

Консультации по вопросам к госэкзамену (дополнительная часть) Для кафедр АСВК, АЯ, СКИ и СП

5. Транзакционное управление в СУБД. Методы сериализации транзакций

http://citforum.ru/database/advanced_intro/39.shtml#13

15. Технологии параллельного программирования

Для подготовки к ответу на вопрос №15 используйте материалы раздела «Технологии» сервера НИВЦ МГУ (parallel.ru) и пятой главы учебника Воеводина В.В. и Воеводина Вл. В. «Параллельные вычисления». Обсуждение задачи об «обедающих философах», к которой могут проявить интерес члены ГЭКа, можно найти в книге Эндрюса Г.Р. «Основы многопоточного, параллельного и распределённого программирования».

22. Унифицированный язык моделирования UML. Основные средства языка

План ответа на вопрос:

- Назначение UML;
- Состав диаграмм UML и их назначение:
 - диаграммы вариантов использования (use case diagrams);
 - диаграммы состояний (statechart diagrams);
 - диаграммы классов (class diagrams);
 - диаграммы деятельности (activity diagrams);
 - диаграммы последовательности (sequence diagrams);
 - диаграммы компонентов (component diagrams);
 - диаграммы коммуникации (communication diagrams);
 - диаграммы размещения (deployment diagrams).

См. литературу (1, 2). Примеры диаграмм см. в методических пособиях:

<http://sp.cmc.msu.ru/ooap/exercises.html>

<http://sp.cmc.msu.ru/courses/prak5/umlpracicum.pdf>

и на сайте: <http://www.uml-diagrams.org/>

23. Основы программной инженерии.

План ответа на вопрос:

1. Понятие программного обеспечения и инженерии программного обеспечения.

Определение 1.1: отдельная программа или логически связанная совокупность программ на носителях данных, снабженная программной документацией, называется *программным обеспечением*.

Определение 1.2: Инженерия программного обеспечения или программная инженерия (software engineering) – это, с одной стороны, совокупность инженерных методов и средств создания программного обеспечения и, с другой стороны, дисциплина, изучающая применение строгого систематического подхода к его разработке, эксплуатации и сопровождению. Фундаментальная идея программной инженерии: программирование является формальным процессом, который можно изучать и совершенствовать.

2. Жизненный цикл программного обеспечения.

Основным понятием программной инженерии является жизненный цикл.

Определение 2.1: Жизненный цикл ПО (software life cycle) – это период времени с момента замысла создания ПО, до момента его полного вывода из эксплуатации.

Жизненный цикл регламентируется международным стандартом ISO 12207. В рамках конкретных технологий программирования понятие ЖЦ уточняется, но указанный стандарт не нарушается.

Два аспекта ЖЦ: статический и динамический. В рамках первого представляет собой ЖЦ совокупность процессов.

2.1. Процессы жизненного цикла по стандарту ISO 12207:1995.

Определение 2.2: Процесс ЖЦ – набор взаимосвязанных действий, преобразующих некоторые входные данные и ресурсы в выходные. Каждый процесс характеризуется задачами, методами их решения, действующими лицами. Процессы ЖЦ протекают параллельно. Состав процессов ЖЦ:

- основные (приобретение, поставка, разработка, эксплуатация, сопровождение);
- вспомогательные (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем);
- организационные (управление, создание инфраструктуры, совершенствование, обучение).

Рассмотрим основные процессы жизненного цикла:

Процесс приобретения включает действия заказчика по приобретению ПО, начиная от подготовки заявки, заканчивая приемкой и завершением работ.

Процесс поставки включает в себя действия поставщика-производителя ПО, начиная с подготовки ответа на заявку заказчика, заканчивая проверкой и оценкой поставки.

Процесс разработки включает в себя действия разработчиков: подготовительную работу; анализ требований; проектирование архитектуры ПО; детальное проектирование; кодирование; тестирование; интеграцию; установку; приемку.

Процесс эксплуатации включает в себя следующие действия: эксплуатационное тестирование; эксплуатацию; поддержку пользователей.

Процесс сопровождения включает в себя следующие действия: анализ проблем и запросов на модификацию ПО; проверку и приемку; перенос ПО в другую среду; снятие ПО с эксплуатации.

2.2. Каскадная и итерационная модели жизненного цикла.

Вторая точка зрения на ЖЦ показывает его устройство во времени. В ее рамках ЖЦ представлен моделью.

Определение 2.3: Модель ЖЦ ПО – это структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении всего ЖЦ.

В любой модели ЖЦ рассматривается как совокупность стадий.

Определение 2.4: Стадия ЖЦ – это часть ЖЦ ограниченная временными рамками, по завершении которой достигается определенный важный результат в соответствии с требованиями для данной части.

Определение 2.5: Моменты времени, определяющие границы стадий ЖЦ называются *контрольными точками (вехами)*.

Особенности *каскадной модели*: фиксация требований к системе в начале проекта; переход со стадии на стадию только после полного завершения работ на текущей стадии; недопустимость возврата на пройденные стадии; жесткая привязка процессов ЖЦ к стадиям ЖЦ. Типичный состав стадий каскадной модели:

- I. Формирование требований.
- II. Проектирование
- III. Реализация.
- IV. Тестирование.
- V. Ввод в действие.
- VI. Эксплуатация и сопровождение.
- VII. Снятие с эксплуатации.

Достоинства каскадной модели: в конце каждой стадии проект находится в согласованном и полном состоянии; легко планировать и управлять. Недостатки: позднее обнаружение проблем; избыточность документации; разработка ПО в целом, без использования преимуществ декомпозиции; неравномерная нагрузка на членов группы, работающей над проектом, в ходе ЖЦ. Особенности *итерационных моделей*: процесс разработки разбивается на последовательность шагов (итераций), выполняемых циклически; с каждой пройденной итерацией (витком спирали) ПО наращивается, в него интегрируются новые разработанные части; в конце каждой итерации осуществляется верификация, т.е. проверка соблюдения требований. Достоинства: полный учет требований заказчика, большее его участие в проекте; более равномерная нагрузка на группу; раннее обнаружение проблем и их разрешение по мере возникновения, уменьшение рисков на каждой итерации. Недостатки: сложность планирования; плохая документированность ПО. См. схему спиральной модели ЖЦ.

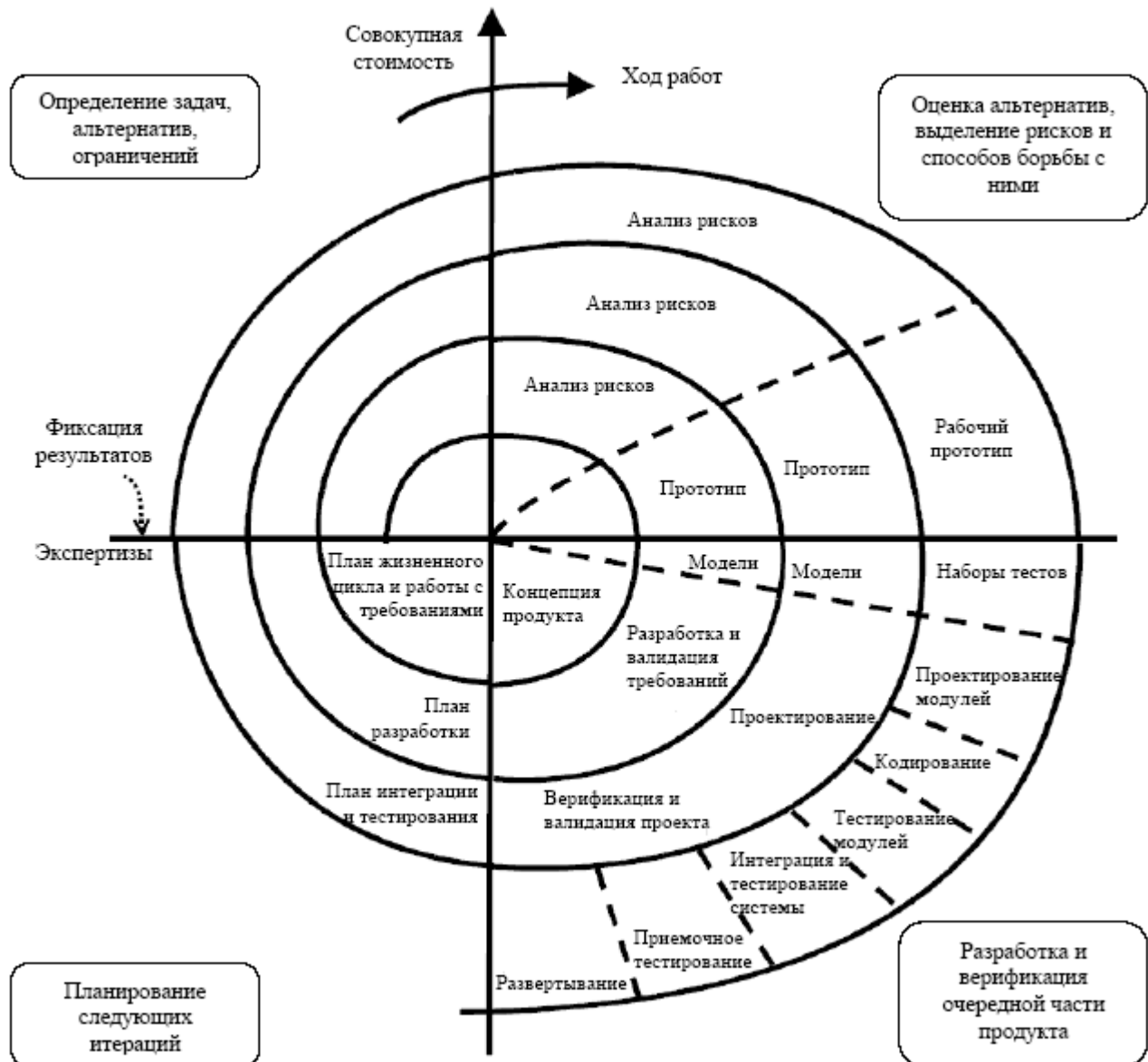


Рис 1. Схема спиральной модели ЖЦ в «полярных координатах». Угол = пройденное время, расстояние = стоимость.

Наряду с каскадной и итерационной моделями ЖЦ примеряется *эволюционная модель*. Она похожа на итерационную тем, что не является водопадной. В эволюционной модели итерации не планируются заранее. Вместо этого вводится рабочий цикл, в котором параллельно ведутся действия по проектированию, разработке и проверке качества. В рамках рабочего цикла

создаются промежуточные версии, до тех пор, пока не будет реализована необходимая функциональность и достигнуто достаточное качество. Тогда рабочий цикл завершается, система поставляется заказчику, вводится в действие и т. п.

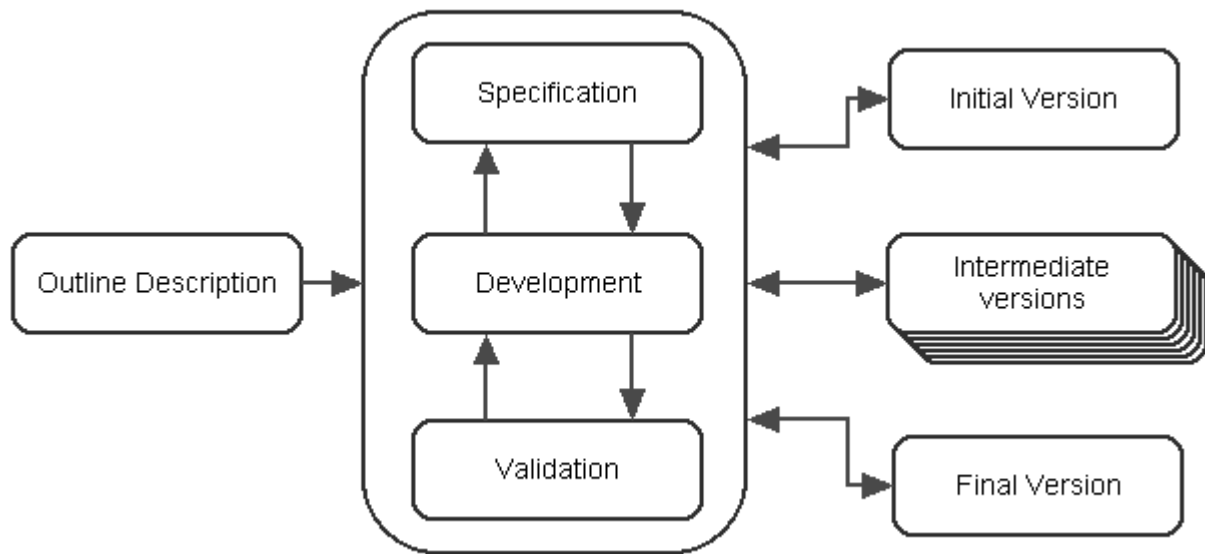


Рис. 2. Схема эволюционной модели ЖЦ.

Литература к вопросам:

1. Фаулер М. UML. Основы. 3-е издание.: Пер. с англ. – СПб: Символ-Плюс, 2004. - 192 с.
2. Конспекты лекций по курсу «Объектно-ориентированный анализ и проектирование» <http://sp.cmc.msu.ru/courses/ooap/>
3. Иан Соммервил. Инженерия программного обеспечения. 6-е издание. М. – СПб. – Киев: 2002. – 623 с