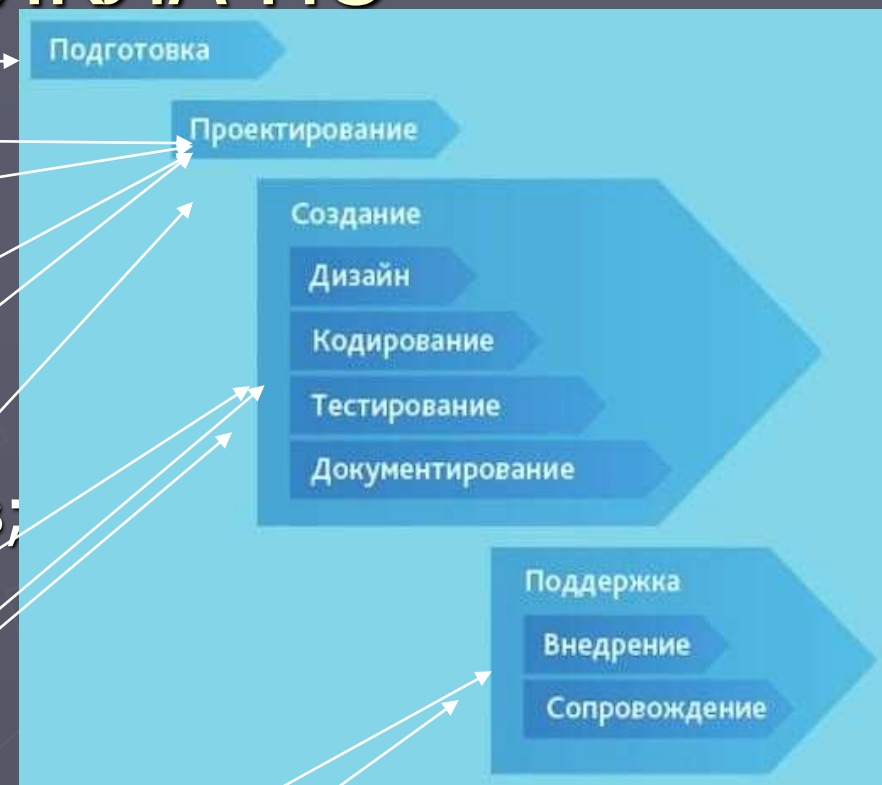


# МЕТОДОЛОГИИ РАЗРАБОТКИ ПО



# МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА ПО

1. подготовка процесса разработки;
2. анализ требований к системе;
3. проектирование системной архитектуры;
4. анализ требований к программным средствам;
5. проектирование программной архитектуры;
6. техническое проектирование программных средств;
7. программирование и тестирование программных средств;
8. сборка программных средств;
9. тестирование программных средств;
10. сборка системы;
11. тестирование системы;
12. внедрение;
13. использование и техническая поддержка.



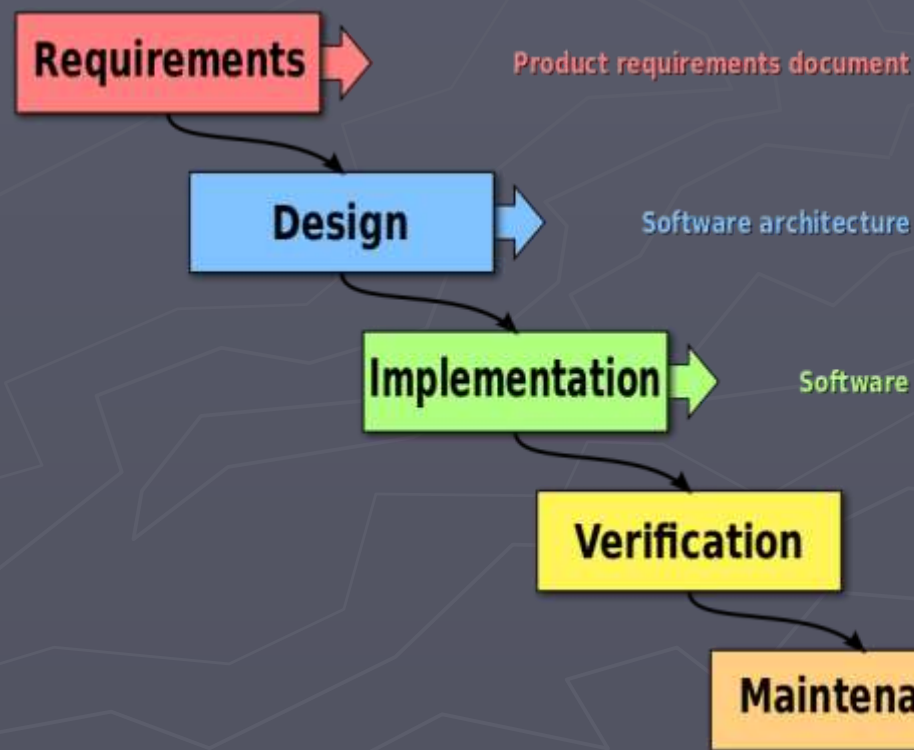
Этапы разработки ПС					Всего
Виды работ	Анализ	Проектирование	Кодирование	Отладка и тестирование	
Анализ требований и разработка спецификаций	13				13
Подготовка данных для отладки		2	2	4	8
Планирование отладки	2		2	4	8
Проектирование		13			13
Тестирование	5	5	4	11	25
Кодирование			8		8
Испытание ПО				17	17
Документирование			4	4	8
Всего	20	20	20	40	100

# Методология разработки ПО

- Методология разработки ПО – это система, определяющая порядок выполнения задач, методы оценки и контроля.

- Зависят от:

- 1) типа проекта
- 2) бюджета
- 3) сроков реализации
- 4) квалификации команды



# МЕТОДОЛОГИИ РАЗРАБОТКИ ПО

- 1) Waterfall
- 2) Rational Unified Process (RUP)
- 3) Agile - общая методология гибкой разработки
- 4) Spiral
- 5) Extreme Programming (XP)
- 6) Structured Analysis and Design Technique (SADT)
- 7) Microsoft Solutions Framework (MSF) Microsoft Operations Framework (MOF)
- 8) Rapid Application Development (RAD)
- 9) Personal Software Process
- 10) Scrum — концепция работы в условиях сорванных сроков и идеологического кризиса

# Waterfall

Стратегическое  
планирование



Анализ  
требований



Проектирование



Реализация



Тестирование и  
отладка



Эксплуатация и  
сопровождение

процесс является жестким и линейным, имеет четкие цели для каждого этапа, новая фаза начинается по завершению предыдущей, нет возврата назад

## Преимущества:

- Полная и согласованная документация на каждом этапе;
- Простота использования;
- Стабильные требования.
- Бюджет и дедлайны заранее определены

## Недостатки:

- Большое количество документации;
- Не очень гибкая система;
- Тестирование начинается на последних стадиях

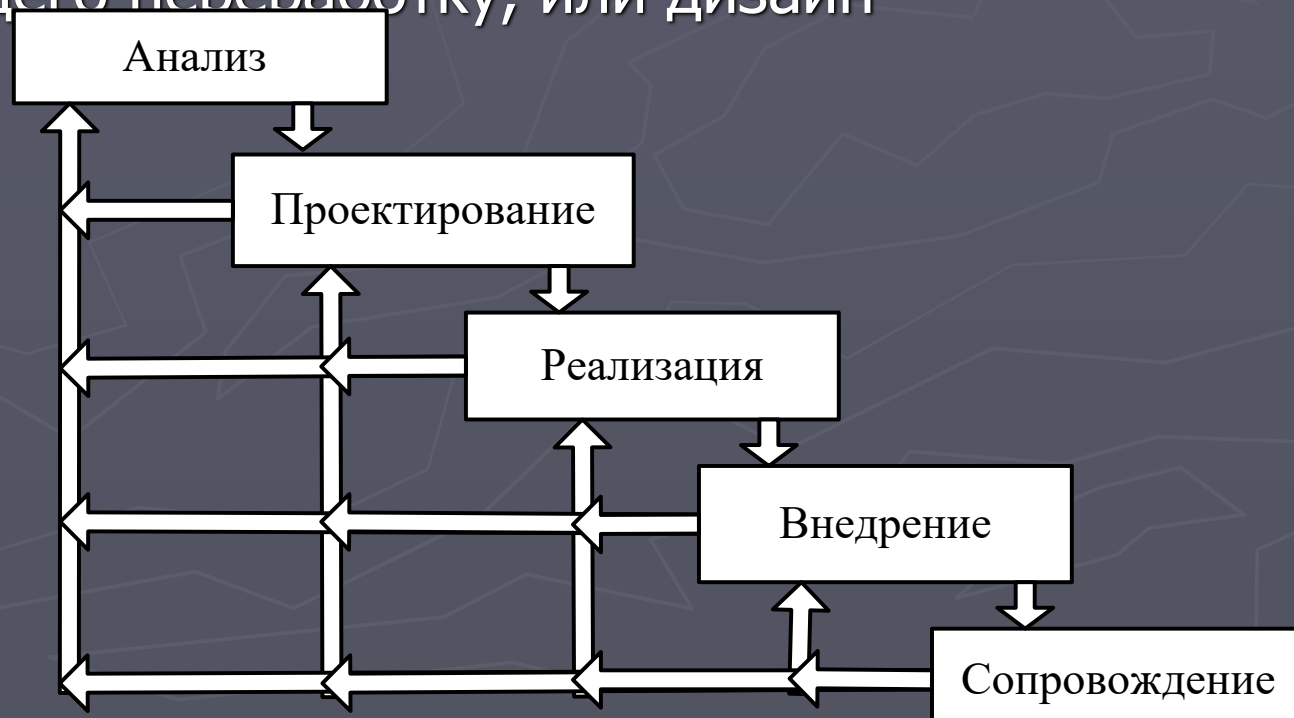
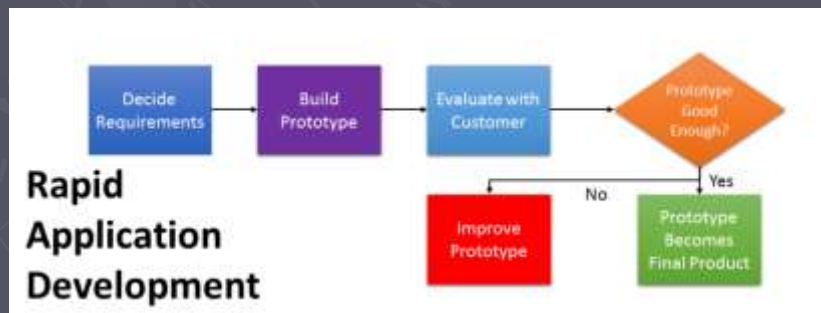


# Rapid Application Development (RAD)

Методология быстрой разработки приложений

## Принципы

- ▶ Использование фокус-групп для сбора требований.
- ▶ Прототипирование и пользовательское тестирование
- ▶ Повторное использование программных компонентов.
- ▶ Использование плана, не включающего переработку, или дизайн следующей версии продукта.
- ▶ Проведение неформальных совещаний по запросу одной из сторон.



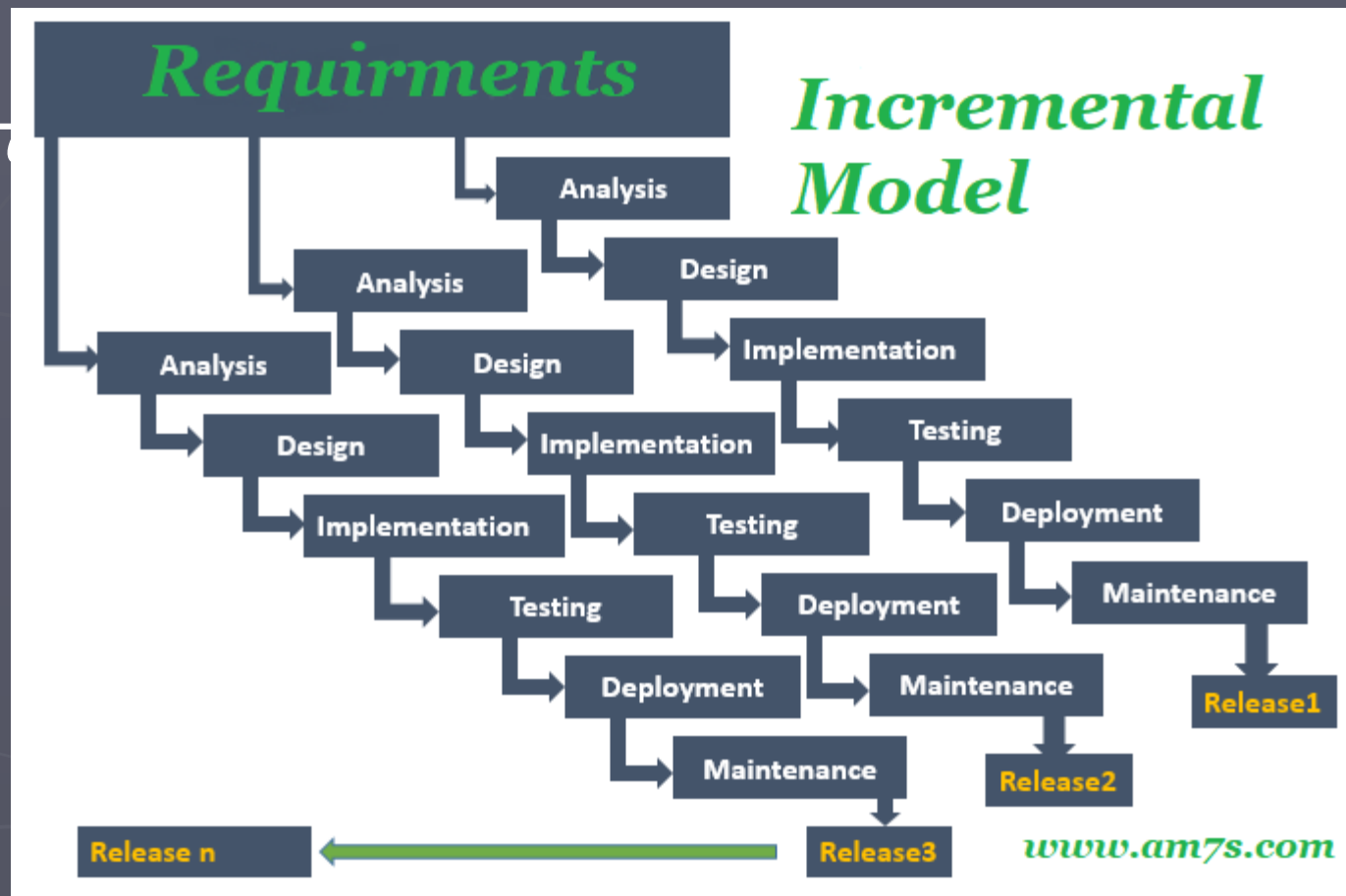
# Incremental Model

## Преимущества модели

- Быстрый выпуск минимального продукта
- Постоянное тестирование пользователями

## Недостатки модели

- Переработка проекта
- Отсутствие фиксированного бюджета и сроков.



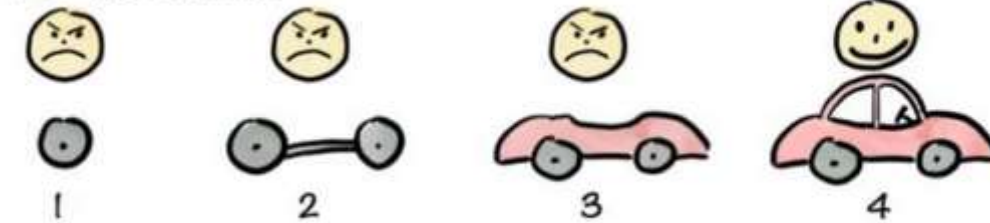


# Iterative Model

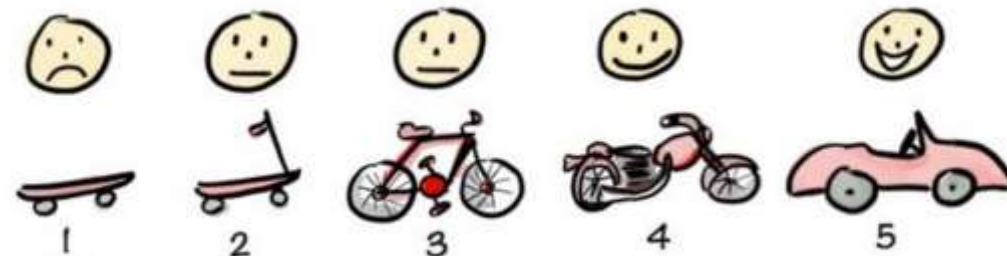
- ▶ – определение и анализ требований;
- ▶ – дизайн и проектирование – согласно требованиям. Причем дизайн может как разрабатываться отдельно для данной функциональности, так и дополнять уже существующий;
- ▶ – разработка и тестирование – кодирование, интеграция и тестирование нового компонента;
- ▶ – фаза ревью – оценка, пересмотр текущих требований и предложения дополнений к ним.



Not like this....



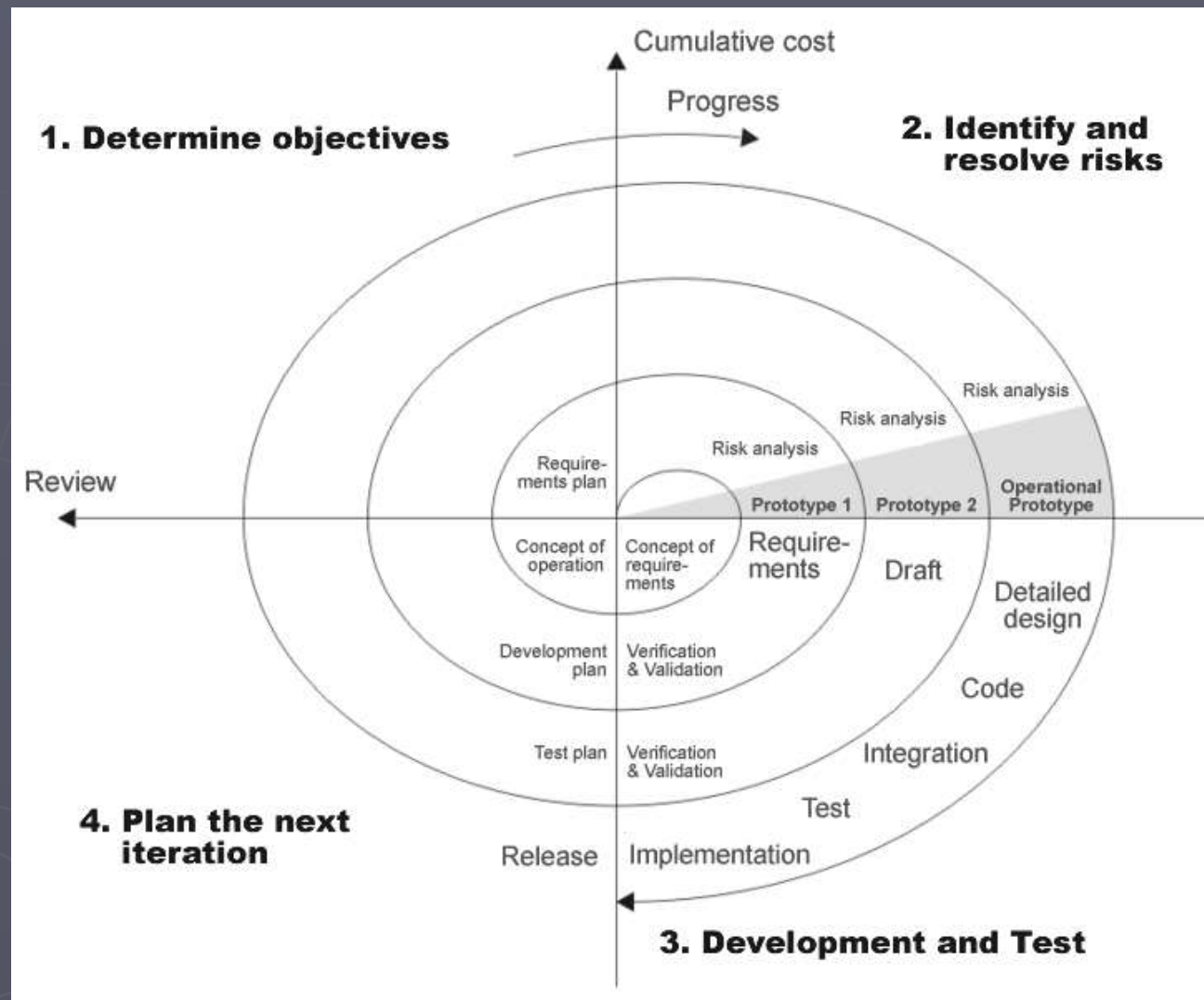
Like this!



# Спиральная модель - Spiral

Спиральная модель предполагает 4 этапа для каждого витка:

- планирование;
- анализ рисков;
- конструирование;
- оценка результата и при удовлетворительном качестве переход к новому витку.



# «V-Model» (validation and verification)

Применяется

- ▶ Если требуется тщательное тестирование
- ▶ Для малых и средних проектов, где требования четко определены и фиксированы



# Rational Unified Process (RUP)

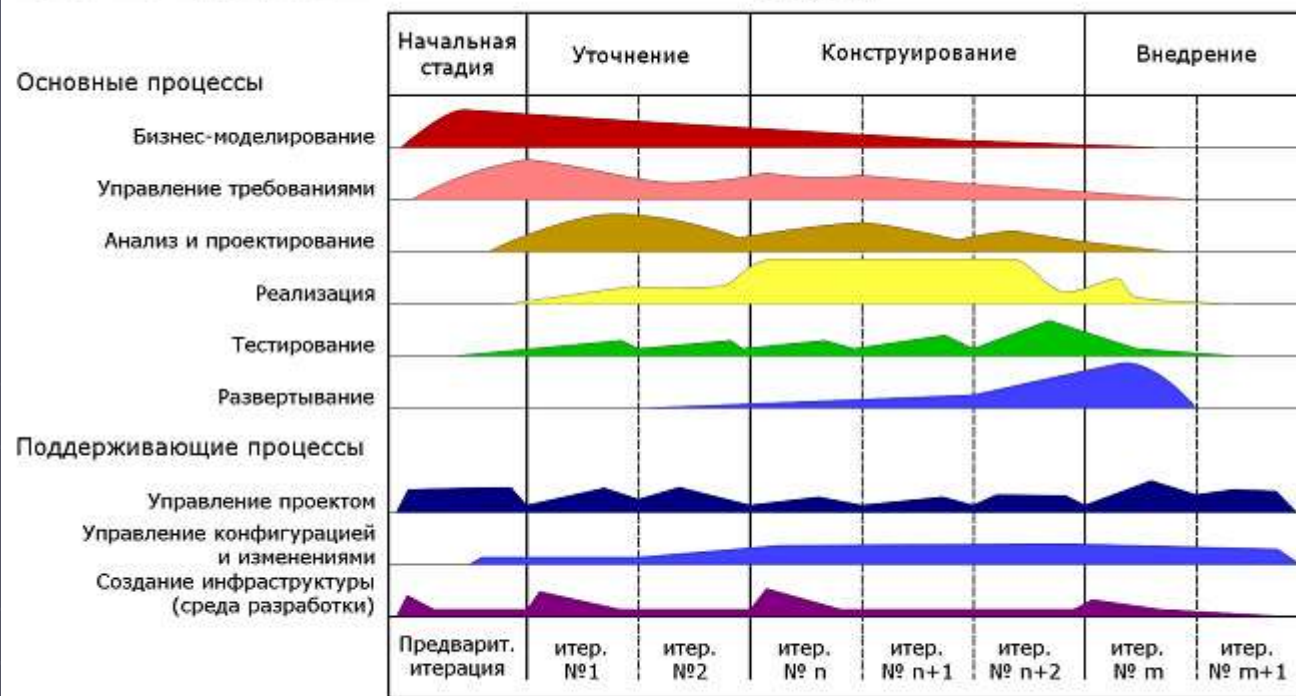
рациональный унифицированный процесс, созданный в 1996 г

Rational Гради Буч, Айвар Якобсон и Джим Румбах

- Учитывает изменяющиеся требования.
- Интеграция функций происходит постепенно, то есть каждая «деталь» проходит цикл разработки, проверки и внедрения в проект.
- Ранний выпуск продукта.
- Повторное использование.
- Постоянное обучение.
- Постоянное улучшение продукта.

Рабочие процессы

Стадии

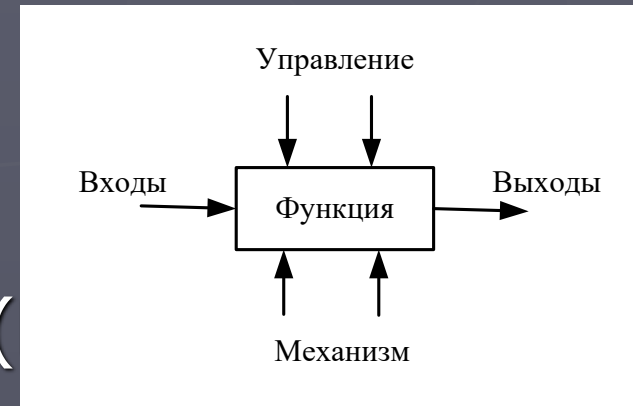


Итерации



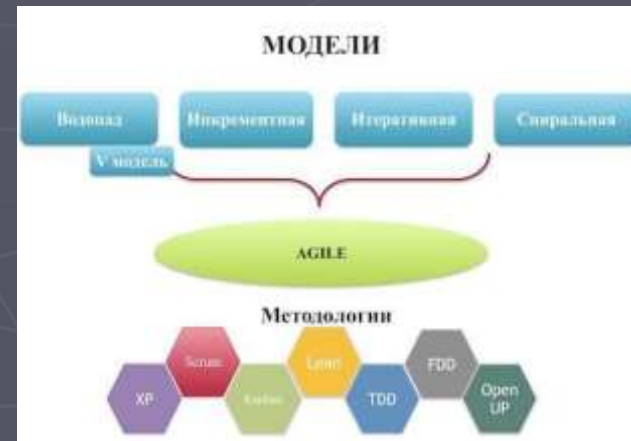
# Structured Analysis and Design Technique (SADT)

- ▶ методология структурного анализа и проектирования
- ▶ Диаграммы IDEF0
- ▶ диаграмм потоков данных IDEF3 (DFD – Data Flow Diagrams)
- ▶ диаграмм «сущность - связь» IDEF1 (Relationship Diagrams)
- ▶ диаграмм переходов состояний IDEF2 (STD – State Transition Diagrams)



# Agile (модель гибкой разработки ПО)

- ▶ экстремальное программирование (Extreme Programming, XP);
- ▶ бережливую разработку программного обеспечения (Lean);
- ▶ разработку, управляемую функциональностью (Feature-driven development, FDD);
- ▶ разработку через тестирование (Test-driven development, TDD);
- ▶ методологию «чистой комнаты» (Cleanroom Software Engineering);
- ▶ итеративно-инкрементальный метод разработки (OpenUP);
- ▶ методологию разработки Microsoft Solutions Framework (MSF);
- ▶ метод разработки динамических систем (Dynamic Systems Development Method, DSDM);
- ▶ фреймворк для управления проектами Scrum;
- ▶ метод управления разработкой Kanban.

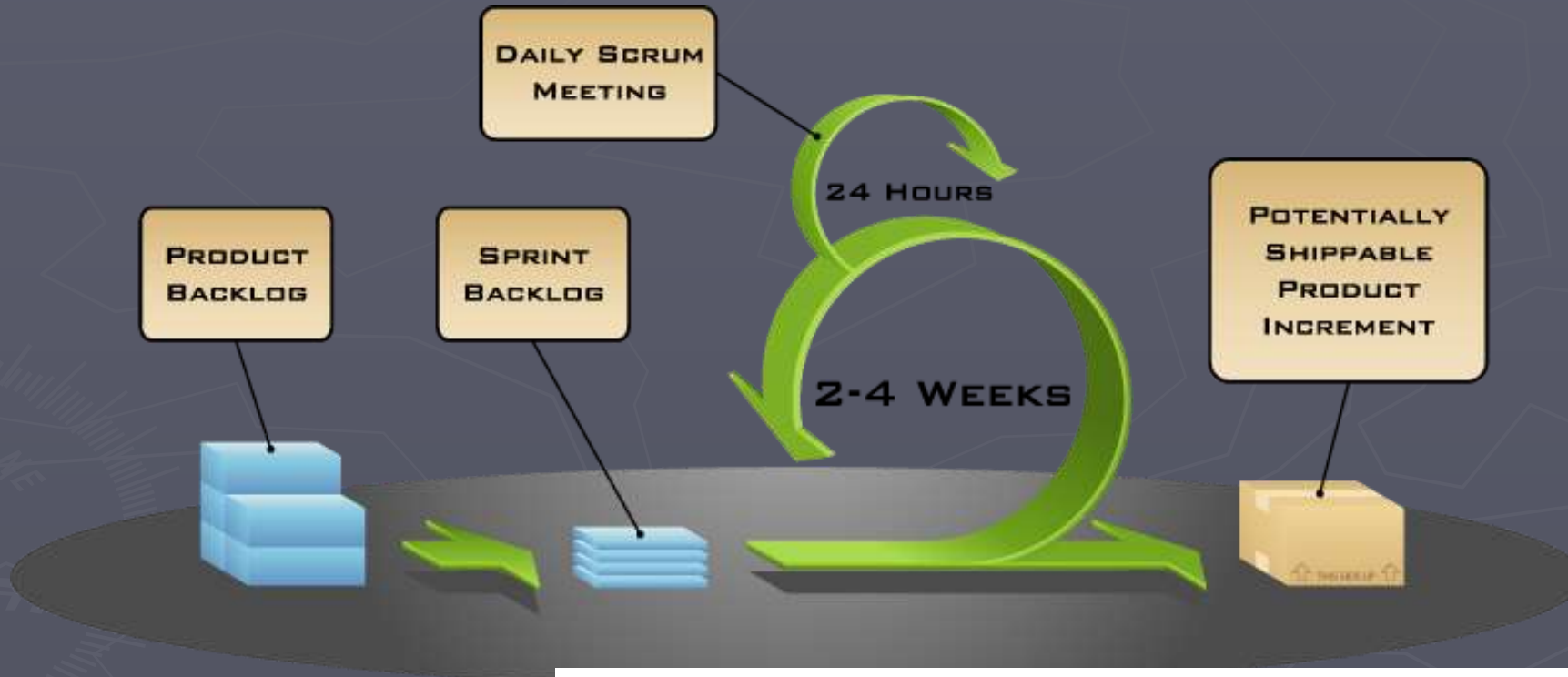




Характеристики проекта	Традиционные модели	Гибкие методологии
Подход	Прогнозирующий	Адаптивный
Критерии успеха	Следование плану	Ценность для бизнеса
Риски	Риски определены	Риски не определены
Контроль	Легко контролировать	Зависит от профессионального уровня специалистов
Заказчики	Низкая вовлеченность	Высокая вовлеченность
Документация	Детальная с начала проекта	Доработка по мере развития проекта
Требования	Известны заранее, стабильны	Не всегда известны заранее, легко изменяемы
Команда проекта	Включение новых специалистов на любом этапе	Опытные специалисты, стабильный состав
Рефакторинг (изменение кода)	Дорого	Недорого

# Agile (модель гибкой разработки ПО)

метод гибкой разработки программного обеспечения, предполагающий большое количество итераций. Документ Agile Manifesto описывает 4 идеи и 12 принципов гибкого подхода,



- Неформальные отношения важнее задокументированных.
- Работающий продукт — главная оценка прогресса.

# Agile/Scrum

- ▶ Scrum-команда – это команда работающая над проектом(разработчики, тестеры, дизайнеры).
- ▶ Scrum-мастер – человек, который следит, чтобы соблюдались принципы скрама.
- ▶ Product owner – заказчик.

- ИТИНГИ:

- Stand-up – короткий митинг, проводится каждый день, задачи должны быть выполнены за следую
- Плэннинг –
- Ретроспектива – проводится в конце спринта

# Agile/Kanban

## Преимущества:

- Простота использования.
- Наглядность.(помогает в нахождении узких мест, упрощает по
- Высокая вовлеченность команды в
- Высокая гибкость в разработке.

## Недостатки:

- Нестабильный список задач.
- Сложно применять на долгосрочных проектах.
- Отсутствие жестких дедлайнов.

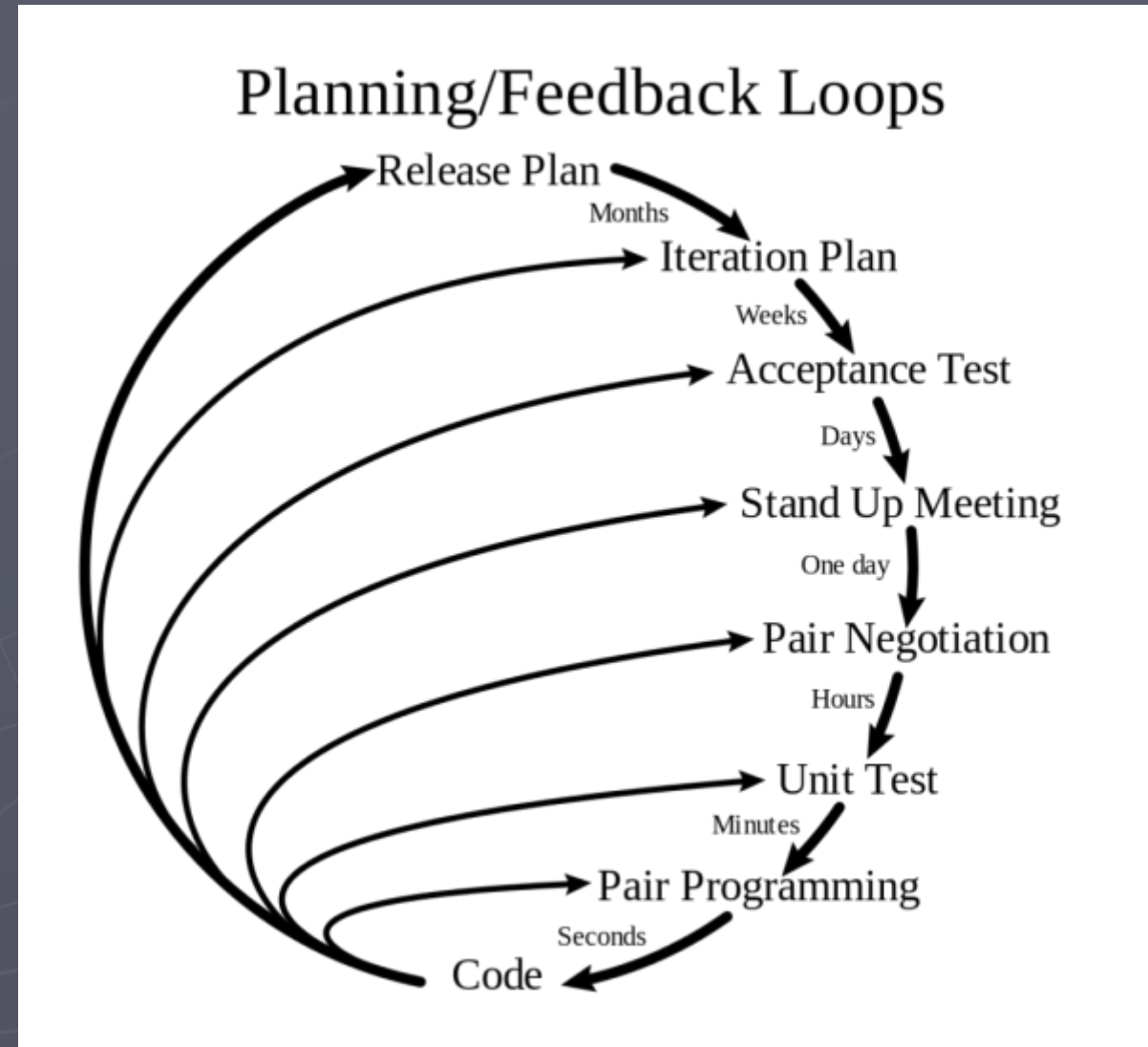


# Extreme Programming (XP)

- ▶ экстремальное программирование - возможность вести разработку в условиях постоянно меняющихся требований

Do the Simplest Thing that Could Possibly Work» («Ищите самое простое решение, которое может сработать»)

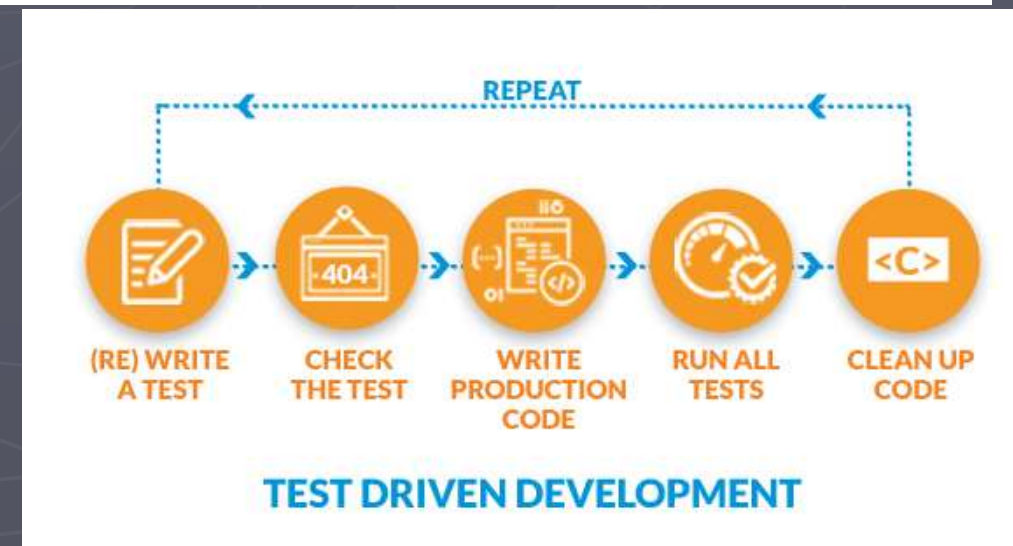
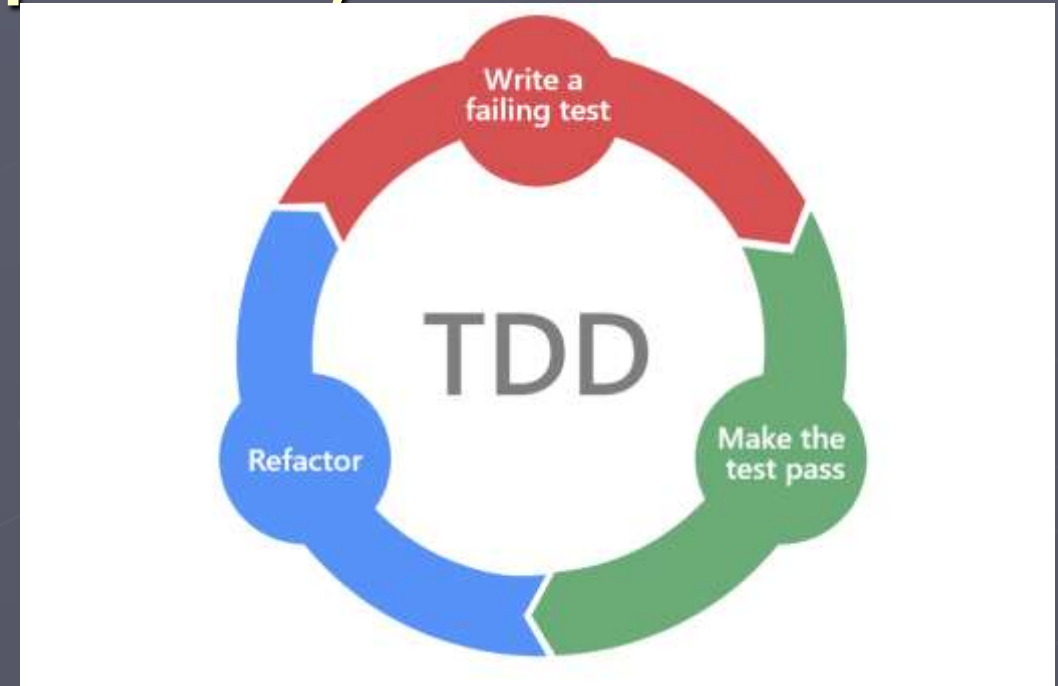
YAGNI («You Aren't Going to Need It» — «Это вам не понадобится»).





# Test-driven development, TDD

- ▶ **Цикл разработки по TDD**
- ▶ Добавить тест для новой (еще не реализованной) функциональности или для воспроизведения существующего бага
- ▶ Запустить все тесты и убедиться, что новый тест не проходит
- ▶ Написать код, который обеспечит прохождение теста
- ▶ Запустить тесты и убедиться, что они все прошли успешно
- ▶ Заняться рефакторингом и оптимизацией
- ▶ Перезапустить тесты и убедиться, что они все ещё проходят успешно
- ▶ Повторить цикл





# Microsoft Solutions Framework (MSF) и Microsoft Operations Framework (MOF)

Основополагающие принципы MSF

- ▶ Способствуйте открытому общению
- ▶ Работайте над общим видением
- ▶ Расширяйте полномочия членов команды
- ▶ Разделяйте ответственность
- ▶ Сотрудничайте с клиентом и сосредоточьтесь на предоставлении бизнес-ценности
- ▶ Будьте готовы к переменам и оставайтесь гибкими
- ▶ Инвестируйте в качество
- ▶ Учитесь на опыте

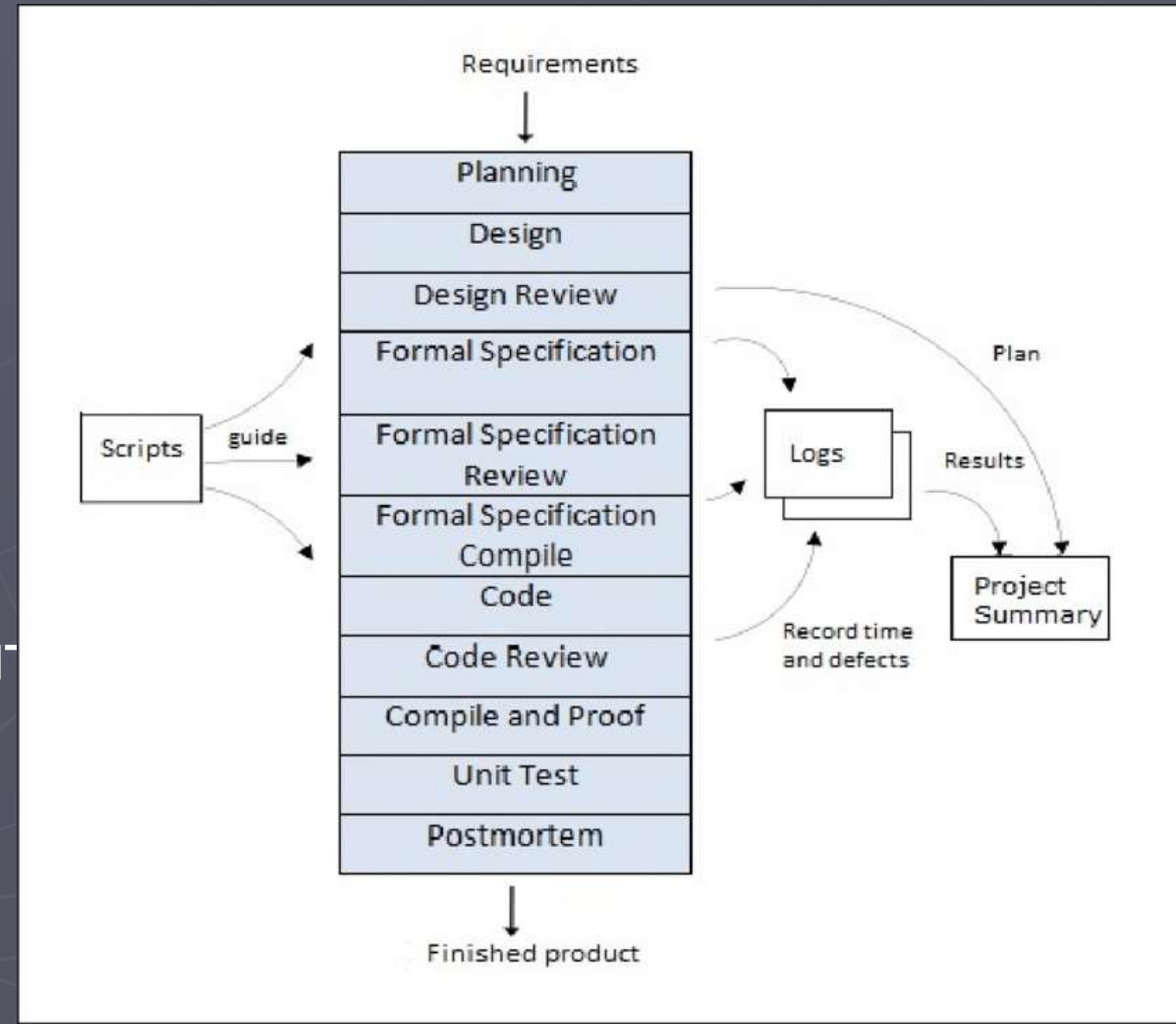


# Personal Software Process

- ▶ персональный процесс разработки программного обеспечения

## Сбор статистики

- ▶ Скрипты.
- ▶ Оценки
- ▶ Стандарты кодирования. Применение стандартов к процессу может обеспечить точные и постоянные данные.
- ▶ Формы.



# Блок-схема выбора оптимальной методологии разработки ПО

