
Технологии программирования

**Кузнецов Алексей Владимирович, к.т.н.
каб. 128.2**

Перечень рекомендуемой литературы

- **Кузнецов А.В.** Технологии и методы программирования. Часть 1. Технологии программирования. Курс-лекций, 2021. 416 с.
- **Анашкина Н.В., Петухова Н.Н., Смольянинов В.Ю.** Технологии и методы программирования. Учебное пособие. М.: Академия, 2012. 384 с.
- **Смольянинов В.Ю.** Технологии и методы программирования. Ч.1: Технология программирования. Курс лекций. М.: [А], 2011-195 с.
- **Кулямин В.В.** Технологии программирования. Компонентный подход. М.: Бином. Лаборатория знаний, 2014. 464 с.

Технология программирования

- система инженерных знаний, методов и средств, используемых для **описания (спецификации)** программ, технологий, процессов и **обеспечения процессов разработки и сопровождения** программ в течение всего периода их существования:
 - принципы и методы организации труда при разработке и сопровождении ПО
 - принципы и методы разработки и сопровождения ПО
 - средства инструментальной поддержки и автоматизации технологического процесса разработки ПО и методики использования этих средств
 - порядок описания ПО и его составных частей в процессе их разработки и сопровождения

Технология программирования

- Технология:
 - последовательность выполнения технологических операций
 - условия выполнения технологических операций
 - описание операций: исходные данные, результаты, инструкции, нормативы, стандарты, критерии и методы оценки, средства и исполнители и т.п

Технология может использоваться:

- на конкретном этапе разработки ПО
- на нескольких этапах или всем процессе разработки

Этапы развития программирования

- **1. «Стихийное» программирование**
(от первых ЭВМ до 60-х годов 20 века)
- **2. Структурный подход к программированию**
(60-70е годы 20 века)
- **3. Объектный подход к программированию**
(80-90е годы 20 века)
- **4. Компонентный подход, CASE-технологии**
(90е годы 20 века – н.в.)

Жизненный цикл программного обеспечения

Лекция 1

Тема 1: Стандарты в области разработки ПО

Жизненный цикл

Жизненный цикл (ЖЦ) ПО – период времени существования ПО с момента принятия решения о необходимости разработки и заканчивая моментом прекращения использования

Стадии жизненного цикла – главные периоды жизненного цикла, описывающие развитие ПО и достигаемые контрольные точки ЖЦ

Основные стадии ЖЦ

- Формирование требований к ПО
- Разработка ПО:
 - Проектирование архитектуры и структуры
 - Реализация
 - Интеграция
 - Тестирование
 - Отладка
- Аттестация ПО
- Внедрение ПО
- Эксплуатация ПО
- Сопровождение ПО



Процесс
создания
ПО

Участники процесса

- **Заказчик**
- **Разработчик**
 - Проектировщик
 - Разработчик пользовательского интерфейса
 - Разработчик модулей
 - Программист
 - Интегратор
 - Разработчик тестов
 - Тестировщик
- **Пользователь**

Модель ЖЦ

- **Модель ЖЦ** – структура процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение ПО, от формирования требований до прекращения использования
- Модель описывает:
 - процессы, которые являются частью технологического цикла создания ПО
 - распределение процессов по стадиям (несколько процессов на стадию или наоборот)
 - порядок и критерии перехода от стадии к стадии
 - вспомогательные методики и практики
 - действия и роли людей, участвующих в создании ПО

Модели ЖЦ

- Каскадная модель
- Поэтапная модель
- Инкрементная модель
- Спиральная модель

Каскадная модель



Каскадная модель

Достоинства:

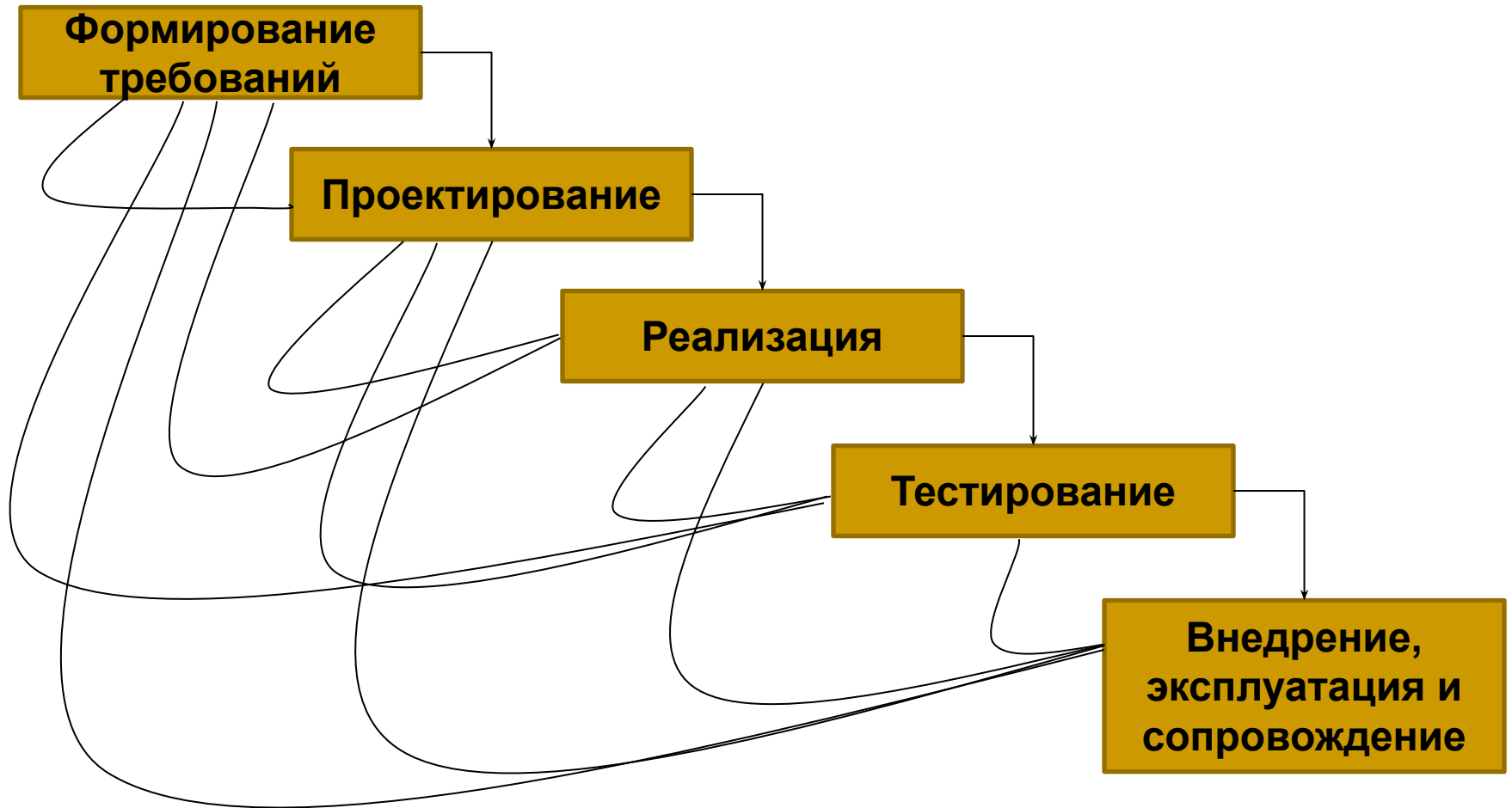
- План и временной график создания, отслеживание ресурсов, рисков, финансов
- Сохранение целостного представления системы и неизменность задач
- Готовность всех результатов к следующей стадии
- Для систем с четко сформулированными требованиями
- Низкий порог вхождения новых участников

Каскадная модель

Недостатки:

- Невозможность возврата к предыдущим стадиям
- Несоответствие необходимых и существующих требований к системе
- Результаты создания доступны только в конце работы
- Реальные проекты часто требуют отклонения от стандартной последовательности шагов
- Большое время обнаружения ошибок
- Сдвиг сроков работы или плохое тестирование при ошибках и новых задачах

Поэтапная модель



Поэтапная модель

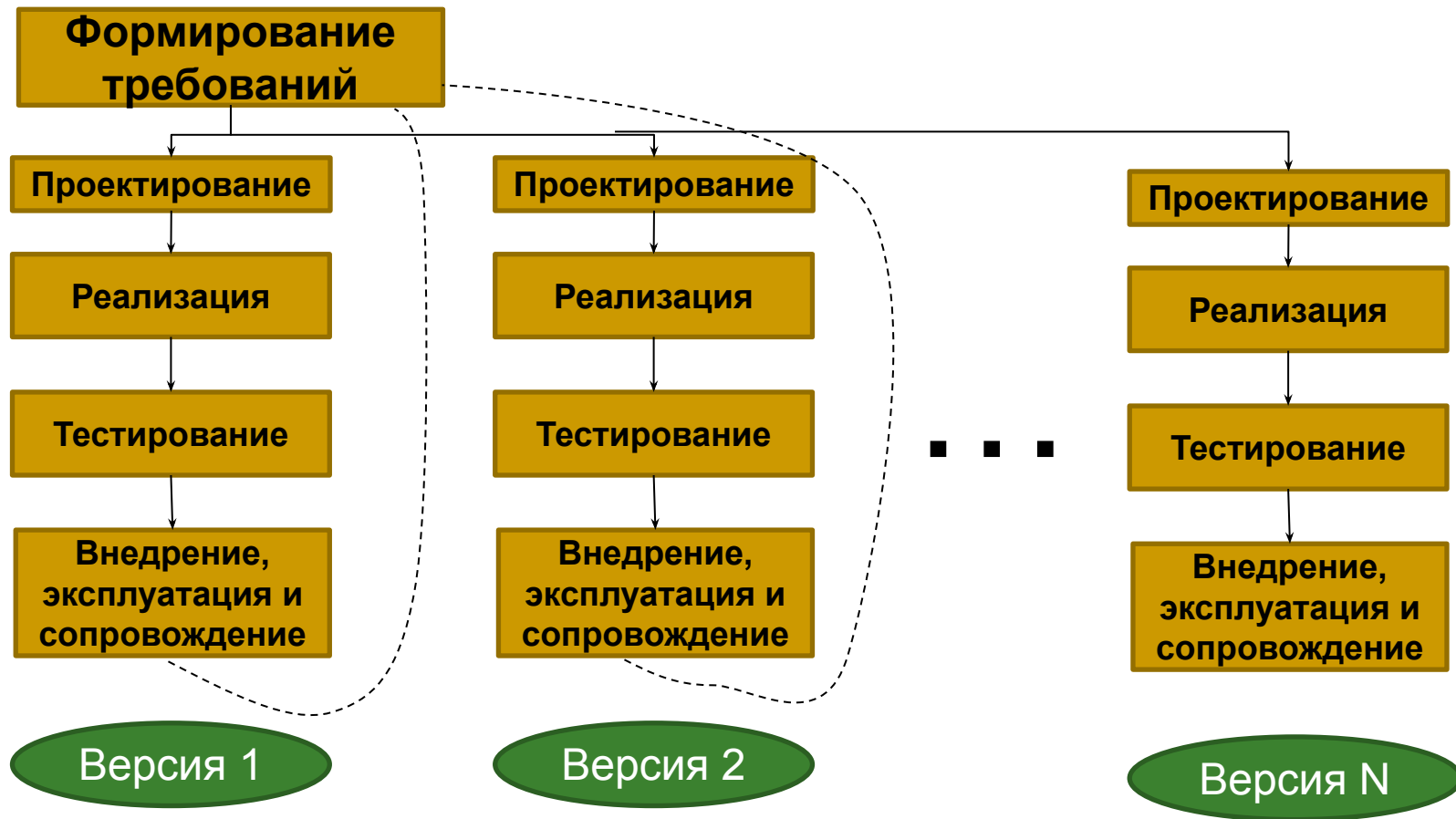
Достоинства:

- Меньшая трудоемкость по сравнению с каскадной моделью
- Возможность вернуться на несколько стадий назад и получить новую усовершенствованную версию системы
- Можно разработать сначала урезанную версию системы

Недостатки:

- Время жизни каждой стадии растягивается на весь период проектирования

Инкрементная модель



Инкрементная модель

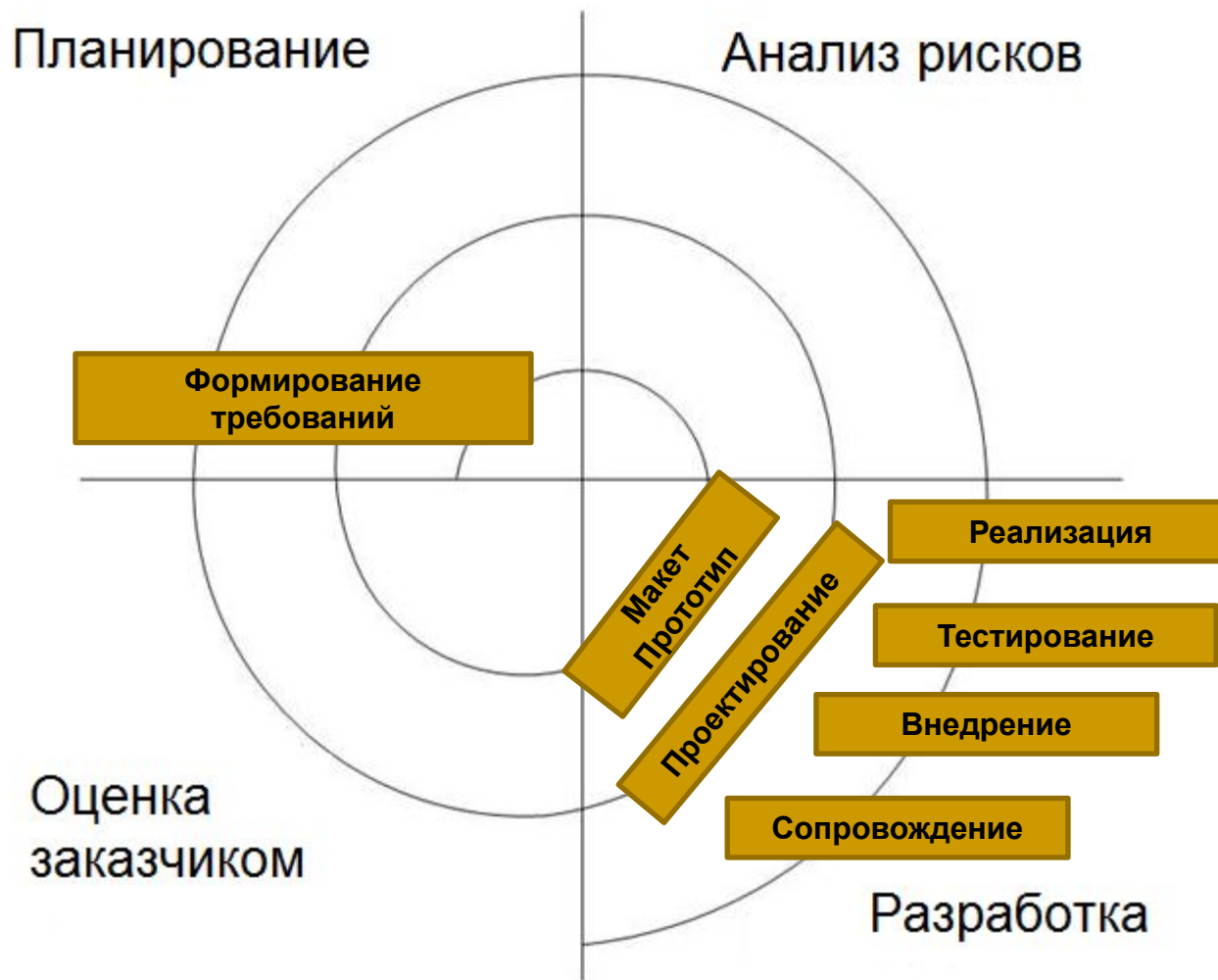
Достоинства:

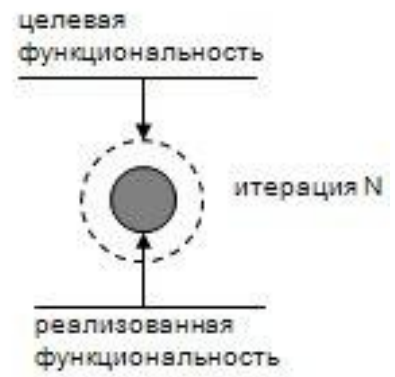
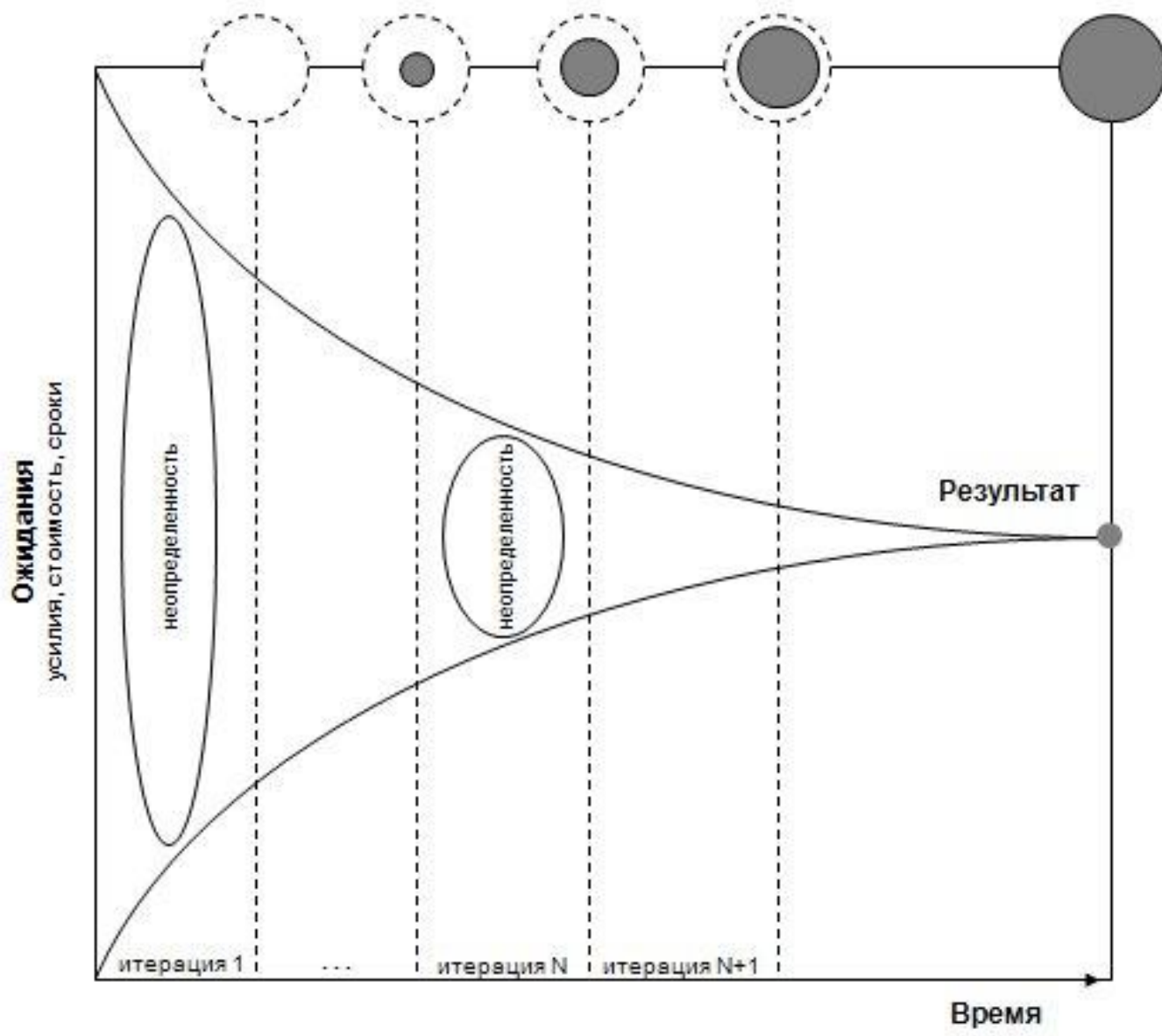
- Меньшая трудоемкость по сравнению с каскадной моделью
- Сначала ПО с урезанным функционалом, затем более полное

Недостатки:

- Разработка очередной версии начинается с начала

Спиральная модель





Риски

- Дефицит специалистов
- Нереалистичные сроки и бюджет
- Реализация функциональности на основе неточных или неполных требований
- Разработка неудачного пользовательского интерфейса
- Ненужная оптимизация
- Непрерывающийся поток изменений
- Отсутствие полной информации о внешних компонентах при интеграции
- Недостатки во внешних работах
- Недостаточная производительность
- Разрыв квалификации специалистов

Спиральная модель

Достоинства:

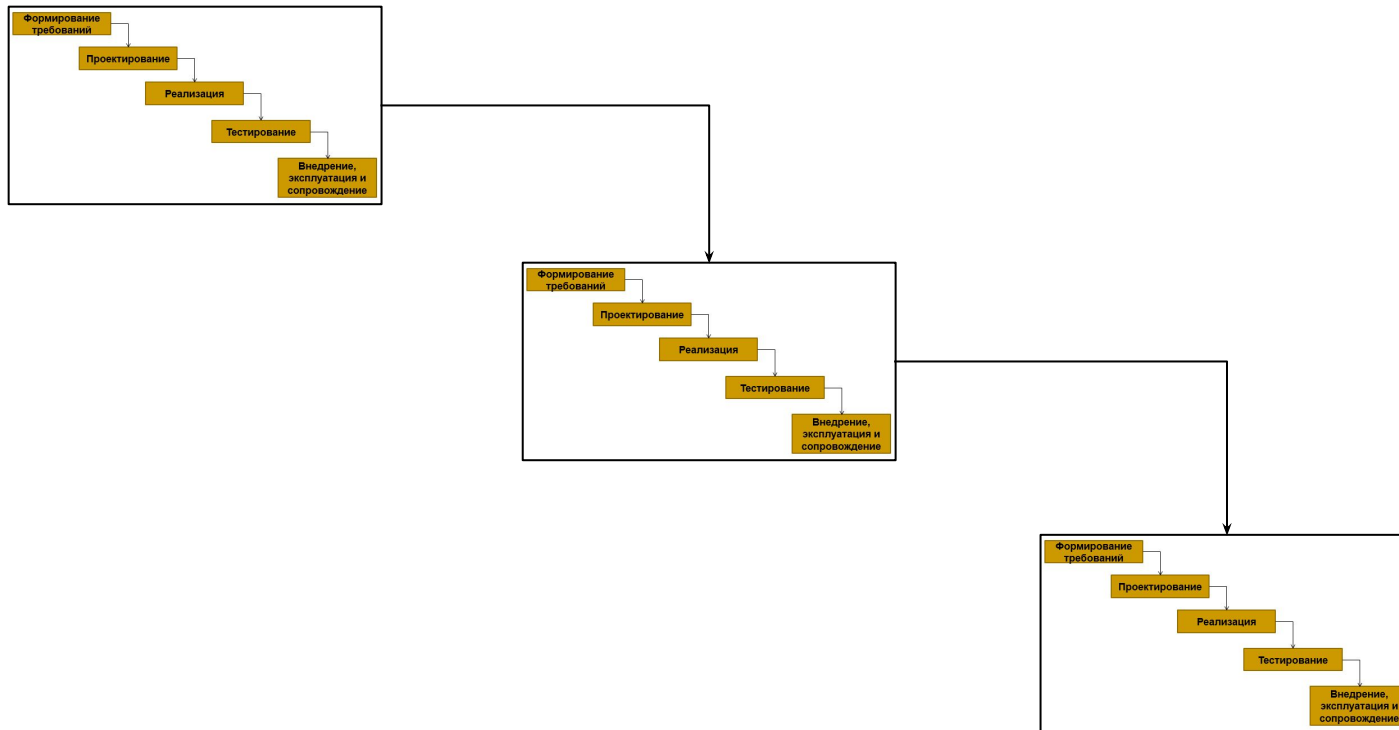
- Эволюционное представление разработки
- Мониторинг рисков
- Макетирование, моделирование, прототипы
- Ранняя доступность системы
- Изменения на поздних стадиях разработки
- Повторное использование проектных решений средств проектирования, моделей системы
- Обратная связь от пользователя

Спиральная модель

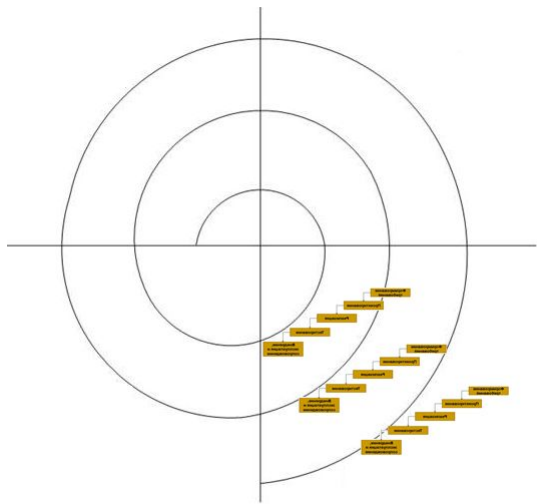
Недостатки:

- Дополнительные ресурсы для анализа
- Отдельная экспертиза для каждой итерации
- Большое количество стадий = большой объем документации
- Повышенные требования к квалификации заказчика
- Трудности контроля и управления временем окончания разработки

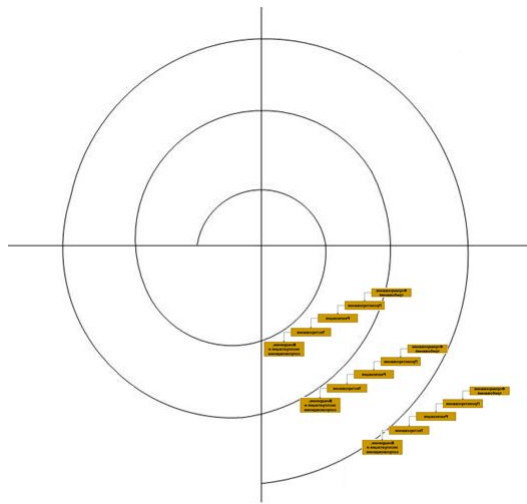
Комбинирование моделей



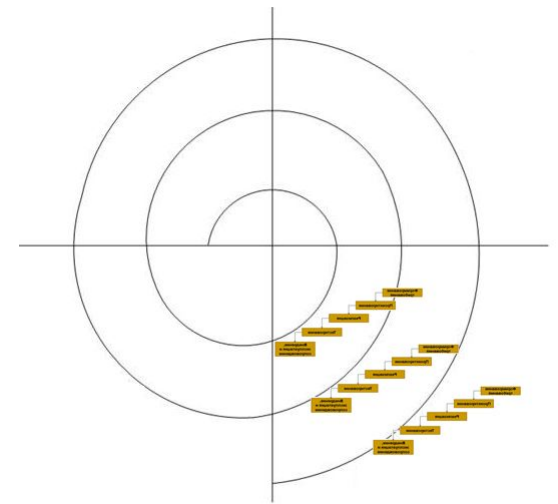
Комбинирование моделей



Подсистема 1



Подсистема 2



Подсистема 3

Стандарт SEI CMM

- CMM (Capability Maturity Model) –
Модель зрелости возможностей
 - Унифицированный подход к оценке
возможностей организации (классификация)

Уровни зрелости организации

1. Начальный
2. Повторяемый
3. Определенный
4. Управляемый
5. Совершенствующийся

Уровни зрелости организации

1. Начальный

Организация

- не имеет осознанного процесса разработки
- не планирующая свои ресурсы и возможности

Зависимость от конкретных разработчиков

Уровни зрелости организации

2. Повторяемый

Организация

- ведет учет ресурсов,
- планирует и отслеживает ход выполнения проектов и их качество,
- устанавливает правила управления проектами, основанные на имеющемся опыте

Организация обладает определенными технологиями

Уровни зрелости организации

3. Определенный

Процесс разработки и сопровождения

- стандартизован и документирован
- включает управленческие и технические процессы
- включает процессы обучения сотрудников
- координирует работы групп сотрудников
- включает экспертизы требований и результатов, управление рисками

Организация перестает зависеть от конкретных разработчиков, нет точного предсказания затрат

Уровни зрелости организации

4. Управляемый

- измеримые показатели качества продукции и результативности процессов, позволяющие предсказывать потребности в ресурсах для разработки продукции с определенным качеством
- управление качеством и процессом, основанное на показателях

Организация может точно предсказать сроки и стоимость работ, но не имеет механизмов улучшения процессов разработки

Уровни зрелости организации

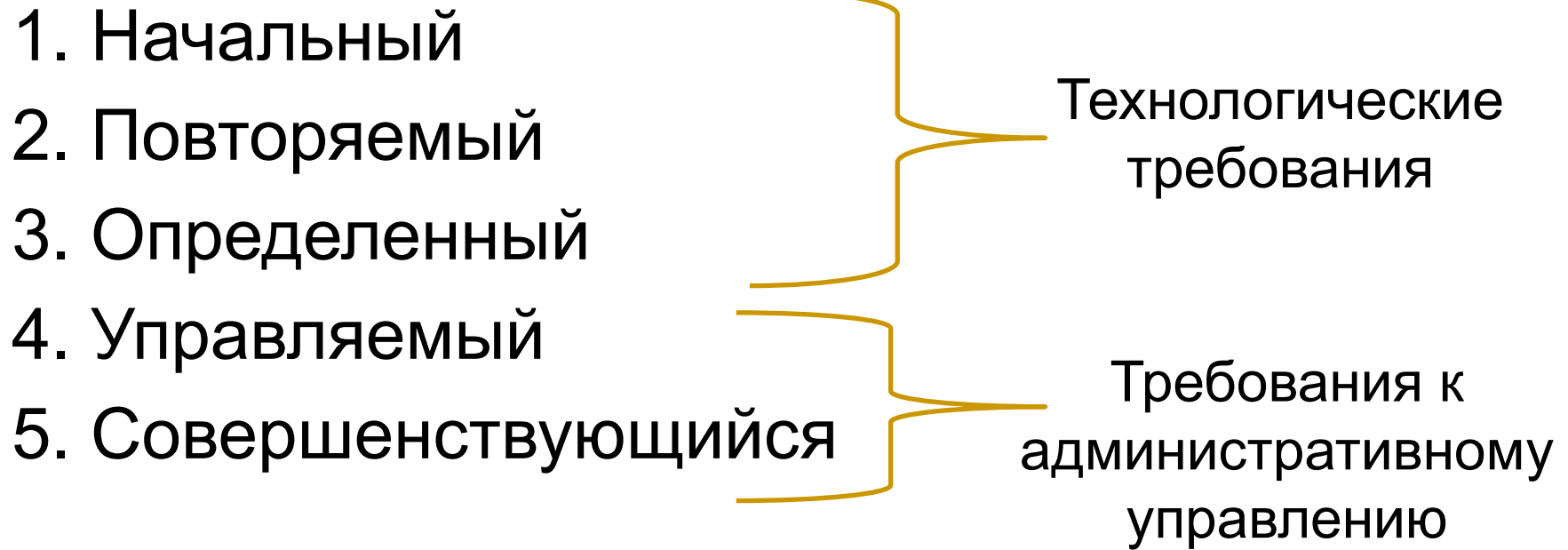
5. Совершенствующийся

- методики определения слабых мест
- поиск и оценка новых технологий разработки и обучения им персонала
- управление изменением процессов и технологий

Организация постоянно

самосовершенствует процесс разработки

Уровни зрелости организации



Уровни зрелости организации

1. Начальный

Зависимость от конкретных разработчиков

2. Повторяемый

Организация обладает определенными технологиями

3. Определенный

Организация перестает зависеть от конкретных разработчиков, нет точного предсказания затрат

4. Управляемый

Организация может точно предсказать сроки и стоимость работ, но не имеет механизмов улучшения процессов разработки

5. Совершенствующийся

Организация постоянно самосовершенствует процесс разработки

Технологические требования

Требования к административному управлению

Жизненный цикл программного обеспечения

Лекция 1

Тема 1: Стандарты в области разработки ПО

Вопросы

1. Технология программирования. Этапы развития ТП.
2. Жизненный цикл ПО. Основные стадии ЖЦ. Участники процесса разработки ПО и их участие на стадиях ЖЦ ПО.
3. Модели жизненного цикла ПО. Достоинства, недостатки, сравнение.
4. Уровни зрелости организации по стандарту SEI CMM.