

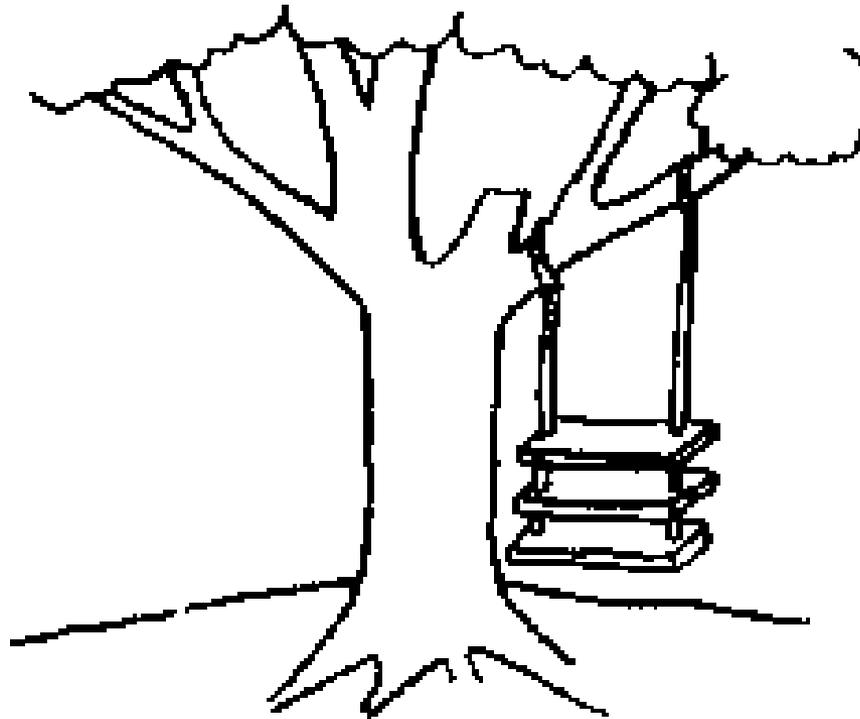
ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Жизненный цикл ПО.

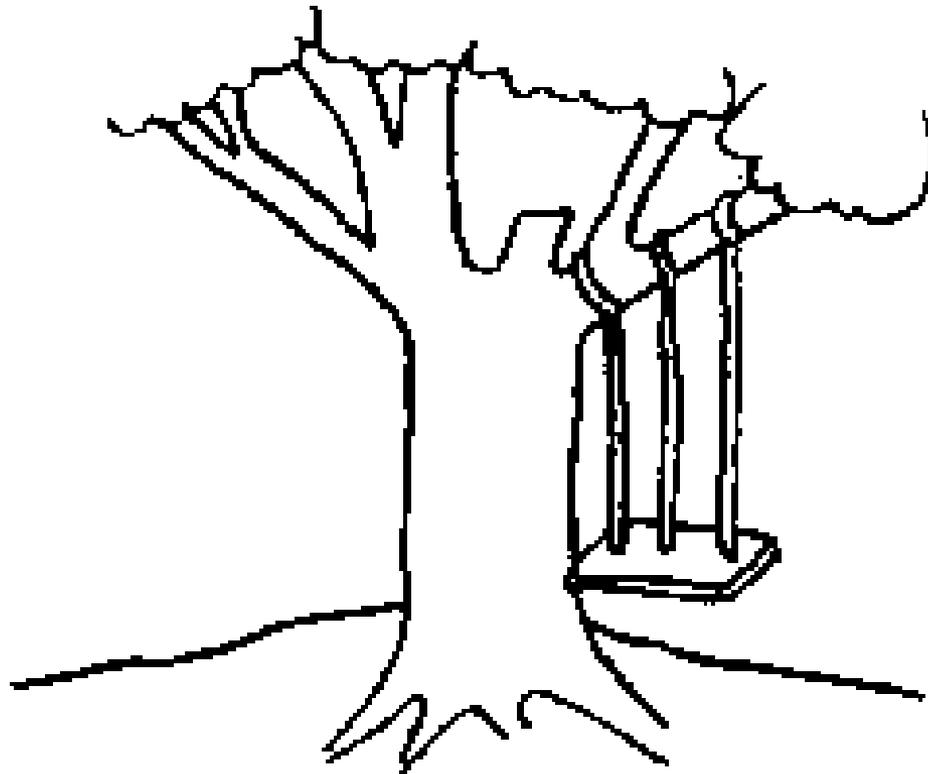
Водопадная и итерационная модель.

ПРОЦЕСС РАЗРАБОТКИ ПО (ЖИЗНЕННЫЙ ЦИКЛ ПО)

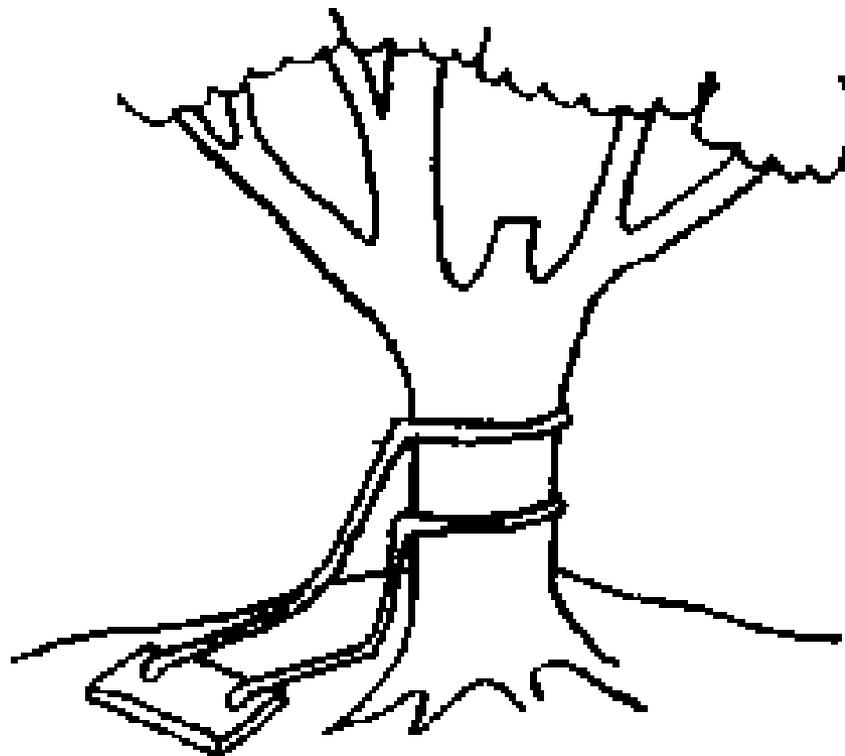
«КАЧЕЛИ» - КАК ПРОЕКТИРУЮТСЯ ПРОГРАММЫ (1975 !)



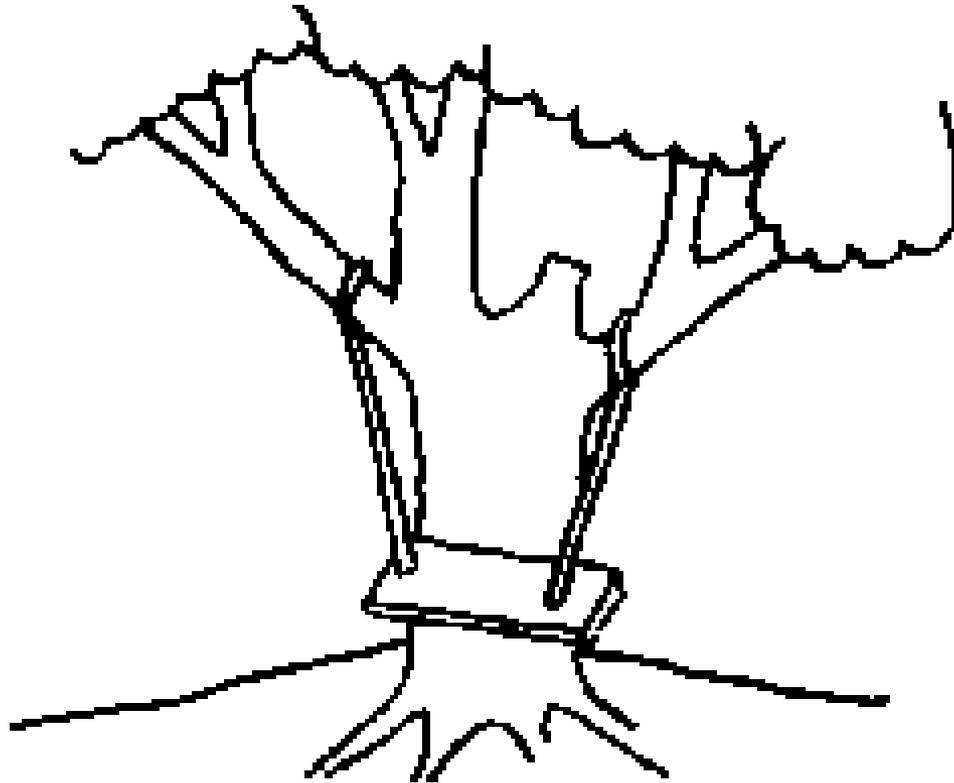
**Как было предложено
организатором разработки**



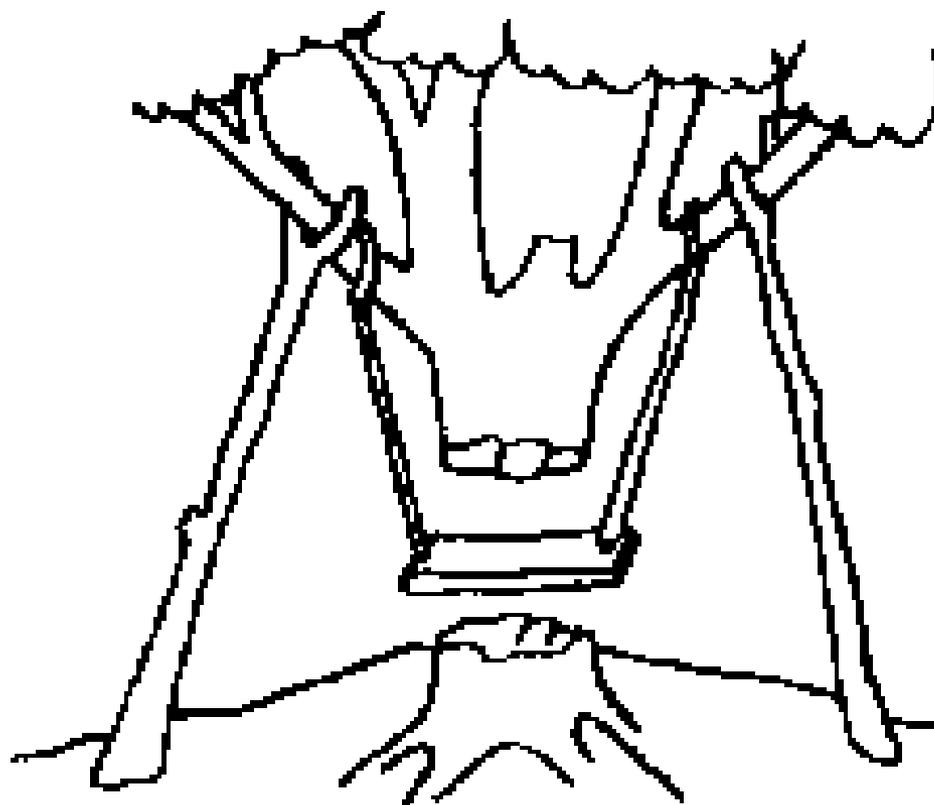
**Как было описано
в техническом задании**



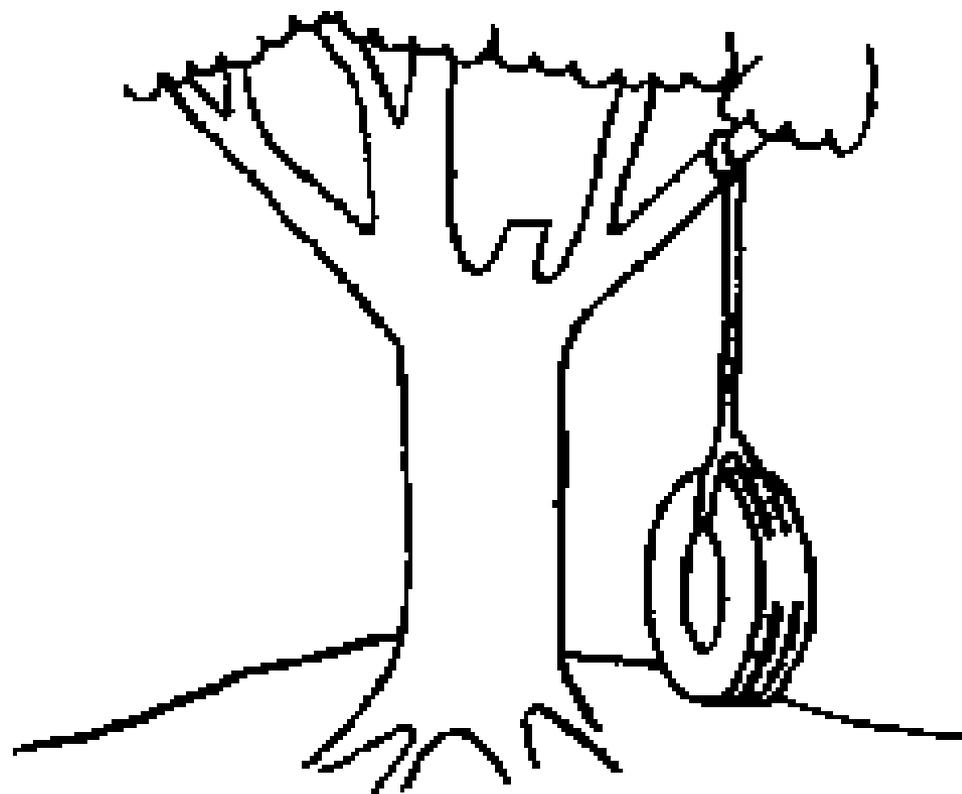
**Как было спроектировано
ведущим системным специалистом**



Как было реализовано программистами



Как было внедрено



Чего хотел пользователь

ПРОЦЕСС РАЗРАБОТКИ ПО

- ◎ **Процесс разработки ПО (жизненный цикл ПО)**
– это набор действий и связанных с ними результатов, направленных на разработку и/или развитие программного продукта:
 1. Спецификация требований
 2. Разработка
 3. Валидация
 4. Развитие

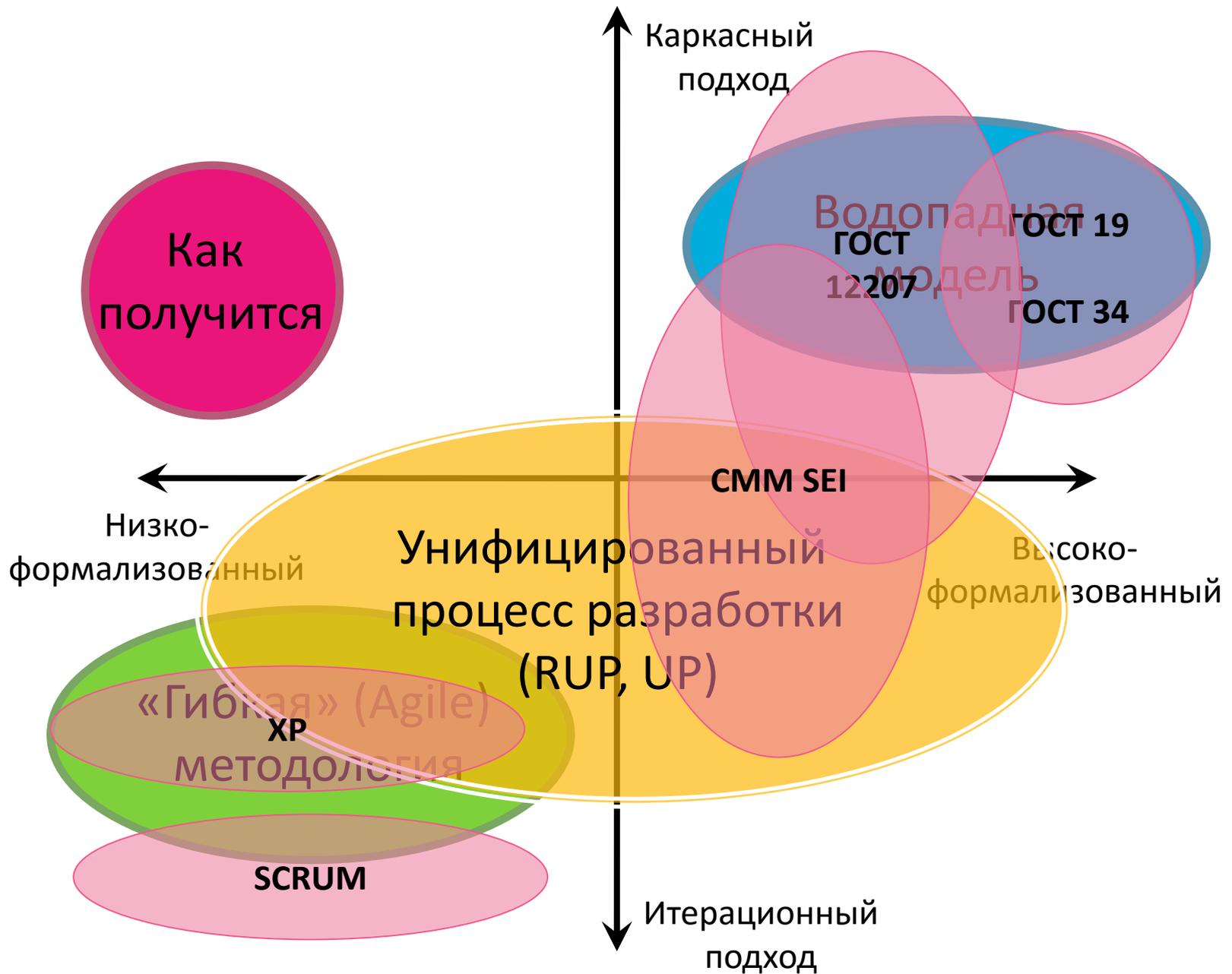
ПРОЦЕСС РАЗРАБОТКИ ПО

- © Не существует «идеального процесса разработки ПО»

НЕ СУЩЕСТВУЕТ!!!!

МОДЕЛЬ ПРОЦЕССА РАЗРАБОТКИ ПО

- ◎ **Модель процесса разработки ПО** – это абстрактная репрезентация процесса разработки ПО, представляющая данный процесс в определенной перспективе.
- ◎ Модель – это не всеобъемлющее описание процесса разработки ПО. Это скорее абстракция, которая позволяет описать различные подходы к процессу разработки.



МОДЕЛИ РАЗРАБОТКИ ПО

Можно выделить 3 основных модели разработки ПО:

 Водопадная модель

 Поэтапная (эволюционная) разработка

 Компонентно-ориентированная разработка

Часто эти модели объединяются в рамках одного проекта (подсистемы могут создаваться посредством различных подходов)



ВОДОПАДНАЯ МОДЕЛЬ

Определение требований

Проектирование системы

Реализация и тестирование

Интеграция и комплексное
тестирование

Функционирование и
поддержка



ФАЗЫ ВОДОПАДНОЙ МОДЕЛИ

- ◎ Результатом каждой фазы в водопадной модели является один или несколько утвержденных документов.
- ◎ Последующая фаза не может начаться пока предыдущая не завершена.
- ◎ Но только в идеальном мире процесс разработки линеен и не обладает обратными связями
- ◎ => в реальном мире в водопадной модели подразумевается несколько итераций, после которых результат «замораживается» и происходит переход ко следующей фазе.



ДОСТОИНСТВА И НЕДОСТАТКИ

Достоинства:

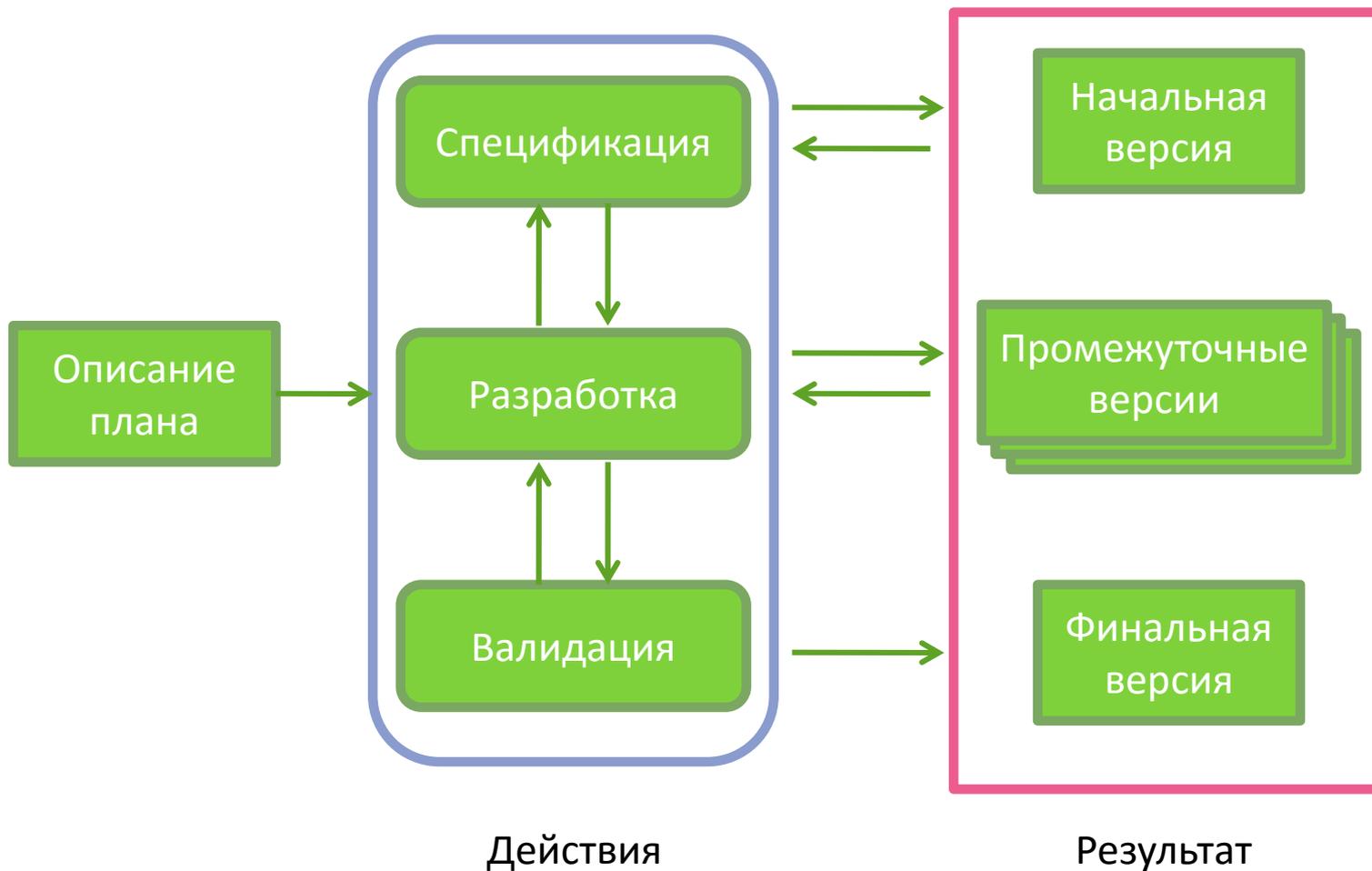
- ◎ Очень формализованная, в результате каждой фазы формируется утвержденный документ
- ◎ Соответствует стандартным моделям инженерной разработки.

Недостатки:

- ◎ Отсутствие гибкости
- ◎ Все договоренности – на ранней стадии, соответственно нет возможности подстроиться под изменяющиеся требования пользователя.



ПОЭТАПНАЯ РАЗРАБОТКА





ПОЭТАПНАЯ РАЗРАБОТКА

- ◎ ***Постепенная (эволюционная) разработка:***
тесная работа с клиентом, для выяснения требований и постепенное создание финальной версии.
- ◎ В начале – те части, которые понятно как делать. Развитие системы: добавление новых возможностей, предлагаемых клиентом.

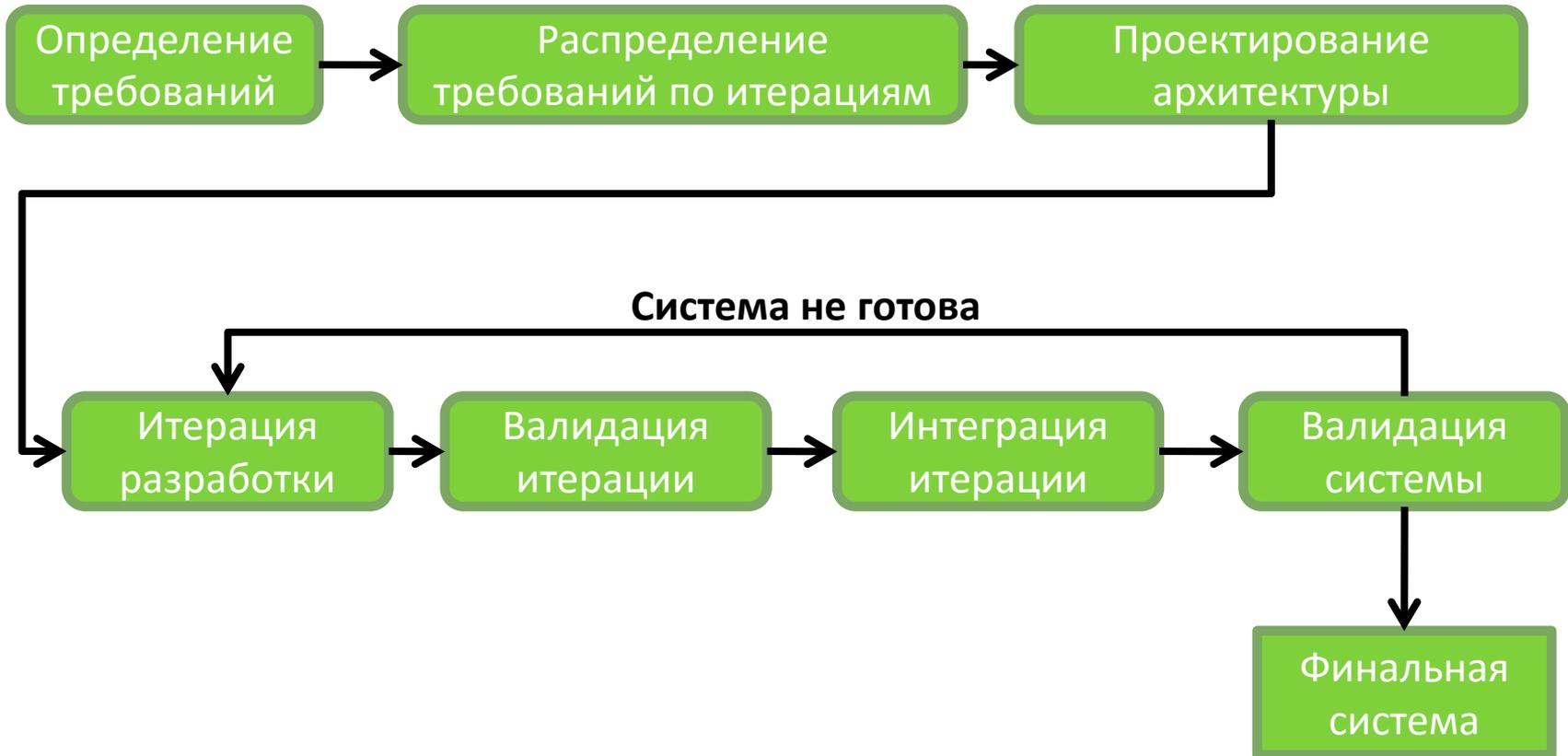


ПОЭТАПНАЯ РАЗРАБОТКА

- ◎ **Разработка на основе прототипов:**
постепенно создаются прототипы разрабатываемой системы, для того чтобы понять конкретные требования заказчика.
- ◎ Как только прототип выполнил свою задачу (требования заказчика стали понятны), он выбрасывается. Его код не используется в конечном продукте.



ИНКРЕМЕНТАЛЬНАЯ РАЗРАБОТКА





ИНКРЕМЕНТАЛЬНАЯ РАЗРАБОТКА

- ③ В начале описываются и ранжируются по значимости базовые сервисы, которые должна предоставлять система.
- ③ Определяется количество итераций, за которые должны создать готовую систему.
- ③ В начале каждой итерации формируются детальные требования к набору сервисов, которые будут созданы в ее рамках (дальнейшее их изменение невозможно).



ДОСТОИНСТВА

- ◎ Не надо ожидать, пока вся система будет полностью готова. Результат первой итерации может быть использован сразу.
- ◎ Ранние итерации могут служить прототипами и давать основу для построения требований ко следующим итерациям.
- ◎ Наиболее важные сервисы создаются в первую очередь, соответственно обеспечивается их всестороннее тестирование на более поздних этапах.



НЕДОСТАТКИ

- ⦿ Необходимо обеспечить малый объем работ в рамках итерации (не более 20 000 строк кода). При этом, должна обеспечиваться определенная функциональность.
- ⦿ Но множество систем обладают большим набором базовых сервисов, необходимых всем дальнейшим надстройкам и разделить их на итерации не всегда легко.

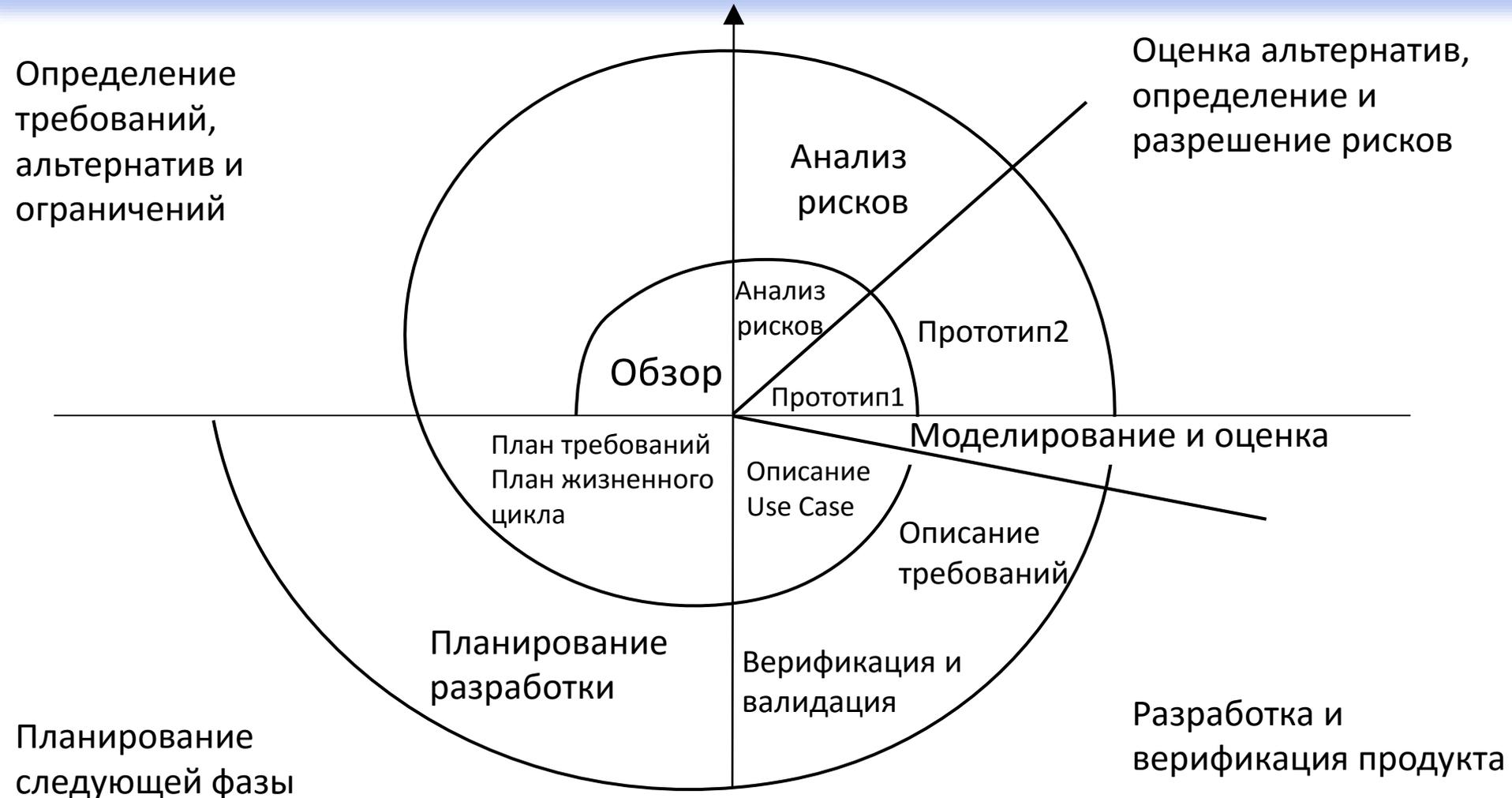


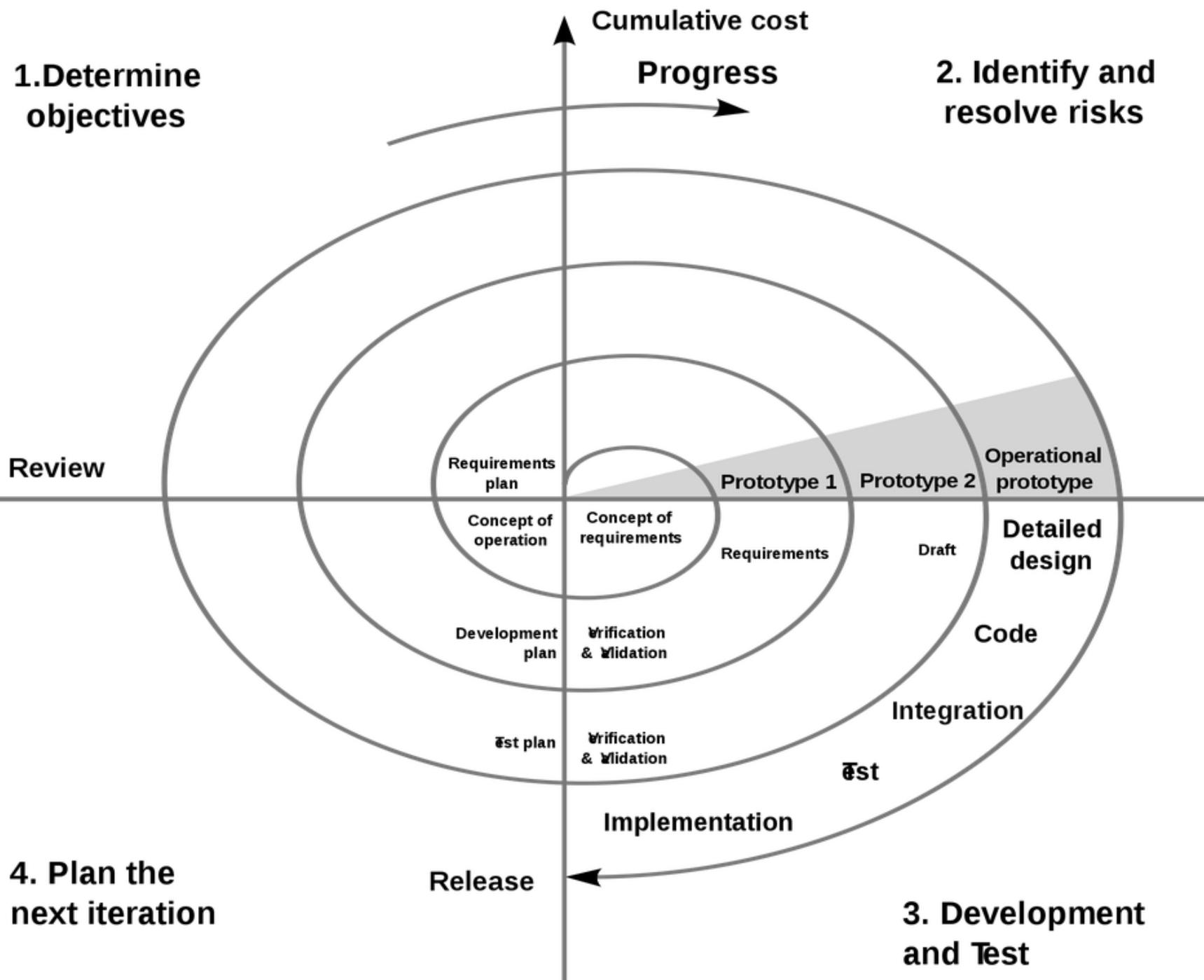
СПИРАЛЬНАЯ РАЗРАБОТКА

- ◎ Каждый шаг спирали – фаза процесса разработки ПО (постановка задачи, определение требований, дизайн архитектуры и т.д.)
- ◎ В каждом шаге выделяют 4 сектора:
 - ◎ Определение требований
 - ◎ Оценка и уменьшение рисков
 - ◎ Разработка и валидация
 - ◎ Планирование



СПИРАЛЬНАЯ РАЗРАБОТКА







- ◎ **Основное отличие спиральной разработки от других методов разработки ПО – это явная оценка рисков.**
- ◎ Риск – это вероятность того, что что-то может пойти не так как хотелось бы (например, при использовании нового языка программирования есть риск, что существующие компиляторы **не** будут создавать высокоэффективный код).



ПРИМЕНЕНИЕ ПОЭТАПНОЙ РАЗРАБОТКИ

- ◎ Поэтапная разработка более гибкая, чем водопадная модель, позволяет легче подстраиваться под изменяющиеся требования заказчика
- ◎ Хорошо подходит для небольших и средних по размерам проектов (порядка 500 000 строк кода)



ИТЕРАЦИИ ПОЗВОЛЯЮТ

- ① контролировать и корректировать ход выполнения проекта
- ① эффективнее работать с изменяющимися требованиями;
- ① эффективнее работать с рисками
- ① на ранних этапах оценивать потенциальные характеристики системы



ПРОБЛЕМЫ ПОЭТАПНОЙ РАЗРАБОТКИ

- ◎ **Процесс разработки не виден**
 - ◎ Ради скорости разработки в жертву приносится формальность: практически отсутствует документация, производимая на каждом этапе водопадной модели

- ◎ **Системы часто слабоструктурированы**
 - ◎ Постоянные изменения приводят к повреждению структуры ПО. Поддержка и дальнейшее изменение становится дорогой и сложной процедурой.