Юлия Николаевна

Технология разработки ПО

Трпо – совокупность процессов и методов программного продукта.

Трпо – система инженерных принципов для создания экономического ПО, которое эффективно и надёжно работает на реальных ПК.

Программный продукт – прога с документацией (полный комплект).

Технология программирования (как термин) был введён «Ершов Андрей Петрович.»

Разработка состоит из:

1. Анализ (требования, рынок и т.д.);
2. Проектирование;
3. Разработка (программирование);
4. Тестирование;
5. Отладка.

**Цели и задачи ТРПО**

Особенности современных информационных систем:

* Сложность описания;
* Наличие совокупности тесно связанных, взаимодействующих компонентов; (типо, есть банк, а есть банкомат, и они связаны)
* Отсутствие прямых аналогов; (нужно придумывать чё-нить новое, а не копипастить уже созданное)
* Необходимость интеграции существующего, и разработанного приложения; (если сделали моб. прогу, не нужно переделывать основу)
* Функционирование в неоднородной среде, на нескольких аппаратных платформах;
* Разобщённость и разнородность групп разработчиков по уровню квалификации и использованию инструментальных средств;
* Временная протяжённость проекта.

ПО – полный набор, или часть программ (процедур, правил) и связанный с ними документации.

ПО – интеллектуальный продукт, не зависящий от среды, на которой он написан.

Программные средства = ПО

Программных продукт – набор компьютерных программ, процедур и док-ции, предназначенный для передачи пользователю.

Смертельный марш – (ввёл Э. Йордан) под ним понимается проект, параметры которого откланяются от нормы на 50%+ (Хотя бы одного):

1. план проекта сжат более чем на половину;
2. кол-во разрабов уменьшено на 50%;
3. Бюджет урезается более чем на половину;
4. Более чем на половину увеличивается функционал.

Кризис программирования – появился в начале 70-х в USA, выражался в том, что большие проекты стали выполнятся с отставанием, с превышением сметы расходов, разработанный продукт не обладал требуемыми функциональными возможностями и херовым качеством. (в 95 году примерно 16% проектов были норм.)

**Сложная система**

Сложная система – система, состоящая из взаимодействующих составляющих (подсистем), в следствии чего, она приобретает новые св-ва.

Признаки сложной системы:

1. Наличие общей задачи;
2. Большое кол-во компонентов;
3. Возможность декомпозиции системы;
4. Иерархическая архитектура системы;
5. Сложность поведения системы;
6. Устойчивость системы к внешним воздействиям;
7. Высокая надёжность системы.

**Понятие и основные этапы жизненного цикла ПО**

Жизненный цикл ПО – момент жизни прогу, который она задумывается, и кончается в тот момент, когда заканчивается её поддержка.

Стандарт ISO/IEC 12207 – (от 1995 года) регламентирует жизненный цикл ЖЦПО

Процесс – совокупность взаимосвязанных действий, которые преобразуют input to output. (согласно iso/iec 12207)

Каждый процесс хар-ся задачами и методами.

Процесс делится на действие, а действие на задачи.

В соответствии со стандартом, процессы делятся на:

1. Основные процессы (процессы без которых ничего работать не будет);
   1. Процесс приобретения (действие заказчика);
   2. Процесс поставки (действие разработчика, а точнее компании по сдаче продукта);
   3. Процесс разработки (создание ПО и компонентов);
   4. Процесс эксплуатации;
   5. Поддержка;
2. Вспомогательные;
   1. Документирование (формализованное описание информации в течении ЖЦПО);
   2. Управление конфигурацией (проджект-мененджеминг);
   3. Процесс обеспечения качества;
   4. Процесс верификации (проверка на соответствие требованиям);
   5. Процесс совместной оценки;
   6. Процесс аудита (Проверка ПО сторонней компанией);
   7. Разрешение проблем;
   8. Процесс аттестации (оценка достоверности проведённого тестирования).
3. Организационные
   1. Процесс управления
   2. Процесс создания инфраструктуры (выбор ПО, раб места и тп)
   3. Процесс усовершенствования (оценка измерение усовершенствование ЖЦПО
   4. Процесс обучения

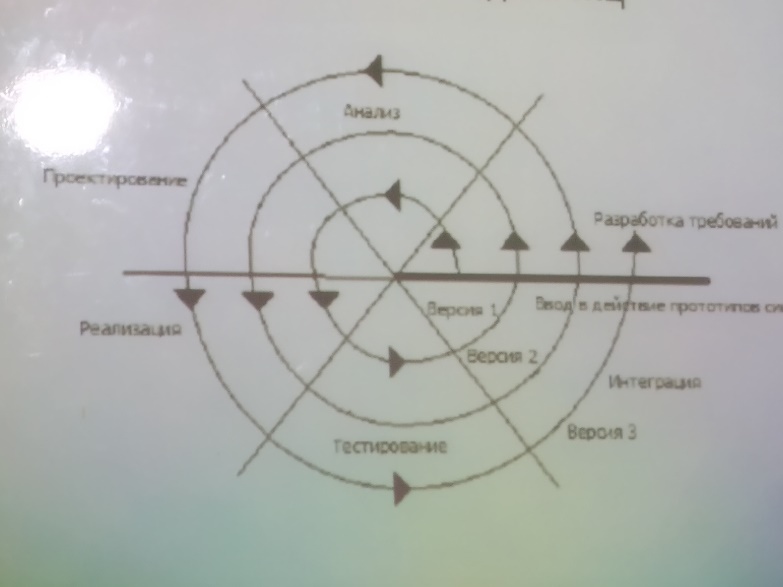
Модели ЖЦПО со стандартом ISO/IEC 12207

Модель ЖЦПО – структура определяющая последовательность выполнения и взаимосвязи процессов, действий задач на протяжении ЖЦ. Она зависит от специфики, масштаба и сложности проекта  
Стадия создания ПО – часть процесса создания ПО, ограниченная временными рамками и заканчивающаяся выпуском конкретного продукта

1. Стадии ЖЦПО:  
   Формирование требований к ПО  
   Проектирование  
   Реализация  
   Тестирование  
   Ввод в действие  
   Эксплуатация и сопровождение  
   Снятие с эксплуатации

Две модели ЖЦПО: Каскадная (1970-80):

**Илья забыл написать**

Спиральная (1986-90):  


Собираем требования – анализ – проектирование – кодируем(реализация) – тестирование – интеграция – ввод в действие прототипов

Прототип – действующий программный компонент, реализующий часть функций и часть интерфейсов программы.

**Управление требованиями системы**

Требования – условие которому должна соответствовать система;

Анализ требований *–* предварительно определение того, что не удастся реализовать, перед тем, как реализация провалится;

Оценка требований – доступное и умелое разъяснение заказчикам требований, которые не будут реализованы;

Управление требованиями – убеждение заказчика в том, что хочет именно он (но то, что вы можете реализовать);

Необходимость требования к системе:

1. Добиться понимания с заказчиком, как должна выглядеть система;
2. Дать разработчикам понимание требований к системе;
3. Определить границы системы;
4. Обеспечить базу для оценки стоимости и времени для разработки
5. Обеспечить базу для планирования итераций;
6. Определить пользовательский интерфейс;

Виды требований:

1. Функциональные – описание поведения и сервисов системы, её функционала. Определяют то, что должна система делать;
2. Нефункциональные – определяет надёжность, особенности поставки уровень качества и т.д.

Уровни требований:

1. Бизнес – определяет цели клиента или заказчика ПО (**ПОЧЕМУ**?);
2. Пользовательские – определяет цели/задачи которые должны выполнятся пользовательской системой при помощи создаваемого ПО (**КТО**? И **ЧТО**?);
3. Функциональные – функциональность ПО, которая должна быть создана разрабами для предоставления возможностей выполнения пользователями своих обязанностей в рамках бизнес-требований (**ЧТО**?);

Свойства требований:

1. Ясность, недвусмысленность;
2. Полнота и непротиворечивость;
3. Необходимый уровень детализации;
4. Прослеживаемость;
5. Тестируемость;
6. Модифицируемость;

Методы выявления требований:

1. Интервьюирование – получение получения инфы от пользователя во время интервью;
2. Сценарии – приведение примеров из реала вместо абстрактных значений. Продумываются сценарии работы;
3. Этнография – аналитик внедряется в будущее окружение системы и наблюдает повседневную работу. Тем самым выявляя не выявленные требования;
4. Прототипы – создаются предварительные модели (вплоть до бумажных);
5. Наблюдение – непосредственное наблюдение аналитиков за пользователем;
6. Выявление требований на основе опорных точек зрения;

**Отличие наблюдения от этнографии в том, что наблюдение со стороны пользователя;**

**Методы выявления требований могут комбинироваться и дополнятся.**

Документирование требований (документы):

1. Обзор продукта (внешнее описание системы);
2. Системные модели (диаграммы прецедентов, деятельности, user story);
3. ТЗ (техническое задание);
4. Частичные ТЗ;

Управление требованиями – системный подход, систематический подход к обнаружению, организации, документированию и сопровождению изменяющихся требований к системе.

Управление изменениями требований:

* Анализ проблем – анализ проблем и предложений их изменений;
* Анализ примечаний – чекается результат изменений;
* Реализация изменений – реализация.

Работа по управлению требований;

* Обнаружение требований и их документирование;
* Установление между заказчиком и исполнителем соглашений;
* Отслеживание изменений и оценка их влияния на процесс.

**RAD**

**(Rapid Application Development)**

**(Методология быстрой разработки)**

RAD Включает:

1. Короткий график разработки;
2. Маленькая команда разрабов;
3. Повторяющиеся циклы (Требования дополняются общением с заказчиком);
4. Команда разработчиков должна быть лютыми многофункциональными профессионалами.

Главная идея методологии RAD – как можно быстрее донести до заказчика рез-ты работы.

ЖЦ ПО в соответствии RAD:

1. Анализ и планирование требований – узнаём, что требует заказчик, и узнаём приоритетность требований, ограничивается масштаб проекта и временные рамки для каждой фазы. Выясняется, если смысл в реализации данного проекта в текущих финансовых условиях.  
   **В результате** фазы мы получаем список и приоритетность функций, предварительная функциональная и информационная модель;
2. Проектирование – Разработчики и конечные пользователи участвуют в техническом проектировании. Используются CASE-средства, для быстрого получения прототипа. Более подробно рассматриваются процессы в системе. Анализируется и корректируется функциональная модель. Каждый процесс рассматривается детально. Определяются требования к разграничению данных. Оценивается кол-во функциональных моментов, разрабатываемых системой, для разбиение её на подсистемы (Прога разбивается на более мелкие подсистемы), для разработки за приемлемое время (60-90 дней)  
   **В результате** – общая информационная модель, нормальная модель системы и подсистемы, определены интерфейсы, между автономно работающими подсистемами, построенные прототипы экранов и отчётов и диалогов с системой;
3. Построение – Пишется код проги. Пишется быстро из-за того, что часть кода генерится с помощью CASE-средств. Во время разработки осуществляется тестирование каждой подсистемы. Завершается физическое проектирование системы. Производится анализ использования данных и делаются БД. Определяются требования к аппаратным рес-ам. Завершается раз-ка документации.  
   **В результате ­**– готовая система, удовлетворяющая всем требованиям;
4. Внедрение – производится обучение пользователей. Фаза короткая, за сим, планируется заранее  
   **В результате**  - внедрённая информационная система;

Методология RAD не универсальная, подходит для небольших проектов, которые разрабатываются для конкретного заказчика.  
Методология RAD не подходит для сложных расчётных программ, ОС.

Оценка размеров приложений в RAD делается на основании функциональных элементов (меньше 1000 элементов – разрабатывает 1 чел, От 1000 до 4000 – тима)  
Основные принципы RAD:

1. Итерационная разработка;
2. Необязательно завершать работу на каждом этапе ЖЦ ПО;
3. Применение CASE-средств;
4. Участие конечных пользователей в разработке;
5. Разработка прототипов;
6. Тестирование параллельно разработке;
7. Разработка подсистем командами;