Планирование разработки программного обеспечения

Подходы к разработке ПО

1) проектирование исходя из выполняемых ПО функций (задач);
2) проектирование, основанное на исследовании потоков данных;
3) проектирование, основанное на структуре данных;
4) проектирование на базе абстрактных типов данных;
5) объектно-ориентированное проектирование.

1 Проектирование исходя из выполняемых функций (метод функциональной декомпозиции)

Метод предназначен для создания программ решения научно-технических задач . Задачи характеризуются небольшим количеством исходных данных, не имеют сложной структуры (данные организованы просто), но характеризуются сложным алгоритмом решения.

Подходы к разработке ПО

IPO (*Input-Processing-Output*) - диаграмма содержит:

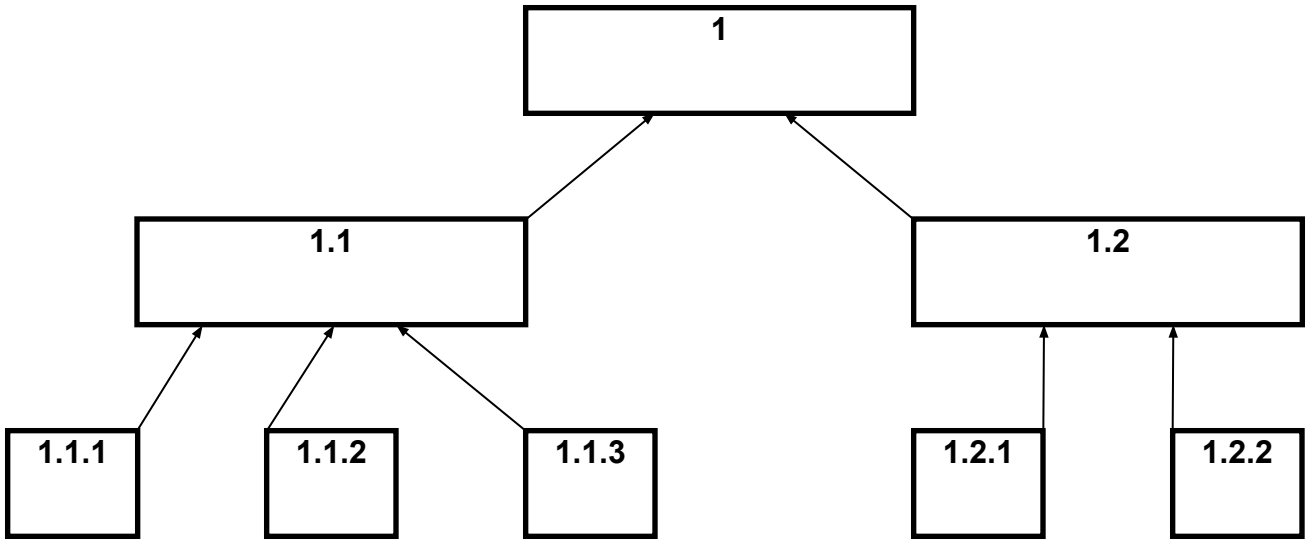
- перечень входных данных;
- описание выполняемой функции;
- перечень выходных данных (и сигнальную переменную)

Этапы проектирования:

1) создается иерархическая диаграмма (функциональная декомпозиция),
2) для каждого элемента иерархической диаграммы составляют IPO-диаграмму.

Подходы к разработке ПО

IPO - диаграмма



Номер: 1.1	Имя:	
Ввод Input	**Процесс Processing**	**Вывод Output**
Описание исходных данных модуля	Описание процесса преобразования вход - выход. Включая обращения к модулям более низкого уровня	Описание выходных данных

Рис. 1. Функциональная декомпозиция

Подходы к разработке ПО

2 Проектирование, основанное на исследовании потоков данных

Применяется при проектировании систем обработки данных

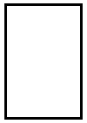
Этапы проектирования:

1. Составление диаграмм потоков данных. Диаграмма потоков данных задает движение данных, а не передачу управления, как на традиционных схемах алгоритмов.

2. Определение структуры данных для каждого источника, потребителя, файла (базы данных). Для каждого источника и потребителя разрабатывается интерфейс пользователя

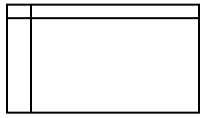
Подходы к разработке ПО

Процесс обработки (5,6)



Источник данных (1,2,3)
Получатель данных (4)

Поток данных



Файл или база данных

Для каждого процесса обработки уточняются состав входных и выходных данных и выполняемая процессом функция (это аналогично составлению IPO-диаграммы). После этого дальнейшее проектирование этого процесса может быть осуществлено методом функциональной декомпозиции.

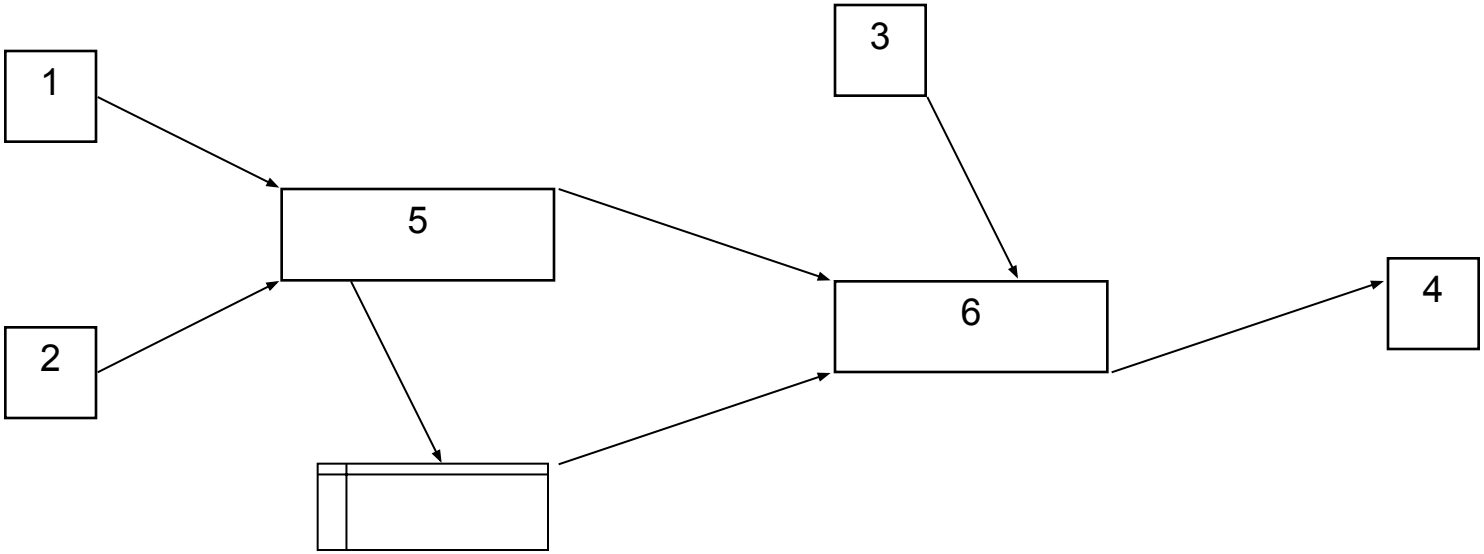


Рис. 2. Диаграмма потоков данных

Подходы к разработке ПО

3 Проектирование, основанное на структуре данных	применяется при проектировании баз данных

Определяется состав данных, продумывается структура базы данных, данные распределяются по таблицам (где они и хранятся, т.е. таблица – источник данных), для каждой таблицы определяются поля, тип данных в них и назначаются ключи, после чего создаются связи между таблицами.

Должен соблюдаться принцип: каждый элемент данных вводится один раз. На основе всех таблиц создаются запросы для выборки и обработки данных, формы для просмотра данных и отчеты для вывода данных на печать.

4 Проектирование на базе абстрактных типов данных

Применяется при разработке трансляторов языков программирования.

Тип данных характеризуется:

1) множеством допустимых значений;
2) множеством операций над данными этого типа и правилами их выполнения.

Проектирование на базе абстрактных типов данных заключается в установлении соответствия между объектами предметной области и имеющимися в среде реализации абстрактными типами данных, а также в доопределении и реализации недостающих данных и операций над ними.

Подходы к разработке ПО

5 Объектно-ориентированное проектирование

Объект = (данные + методы (методы – функции, процедуры, конструкторы, деструкторы))

Описание объекта включает:

- описание данных, характеризующих объект;
- описание процедур и функций для обработки этих данных.

Объектно-ориентированный подход

Объектно-ориентированный подход включает в себя:

1) объектно-ориентированный анализ;
2) объектно-ориентированное проектирование;
3) объектно-ориентированное программирование.

На стадии **анализа** путем исследования предметной области выявляют, какие объекты в ней существенны, как они взаимодействуют.

На стадии **проектирования** создают проект будущего программного комплекса в терминах объектов и передаваемых между ними сообщений. Объект включает в себя *данные* и *процедуры* для их обработки, а *передача сообщения* от одного объекта к другому с т.з. программиста означает вызов процедуры, входящей в состав объекта-адресата.

На стадии **программирования** выполняется реализация проекта на языке программирования, имеющего средства объектно-ориентированного программирования.

Подходы к разработке ПО

Инкапсуляция - это возможность закрыть часть содержания объекта от пользователей или других объектов.

Наследование свойств - это возможность связать объекты отношениями подчинения (предшественник и наследник) и, таким образом, передать все свойства одного объекта другому. Т.е. объекту-наследнику доступны все данные и методы объекта-предшественника, поэтому их не нужно объявлять заново. Кроме того, объект-наследник может иметь дополнительные данные и методы.

Полиморфизм - это возможность разной реализации одних и тех же операций у разных объектов (т.е. один и тот же метод может быть реализован по-разному, в зависимости от того, какому объекту он принадлежит).

Этапы проектирования:

- Выделение объектов
- Определение их свойств
- Определение задач, выполняемых каждым объектом
- Определение наследования (установление иерархических связей между объектами)

Модели процесса разработки ПО

1) "Водопад" (Waterfall);
2) Прототипирование;
3) Итерация;
4) Спираль.

1 Модель "Водопад"

Характерная черта этой модели - полное завершение предыдущего этапа до начала следующего, потому что к законченному этапу больше не возвращаются (отсутствует обратная связь). В связи с этим имеются следующие ограничения:

1. Требования к разрабатываемой системе стабилизированы до начала проектирования (требования не меняются).

2. Стабилизация требований к системе обычно влечет за собой и выбор технических средств в начале разработки, т.е. замораживаются требования к техническим средствам (они являются частью требований к системе).

Модели процесса разработки ПО

Используется для разработки простых систем или новых версий уже внедренных систем. В последнем случае (разработка новых версий) процесс разработки, внедрения и дальнейшего усовершенствования имеет циклический характер, как показано на следующем слайде …

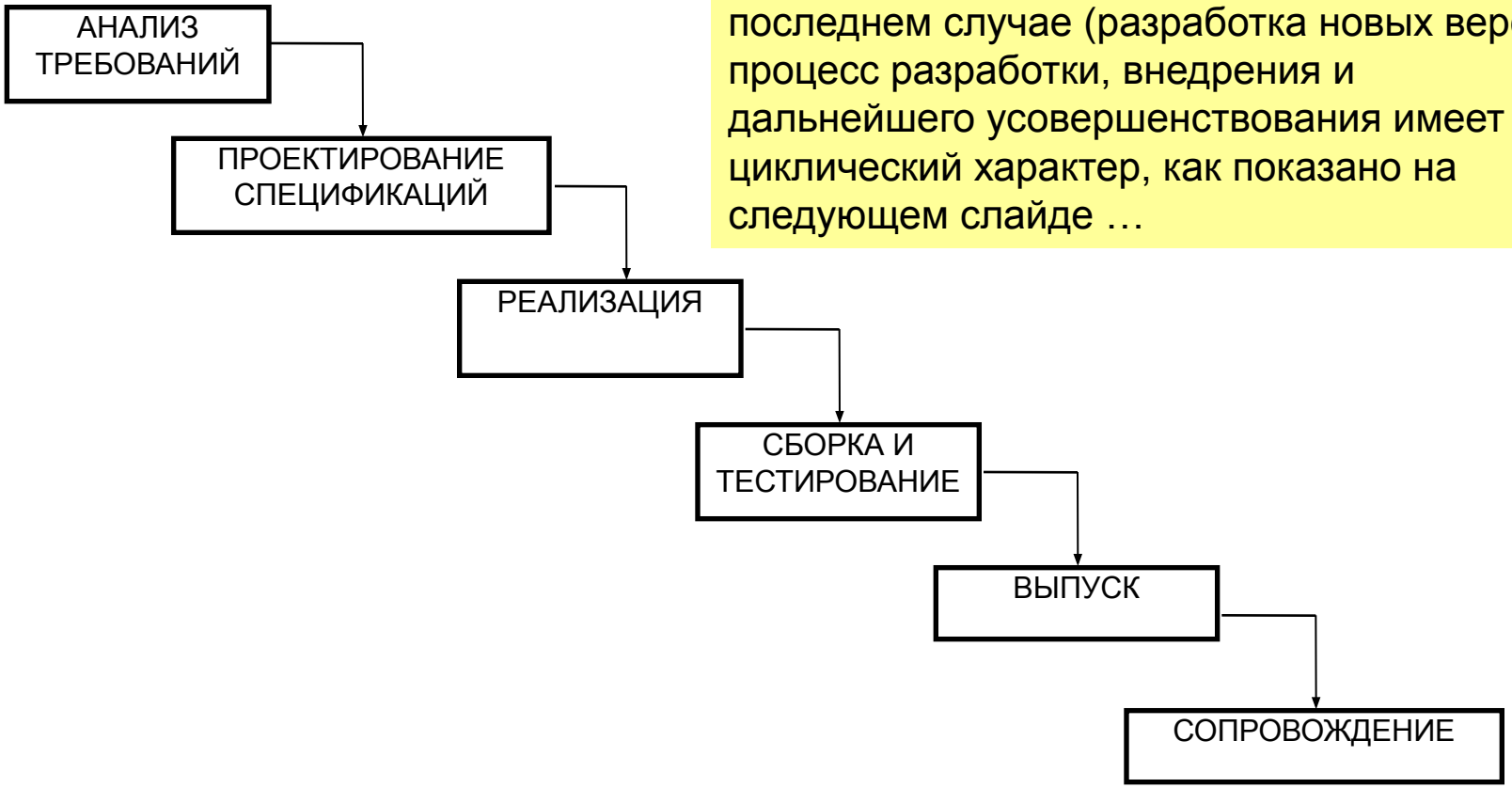


Рис. 3. Модель «Водопад»

Модели процесса разработки ПО

Сопровождение – это:

1) устранение замеченных в ходе эксплуатации ошибок и недостатков
2) незначительные улучшения
3) накопление предложений для разработки новых версий продукта
4) если программный продукт разрабатывается для рынка, то сопровождение включает создание и поддержание дилерской сети (туда обращаются по вопросам приобретения и эксплуатации продукта).

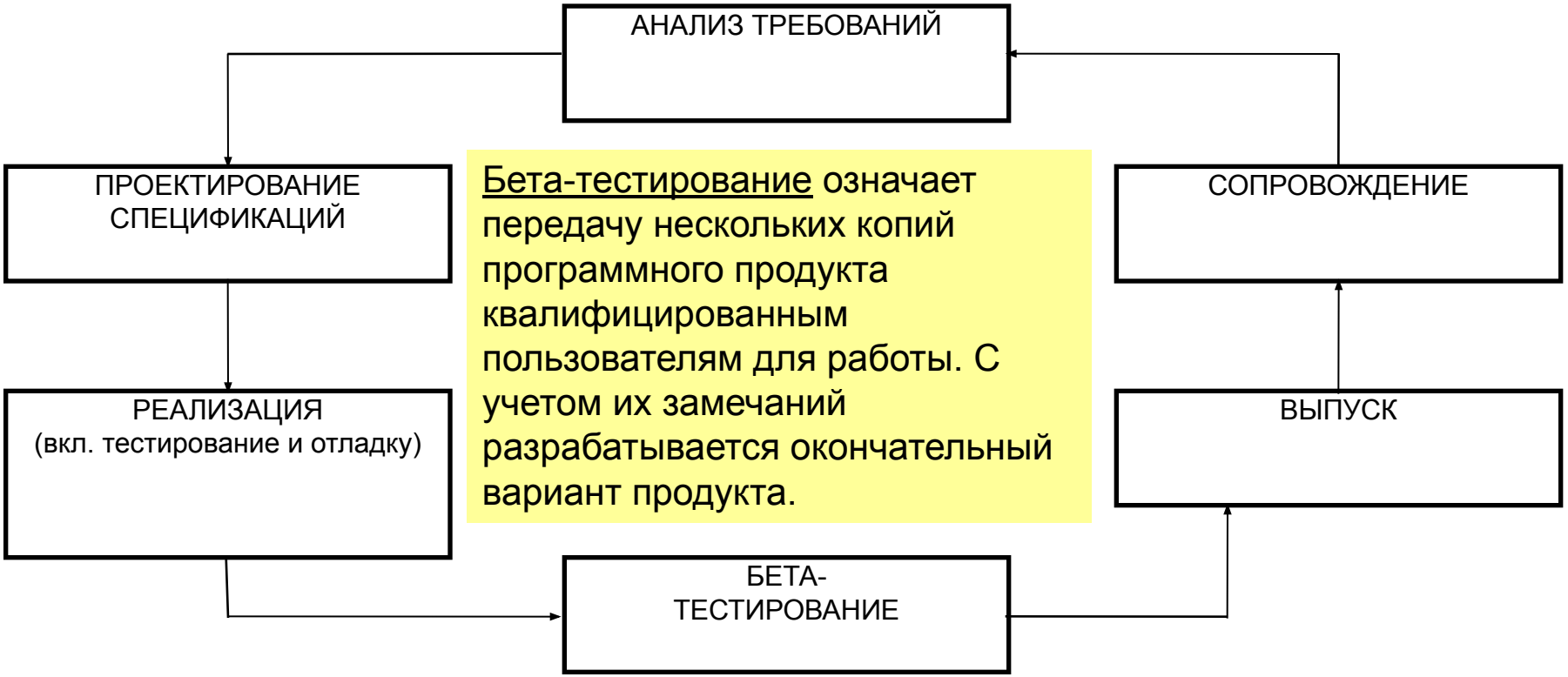


Рис. 4. Цикл создания и усовершенствования программного обеспечения

Модели процесса разработки ПО

2 Метод прототипирования

Идея метода заключается в том, что сначала разрабатывается не сам программный продукт, а его прототип, содержащий решение главных проблем, стоящих перед разработчиками. После успешного завершения разработки прототипа по тем же принципам разрабатывается и настоящий программный продукт.

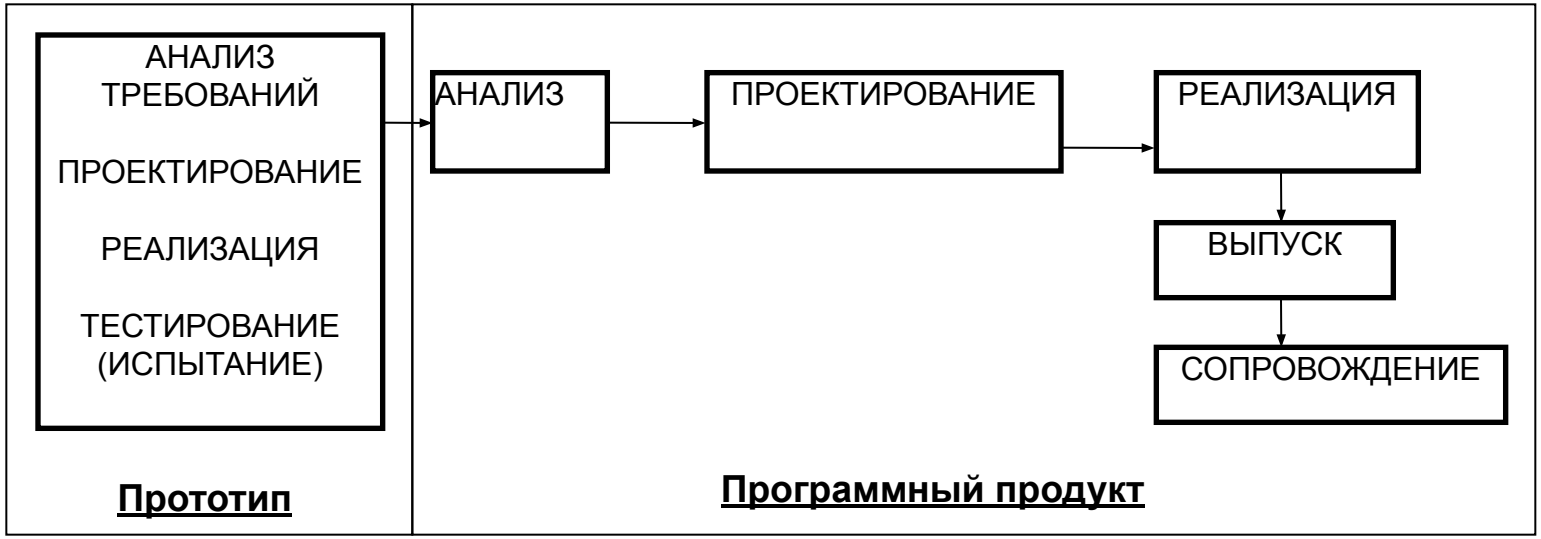


Рис. 5. Прототипирование

Модели процесса разработки ПО

3 Итерационная модель (*«водопад» + «прототипирование»*)

Процесс разработки выполняется как наращивание новых функциональных возможностей в разрабатываемую систему до тех пор, пока не будут реализованы все требуемые функции программного продукта (*начинают с самого сложного*).

Рис. 6. Итерационная модель

Преимущества данной модели:

1. Возможность активного участия заказчика в разработке, он имеет возможность уточнять свои требования в ходе разработки (работая шаг за шагом, разработчик и заказчик лучше понимают друг друга).
2. Возможность тестирования вновь разрабатываемых частей совместно с ранее разработанными, это уменьшит затраты на комплексную отладку (т.е. тестировать не весь продукт сразу, а по частям).
3. Уже во время разработки можно начинать внедрение по частям.

Модели процесса разработки ПО

4 Модель "Спираль"

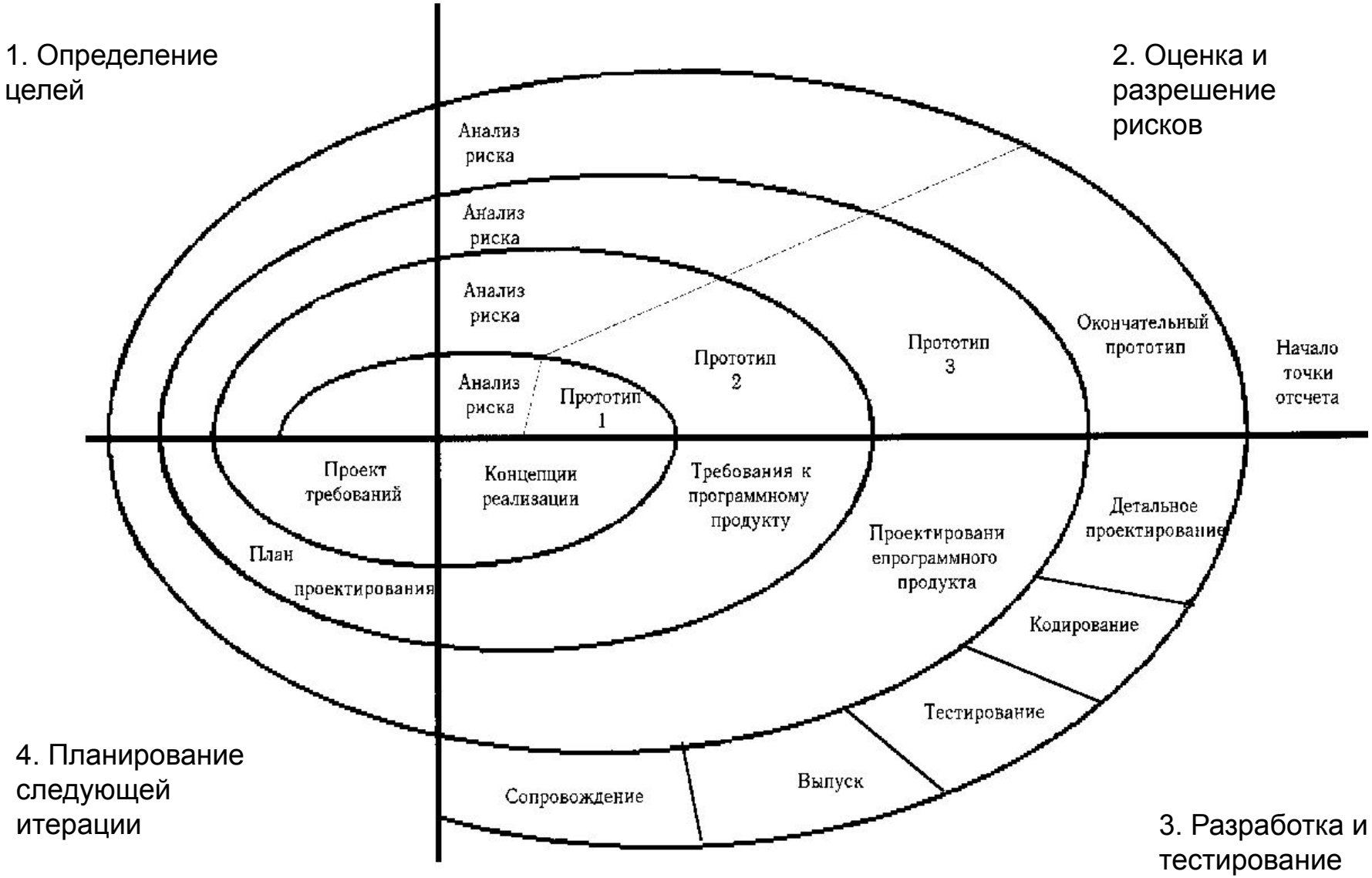


Рис. 7. Модель "спираль"

4 Модель "Спираль"

Состоит из циклов. Каждый цикл в модели начинается с определения цели этого цикла, анализа разных путей ее достижения и возможных ограничений, оценивается степень неопределенности и риска. Выбирается стратегия проектирования, позволяющая их уменьшить. Это относится к первому участку – анализу риска.

Далее разрабатывается первый прототип, следует уточнение требований, снова анализ, и т.д. Последовательно получается окончательный прототип.

Далее следуют пять этапов разработки программного продукта по прототипу. Такая модель применяется для проектов с большой неопределенностью и риском.

Категории рисков (по приоритету):

1. Дефицит специалистов.
2. Нереалистичные сроки и бюджет.
3. Реализация несоответствующей функциональности.
4. Разработка неправильного пользовательского интерфейса.
5. «Золотая сервировка», перфекционизм, ненужная оптимизация и оттачивание деталей.
6. Непрекращающийся поток изменений.
7. Нехватка информации о внешних компонентах, определяющих окружение системы или вовлечённых в интеграцию.
8. Недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами.
9. Недостаточная производительность получаемой системы.
10. Разрыв между квалификацией специалистов и требованиями проекта.