осуществимости или промежуточных продуктов;  3) при разработке проектов, использующих новые технологии. |
| 5.RAD | концепция организации технологического процесса разработки программных продуктов, ориентированная на максимально быстрое получение качественного результата в условиях сильных ограничений по срокам и бюджету и нечётко определённых требований к продукту. |  | **Достоинства:**  1.Быстрое развитие продукта.  2.Разработка многоразовых мелких компонентов.  3.Повторный обзор в процессе разработки.  4.Интеграция повторно используемых компонентов на начальном уровне, следовательно, экономит усилия, несмотря на то, что не добавляются более крупные модули.  5.Конструктивная реакция.  **Недостатки:**  1.Требуется много усилий для сбора всех требований на начальном этапе.  2.Навыки моделирования имеют много зависимостей.  3.Не подходит для малобюджетного проекта. | RAD предполагает, что разработка ПО осуществляется небольшой командой разработчиков за срок порядка трёх-четырёх месяцев путём использования инкрементного прототипирования с применением инструментальных средств визуального моделирования и разработки. |
| 6.Быстрого прототипирования | Модель быстрого протитипирования предназначена для быстрого создания прототипов продукта с целью уточнения требований и поэтапного развития прототипов в конечный продукт. Скорость (высокая производительность) выполнения проекта обеспечивается планированием разработки прототипов и участием заказчика в процессе разработки. |  | **Достоинство:**  1.конечный пользователь может "увидеть" системные требования в процессе их сбора командой разработчиков; таким образом, взаимодействие заказчика с сис­темой начинается на раннем этапе разработки;  2.исходя из реакции заказчиков на демонстрации разрабатываемого продукта, разра­ботчики получают сведения об одном или нескольких аспектах поведения системы, благодаря чему сводится к минимуму количество неточностей в требованиях;  3.снижается возможность возникновения путаницы, искажения информации или недоразумений при определении системных требований, что несомненно приво­дит к созданию более качественного конечного продукта;  4.в процесс разработки можно внести новые или неожиданные требования пользо­вателя, что порой необходимо, так как реальность может отличаться от концепту­альной модели реальности;  5.модель представляет собой формальную спецификацию, воплощенную в рабо­чую модель;  6.модель позволяет выполнять гибкое проектирование и разработку, включая не­сколько итераций на всех фазах жизненного цикла;  7.при использовании модели образуются постоянные, видимые признаки прогресса в выполнении проекта, благодаря чему заказчики чувствуют себя уверенно;  возможность возникновения разногласий при общении заказчиков с разработчи­ками минимизирована;  8.ожидаемое качество продукта определяется при активном участии пользователя в процесс на ранних фазах разработки;  9.возможность наблюдать ту или иную функцию в действии пробуждает очевидную необходимость в разработке функциональных дополнительных возможностей;  10.благодаря меньшему объему доработок уменьшаются затраты на разработку;  11.благодаря тому что проблема выявляется до привлечения дополнительных ресур­сов сокращаются общие затраты;  обеспечивается управление рисками;  12.документация сконцентрирована на конечном продукте, а не на его разработке;  13.принимая участие в процессе разработки на протяжении всего жизненного цикла, пользователи в большей степени будут довольны полученными результатами.  **Недостатки:**  **1.**модель может быть отклонена из-за создавшейся среди консерваторов репутации о ней как о "разработанном на скорую руку" методе;  2.разработанные "на скорую руку" прототипы, в отличие от эволюционных уско­ренных прототипов, страдают от неадекватной или недостающей документации;  3.если цели прототипирования не согласованы заранее, процесс может превратить­ся в упражнение по созданию хакерского кода;  4.с учетом создания рабочего прототипа, качеству всего ПО или долгосрочной эксплуатационной надежности может быть уделено недостаточно внимания.  5.иногда в результате использования модели получают систему с низкой рабочей харак­теристикой, особенно если в процессе ее выполнения пропускается этап подгонки;  6.при использовании модели решение трудных проблем может отодвигаться на бу­дущее. В результате это приводит к тому, что последующие полученные продукты могут не оправдать надежды, которые возлагались на прототип;  7.если пользователи не могут участвовать в проекте на итерационной фазе быст­рого прототипирования жизненного цикла, на конечном продукте могут отра­зиться неблагоприятные воздействия, включая проблемы, связанные с его каче­ственной характеристикой;  8.на итерационном этапе прототипирования быстрый прототип представляет со­бой частичную систему. Если выполнение проекта завершается досрочно, у ко­нечного пользователя останется только лишь частичная система;  9.несовпадение представлений заказчика и разработчиков об использовании прото­типа может привести к созданию другого пользовательского интерфейса;  10.заказчик может предпочесть получить прототип, вместо того, чтобы ждать появ­ления полной, хорошо продуманной версии;  11.если язык или среда прототипирования не сочетаются с производственным языком или окружением, всесторонняя реализация продукционной системы может быть задержана;  12.прототипирование вызывает зависимость и может продолжаться слишком долго. Нетренированные разработчики могут попасть в так называемый цикл "кодирование — устранение ошибок" (code-and-fix cycle), что приводит к дорого­стоящим незапланированным итерациям прототипирования;  13.разработчики и пользователи не всегда понимают, что когда прототип превра­щается в конечный продукт, все еще существует необходимость в традиционной Документации. Если она отсутствует, модифицировать модель на более поздних этапах может оказаться более дорогостоящим занятием, чем просто не восполь­зоваться созданным прототипом;  14.когда заказчики, удовлетворенные прототипом, требуют его немедленной поставки, перед менеджером программного проекта возникает соблазн пойти им навстречу;  15.на заказчиков могут неблагоприятно повлиять сведения об отличии между прото­типом и полностью разработанной системой, готовой к реализации;  16.на заказчиков может оказать негативное влияние тот факт, что они не располагают информацией о точном количестве итераций, которые будут необходимы;  17.на разработку системы может быть потрачено слишком много времени, так как итерационный процесс демонстрации прототипа и его пересмотр могутпродолжаться бесконечно без надлежащего управления процессом. У пользователей может воз­никнуть стремление пополнять список элементов, предназначенных для прототи­пирования, до тех пор, пока проект ни достигнет масштаба, значительно превы­шающего рамки, определенные анализом осуществимости проектного решения;  18.при выборе инструментальных средств прототипирования (операционные систе­мы, языки и малопродуктивные алгоритмы) разработчики могут остановить свой выбор на менее подходящем решении, только чтобы продемонстрировать свои способности;  19.структурные методы не используются, чтобы не помешать выполнению анализа. При прототипировании необходимо провести "реальный" анализ требований, осуществить проектирование и обратить внимание на качество с целью создания программы, до­пускающей сопровождение, точно так же, как и в любой другой модели жизненного цикла (хотя на эти действия может понадобиться меньше времени и ресурсов). | Модель быстрого протитипирования предназначена для быстрого создания прототипов продукта с целью уточнения требований и поэтапного развития прототипов в конечный продукт. Скорость (высокая производительность) выполнения проекта обеспечивается планированием разработки прототипов и участием заказчика в процессе разработки. |

Задание 2. Выбор ЖЦ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипированния | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Нет | Нет | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Нет | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы(программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |
|  |  | Да:0 Нет:1 | Да:0 Нет:1 | Да:2 Нет:2 | Да:2 Нет:3 | Да:4 Нет:2 | Да:4 Нет:2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипированния | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5. | Важна ли лёгкость распределения человеческих ресурсов? | Да | Да | Да | Да | Нет | Нет |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки | Да | Да | Нет | Да | Да | Да |
|  |  | Да:2 Нет:1 | Да:3 Нет:1 | Да:1 Нет:2 | Да:3 Нет:1 | Да:2 Нет:1 | Да:1 Нет:1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипированния | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта(системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Да |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |
|  |  | Да:1 Нет:2 | Да1: Нет:2 | Да:0 Нет:1 | Да:1 Нет:1 | Да:1 Нет:0 | Да:2 Нет:0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипированния | Эволюционная |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3. | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4. | Ожидается ли длительная эксплуатация проекта? | Да | Да | Нет | Да | Нет | Да |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 5. | Необходим ли высокий уровень надёжности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6. | Предполагается ли эволюция проекта в течении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8. | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9. | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы(время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |
|  |  | Да:1 Нет:1 | Да:2 Нет:1 | Да:1 Нет:0 | Да:7 Нет:0 | Да:6 Нет:1 | Да:8 Нет:1 |

Итог:

Каскадная: 4

V-образная: 6

RAD: 4

Инкрементная: 13

Быстрого прототипирования: 13

Эволюционная: 15

Исходя из вычислений для разработки данного ПО наиболее подходящей моделью ЖЦ является «Эволюционная» модель.