**Лабораторная работа №3.**

**Тема**: Выбор модели, метода и подхода разработки программы.

**Цель работы:**

1. Изучить основные стратегии разработки и модели жизненного цикла.

2. Научиться осуществлять выбор стратегии и модели жизненного цикла для разработки

конкретного проекта и обосновывать свой выбор.Процедура выбора модели жизненного цикла программных средств и систем

Для выбора подходящей к условиям конкретного проекта модели ЖЦ ПС и систем Институтом качества программного обеспечения SQI рекомендуется использовать специальную процедуру. Данная процедура базируется на применении четырех таблиц вопросов.

Каждая из табл. 3.1 – 3.4 представляет одну из категорий классификации проектов. Каждый из вопросов (строка в таблице) предназначен для классификации анализируемого проекта по определенному критерию категории. Столбцы данных таблиц соответствуют обобщенным моделям ЖЦ, фактически представляющим стратегии разработки ПС. При этом под RAD-моделью подразумевается независимая RAD-модель, не встроенная в другие модели жизненного цикла.

***Изучение моделей ЖЦ.***

1. **Каскадная модель (водопад) и каскадная модель с возвратом.**

Преимущества: Высокая прозрачность разработки и фаз проекта

* Чёткая последовательность
* Стабильность требований
* Строгий контроль менеджмента проекта
* Облегчает работу по составлению плана проекта и сбора команды проекта
* Хорошо определяет процедуру по контроля качества

Недостатки:

* Очень не гибкая методологии
* Может создать ошибочное впечатление о работе над проектом (например фраза «45% выполнено» не несёт за собой никакой [полезной информации](https://konkurs-uspeh.ru/chto-proveryaet-git-v-godu-chto-proveryaet-inspekciya-po-trudu-i/), а является всего лишь инструментов для менеджера проекта)
* У Заказчика нет возможности ознакомиться с системой заранее и даже с «Пилотом» системы
* У Пользователя нет возможности привыкать к продукту постепенно
* Все требования должны быть известны в начале жизненного цикла проекта

1. **Спиральная модель.**

Преимущества:

* наличие действий по анализу рисков, что обеспечивает их сокращение и заблаговременное определение непреодолимых рисков;
* обеспечение разбиения большого потенциального объема работ по выполнению проекта на небольшие части;
* первоочередность реализации решающих функций с высокой степенью риска, что позволяет при необходимости остановить работы над проектом на ранних циклах модели и уменьшить расходы;
* возможность гибкого проектирования, основанная на преимуществах каскадной модели при одновременном разрешении итераций;
* реализация преимуществ инкрементной модели (выпуск инкрементов, сокращение графика работ, неизменяемость ресурсов при постепенном росте системы);

Недостатки:

* высокая стоимость модели за счет стоимости и дополнительных временных затрат на планирование, определение целей, выполнение анализа рисков и прототипирование при прохождении каждого цикла спирали;
* неоправданно высокая стоимость модели для проектов, имеющих низкую степень риска или небольшие размеры;
* усложненность структуры модели, что приводит к сложности ее использования разработчиками, менеджерами и заказчиками;
* необходимость в высокопрофессиональных знаниях для оценки рисков;
* возможность отдаления окончания работы над проектом в связи с желанием заказчика улучшать каждую созданную версию

1. **V-образная модель.**

Преимущества:

* планирование на ранних стадиях разработки системы ее тестирования;
* обеспечение аттестации и верификации всех промежуточных результатов разработки;
* упрощение (по сравнению с каскадной моделью) отслеживания хода процесса разработки, возможность более реального использования графика проекта;
* простота в использовании

Недостатки:

* сложность поддержки параллельных событий;
* непредусмотренность итераций между фазами;
* невозможность внесения динамических изменений в требования на разных этапах жизненного цикла;
* поздние сроки тестирования требований в жизненном цикле, что оказывает существенное влияние на график выполнения проекта при необходимости выполнить их изменения;
* отсутствие в модели действий, направленные на анализ рисков.

1. **Модель быстрой разработки приложений (RAD-модель)**

Преимущества:

* Быстрое развитие продукта.
* Разработка многоразовых мелких компонентов.
* Повторный обзор в процессе разработки.
* Интеграция повторно используемых компонентов на начальном уровне, следовательно, экономит усилия, несмотря на то, что не добавляются более крупные модули.
* Конструктивная реакция.

Недостатки:

* Требуется много усилий для сбора всех требований на начальном этапе.
* Навыки моделирования имеют много зависимостей.
* Не подходит для малобюджетного проекта.

1. **Многопроходная модель.**

Преимущества:

* модель хорошо известна потребителям;
* она упорядочение справляется со сложностями и хорошо срабатывает для тех проектов, которые достаточно понятны, но все же трудно разрешимы;
* она весьма доступна для понимания
* она проста и удобна в применении, так как процесс разработки выполняется поэтапно;
* ее структурой может руководствоваться даже слабо подготовленный в техническом плане или неопытный персонал;

Недостатки:

* в основе модели лежит последовательная линейная структура, в результате чего каждая попытка вернуться на одну или две фазы назад, чтобы исправить какую-либо проблему или недостаток, приведет к значительному увеличению затрат и сбою в графике;
* пользователи не могут убедиться в качестве разработанного продукта до окончания всего процесса разработки. Они не имеют возможности оценить качество, если нельзя увидеть готовый продукт разработки;
* у пользователя нет возможности постепенно привыкнуть к системе. Процесс обучения происходит в конце жизненного цикла, когда ПО уже запущено в эксплуатацию;
* каждая фаза является предпосылкой для выполнения последующих действий, что превращает такой метод в рискованный выбор для систем, не имеющих аналогов, так как он не поддается гибкому моделированию;

1. **Модель прототипирования.**

Преимущества:

* начало взаимодействия заказчика с системой на раннем этапе разработки; то есть пользователь участвует в разработке всех требований, что повышает качество.
* снижение возможности искажения информации при определении требований, что приводит к созданию более качественного конечного продукта;
* возможность внесения в процессе разработки новых требований пользователя;
* возможность выполнения гибкого проектирования и разработки за счет итераций на всех фазах жизненного цикла;
* уверенность заказчиков в результатах выполнения проекта за счет постоянных признаков прогресса в последовательных итерациях;

Недостатки:

* недостаточность или неадекватность документации по ускоренным прототипам;
* вероятность недостаточного качества или долгосрочной эксплуатационной надежности результирующей системы за счет ее создания из рабочего прототипа; так как изначально разрабатываются основные функции, остальные более сложные – на потом.
* вероятность получения системы с низкой рабочей характеристикой при пропуске этапа подгонки;
* возможность отодвигания решения трудных проблем на будущее, что может привести к несоответствию полученных продуктов требованиям заказчиков;
* возможность задержки реализации конечной версии системы при несочетании языка или среды прототипирования с производственным языком или средой;

**Рассматриваемая процедура состоит из следующей последовательности шагов:**

***1-й шаг****.*Проанализировать отличительные черты проекта по критериям категорий, представленным в виде вопросов.

***2-й шаг****.*Ответить на вопросы по анализируемому проекту, отметив слова «да» или «нет» в соответствующих строках табл. 3.1 – 3.4. Если слов «да» или «нет» в строке несколько, необходимо отметить все из них (все «да» или все «нет»).

В качестве примера в табл. 3.1 выделены варианты ответов для проекта разработки сложного и критичного программного средства, требования к которому заранее не известны и будут уточняться по ходу разработки.

***3-й шаг****.*Расположить по степени важности категории (таблицы) и/или критерии, относящиеся к каждой категории (вопросы внутри таблиц), относительно проекта, для которого выбирается модель ЖЦ.

***4-й шаг.***Выбрать из моделей (см. табл. 3.1 – 3.4) ту модель, которая соответствует столбцу с наибольшим количеством отмеченных ответов с учетом их степени важности (с наибольшим количеством отмеченных ответов в верхней части приоритетных таблиц). Выбранная модель ЖЦ является наиболее приемлемой для анализируемого проекта.

Таблица 3.1 – Выбор модели жизненного цикла на основе характеристик требований

| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |

Продолжение таблицы 3.1

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | Нет |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Нет |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

V-образная – 5, RAD – 5, инкрементная – 3, эволюционная – 3, эволюционная – 3,быстрого прототипирования – 2, Каскадная – 2,.

На основе результатов заполнения табл. 3.1 наиболее подходящей является инкрементная модель.

Таблица 3.2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

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

RAD – 4, инкрементная – 3, быстрого прототипирования – 3, Каскадная – 3, эволюционная – 2, V-образная – 1,.

На основе результатов заполнения табл. 3.2 наиболее подходящими являются RAD и инкрементнаямодели.

Таблица 3.3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | Да | Да | Да | Да |

Продолжение таблицы 3.3

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Да | Нет | Да | Нет |

RAD – 3, быстрого прототипирования – 3, инкрементная – 2, эволюционная – 2, Каскадная – 1, V-образная – 1.

На основе результатов заполнения табл. 3.3 наиболее подходящими являются RAD и быстрого прототипирования модели.

Таблица 3.4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Нет |
| 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 |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Нет | Да | Нет | Нет | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Да | Нет | Нет | Да |

Продолжение таблицы 3.4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Нет |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Да | Нет | Нет | Да |
| 8. | Является ли график сжатым? | Нет | Нет | Да | Нет | Нет | Да |
| 9. | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Нет | Нет |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Нет |

RAD – 6, инкрементная – 5, эволюционная – 5, быстрого прототипирования – 4, Каскадная – 2, V-образная – 2,.

На основе результатов заполнения табл. 3.4 наиболее подходящей является RAD модель.

Исходя из результатов заполнения табл. 3.1 – 3.4 (каскадная – 8, V-образная – 9, RAD – 18, инкрементная – 14, быстрого прототипирования – 12, эволюционная – 13) наиболее подходящей моделью, для разработки данного ПП, является RAD модель.