1. **Дайте определение понятия «жизненный цикл программного средства».**

Период, начинающийся с момента принятия решения о необходимости создания средства программного решения, и заканчивающийся в момент полного изъятия программы из эксплуатации

1. **Укажите основные отличительные особенности жизненного цикла Программного Средства по сравнению с жизненным циклом технических объектов.**
2. В начале ЖЦ программного обеспечения не удаётся чётко определить неизменный набор требований;
3. Этап серийного производства программного обеспечения не требует значительных материальных затрат;
4. Физического износа программного обеспечения не происходит;

Моральное устаревание:

- перестала удовлетворять актуальным требованиям;

- дальнейшая модификация нецелесообразная;

1. **Дайте определения следующих понятий, относящихся к элементам деятельности:** Ресурсы и материалы, Средства и инструменты, Методы

**Ресурсы и материалы** – ответ на вопрос **ИЗ ЧЕГО** продуцируется результат

- **Ресурсы** – аспект **расходования** (время, электроэнергия, оборудование)

- **Материалы** – то, что используется и перерабатывается, **но без расходования** (сведения о предметной области, требования)

**Средства и инструменты** – **С ПОМОЩЬЮ ЧЕГО** продуцируется результат

- **Средства** – более **общее** понятие (системы программирования, CASE-средства)

- **Инструменты** – **узко** **спец**иализированные средства (Word, Notepad)

**Методы** – способ выполнения некоторой деятельности

- **приёмы** – разные методологии (ООП, функциональное программирование)

- **разные предписания и соглашения** (разные протоколы)

- **регламенты,** ограничивающие / запрещающие

1. **Дайте определение понятия «модель жизненного цикла программного средства». Перечислите известные Вам модели жизненного цикла ПС.**

Модель ЖЦ – структура, которая определяет **последовательность взаимосвязи** процессов, действий и задач, выполняемых на протяжении ЖЦПО

Делятся на 2 категории:

1. **Иллюстративные** – отображает ЖЦ программного обеспечения, указывают на отдельные его аспекты;

(Обобщённая, Каскадная, Спиральная, Гантера и др.)

1. **Инструментальные** – служат основой для организации всех работ на протяжении ЖЦ программного обеспечения, а также являются средствами систематизации работ в соответствии с производственным процессом

(календарные графики, графики сетевого планирования, сетевые диаграммы и др.)

1. **Опишите основные свойства и возможности инструментальных моделей жизненного цикла ПС**

**Атрибутивность** — с элементами модели связаны определенные атрибуты, необходимые для управления проектом. Эти атрибуты можно задавать или извлекать, т.е. размещать информацию о проекте в некотором хранилище или получать ее из хранилища.

**Расширяемость** — элементы модели допускают пополнение, в результате чего модель становится более детализированной, точнее отражающей реальный процесс.

**Масштабируемость** — возможность увидеть модель с разной степенью детализации, от охвата всего процесса и до конкретной работы.

**Интегрированность с инструментами поддержки.** Это качество не самой модели, а CASE-средств, совместно с которыми она используется.

**Инструментальная модель должна:**

1) давать полную картину процесса разработки проекта (историю развития, текущее состояние и варианты продвижения вперед);

2) предоставлять средства декомпозиции ЖЦПО на этапы, процессы и работы с возможностью их уточнения;

3) предоставлять средства для организации планирования процесса разработки включая:

– установление сроков выполнения работ;

– выделение и отслеживание ресурсной обеспеченности работ;

– распараллеливание производственных операций;

– распределение производственных операций между исполнителями.

1. **Каковы принципиальные особенности Каскадной Модели жизненного цикла ПС? В чем состоят преимущества и недостатки данной модели?**

**Особенности:**

1) **переход с одного этапа на следующий** должен осуществляться только после того, как будет полностью завершена работа на текущем этапе;

2) каждый **этап** **заканчивается** получением некоторых **результатов**, которые служат в качестве **исходных** **данных** для **следующего** **этапа**;

3) при **завершении** каждого этапа должен **выпускаться** **полный комплект документации** (для др. команды разработчиков)

**Преимущества:**

1) на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;

2) выполняемые в логичной последовательности этапы работ позволяют планировать сроки их завершения и соответствующие затраты.

**Недостатки:**

1) определение всех требований к ПС в начале разработки, что в принципе невозможно

2) возврат на этап анализа при проектировании (при обсуждении с пользователями, **появляются новые требования к архитектуре проекта**)

3) этап тестирования, при котором может выясниться, что требования не были достаточно детализированы или их реализация некорректна

4) «Да, это то, что **я просил**, **но** **не** то, что я **хочу**!»

1. **Каковы принципиальные особенности модели жизненного цикла ПС Гантера? В чем состоят преимущества и недостатки данной модели?**

Необходимость определения в модели:

1) контрольных точек, задающих организационно-временные

рамки проекта;

2) организационно-технических (производственных) функций,

которые выполняются при развитии проекта.

Данный подход реализован в модели жизненного цикла ПС

Гантера в виде матрицы «фазы – функции».

Модель Гантера имеет два измерения:

1) **фазовое**, отражающее **этапы** **выполнения** проекта и сопутствующие им события;

2) **функциональное**, показывающее, какие **производственные**

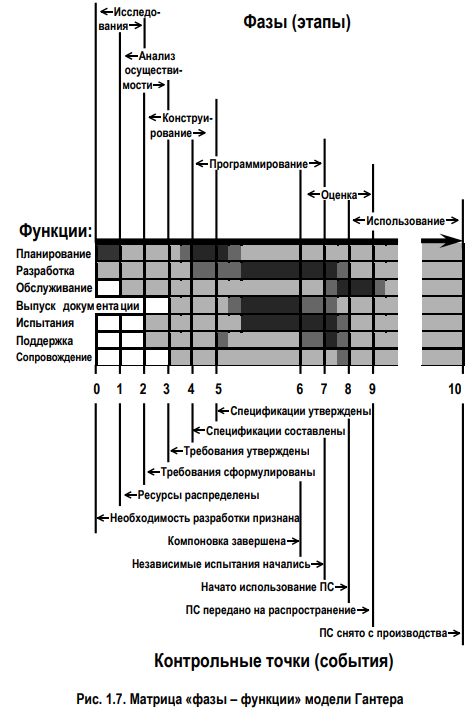
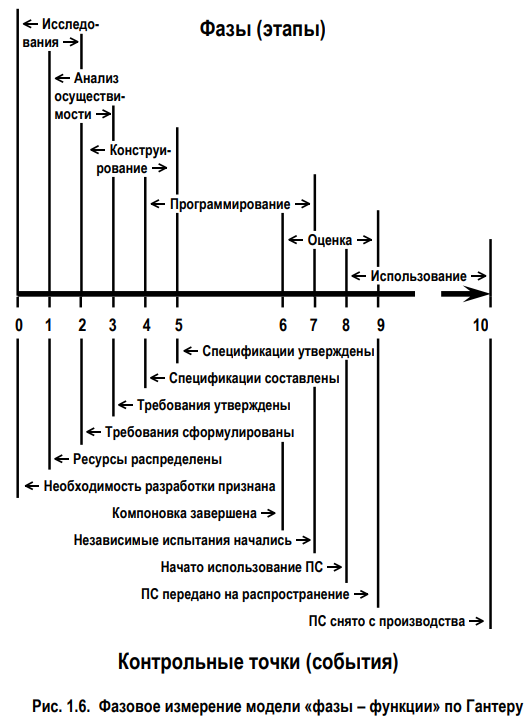
**функции** **выполняются** в ходе развития проекта, и какова **их** **интенсивность** на каждом из этапов.

Модель Гантера учитывает **соотношение** **производственных**

**функций** (технологических операций) и **фаз** **жизненного** цикла (этапов), этим она отличается от других моделей.

В то же время задача отражения итеративности в модели Гантера в явном виде не предусматривается.

Хотя само по себе перекрытие смежных фаз проекта и выпуск соответствующей событиям документации есть путь к минимизации возвратов к выполненным этапам, более содержательные средства описания итераций в модель не закладываются.



1. **Каковы принципиальные особенности спиральной модели жизненного цикла ПС? В чем состоят преимущества и недостатки данной модели?**

Спиральная модель поддерживает **итеративный процесс разработки**: ПС создается **не сразу**, как в каскадной модели, **а по частям** и **возврат** на предыдущие этапы **планируется** **заранее**. Анализ, проектирование, реализация, отладка и установка системы выполняются по несколько раз.

Спиральная модель **в большей степени**, чем каскадная, **гарантирует**, что созданное **ПС оправдает ожидания заказчика**.

**Преимущества:**

1. В каждом цикле принятые решения развиваются, приближаясь к требованиям конечного пользователя (часто даже не высказанным), оставаясь при этом простыми, надежными и открытыми для дальнейшего изменения.

2. Каждый виток спирали соответствует созданию версии ПС. Главная же задача – как можно быстрее показать пользователям очередную версию системы, тем самым, активизируя процесс уточнения и дополнения требований.

3. Разработка итерациями отражает объективно существующий спиральный цикл создания систем. На каждом этапе не требуется полного завершения работ. Это позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем; недостающая работа выполняется на следующей итерации.

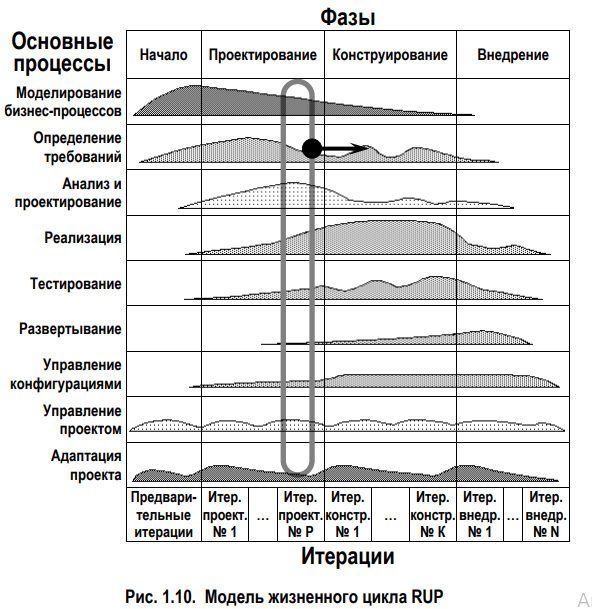
**Недостатки:**

1. Основная проблема – определение момента перехода на следующий этап. Необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом. План составляется на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков.



1. **Каковы принципиальные особенности модели жизненного цикла ПС RUP?**

Модель RUP задается в виде **матрицы интенсивностей функций**, выполняемых на этапах (фазах), которые проецируются на итерации (рис. 1.10).

Авторы CASE-средств, поддерживающих RUP, неизменно подчеркивают иллюстративный стиль изображения интенсивностей. Для каждой итерации можно указать: в какой фазе она находится в данный момент (серый овал на рис. 1.10), а также какая рассматривается функция (жирная точка и стрелка, ведущая к очередной фазе). Для каждой функции показана интенсивность ее выполнении в виде фигур разной формы (больший объем фигуры на определенной фазе свидетельствует о большей интенсивности выполнения функции).

16. Каковы принципиальные особенности адаптивной разработки по Хайсмиту?

Основу ASD составляют три нелинейные, перекрывающие друг друга фазы: обдумывание, сотрудничество и обучение, относящиеся к каждому периоду разработки, который завершается выпуском релиза. При обычном планировании отклонения от плана воспринимаются как ошибки, которые нужно исправлять. В адаптивных разработках считается, что отклонения ведут к решениям, которые объективно обусловлены, следовательно, они являются нормой. Неопределенность в столь непредсказуемой среде преодолевается за счет активного сотрудничества разработчиков. При этом внимание руководства направлено не столько на объяснения, что именно нужно делать, сколько на обеспечение коммуникации, при которой разработчики сами находят ответы на возникающие вопросы. Отсюда следует повышенное внимание к обучению, значение которого в предсказуемых методологиях часто занижается. Хайсмит пишет, что «в адаптивном окружении обучения не избежать всем участникам проекта – и разработчикам, и их заказчикам, поскольку и те и другие в процессе работы должны пересматривать собственные обязательства, а также использовать итоги каждого цикла разработки для того, чтобы подготовиться к следующему».

Автор ASD освещает сложные моменты адаптивных разработок, в частности вопросы обеспечения сотрудничества и обучения во время реализации проекта. И в этом ценность работы Хайсмита, поскольку полученные результаты применимы в самых разных случаях. ASD не является методологией, это некоторая концепция различных адаптивных разработок. Схема жизненного цикла ПС не является определяющим фактором ASD, могут применяться различные методики и стратегии развития проекта.