



ЖИЗНЕННЫЙ ЦИКЛ (SOFTWARE LIFE CYCLE)

Митин Кирилл 20П-3



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.



1 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 её функционирования в интересах пользователей.

Процесс сопровождения. Определяет действия персонала сопровождения, который обеспечивает установку и удаление программного продукта, его сопровождение, что представляет собой поддержку текущего состояния и функциональную пригодность, а также действия по управлению модификациями.

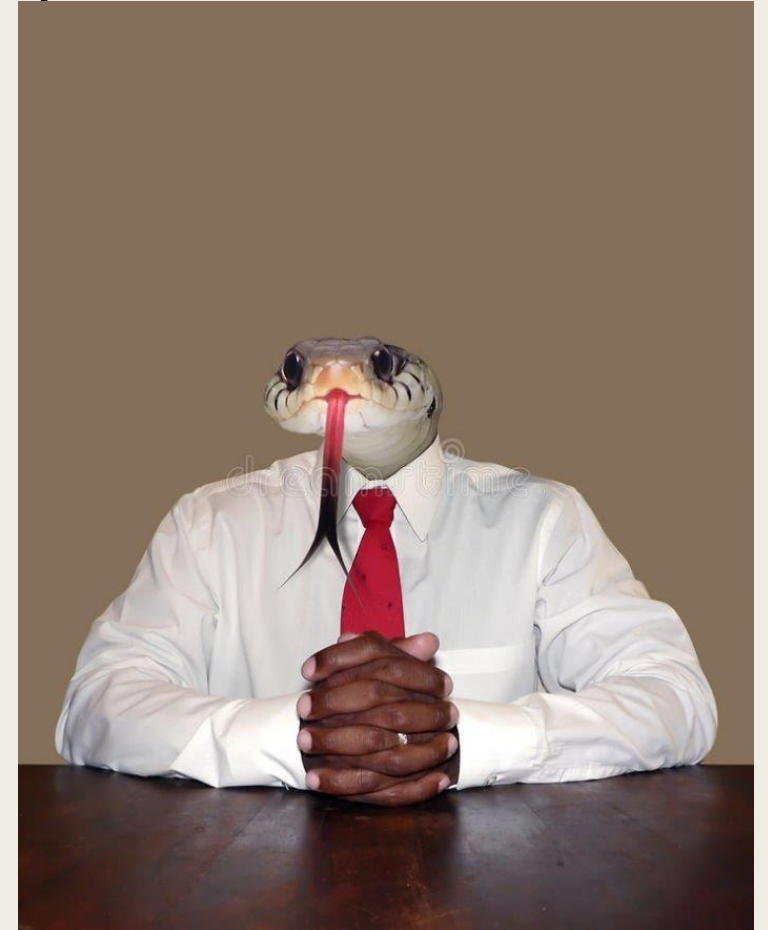
2. Вспомогательные процессы жизненного цикла

Вспомогательные процессы используются и выполняются по мере необходимости. Вспомогательный процесс инициируется другим процессом и поддерживает реализацию этого процесса с определённой целью, чтобы обеспечивать должное качество проекта программного обеспечения.

Процесс документирования. Определяет действия для записи результатов выполнения какого-либо из процессов жизненного цикла.

Процесс управления конфигурацией. Определяет действия по управлению конфигурацией.

Процесс обеспечения качества. Определяет действия для объективной гарантии, что программный продукт и процессы жизненного цикла соответствуют определённым требованиям к ним и придерживаются установленных замыслов. Совместная оценка, верификация, проверки, аттестации могут быть использованы как способы гарантии качества.



Процесс верификации. Определяет действия (для покупателя, поставщика, независимой стороны) для верификации программного продукта с различной глубиной зависимости от проекта программного обеспечения.

Процесс аттестации. Определяет действия (покупателя, поставщика, независимой стороны) для подтверждения качества программного продукта по общепринятой или официальной процедуре.

Процесс совместной оценки. Определяет действия для оценки состояния и результатов чего-нибудь в программном продукте или проекте программного обеспечения. Этот процесс может быть использован любыми двумя сторонами, где одна сторона (проверяющая, рецензирующая) проверяет (рецензирует) другую сторону (проверяемую) на совместном форуме.

Процесс проверки. Определяет деятельность для определения соответствия с требованиями, замыслами и контрактом. Этот процесс может быть использован любыми двумя сторонами, где одна сторона (проверяющая) проверяет программный продукт или деятельность другой стороны (проверяемой).

Процесс решения проблем. Определяет процесс анализа и устранения проблем (включая несоответствия), какова бы ни была их природа или источник, которые были обнаружены на протяжении разработки, эксплуатации, сопровождения или других процессов жизненного цикла.

3. Организационные процессы жизненного цикла

Организационные процессы выполняются с целью создания и обеспечения деятельности, включающей в себя связанные процессы жизненного цикла и персонал, а также совершенствование структуры и процессов. Организационные процессы инвариантны относительно конкретных проектов, однако извлечённые из этих проектов уроки способствуют совершенствованию организации-разработчика.



4. Модели жизненного цикла

Модель жизненного цикла – формализованная упрощённая структура, которая определяет последовательность выполнения практических этапов и их взаимосвязи на протяжении жизненного цикла программного средства.

Этап представляет собой логически завершённую часть жизненного цикла.

4.1. Водопадная модель жизненного цикла

Водопадная модель (*waterfall model*) является классической моделью жизненного цикла. Эта модель (рис. 5-2) предполагает, что переход к следующему этапу осуществляется только после полного завершения работ предыдущего этапа.



5. Основные сведения об этапах разработки

Разработка ПС начинается с этапа формулирования требований к нему. На этом этапе необходимо получить документ, достаточно точно определяющий задачи разработчиков. Этот документ часто называют спецификацией требований, или внешним описанием ПС.

5.1.1. Функциональная спецификация

Функциональная спецификация состоит из трех частей:

1. Описание внешней информационной среды.
2. Определение функций ПС, определённых на множестве состояний этой информационной среды (такие функции называют внешними).
3. Описание нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ ПС, и реакций на эти ситуации.

5.2. Анализ

Анализ направлен на описание задачи. Описание должно быть полным, непротиворечивым, реально проверяемым, пригодным для чтения и обозрения всеми заинтересованными сторонами.

Этап требует очень тесного контакта разработчиков с пользователями, так как возможные неточности и ошибки, вероятнее всего, приведут к краху проекта.

Анализ объясняет, что делает система, а не то, как она это делает.

5.3. Проектирование

Процесс проектирования не должен предшествовать анализу, но должен начинаться сразу после появления некоторой приемлемой модели поведения.

Проектирование объясняет, как ПС будет удовлетворять предъявленным к нему требованиям.

5.4. Реализация (кодирование) и эволюция

Реализация и эволюция – это создание компонентов и итеративное совершенствование для получения целевого ПС.

На этом этапе разработанные ранее логические и физические модели системы реализуются в виде реальных программных объектов, объектов пользовательского интерфейса, баз данных и других программных объектов.

5.5. Сопровождение

Сопровождение – это деятельность по управлению эволюцией ПС в ходе его эксплуатации.

5.6. Тестирование

При планировании работ по тестированию ПС из огромного количества возможных тестов выбирается малая, но реально выполнимая часть. И как бы тщательно ни выполнились эти отобранные тесты, все до единой ошибки всё равно не найти. Даже если в программе действительно не останется ошибок, об этом никогда не узнать: ведь этого нельзя ни доказать, ни проверить.

Полностью протестировать программу означает, что до окончания тестирования в ней не должно остаться ни одной не выявленной ошибки. Будут ли ошибки исправлены – это уже другой вопрос, но все имеющиеся проблемы должны быть известны и правильно поняты.

КОНЕЦ