Лекция 6
Технология разработки ПО

О.В. Федорова, доцент каф. ПМИ

Модели жизненного цикла ПО

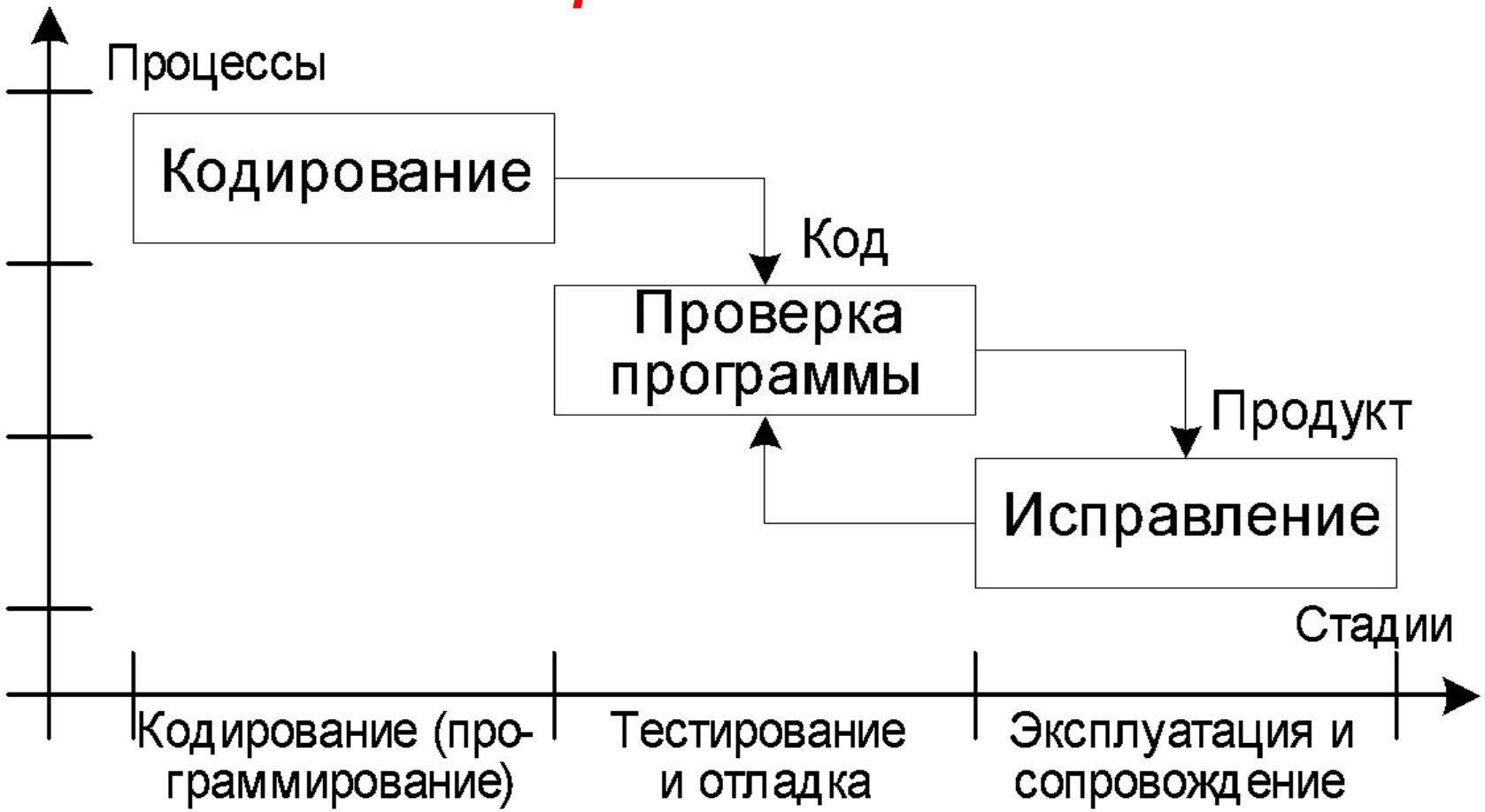
Основными моделями ЖЦ ПО считаются:

- Каскадная модель,
- Итеративная инкрементная модель,
- Эволюционная модель,
- Спиральная модель.

Непланируемая модель или *модель «кодирование – исправление»* является самой простой моделью ЖЦ ПО.

Принцип модели (рис.6.1) заключается в написании кода программы без какого-либо серьёзного предварительного анализа требований и проектирования, запусках программы для проверки его работоспособности и последующем исправлении ошибок и/или добавлении функциональности до получения варианта программы, удовлетворяющего пользователя.

Непланируемая модель или модель «кодирование – исправление»



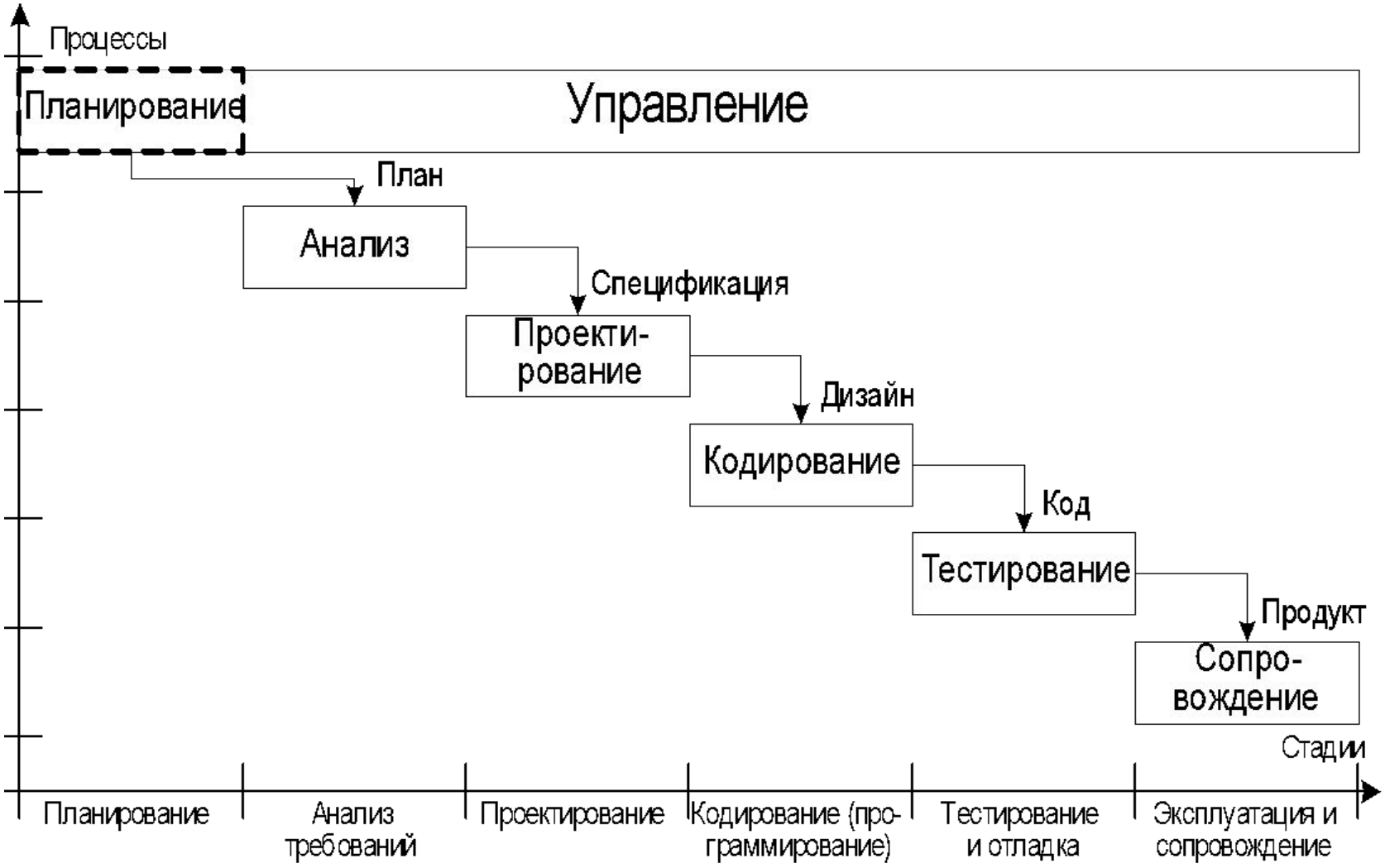
Классическая каскадная модель или *модель водопада*

Классическая каскадная модель или *модель водопада* создана по аналогии с методиками из других инженерных областей, где существует практика поэтапного создания продукта от составления спецификаций до поставки заказчику.

Принцип модели (рис.6.2) заключается в однократном выполнении процессов в виде заранее ограниченных и однозначно упорядоченных во времени стадий, осуществляемых как бы в их естественных границах.

Все процессы, действия, задачи выполняются чётко последовательно, не допускается никаких перекрытий, параллелизма и возвратов назад. Это жёсткое ограничение делает такую модель практически нереализуемой, так как на уже законченных стадиях не допустимы никакие ошибки, что требует получения артефактов идеального качества.

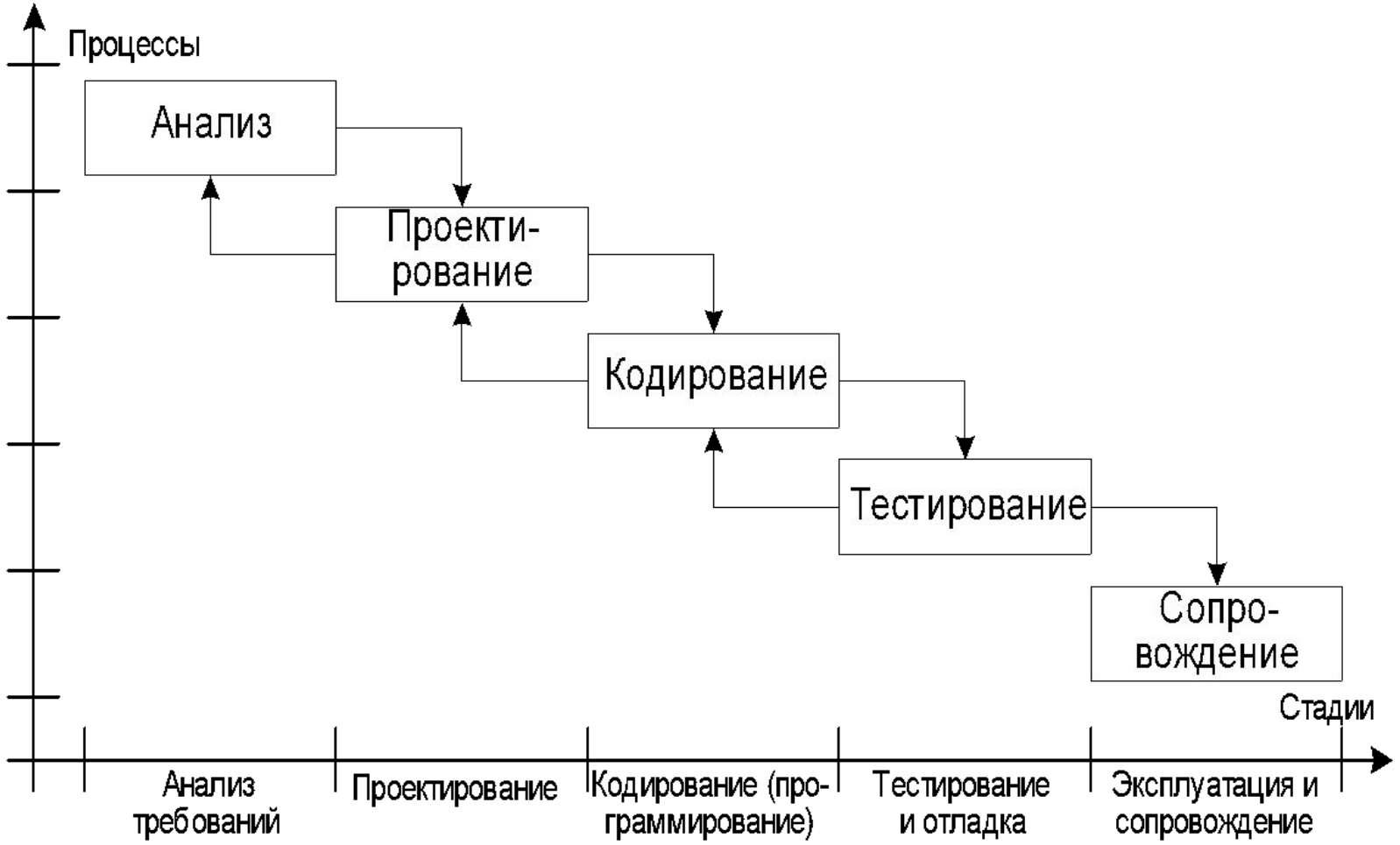
Классическая каскадная модель или модель водопада



Модифицированная каскадная модель или *модель водоворота*

- На практике применяются каскадные модели без учёта этих ограничений.
- *Модифицированная каскадная модель* или *модель водоворота* является простейшим случаем таких моделей. Принцип модели (рис.6.3) заключается в возможности возвращения на предыдущую стадию в случае нахождения ошибки на текущей стадии и пересмотре или уточнении ранее принятых решений.

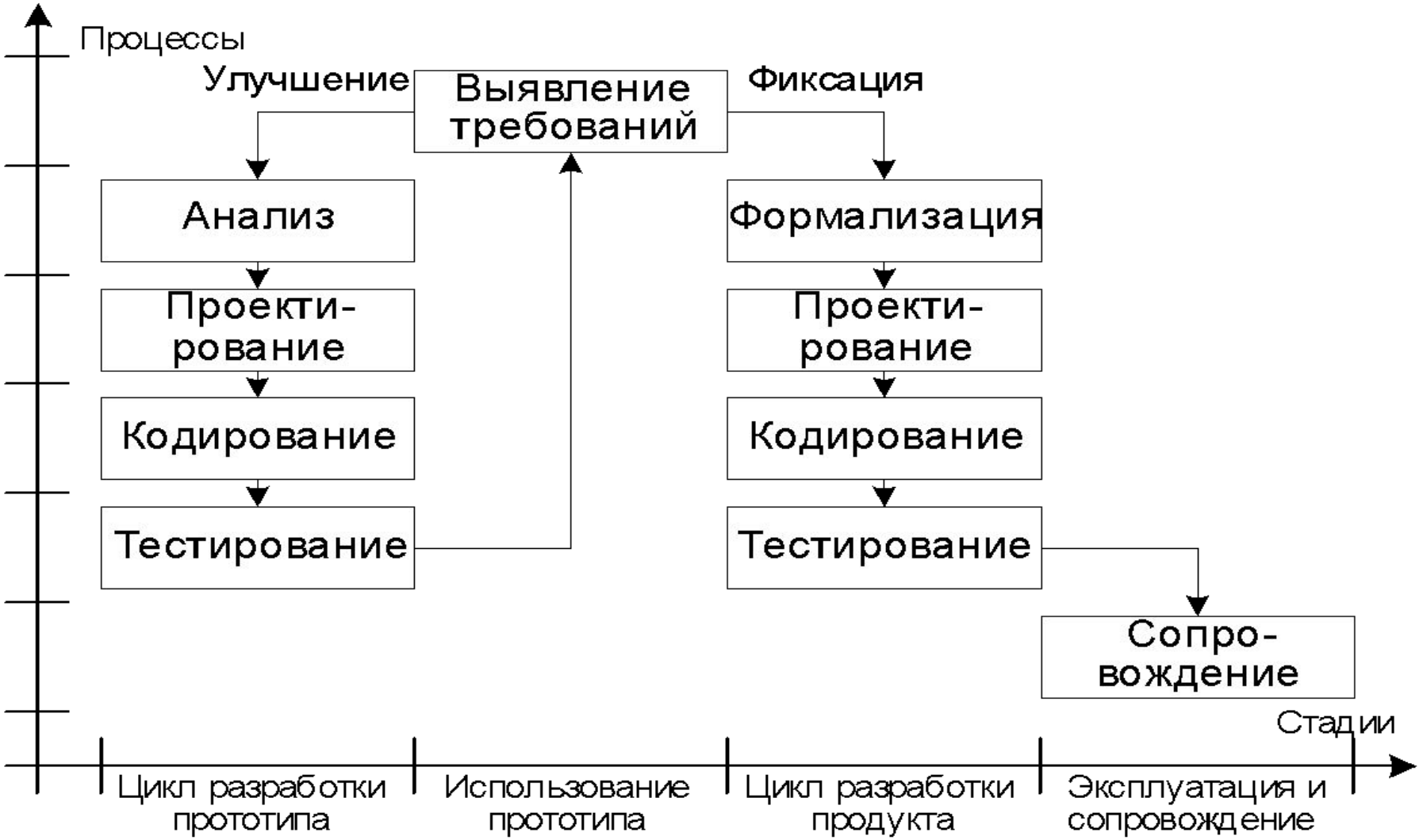
Модифицированная каскадная модель или модель водоворота



Прототипируемая модель или модель прототипирования

- ***Прототипируемая модель или модель прототипирования*** создана для решения проблем при разработке в условиях неопределённости исходных требований путём разработки прототипов требуемого продукта. Модель требует быстрого построения множества прототипов, поэтому реализация этой модели возможна только при использовании соответствующего инструментария автоматизации.
- ***Классическая модель прототипирования*** использует разработку прототипов для постепенного выявления всех требований. Принцип модели (рис.6.4) заключается в первоначальной разработке сильно упрощённого прототипа, который в соответствии с пожеланиями пользователя циклически усовершенствуется и усложняется до тех пор, пока требования к ПО не становятся очевидными. После этого выполняется формализация выявленных требований (составляется спецификация) и ведётся собственно разработка продукта по какой-либо модели. Разработка прототипа и

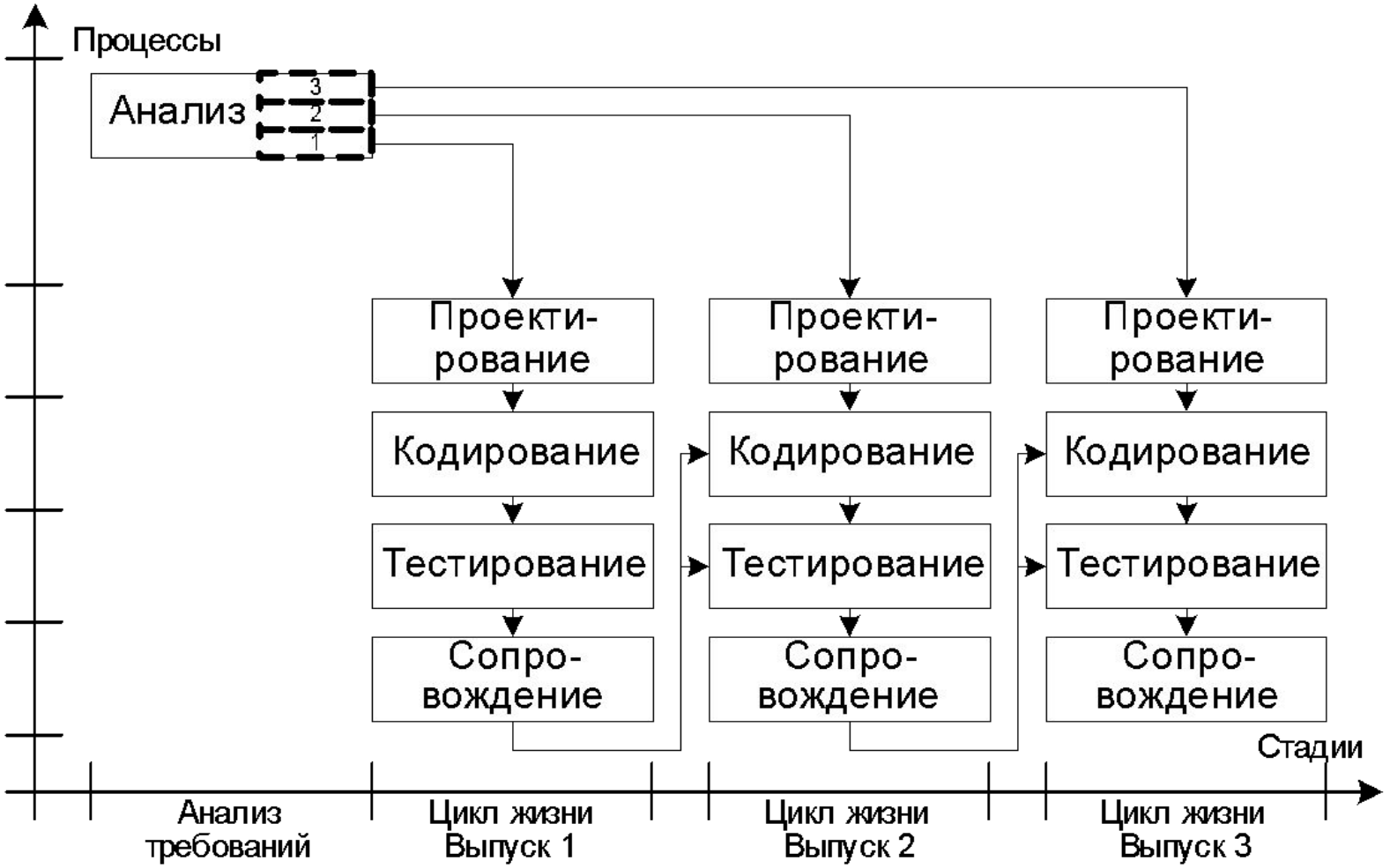
Прототипируемая модель или модель прототипирования



Модель одноразового прототипирования

- Модификацией классической модели прототипирования является *модель одноразового прототипирования*. ***Принцип этой модели заключается в том, что начальный прототип является одноразовым.***
- Благодаря прототипированию реализованные требования к ПО приближаются к целевым требованиям и снижаются неопределённости разработки (усилия, стоимости, сроки) вплоть до получения конечного продукта.
- В целом проекты с моделями прототипирования обычно завершаются на 40% раньше аналогичных проектов с каскадными моделями, а код получается на 45% более коротким. Однако этот код обычно оказывается более запутанным и сложным для сопровождения, а документация – не столь полной и качественной.

Итеративная инкрементная модель



Итеративная инкрементная модель

- *Итеративная инкрементная модель* или модель запланированного усовершенствования продукта использует разработку прототипов (выпусков) для последовательной реализации групп требований. *Принцип модели (рис.6.5) заключается в предварительном выделении требований и разработке прототипов, по функциональности всё более приближающихся к продукту.* Первый прототип-выпуск основывается на наиболее понятной группе требований, в последующие реализации добавляются всё новые группы требований, пока не будет закончено создание продукта. Для каждого прототипа выполняются необходимые процессы, причём анализ требований и проектирование архитектуры выполняются одновременно, а остальные процессы – индивидуально для каждого прототипа.

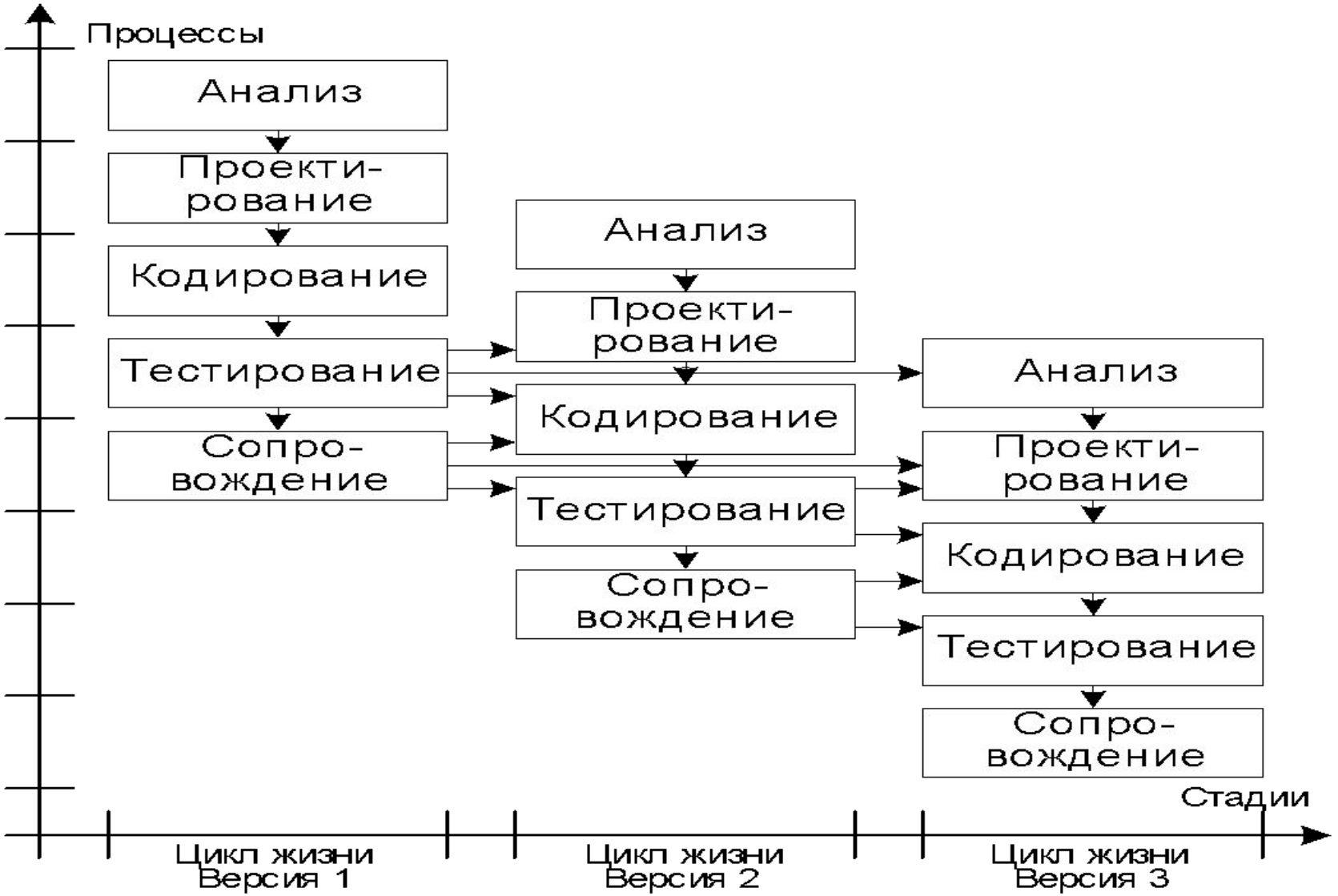
Итеративная инкрементная модель

- Рассматриваемая модель в явном виде включает в своё название два принципа, характерные в том или ином виде для многих моделей прототипирования: ***итеративность и инкрементность разработки.***
- *Итеративность означает разбиение ЖЦ на последовательность итераций, каждая из которых напоминает мини-проект.* Цель каждой итерации – разработка прототипа, результатом последней итерации является продукт.
- *Инкрементность означает разработку продукта путём постепенного учёта требований к системе.* Фактически это также приводит к разработке прототипов, причём последний (часто лишь по срокам) прототип

Эволюционная модель

- *Эволюционная модель использует разработку прототипов (версий) для реализации частично установленных требований при последовательном уточнении и расширении этих требований.*
- Принцип модели (рис.6.6) заключается в постепенной формулировке требований и разработке прототипов, по требованиям всё более приближающихся к продукту.
- Первый прототип-версия основывается на наиболее сложной и непонятной группе требований, в последующих версиях эти требования уточняются и расширяются с учётом разработки ранних версий.

Эволюционная модель



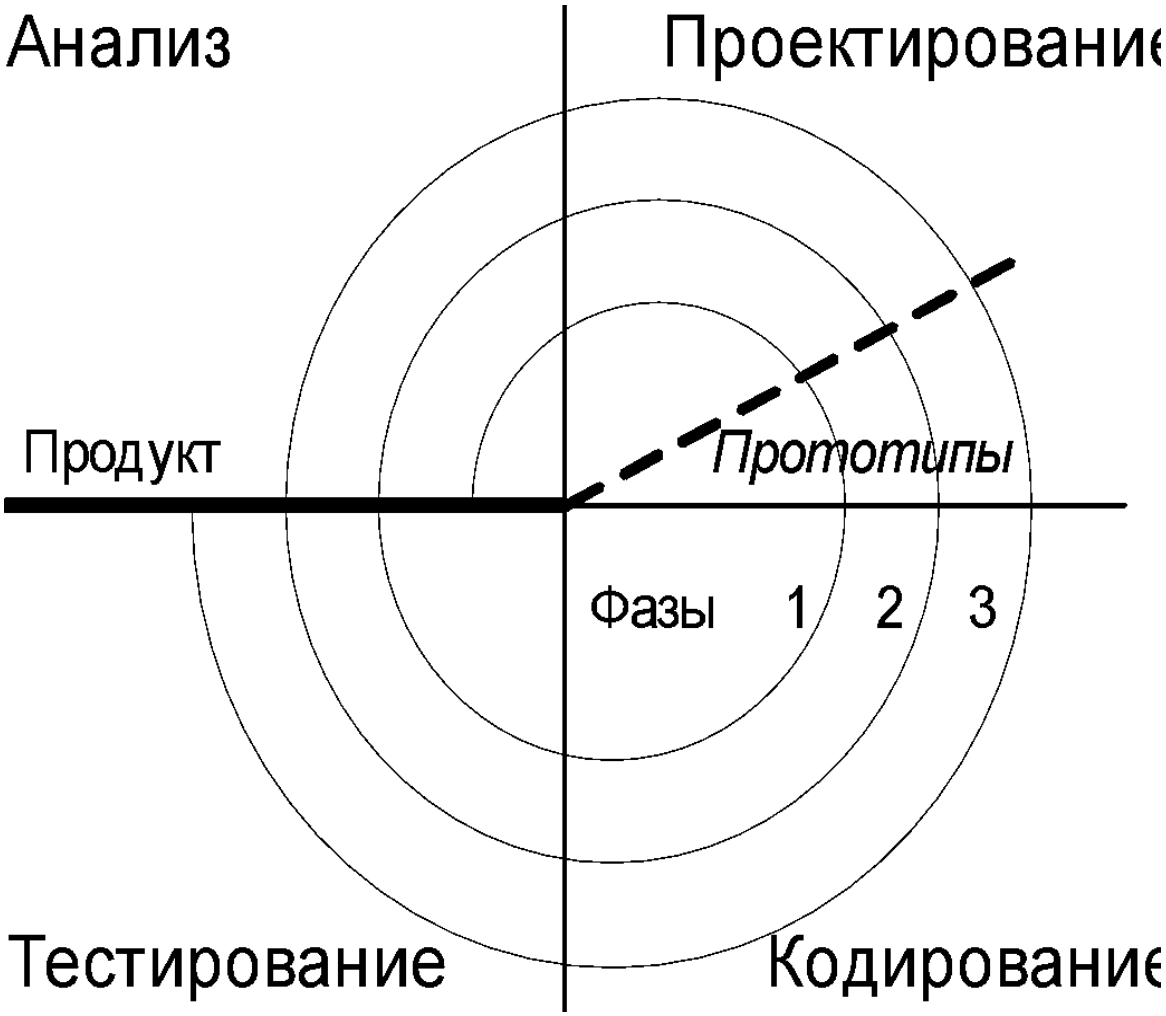
Эволюционная модель

- Рассматриваемая модель в явном виде включает в своё название принцип, характерный для ряда моделей прототипирования: эволюционность разработки.
- *Эволюционность* означает разработку продукта путём включения и доработки реализации требований по мере их прояснения.

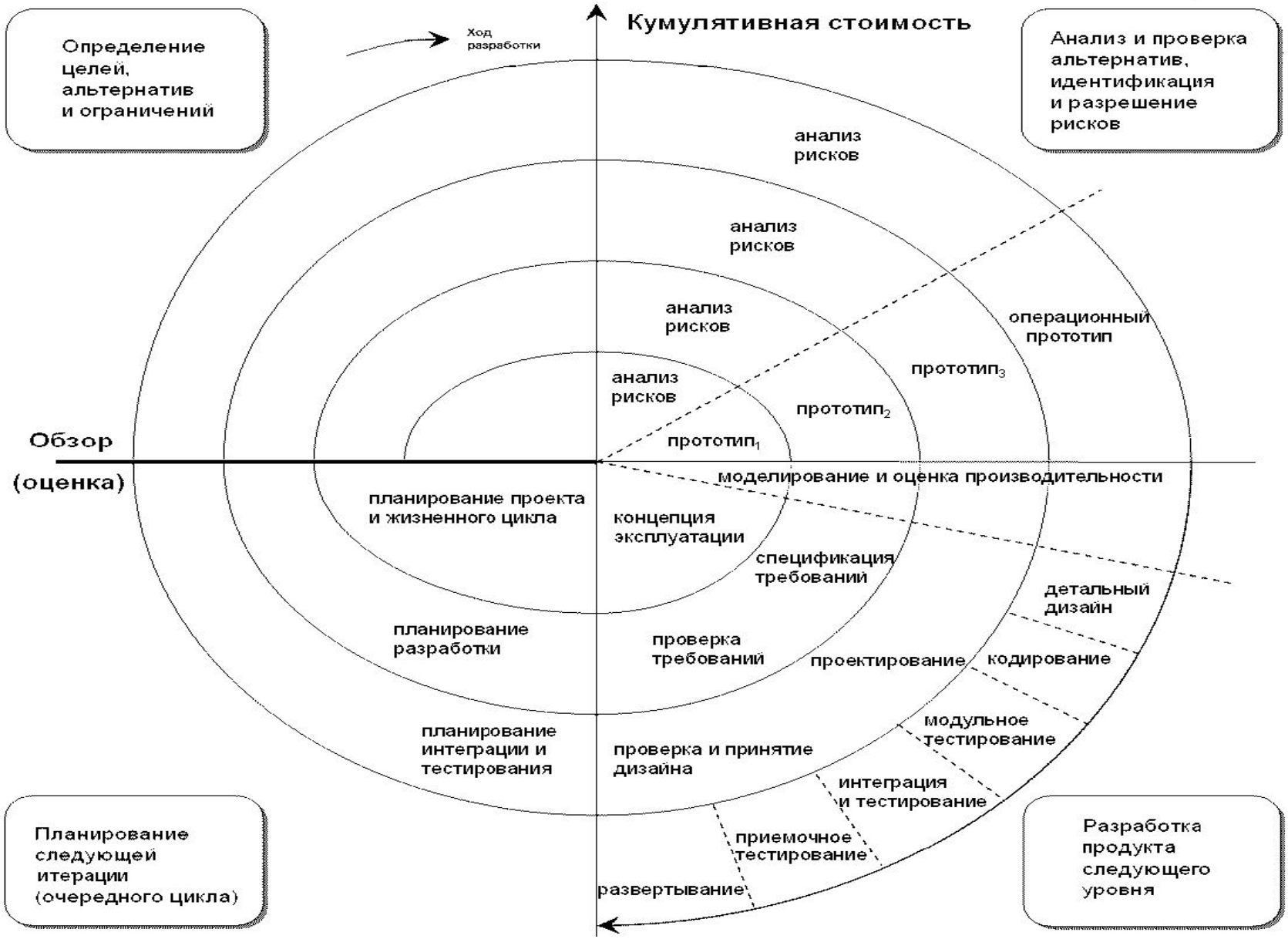
Спиральная модель

- *Спиральная модель* является результатом анализа и адаптации известных моделей: непланируемой, каскадной и прототипируемой. Модель получила своё название из-за графического представления в виде спирали, проходящей через 4 стадии разработки (рис.6.7). Каждое их прохождение есть фаза разработки.

Спиральная модель



Модель Боэма



Модель Боэма

- В графическом представлении модели используются полярные координаты (рис.6.8). При этом в заданный момент времени полярный угол соответствует успешности выполняемого проекта, а полярный радиус, точнее удаление по нему от полюса,– совокупной стоимости разработки.
- Отличительной особенностью этой модели является специальное внимание рискам, влияющим на организацию ЖЦ. *Риск – это некоторое событие или обстоятельство, препятствующее нормальному достижению цели проекта.* Большая часть рисков связана с организационными и процессными аспектами взаимодействия специалистов в команде.

Модель Боэма

Принцип модели (рис.6.8) заключается в разработке ПО путём прототипирования за несколько витков спирали, именуемых итерациями, циклами, фазами.

Каждая итерация состоит из 4 стадий:

1. Определение целей, альтернатив и ограничений;
2. Анализ и проверка альтернатив, идентификация и разрешение рисков;
3. Разработка продукта следующего уровня;
4. Планирование следующей итерации (очередного цикла).

Каждый процесс или группа процессов разработки в рамках итерации предваряются анализом рисков и завершаются проверкой.

6 ключевых практик

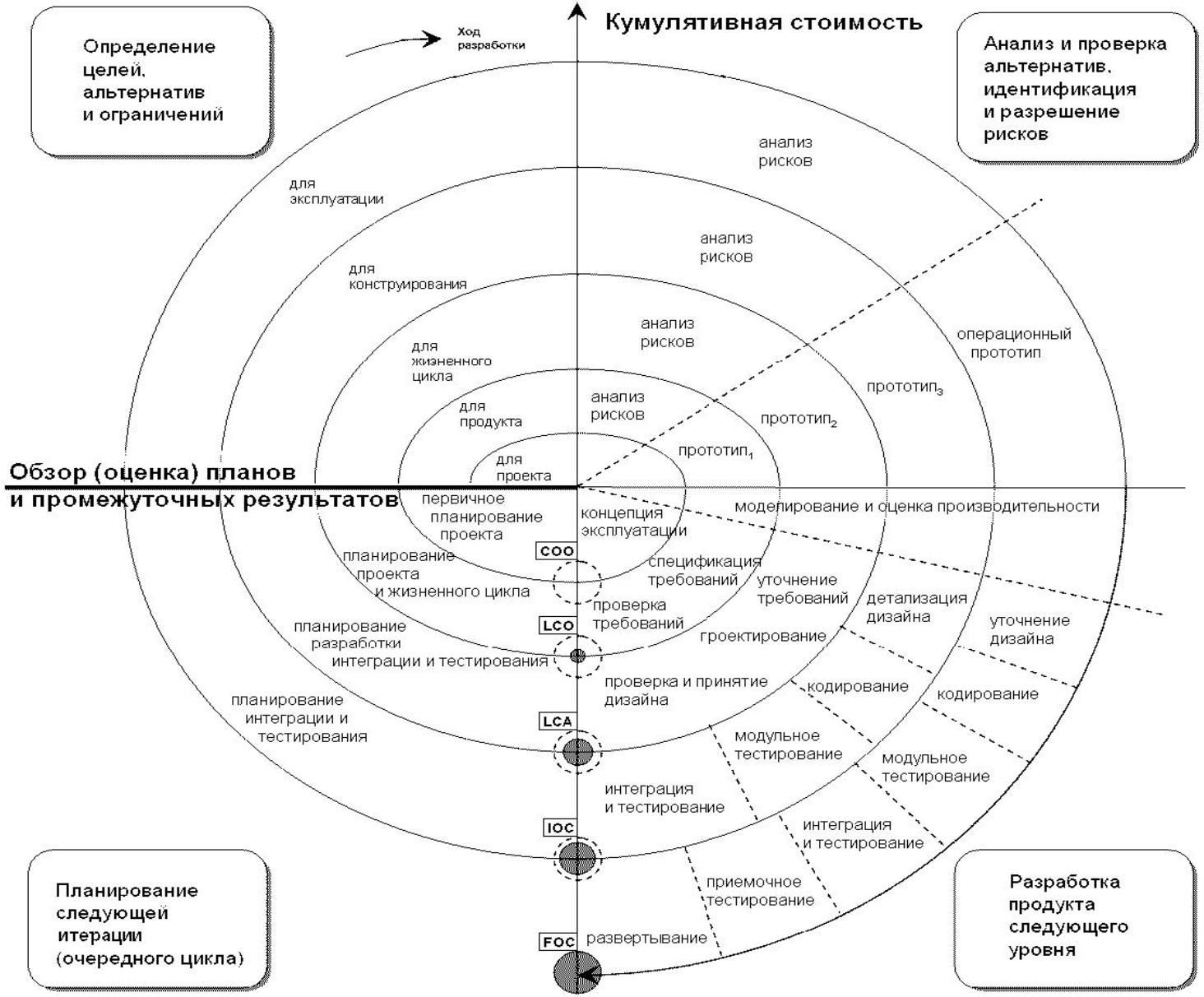
В 2000 г. Боэм на основе опыта использования спиральной модели сформулировал 6 ключевых практик, обеспечивающих успешное её применение:

1. Параллельное, а не последовательное определение артефактов проекта.
2. Согласие в том, что на каждом цикле уделяется внимание: поставленным целям и ограничениям, альтернативам организации процесса и технологических решений, закладываемых в продукт, идентификации и разрешению рисков, оценки со стороны заинтересованных лиц, достижению согласия в том, что можно и необходимо двигаться дальше.
3. Использование соображений, связанных с рисками, для определения уровня усилий, необходимого для каждой работы на всех циклах спирали.
4. Использование соображений, связанных с рисками, для определения уровня детализации каждого артефакта, создаваемого на всех циклах спирали.
5. Управление ЖЦ в контексте обязательств на основе 3 контрольных точек:
    1. Цели ЖЦ (LCO);
    2. Архитектура ЖЦ (LCA);
    3. Начальный операционный вариант (IOC).
6. Уделение внимания проектным работам и артефактам ПО и ЖЦ.

Модифицированная спиральная модель

Модифицированная спиральная модель представляет собой один из промежуточных вариантов по уровню детализации. Детализация в этой модели (рис.6.9) связана с уточнением некоторых процессов, увеличением числа итераций при сокращении их длительности и определением контрольных точек.

Модифицированная спиральная модель



Модифицированная спиральная модель

Данная модель содержит следующий общий набор контрольных точек:

1. Концепция эксплуатации (COO);
2. Цели ЖЦ (LCO), включая содержание ЖЦ;
3. Архитектура ЖЦ (LCA), здесь же можно говорить о готовности концептуальной архитектуры целевого ПО;
4. Начальный операционный вариант (IOC) – вариант ПС, готовый для опытной эксплуатации;
5. Конечный операционный вариант (FOC) – вариант ПО в виде продукта, готового для реальной эксплуатации.

Фактически получается эволюционный ЖЦ в форме спиральной модели.