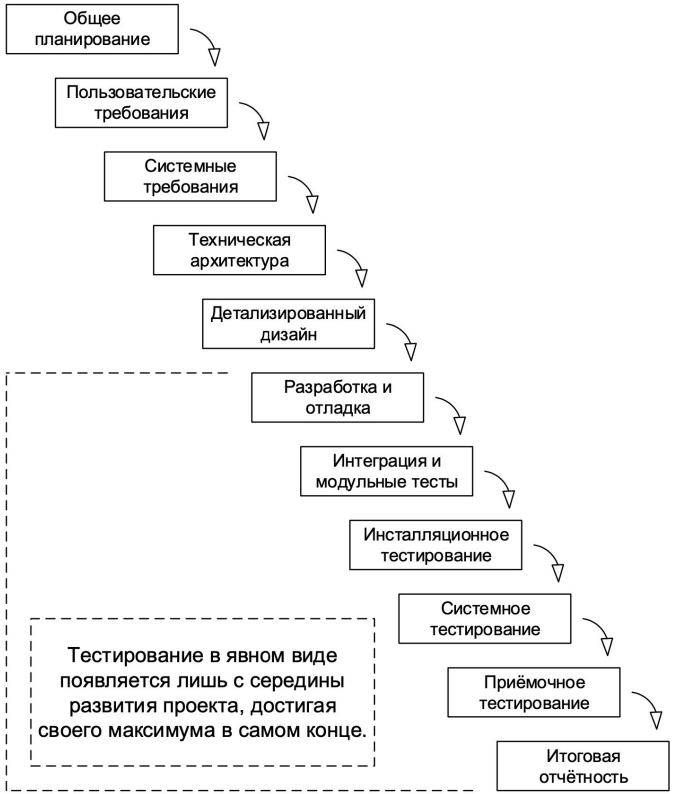
Процессы тестирования и разработки ПО

Модели разработки ПО

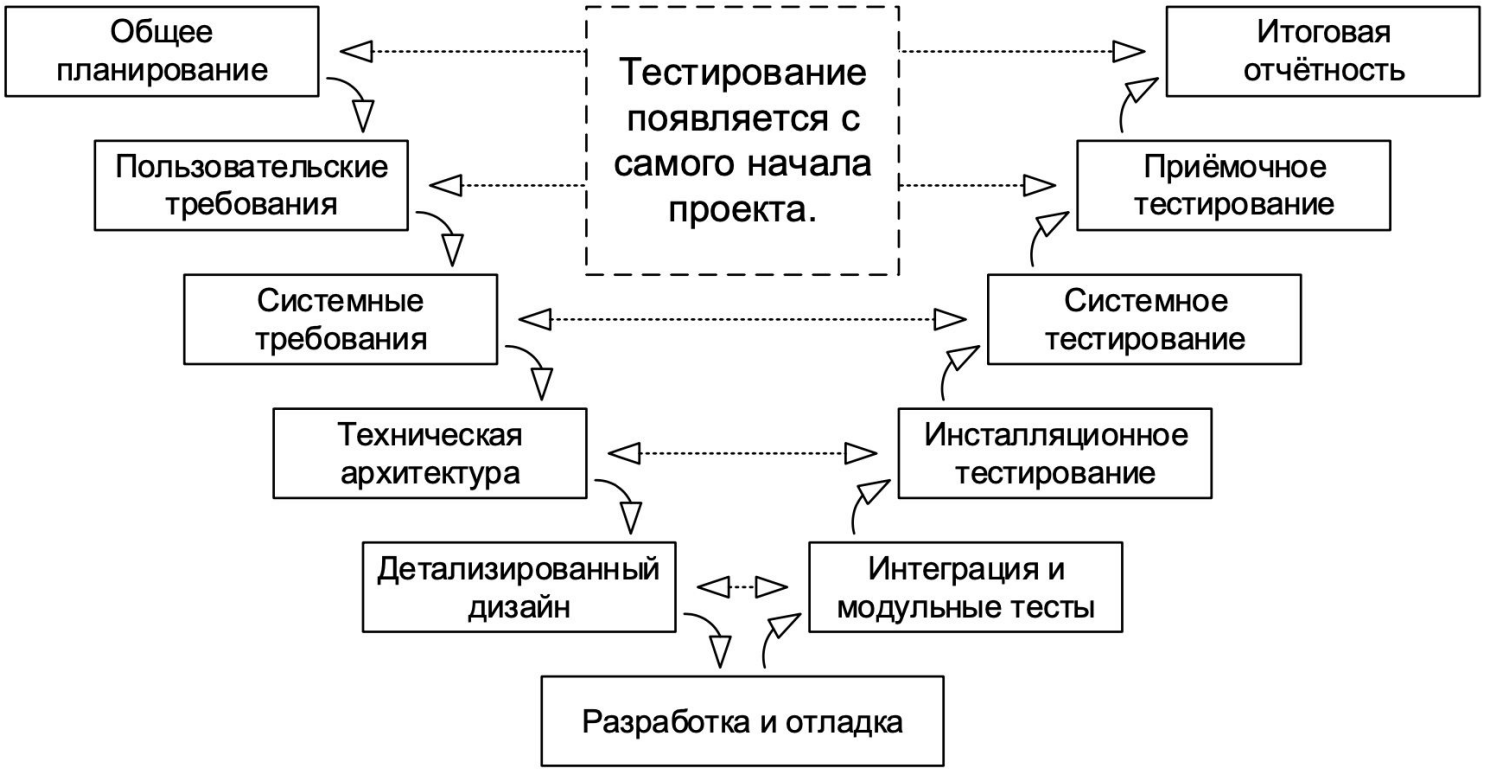
Модель разработки ПО (Software Development Model, SDM) — структура, систематизирующая различные виды проектной деятельности, их взаимо- действие и последовательность в процессе разработки ПО. Выбор той или иной модели зависит от масштаба и сложности проекта, предметной области, доступных ресурсов и множества других факторов.

- Водопадная
- V-образная
- Итерационная инкрементальная
- Спиральная
- Гибкая

Водопадная модель

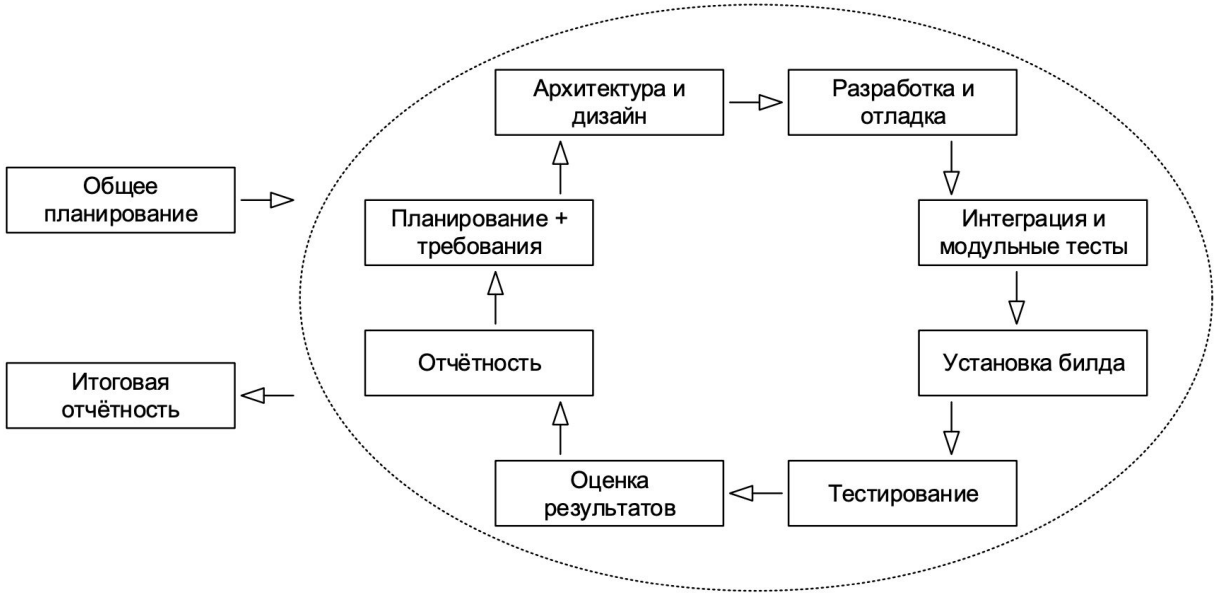


V-образная модель

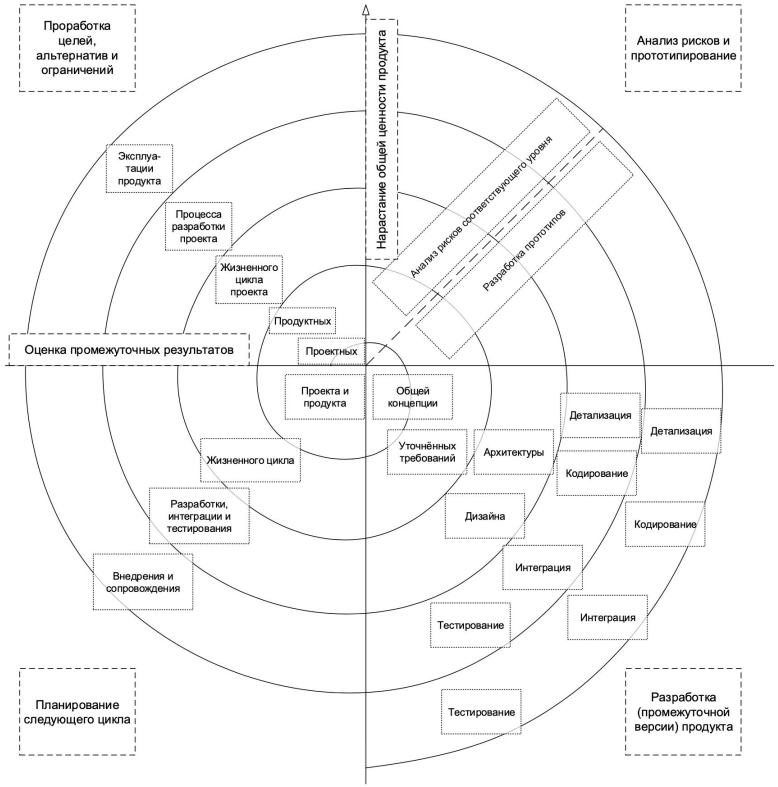


Итерационная инкрементная модель

- с точки зрения жизненного цикла - итерационная
- с точки зрения развития продукта - инкрементная



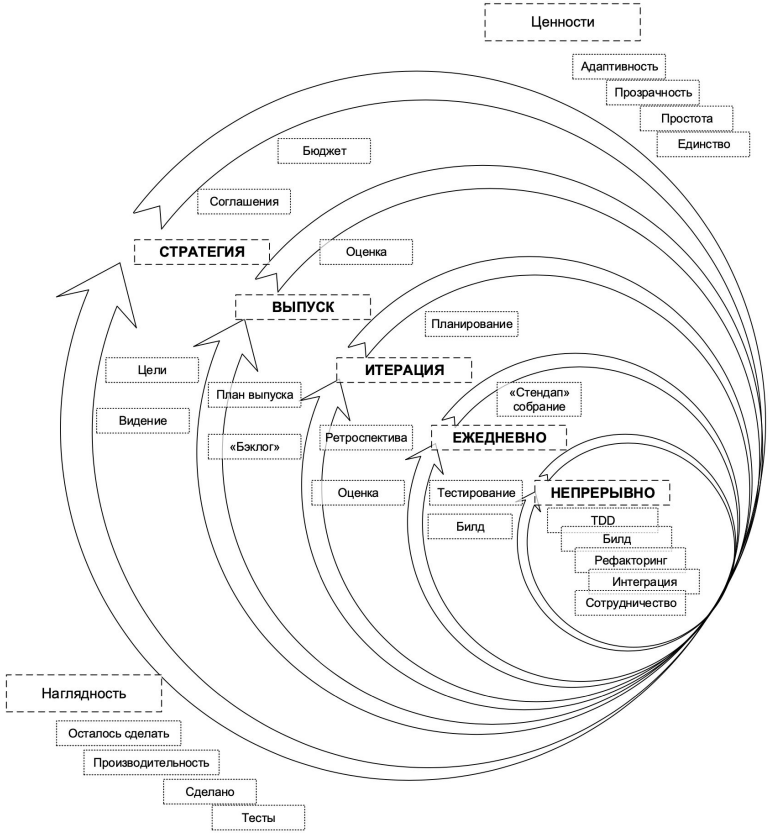
Спиральная модель



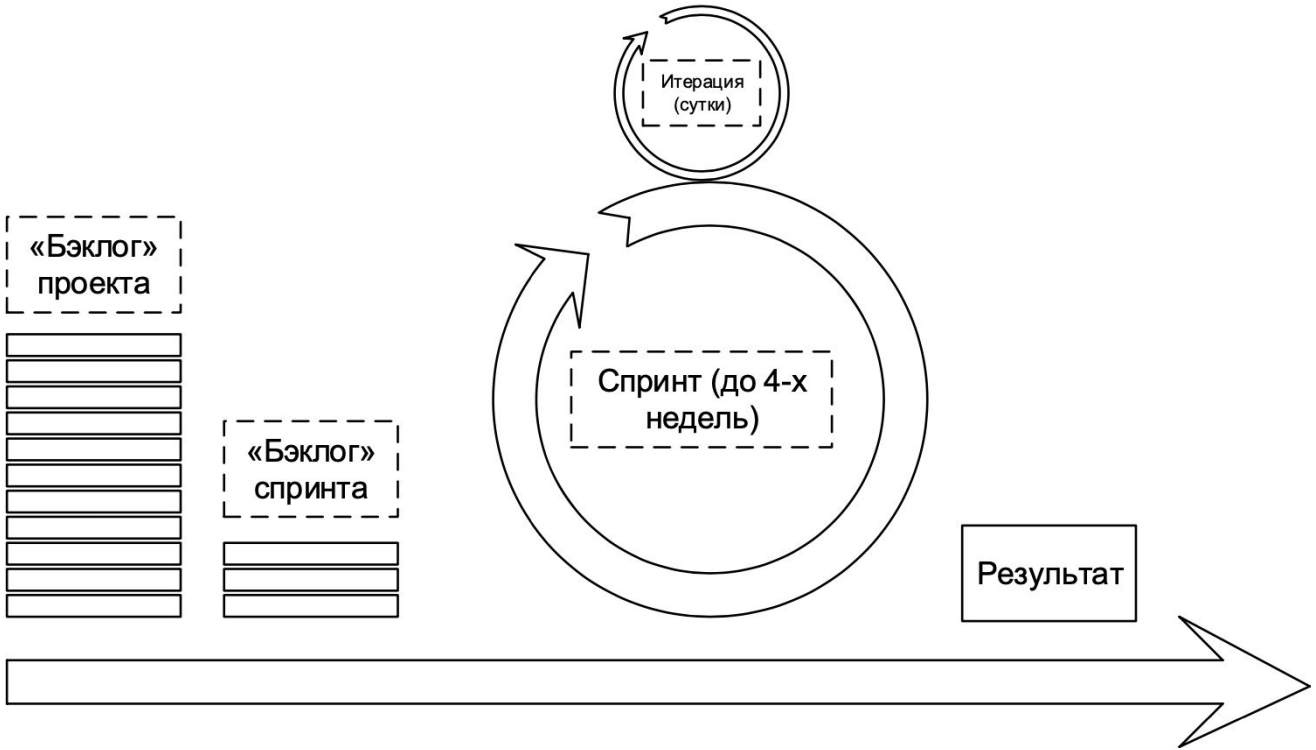
Гибкая (agile) модель

- **Люди и взаимодействие** важнее процессов и инструментов.
- **Работающий продукт** важнее исчерпывающей документации.
- **Сотрудничество с заказчиком** важнее согласования условий контракта.
- **Готовность к изменениям** важнее следования первоначальному плану.

Гибкая (agile) модель



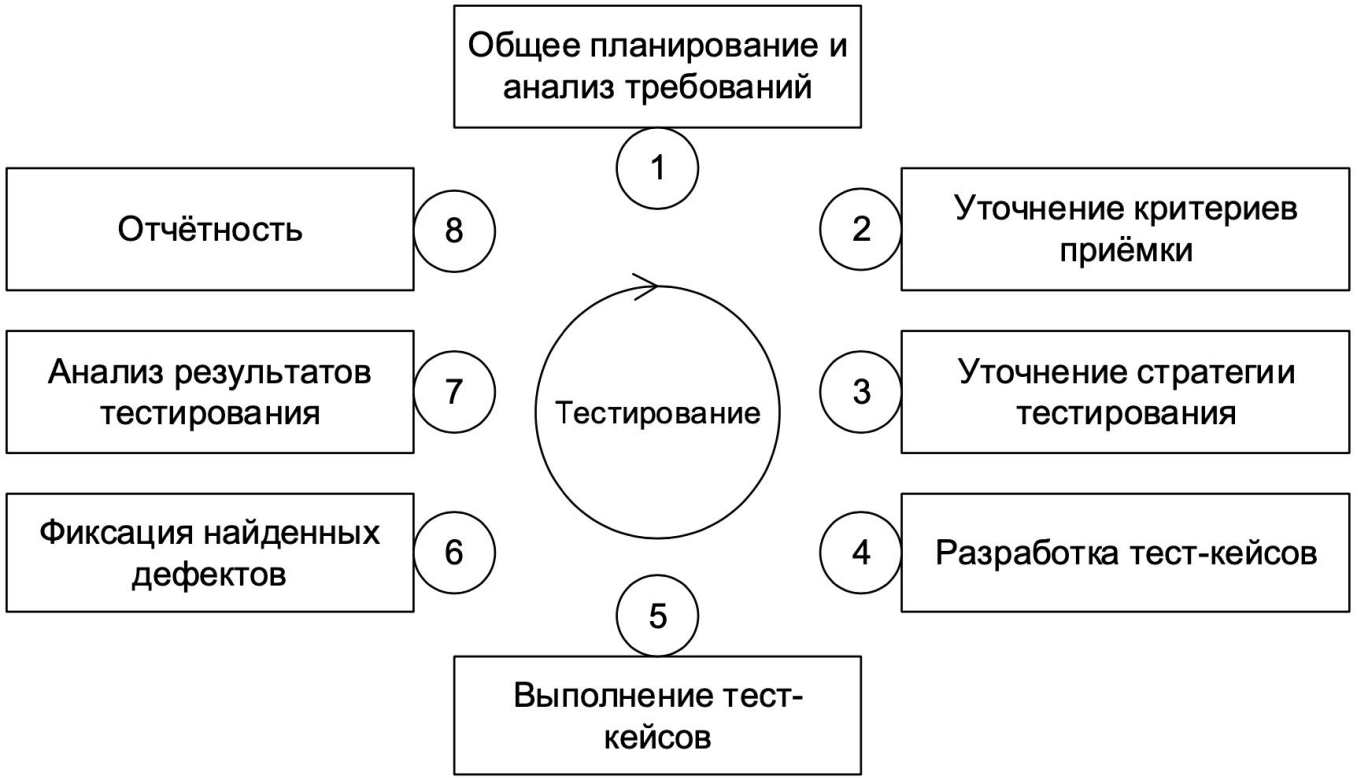
Итерационный подход в рамках гибкой модели



Сравнение моделей

Модель	Преимущества	Недостатки	Тестирование
Водопадная	• У каждой стадии есть чёткий проверяемый результат. • В каждый момент времени команда выполняет один вид работы. • Хорошо работает для небольших задач.	• Полная неспособность адаптировать проект к изменениям в требованиях. • Крайне позднее создание работающего продукта.	С середины проекта
V-образная	• У каждой стадии есть чёткий проверяемый результат. • Внимание тестированию уделяется с первой же стадии. • Хорошо работает для проектов со стабильными требованиями.	• Недостаточная гибкость и адаптируемость. • Отсутствует раннее прототипирование. • Сложность устранения проблем, пропущенных на ранних стадиях развития проекта.	На переходах между стадиями
Итерационная инкрементальная	• Достаточно раннее прототипирование. • Простота управления итерациями. • Декомпозиция проекта на управляемые итерации.	• Недостаточная гибкость внутри итераций. • Сложность устранения проблем, пропущенных на ранних стадиях развития проекта.	• В определённые моменты итераций. • Повторное тестирование (после доработки) уже проверенного ранее.
Спиральная	• Глубокий анализ рисков. • Подходит для крупных проектов. • Достаточно раннее прототипирование.	• Высокие накладные расходы. • Сложность применения для небольших проектов. • Высокая зависимость успеха от качества анализа рисков.	• В определённые моменты итераций. • Повторное тестирование (после доработки) уже проверенного ранее.
Гибкая	• Максимальное вовлечение заказчика. • Много работы с требованиями. • Тесная интеграция тестирования и разработки. • Минимизация документации.	• Сложность реализации для больших проектов. • Сложность построения стабильных процессов.	В определённые моменты итераций и в любой необходимый момент.

Жизненный цикл тестирования



Основные принципы тестирования

- Тестирование показывает наличие дефектов, а не их отсутствие
- Исчерпывающее тестирование невозможно
- Тестирование тем эффективнее, чем раньше оно выполняется
- Кластеризация дефектов
- “Парадокс пестицида”
- Тестирование зависит от контекста
- Отсутствие дефектов - не самоцель