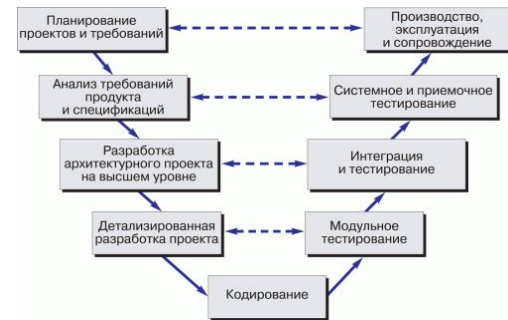


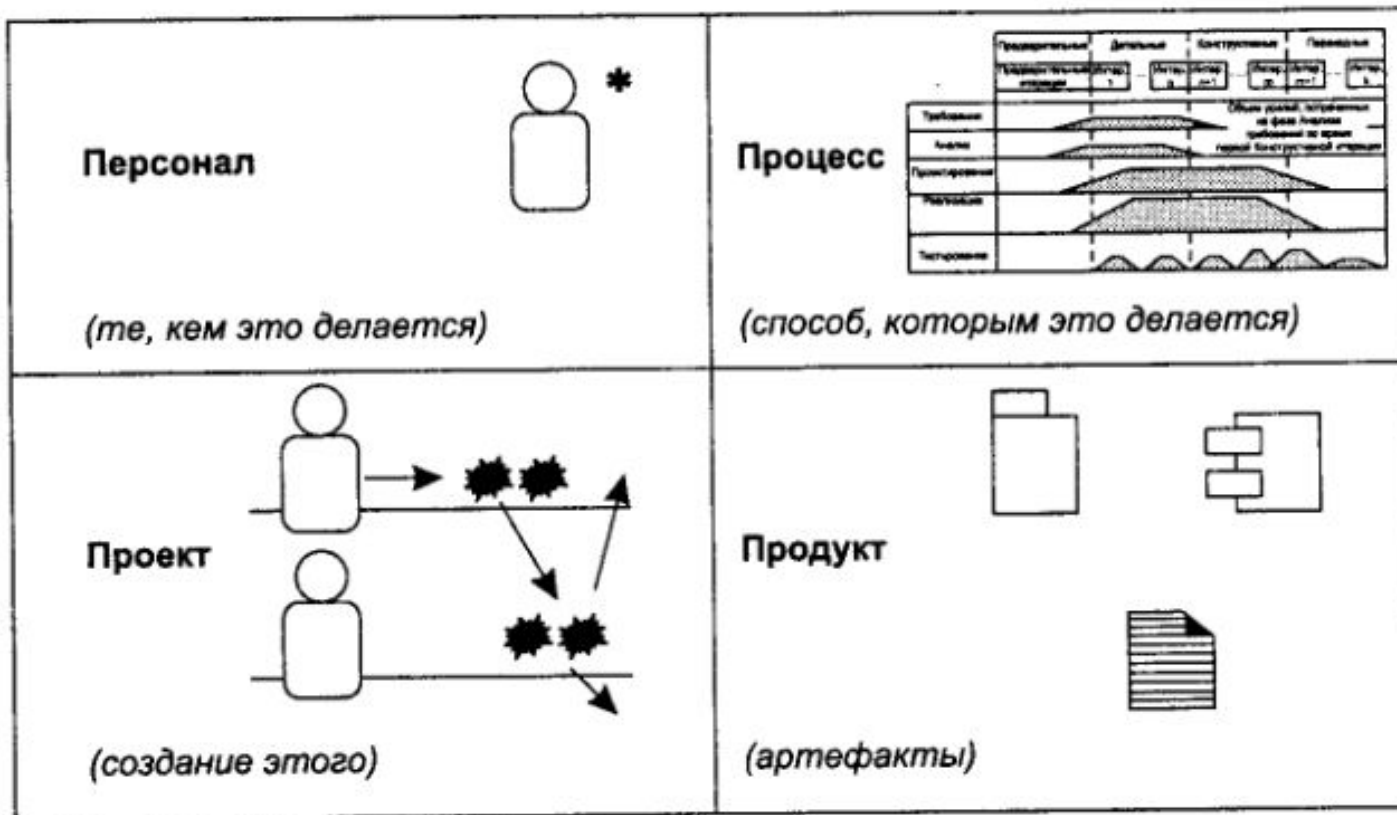
## 4P's

# Lecture Outline



- 1) **P #1** : Процесс разработки ПО
- 2) **P #2** : Персонал разработки ПО
- 3) **P #3** : Продукт (собственно ПО)
- 4) **P #4** : Проект по разработке ПО

# 4P's - Project, Process, People, Product



# Process

	Основные этапы разработки ПО
1)	Планирование проекта
2)	Составление требований
3)	Проектирование
4)	Кодирование
5)	Тестирование
6)	Сопровождение

Процессы жизненного цикла ПО:

- Основные
- Поддерживающие
- Организационные

## Стратегии

### разработки

- Водопадная
- Инкрементная
- Эволюционная
- RAD
- V-Model
- Prototyping

## Принципы

### разработки

- PSP
- TSP
- CMM

## Стандарт

ы

- ISO
- IEEE

# Software Lifecycle Processes (ISO)

- Основные
  - Приобретение
  - Поставка
  - Разработка
  - Эксплуатация
  - Сопровождение
- Поддерживающие
  - Документирование
  - Управление конфигурацией
  - Обеспечение качества
  - Верификация и валидация
  - Совместная оценка
  - Аудит
- Организационные
  - Управление
  - Создание инфра-структуры
  - Усовершенствование
  - Обучение

# Waterfall Model



# V-shaped Model



# Incremental Model

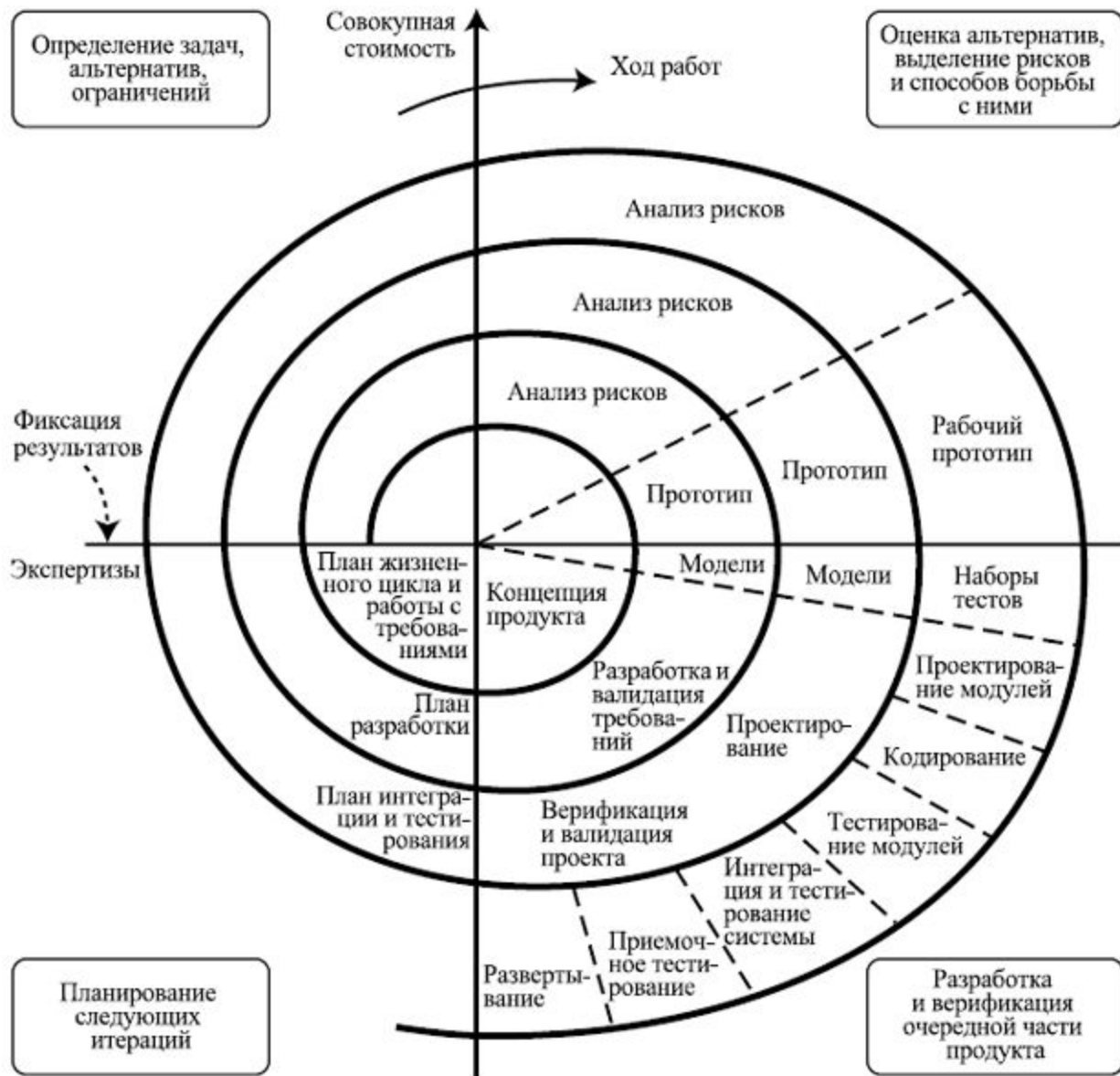


<sup>1</sup> План управления программным проектом (SPMP)

<sup>2</sup> Проектная документация программного обеспечения (SDD)

<sup>3</sup> Спецификация требований к программному обеспечению (SRS)

# Spiral Model





# Discussion

## Издержки процессов разработки

Фактор	Чистый водопадный процесс	Итеративные процессы	
		Спиральный	Инкрементальный
Легкость контроля документации	Легче	Тяжелее	Тяжелее/Средне (пояснение 1)
Возможность взаимодействия с заказчиком	Тяжелее	Легче	Легче
Поддержание хорошего проектирования	Средне/Легче	Легче (пояснение 2)	Тяжелее
Сбор метрических данных, собранных в ходе проекта	Тяжелее	Средне/Легче	Средне/Легче

1. Инкрементальный процесс осуществим, если документация изначально полна и непротиворечива. Если документация полна и непротиворечива, то относительно небольшие шаги разработки достаточно легко документируются. При этом команда разработчиков получает прекрасную возможность попрактиковаться в обновлении документации, так как процесс повторяется много раз.
2. Шаги спиральной разработки достаточно немногочисленны, что позволяет проектировать на весьма высоком уровне, но в то же время их достаточно много, чтобы обеспечить проектировщикам растущее понимание проблем проекта. Это преимущество объясняет широкое использование спирального метода.

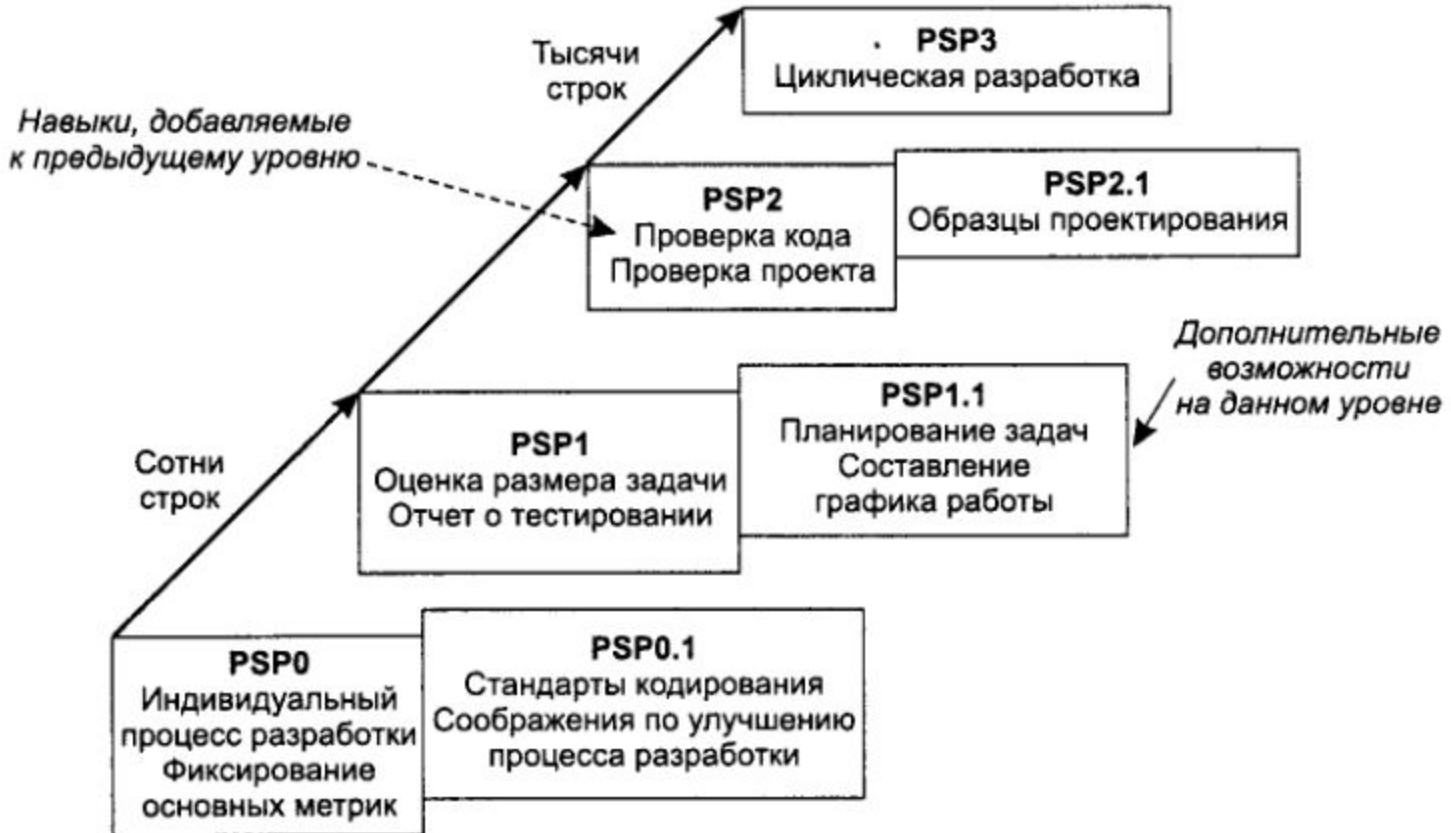
# Prototyping Model



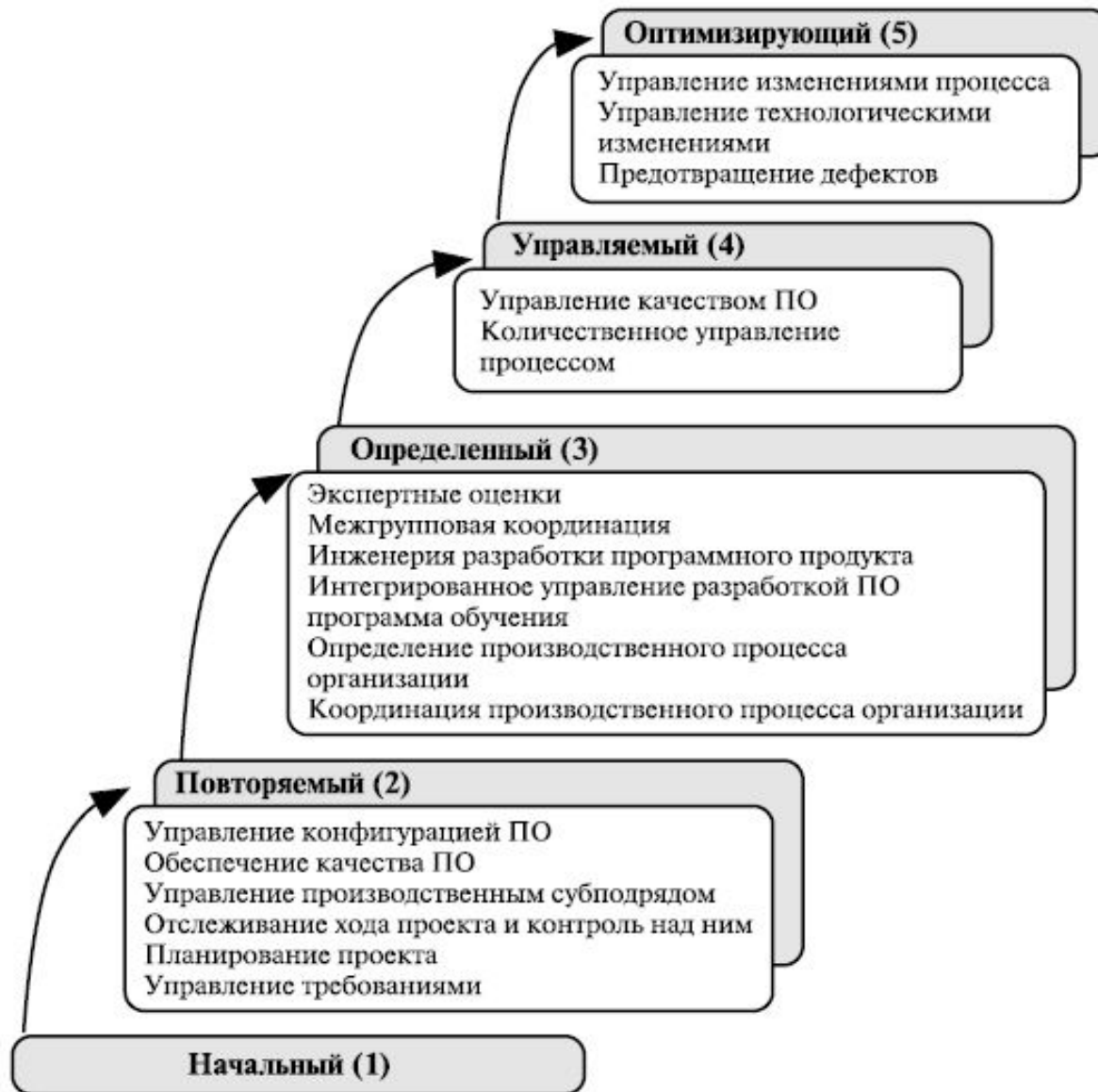
# RAD (Rapid Application Development)



# PSP (Personal Software Process)



# CMM (Capability Maturity Model)



# People

## Stakeholders



Senior Manager

Project Manager



End User

Customer

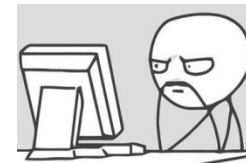


## Team Leader



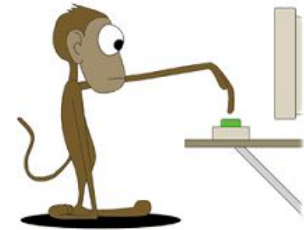
## Software Team

Business Analyst



Junior Developer  
Middle Developer  
Senior Developer

Tester



Quality Assurance

# Product Scope

Аспекты продукта и предметной области должны быть исследованы в самом начале разработки.

Необходимо с самого начала:

- установить количественные показатели (такие как: максимальное время обработки запросов, максимальное число пользователей и т.д.)
- определить границы продукта (product scope):
  - Контекст (место продукта в общей прог. системе или контексте бизнеса)
  - Информация (какие объекты требуются в качестве входа и выхода продукта)
  - Функционал и производительность

# Product Artifacts



Установочные файлы

setup.msi



Исполняемые сценарии

run.bat



Конфигурационные файлы

config.ini



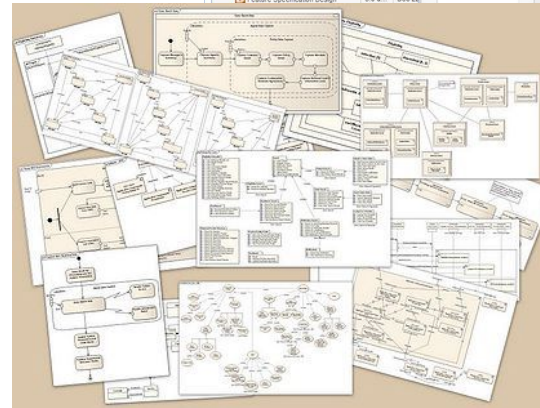
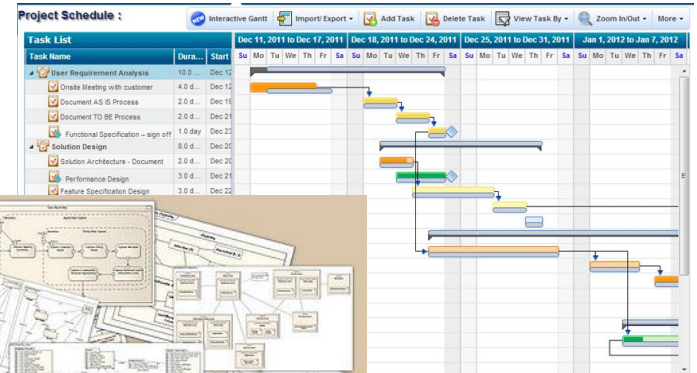
Файлы с исходными кодами  
Файлы с тестовыми сценариями

code.cpp



Разные разворачиваемые данные  
(базы данных, xml-файлы и др.)

Data



Проектная и техническая документация:

- Рабочий план проекта и расчеты рисков
- Модели и диаграммы
- Форматы сообщений, схемы базы данных
- Справочная система



History.log



README.txt



Help.chm



# Product Quality

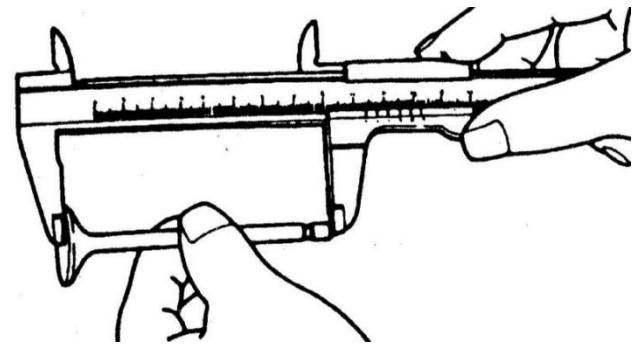


Методы достижения качества ПО:

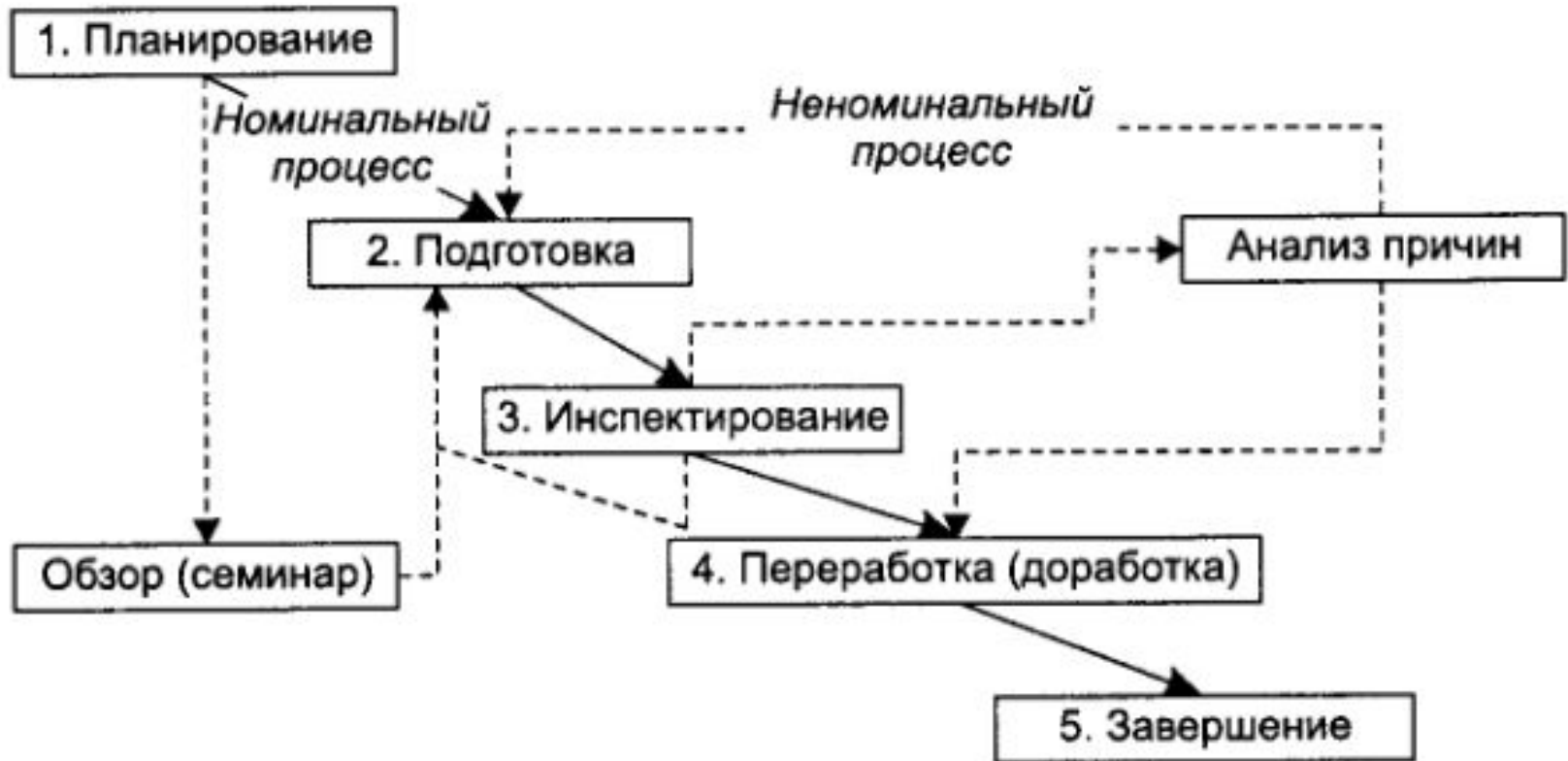
- Инспектирование
- Формальные методы
- Тестирование
- Методы управления проектом

Метрики продукта:

- ✓ Число строк кода
- ✓ Затраченное время
- ✓ Число тестов
- ✓ Плотность дефектов (число дефектов на 1000 строк кода; число дефектов на страницу документации)
- ✓ Число неустранимых дефектов
- ✓ Число компонентов системы



# Product Inspection



# Signs of Project Risk



- Команда разработчиков не понимает нужд заказчика
- Плохо определены границы продукта
- Плохо проводятся изменения
- Выбранная технология меняется
- Бизнесу необходимо подстроиться под новые условия
- Пользователи сопротивляются
- Спонсорство прекращено
- Недостаточно профессиональная команда
- Менеджеры избегают лучших практик

# Good 4P's

