

Технологии разработки программного обеспечения

Лекция 1. Введение. SDLC.

Авласевич Наталия Тадеушевна
(ст.пр. кафедры ИСиТ)

Терминология

Программное обеспечение (ПО) – программа или множество программ, используемых для управления компьютером.

Разработка ПО :

- Разработка нового программного продукта (ПП)
- Изменение существующего продукта (legacy)
- Поддержка существующего продукта (вносить изменения).

Что главное в разработке ПП?

- 1- мы должны разработать правильный продукт (который нужен пользователю или заказчику) - **валидация**
- 2 – мы должны разработать продукт правильно (отвечает требованиям, архитектура , код, документация) – **верификация**

Этапы разработки ПП

- ▶ Исследование рынка и пользователей
- ▶ Сбор и анализ требований
- ▶ Выбор подходящего дизайна
- ▶ Подбор технологий
- ▶ Расчет экономической эффективности
- ▶ Разработка
- ▶ Тестирование
- ▶ Анализ
- ▶ Сопровождение
- ▶ **Жизненный цикл программного средства**
- ▶ Снятие с эксплуатации

Терминология

ЖЦ ПС – совокупность процессов , работ задач , включающих в себя разработку , эксплуатацию и сопровождение ПС или системы, охватывающая их жизнь от формулировки концепции до прекращения использования.

На уровне международных стандартов ЖЦ ПС отражен в стандарте ISO/IEC 12207:1995 «Information technology–Software life cycle processes» («Информационные технологии – Процессы программного жизненного цикла»)

В РБ этот стандарт введен в 2004 г. под СТБ ИСО/МЭК 12207-2003.

SDLC - Software Development Life Cycle

жизненный цикл программного обеспечения



Фазы жизненного цикла программного обеспечения

- ▶ **Планирование и анализ требований (Planning)**. Этот этап осуществляется старшими специалистами команды с участием заказчиков, отдела продаж, маркетинга и т.д. На основе полученных в ходе него сведений планируется базовый подход к проекту, проводится техническое и экономическое обоснование продукта, прогнозируются риски, определяются требования к качеству. Итог этого этапа заключается в определении подходов, которые можно применить для успешного создания проекта с наименьшими издержками.

- ▶ **Определение требований (Defining)**. После завершения предыдущего этапа четко определяются и документируются конкретные требования к продукту. Они направляются клиенту и рыночным аналитикам для согласования и утверждения. Для этого используется документ SRS (Спецификация требований к программному обеспечению), содержащий все нормы, которым должен соответствовать продукт.

- ▶ **Проектирование архитектуры (Designing).** SRS — это «дорожная карта» для разработчиков, с помощью которой они предлагают оптимальную архитектуру для будущего продукта. На базе требований из этого документа, как правило, определяется несколько подходов к разработке, которые фиксируются в DDS, документе проектирования. Он предлагается на рассмотрение всем участвующим в проекте сторонам, которые совместно оценивают такие критерии, как риски, прогнозируемая надежность продукта, модульность внутреннего дизайна, ограничения финансов и времени, и выбирают подходящий подход к проектированию. Он, в свою очередь, содержит четко определенные архитектурные блоки продукта, его связь и представление потока данных с внешними модулями (при их наличии).

- ▶ **Разработка (Building)**. На этой стадии жизненного цикла осуществляется непосредственная работа по созданию и сборке продукта в соответствии с DDS. При наличии детализированного и организованного дизайна написание кода обычно не вызывает серьезных затруднений. В разработке применяются такие средства программирования, как компиляторы, интерпретаторы, отладчики и т.д. Код пишется на различных языках программирования высокого уровня — например C, C++, Java и PHP. Его выбор зависит от типа разрабатываемого ПО.

- ▶ **Тестирование продукта (Testing)**. В том или ином виде проверка продукта осуществляется на всех этапах его жизненного цикла, от анализа до развертывания. На стадии непосредственно технической проверки выявляются, отслеживаются и исправляются дефекты продукта. Эти процедуры проводятся до тех пор, пока продукт не станет полностью соответствовать стандартам, указанным в SRS

- ▶ **Развертывание и сопровождение (Deployment)**. После проверки продукта на ошибки и их устранения он готов к релизу. Развертывание может быть единовременным или поэтапным — в зависимости от того, какую бизнес-стратегию выбрали заказчик и разработчик. Часто первый релиз выпускается в ограниченном сегменте рынка для проведения пользовательского тестирования (UAT) в реальной бизнес-среде. Получив отзывы от представителей целевой аудитории, разработчик далее публикует полноценный релиз без изменений или после соответствующей доработки. Обслуживание уже выпущенного на рынок продукта осуществляется в расчете на существующую клиентскую базу.

Модели жизненного цикла ПО

Модель жизненного цикла программного обеспечения — обобщенное описание действий и задач, осуществляемых в ходе разработки, внедрения и сопровождения информационной системы. Это абстракция реального процесса создания продукта, в которой опущены многие мелкие нюансы. Такое обобщение нужно, чтобы разработчикам было удобнее выбрать подходящую модель под свой проект, не запутавшись в несущественных деталях.

Условно модели разделяются на три основных категории:

- ▶ **Последовательные** — представляют собой строгие последовательности выполнения этапов, каждый из которых начинается только тогда, когда закончился предыдущий.
- ▶ **Итерационные** — такие модели подразумевают, что продукт проходит несколько итераций, каждая из которых содержит все фазы разработки, до тех пор пока не будет полностью соответствовать требованиям, необходимым для выпуска на рынок.
- ▶ **Гибкие** — в их основе лежит итерационный принцип разработки ПО, но структура циклов может меняться в зависимости от текущей задачи и/или степени готовности продукта.

Модели SDLS:

1. Каскадная модель (Waterfall)
2. V- модель
3. Инкрементная модель
4. Эволюционная (спиральная) модель
5. Итерационная модель
6. **Agile**

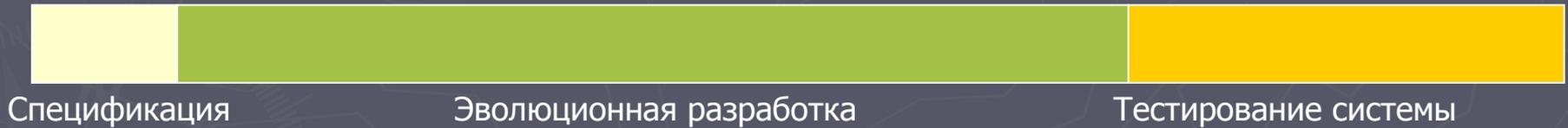
Структура затрат на создание ПО



Распределение стоимостей отдельных этапов создания ПО



Структура затрат при использовании эволюционного подхода



Затраты на разработку и совершенствование системы



Методы создания ПО

Методы представляют собой структурный подход к созданию ПО, который способствует производству ПО эффективным, с экономической точки зрения, способом.

Все основаны на использовании моделей системы в качестве спецификации ее структуры

1. **Функционально-ориентированные** (структурный анализ, JSD, 70-е годы) основаны на определении основных функциональных компонент системы.
2. **Объектно-ориентированные** (Booch, Rumbaugh) используют подходы, основанные на использовании унифицированного языка моделирования UML.

CASE-технологии

Computer-Aided Software Engineering – автоматизированная разработка ПО.

Широкий спектр программ, применяемых для поддержки и сопровождения различных этапов создания ПО:

- ▶ Анализ системных требований
- ▶ Моделирование системы
- ▶ Отладка и тестирование
- ▶ Создание документации
- ▶ Генерация исходного кода программ
- ▶ Др...

Верхний
уровень

Нижний
уровень

Характеристики качества ПО

Удобство сопровождения

усовершенствование в ответ на измененные требования заказчика

Надежность

безотказность

защищенность

безопасность

...

Эффективность

скорость выполнения

процессорное время

объем требуемой памяти

...

Удобство использования

не требует чрезмерных усилий пользователя
соответствующий пользовательский интерфейс
документация

...

Основные проблемы, стоящие перед специалистами по ПО

- ▶ Исследования ранее созданного ПО
- ▶ Разнородность программных систем
- ▶ Требование уменьшения времени на создание ПО