

Модель Хаоса

- Главное правило - всегда решать наиболее важную задачу первой
- Задача - это незавершенная частная задача программирования.
- Наиболее важная задача это комбинация большого размера, срочности и устойчивости.
 - Задачи большого размера ценны для пользователей настолько, насколько они функциональны.
 - Срочные задачи своевременны настолько, насколько должны быть, иначе задерживается остальная работа.
 - Устойчивые задачи проверены и испытаны. Разработчики могут благополучно сфокусироваться на другом.
- Решить, означает привести в состояние стабильности.

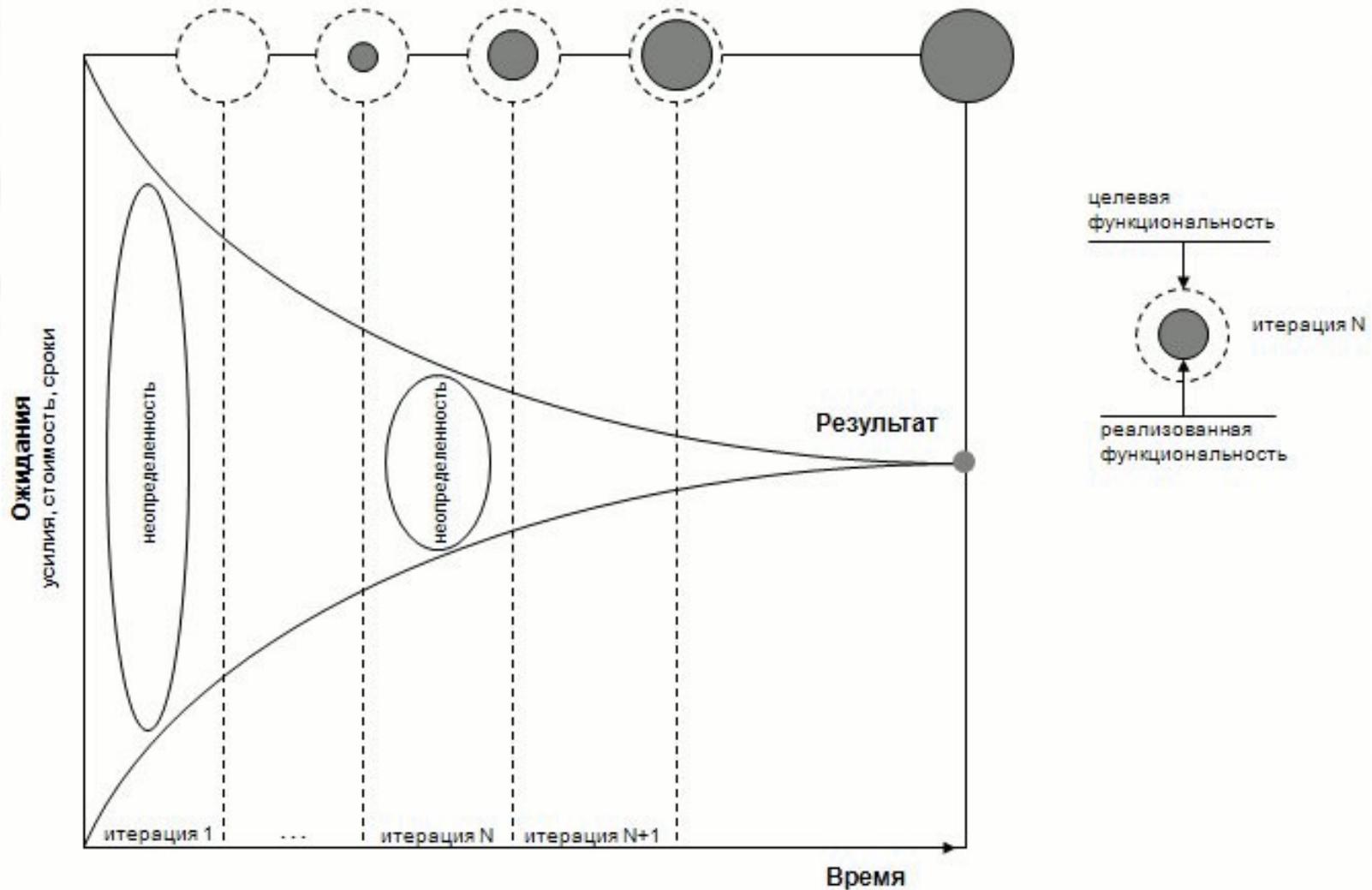
Рациональный унифицированный процесс

- начало — определение бизнес-целей проекта;
- исследование — разработка плана и архитектуры проекта;
- построение — постепенное создание системы;
- внедрение — поставка системы конечным пользователям.

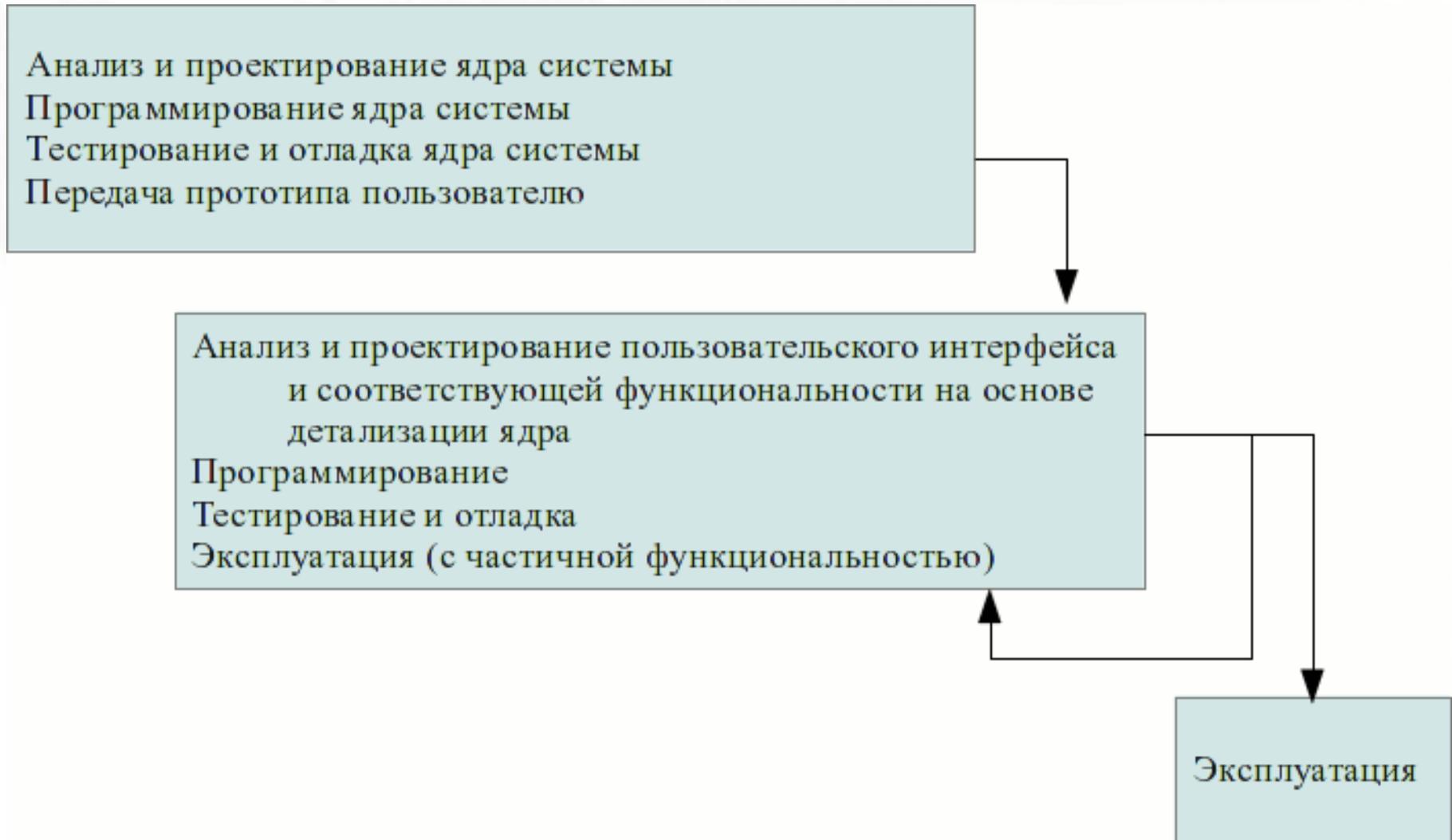
OpenUP

- Совместная работа с целью согласования интересов и достижения общего понимания;
- Развитие с целью непрерывного обеспечения обратной связи и совершенствования проекта;
- Концентрация на архитектурных вопросах на ранних стадиях для минимизации рисков и организации разработки;
- Выравнивание конкурентных преимуществ для максимизации потребительской ценности для заинтересованных лиц.

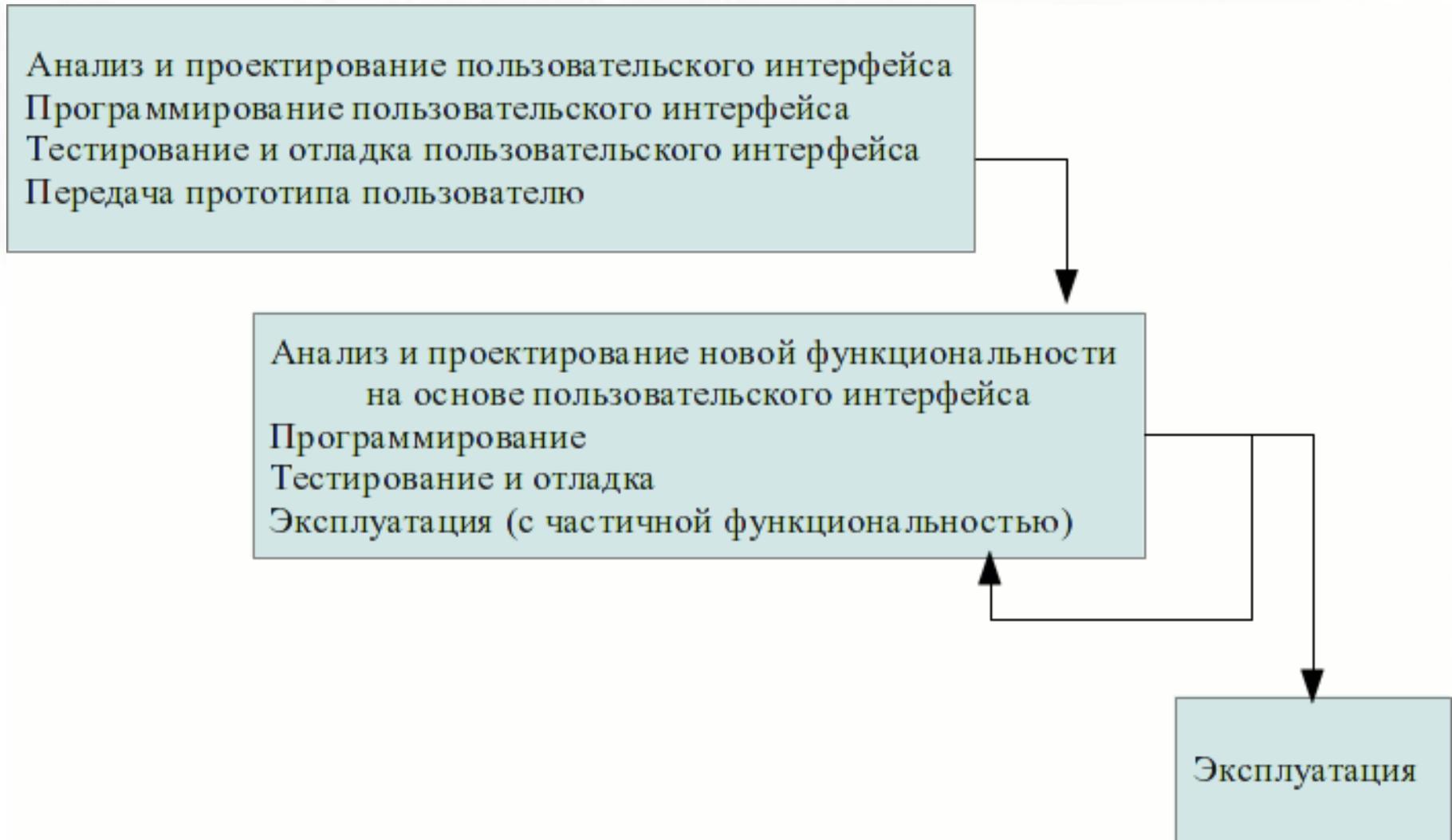
Итерационные методы



Итерационный метод разработки



Эволюционное прототипирование



Гибкие методологии разработки

- минимизация рисков
- интеративный подход
- постоянное общение с заказчиком

Экстремальное программирование

- Короткий цикл обратной связи
- Непрерывный, а не пакетный процесс
- Понимание, разделяемое всеми
- Социальная защищенность программиста

SCRUM

- резерв (история) проекта
- спринт
- совещания

LEAN

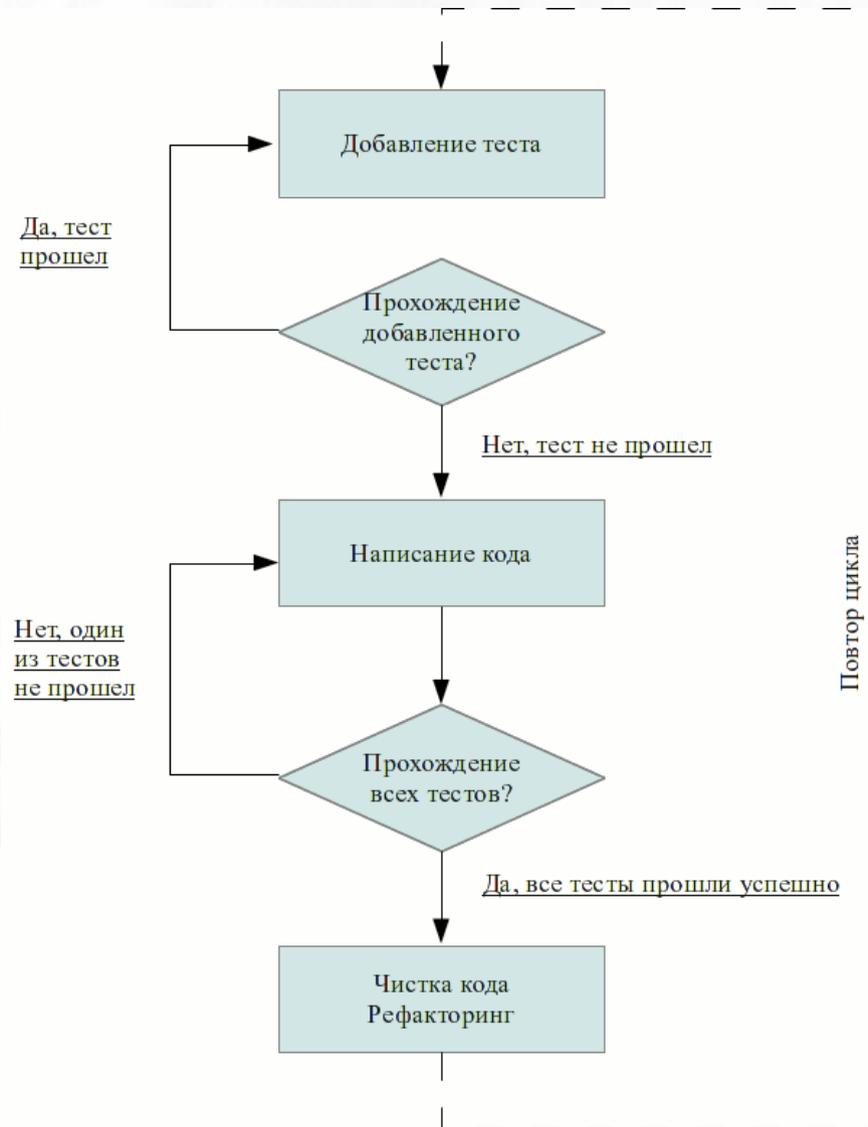
- Исключение затрат.
- Акцент на обучении.
- Предельно отсроченное принятие решений.
- Предельно быстрая доставка заказчику.
- Мотивация команды.
- Интегрирование.
- Целостное видение.

TDD

Жизненный цикл:

- Добавление теста
- Запуск всех тестов: новые тесты не проходят
- Написание кода
- Запуск всех тестов: все тесты проходят
- Рефакторинг
- Повтор цикла

TDD



FDD

Пять базовых видов деятельности:

- разработка общей модели;
- составление списка необходимых функций системы;
- планирование работы над каждой функцией;
- проектирование функции;
- реализация функции.

FDD: практические методы

- Объектное моделирование области
- Разработка по функции
- Индивидуальное владение классом (кодом)
- Команда по разработке функций (свойств)
- Проверка кода (code review)
- Конфигурационное управление
- Регулярная сборка
- Обозримость хода работ и результатов

Технологии коллективной разработки. Авторская разработка

- Авторская разработка — принцип создания программных продуктов, при котором весь жизненный цикл разработки поддерживается одним программистом

Технологии коллективной разработки. Коллективная разработка 1/2

- равноправные исполнители
- программирование в парах
- ядерная модель

Технологии коллективной разработки. Коллективная разработка 2/2

Бригада главного программиста. Роли:

- Главный программист
- Дублер
- Администратор
- Редактор
- Языковед
- Инструментальщик
- Отладчик
- Делопроизводитель

Технологии коллективной разработки. Общинная разработка

- Децентрализованность разработки
- Разработка на базе открытых исходных текстов
- Большое количество внешних тестеров

Оффшорное программирование

- Аутстаффинг
- Аутсорсинг
- Полная разработка