

ЛАБОРАТОРНАЯ РАБОТА № 2

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Выполнил: студент гр. БИН-22(оз)

Сидикова Д

Проверил: преп. Петрова Л.В

ЦЕЛЬ РАБОТЫ

Рассмотреть основные понятия, используемые в области разработки ПО:

- жизненный цикл;
- модели и методологии разработки;
- управление проектом;
- анализ, проектирование, кодирование, тестирование и документирование программных средств;
- оценка качества ПО.

ОСОБЕННОСТИ ПРОМЫШЛЕННОГО ПО И КРИЗИС ЕГО РАЗРАБОТКИ

Во многих случаях программы создаются:

- в единственном экземпляре;
- для решения частных задач;
- не имеют массового применения;
- доступны только тем, кто их разработал.

К сожалению, сами разработчики и их руководители часто не осознают этого факта.

КРИЗИС РАЗРАБОТКИ ПО

Разработка ПО не доводится до выпуска продукта, или с неоправданно большими затратами выпускается «сырая», ненадежная программа. Эта ситуация получила название «кризис разработки ПО».

Вывод: подход к разработке программного продукта качественно не должен отличаться от подхода к разработке любого промышленного продукта.

Во всех случаях разработки требуется:

- техническая компетентность;
- грамотное управление.

СЛОЖНОСТЬ РАЗРАБОТКИ ПО

- Существенная черта серьезного программного продукта — уровень сложности.
- Сложности разработки делятся на:
 - 1) внутренне присущие природе ПО
 - 2) сопутствующие производству ПО, но не внутренне ему присущие
- сущие

ПРИЧИНЫ СЛОЖНОСТИ РАЗРАБОТКИ ПО

- 1. Сложность предметной области.
- 2. Внутренняя сложность программ.
- 3. Отсутствие хороших способов представления больших систем.
- 4. Трудности управления процессом разработки.
- 5. Изменение требований к программе в процессе её разработки.

ХАРАКТЕРИСТИКИ ПРОГРАММНОГО ПРОДУКТА

- «верхний» уровень;

ВЫСОКОУРОВНЕВЫЕ СВОЙСТВА ПРОГРАММЫ

- функциональные возможности;
- надёжность;
- практичность;
- эффективность;
- сопровождаемость;
- мобильность.

Указанные свойства являются свойствами «верхнего» уровня, то есть каждое из них подразумевает наличие множества второстепенных свойств.

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ПРОДУКТА

- Жизненный цикл (ЖЦ) программного средства – совокупность взаимосвязанных процессов создания и последовательного изменения его состояния.
- Процесс ЖЦ — конкретный вид деятельности, систематически выполняющийся для решения определенных задач ЖЦ.

ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА

Основные процессы:

- приобретение;
- поставка;
- разработка;
- функционирование;
- сопровождение.

Организационные процессы:

- управление;
- усовершенствование;
- обучение;
- создание инфраструктуры.

Вспомогательные процессы:

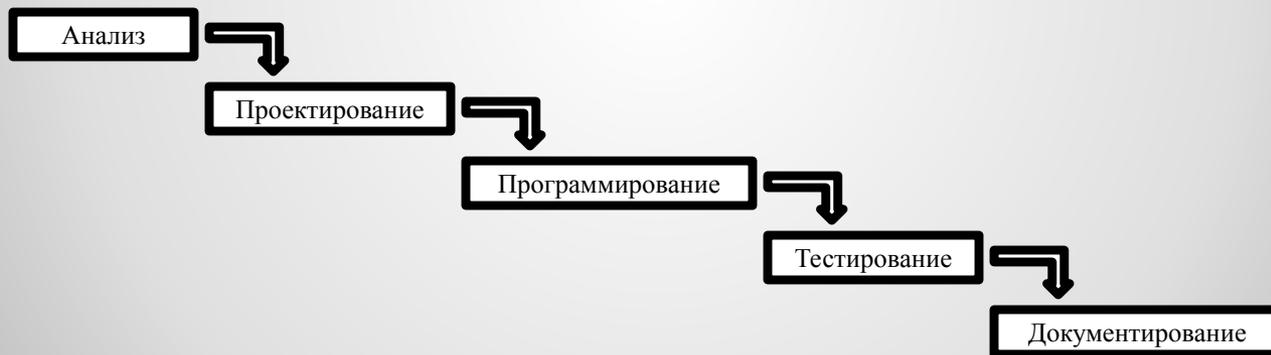
- документирование;
- управление конфигурацией;

ПРОЦЕССЫ РАЗРАБОТКИ

- анализ;
- проектирование;
- программирование (кодирование, реализация);
- тестирование;
- документирование1.

МОДЕЛИ РАЗРАБОТКИ

Водопадная модель.



Главная характеристика — разделение процесса на этапы; последующий этап начинается после завершения предыдущего.

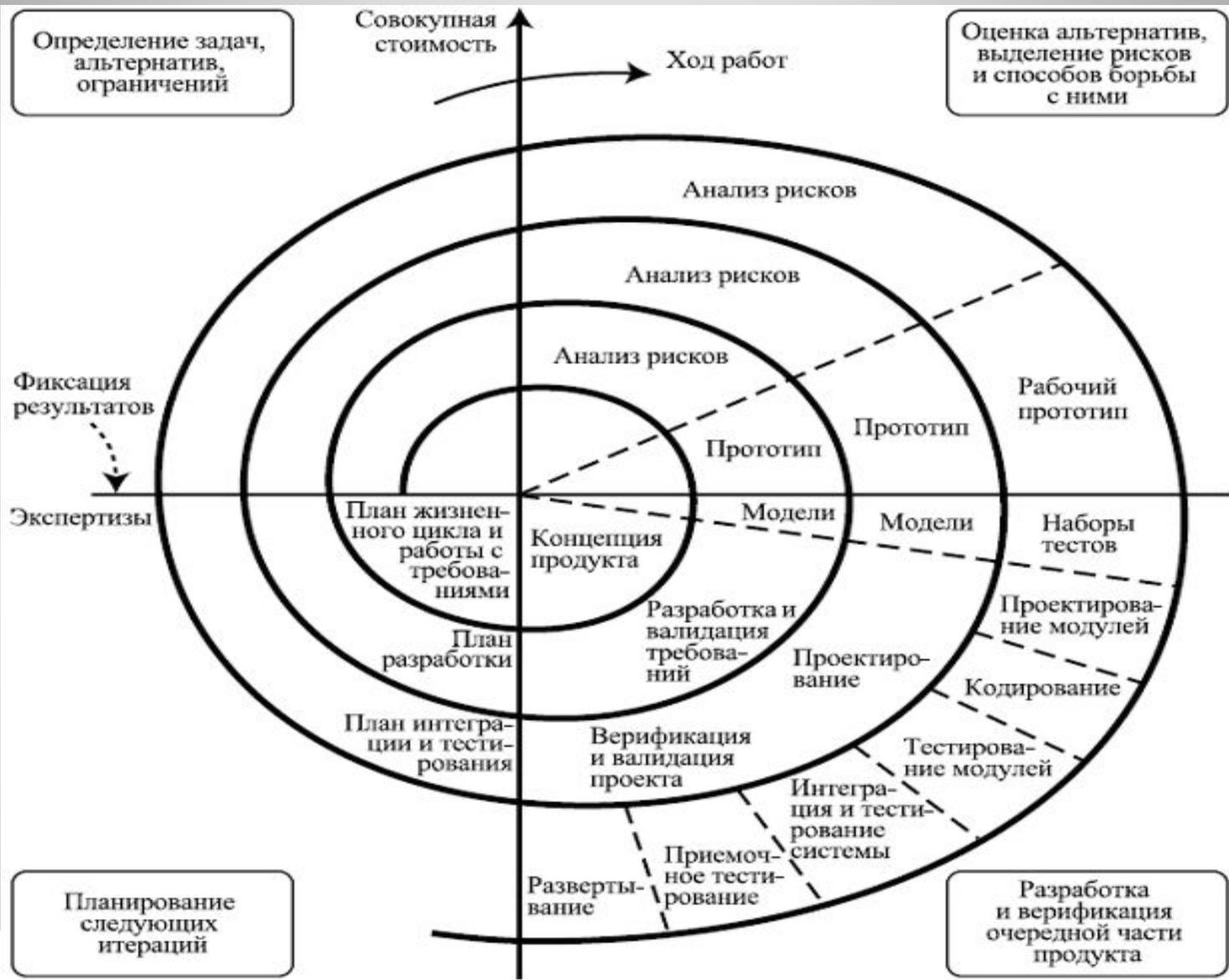
ЭВОЛЮЦИОННАЯ МОДЕЛЬ



СТОРОНЫ ИД

Отрицательные стороны:

Положительные стороны:



Определение задач, альтернатив, ограничений

Оценка альтернатив, выделение рисков и способов борьбы с ними

Фиксация результатов
Экспертизы

Планирование следующих итераций

Разработка и верификация очередной части продукта

Совокупная стоимость
Ход работ

Анализ рисков

Анализ рисков

Анализ рисков

Прототип

Рабочий прототип

Концепция продукта

План жизненного цикла и работы с требованиями

Модели

Модели

Наборы тестов

План разработки

Разработка и валидация требований

Проектирование

Проектирование модулей

План интеграции и тестирования

Верификация и валидация проекта

Кодирование

Тестирование модулей

Развертывание

Приемочное тестирование

Интеграция и тестирование системы

Разработка и верификация очередной части продукта

РИСКИ ВЛИЯЮЩИЕ НА ОРГАНИЗАЦИЮ ЖЦ

- й.

НАБОР КОНТРОЛЬНЫХ ТОЧЕК

Concept of Operations (COO) — использование системы;

Life Cycle Objectives (LCO) — цели жизненного цикла (ЖЦ);

Life Cycle Architecture (LCA) — архитектура ЖЦ;

Final Operational Capability (FOC) — готовый продукт.

АСПЕКТЫ МЕТОДОЛОГИИ РАЗРАБОТКИ

- *последовательность работ и их содержание;*
- *артефакты, которые необходимо создавать в процессе работы:*
- *документы, диаграммы, исходные тексты и т. д.;*
- *организацию команды и ролевую ответственность специалистов;*
- *лучшие практики (best practices), позволяющие максимально эффективно воспользоваться методологией и её моделью*

ЕДИНАЯ СИСТЕМА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

- *Единая система программной документации (ЕСПД)* — комплекс
- государственных стандартов РФ, устанавливающих взаимосвязанные
- правила разработки, оформления и обращения программ и программной
- документации в РФ. Стандарты ЕСПД в основном охватывают ту часть
- документации, которая создается в процессе разработки ПС, но ЕСПД
- специфицирует и последовательность этапов разработки, и их содержа-
- ние, что и позволяет назвать её в определённой мере *методологией*.

MICROSOFT SOLUTIONS FRAMEWORK

Методология содержит следующие части:

- модель проектной группы;
- модель процессов;
- дисциплина управления проектами;
- дисциплина управления рисками;
- дисциплина управления подготовкой

Отличительная черта MSF - Общеизвестно, что MSF уделяет значительное внимание вспомогательным и организационным процессам. В то же время, слабостью MSF всегда были именно технологические процессы разработки, которым в первых версиях методологии не уделялось вообще никакого внимания.

ЭКСТРЕМАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Принципы *agile-методологий* :

1. Взаимодействие людей важнее процессов и инструментов.
2. Работающая программа важнее исчерпывающей документации.
3. Сотрудничество с заказчиком важнее обсуждения контрактов.
4. Адаптация к изменениям важнее следования планам.

МЕТОДОЛОГИЯ XP

Практики методологии:

1. Планирование замысла (planning game1).
2. Частые выпуски (small releases).
3. Метафора системы (system metaphor).
4. Простой проект (simple design).
5. Разработка, управляемая тестированием (test-driven development).
6. Переработка системы (refactoring).

Продолжение.

- Парное программирование (pair programming).
- Непрерывная интеграция (continuous integration).
- Коллективное владение кодом (collective code ownership).
- Заказчик всегда рядом (on-site customer).
- 40-часовая неделя (40-hour weeks).
- Открытое рабочее пространство (open workspace).
- Всего лишь правила (just rules).

Оценка качества процесса разработки

Более *зрелая* организация с более *качественным* процессом будет выпускать и более *качественные* продукты.

Одной из первых известных моделей качества стали стандарты ISO серии 9000, первая версия которых была выпущена в 1987 г.

Недостатки:

- недостаточная подробность стандарта;
- отсутствие разных уровней оценки;
- неточность оценки качества процессов.

Серия ISO 9000 объединяет три стандарта:

- ISO 9000:2005 — Системы менеджмента качества. Основные положения и словарь;
- ISO 9001:2000 — Системы менеджмента качества. Требования;
- ISO 9004:2000 — Системы менеджмента качества. Рекомендации по улучшению деятельности.

CMM (*Capability Maturity Model*) — модель зрелости возможностей, созданная с целью упорядочения выбора поставщиков крупных правительственных организаций.

Ключевым понятием стандарта CMM является *зрелость* организации.

Продолжение

Уровни зрелости организаций:

- *Начальный уровень.* На этот уровень не сертифицируются, он описан как отправная точка.
- *Повторяемый уровень.* Относятся компании, в которых внедрено управление проектами.
- *Определенный уровень.* Процессы управления интегрированы с инженерными процессами в единый процесс.
- *Управляемый уровень.* Устанавливаются *количественные* показатели качества.
- *Оптимизирующий уровень.* Улучшение существующих процессов. Организация нацелена на *предупреждение дефектов*.

В 2000 г. SEI предложил сменить CMM новой, улучшенной моделью — CMMI.

Продолжение