

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Software Development Life Cycle, SDLC

Жизненный цикл разработки программного обеспечения – это процесс, используемый индустрией программного обеспечения для того, чтобы проектировать, разворачивать, разрабатывать и тестировать высококачественное программное обеспечение.

- направлен на создание **высококачественной системы**, которая отвечает ожиданиям клиентов, **эффективно** и **действительно** работает *в текущей и планируемой инфраструктуре* информационных технологий и является недорогой в обслуживании и экономически эффективной.

ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

- **SDLC** является общепризнанной аббревиатурой жизненного цикла разработки программного обеспечения.
- **SDLC** также называется процессом разработки программного обеспечения.
- **SDLC** – это структура, определяющая задачи, которые выполняются на каждом этапе процесса разработки программного обеспечения.
- **ISO / IEC 12207** является международным стандартом для процессов жизненного цикла программного обеспечения. Он является стандартом, который определяет все задачи, необходимые для разработки и поддержки программного обеспечения.

ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

- Работа **SDLC** направлена на
 - **снижение стоимости** разработки программного обеспечения,
 - **одновременно улучшая качество** и
 - **сокращая время** разработки.

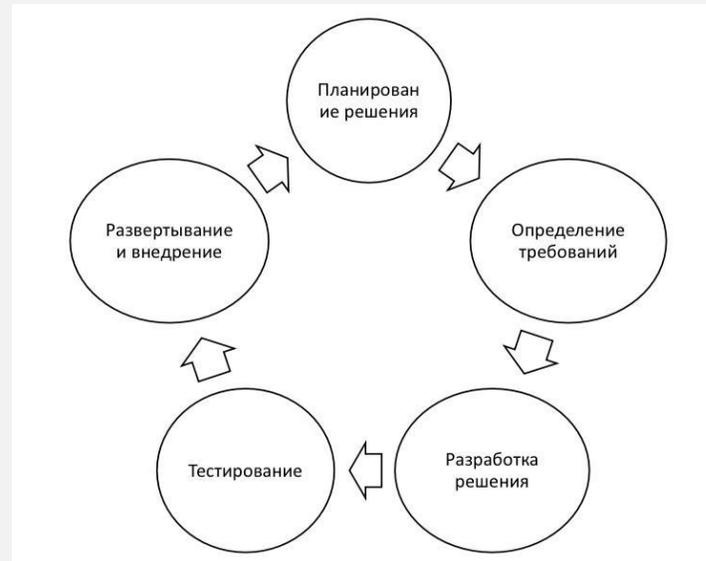
ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Достижение этих явно расходящихся **целей** происходит благодаря следованию **плана**, который устраняет типичные подводные камни для проектов по разработке программного обеспечения:

1. Оценку существующих **аналогичных систем** на предмет их недостатков.
2. Определение **требования** к новой системе.
3. Создание **программного обеспечения** на этапах *проектирования, разработки, тестирования и развертывания*.

ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Основные стадии жизненного цикла разработки программного обеспечения



ПЛАНИРОВАНИЕ И АНАЛИЗ ТРЕБОВАНИЙ

- **Анализ требований** – наиболее важный и фундаментальный этап жизненного цикла.
- Выполняется *старшими членами команды* с входными данными
 - от клиента,
 - отдела продаж,
 - исследований рынка и экспертов в целевой отрасли.
- Эта информация затем используется для **планирования** базового проектного подхода и проведения **технико-экономического обоснования** продукта в экономической, эксплуатационной и технической областях.

ПЛАНИРОВАНИЕ И АНАЛИЗ ТРЕБОВАНИЙ

- **Планирование требований** для обеспечения качества и выявление рисков, связанных с проектом, также осуществляется на этом этапе.
- **Результатом технико-экономического обоснования** является определение *разных технических подходов*, которые могут быть использованы для успешной реализации проекта с минимальными рисками.

ОПРЕДЕЛЕННЫЕ ТРЕБОВАНИЯ

- **Спецификация требований к программному продукту** (Software Requirement Specification, **SRS**) состоит из всех требований к продукту, которые должны быть разработаны в течение *жизненного цикла проекта*.
- Это следующий шаг после **проведения анализа требований**, который заключается в четком определении и документировании требований к продукту и их утверждение заказчиком или аналитиками рынка.

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОДУКТА

- Спецификация требований **SRS** – это справочная информация для архитекторов, разрабатывающих архитектуру продукта.
- На основе требований, указанных в **Спецификации**, архитекторами обычно предлагается и документируется более одного подхода к проектированию архитектуры продукта в спецификации разрабатываемого решения (Design Document Specification, **DDS**).

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОДУКТА

- **Спецификация** разрабатываемого решения рассматривается всеми важными заинтересованными сторонами:
 - заказчики проекта,
 - субподрядчики,
 - разработчики,
 - специалисты по качеству и внедрению будущего продукта.
- На основе разных параметров, таких как *оценка риска, надежность продукта, модульность проектирования, бюджетные и временные ограничения* – для продукта выбирается наиболее **оптимальный подход** к проектированию.

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОДУКТА

- Подход к проектированию четко определяет все **архитектурные модули продукта**, а также их связь и представление потоков данных с внешними и сторонними модулями (если таковые имеются).
- Внутренний дизайн всех модулей предлагаемой архитектуры должен быть четко определен с мельчайшими деталями в **DDS**.

СОЗДАНИЕ ИЛИ РАЗРАБОТКА ПРОДУКТА

- На этом этапе **SDLC** начинается **фактическая разработка** и создается продукт. Программный код создается в соответствии с **DDS**.
- Если конструкция выполнена в детальном и организованном формате, то разработку кода можно выполнить без дополнительных затрат ресурсов и времени на разные согласования.

СОЗДАНИЕ ИЛИ РАЗРАБОТКА ПРОДУКТА

- Разработчики должны следовать **рекомендациям по кодированию**, определенным их организацией и инструментами программирования, такими как компиляторы, интерпретаторы, отладчики и т. д., используемым для написания кода.
- Язык программирования выбирается в зависимости от типа разрабатываемого программного обеспечения и требований к нему.

ТЕСТИРОВАНИЕ ПРОДУКТА

- Этот этап обычно является подмножеством всех этапов; как и в современных моделях **SDLC**, тестовые мероприятия в основном участвуют во всех этапах **SDLC**.
- Этот этап относится только к этапу **тестирования продукта**, на котором дефекты продукта *озвучиваются, отслеживаются, исправляются и повторно тестируются* до тех пор, пока продукт не достигнет **стандартов качества**, определенных в спецификации требований.

РАЗВЕРТЫВАНИЕ И ОБСЛУЖИВАНИЕ

- Как только продукт протестирован и готов к развертыванию, он официально выпускается на соответствующем рынке.
- Иногда развертывание продукта происходит поэтапно в соответствии с бизнес-стратегией организации.
- Сначала продукт может быть выпущен в ограниченном сегменте и протестирован в реальной бизнес-среде (приемочное тестирование – User Acceptance Testing, UAT).

РАЗВЕРТЫВАНИЕ И ОБСЛУЖИВАНИЕ

- Затем, основываясь на обратной связи, продукт может быть выпущен «**как есть**» или с предлагаемыми улучшениями на целевой сегмент рынка.
- После того как продукт выпущен на рынок, его жизненный цикл переходит на стадию **поддержки**.
- **Поддержка** программного продукта включает в себя
 - **устранение дефектов**, найденных пользователями, или добавление улучшений, предложенных ими,
 - **добавление новой функциональности**, обеспечивающей развитие продукта.

МОДЕЛИ РЕАЛИЗАЦИИ ЖИЗНЕННОГО ЦИКЛА

- Существует **ряд моделей**, реализующих жизненный цикл разработки программного обеспечения и дающих конкретные рекомендации о том, как именно реализовывать тот или иной этап **жизненного цикла**.
- Эти модели также называются **моделями процессов разработки программного обеспечения**.
- Каждая модель процесса состоит из **серии шагов**, уникальных для своего типа, чтобы обеспечить успех в процессе разработки программного обеспечения.

МОДЕЛИ РЕАЛИЗАЦИИ ЖИЗНЕННОГО ЦИКЛА

Наиболее известные и популярные модели SDLC, используемые в отрасли:

- водопадная модель (**Waterfall**),
- итерационная модель (**Iterative model**),
- спиральная модель (**Spiral model**),
- V-модель (**V-model**),
- модель большого взрыва (**Big-bang model**).