

Другие типы моделей

Существуют некоторые другие типы моделей, которые можно рассматривать, как промежуточные между каскадной и спиральной моделями. Эти модели используют отдельные преимущества каскадной и спиральной моделей и достигают успеха для определенных типов задач.

Итерационная модель ЖЦ(рис.2.6.) является развитием классической каскадной модели и предполагает возможность возвратов на ранее выполненные этапы.

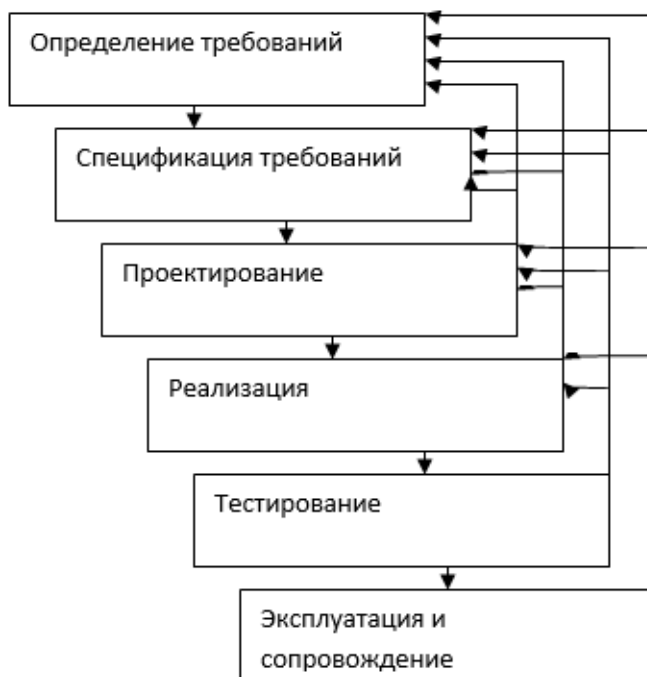


Рис.2.6. Схема итерационной модели ЖЦ ПО

Причинами возвратов в классической итерационной модели являются выявленные ошибки, устранение которых и требует возврата на предыдущие этапы в зависимости от типа ошибки – ошибки кодирования, проектирования, спецификации или определения требований. Реально итерационная модель является более жизненной, чем классическая каскадная модель, т.к. создание ПО всегда связано с устранением ошибок. Практически все применяемые модели ЖЦ имеют итерационный характер, но цели итераций могут быть разными.

V-образная модель (рис.2.7) была связана как итерационная разновидность каскадной модели. Целями итераций в этой модели является обеспечение процесса тестирования. Тестирование продукта обсуждается, проектируется и планируется на ранних этапах ЖЦ разработки. План испытания приемки заказчиком разрабатывается на этапе планирования, а компоновочного испытания системы - на фазах анализа, разработки проекта и т.д. Этот процесс разработки планов испытания обозначен пунктирной линией между прямоугольниками V-образной модели. Помимо планов на ранних этапах разрабатываются также и тесты, которые будут выполняться при завершении параллельных этапов.

Инкрементная (пошаговая) модель (рис 2.8.) представляет собой процесс поэтапной реализации всей системы и поэтапного наращивания функциональных возможностей. На первом шаге необходим полный заранее сформулированный набор требований, которые делятся по некоторому признаку на части. Далее выбирается первая группа требований и выполняется полный проход по каскадной модели.

После того, как первый вариант системы, выполняющий первую группу требований, сдан заказчику, разработчики переходят к следующему шагу (второму инкременту) по разработке варианта, выполняющего вторую группу требований. Особенностью инкрементной модели является разработка приемочных тестов на этапе анализа требований, что упрощает приемку варианта заказчиком и устанавливает четкие цели разработки очередного варианта системы. Кроме того, инкрементная модель для внутренней итерации может использовать не только каскадную, но и другие типы моделей.

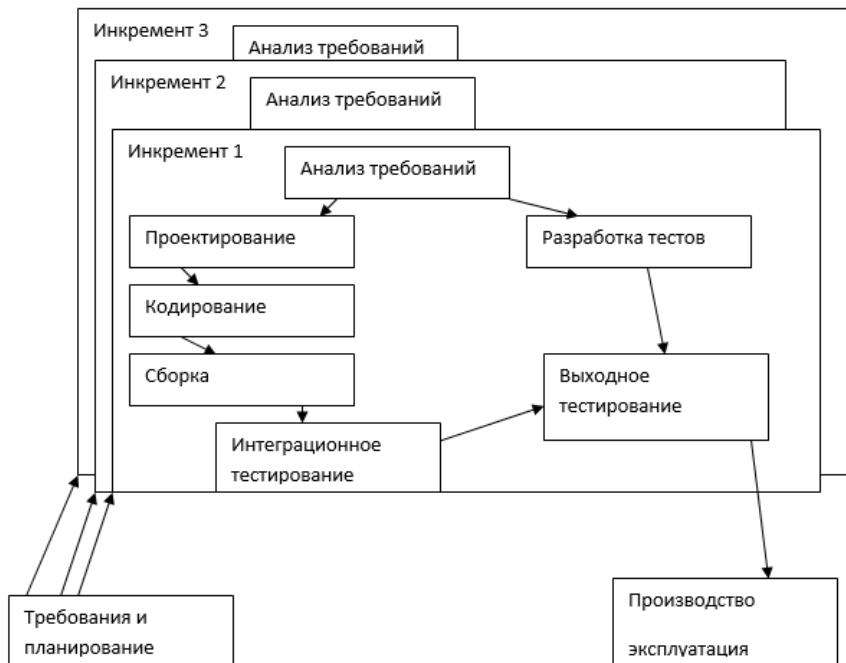


Рис.2.8.Схема инкрементной модели ЖЦ ПО

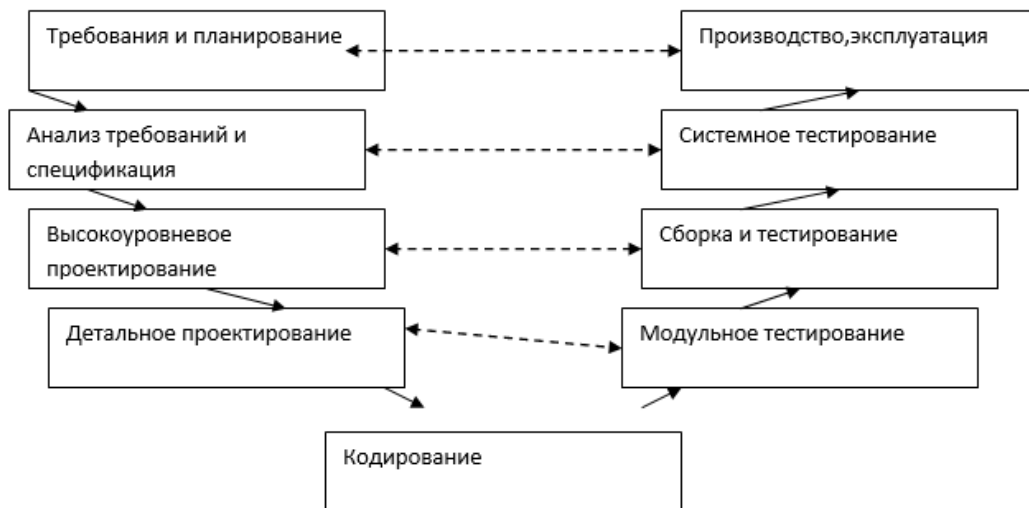


Рис.2.7.Схема V-образной модели ЖЦ ПО

Инкрементная модель особенно эффективна в случае, когда задача разбивается на несколько относительно независимых подзадач (разработка подсистем *Зарплата*, *Бухгалтерия*, *Склад*, *Поставщики

Модель быстрого прототипирования (рис 2.9) предназначена для быстрого создания прототипа продукта с целью уточнения требований и поэтапного развития прототипа в конечный продукт. Скорость выполнения проекта обеспечивается планированием разработки прототипов и участием заказчика в процессе разработки.

Начало ЖЦ разработки помещено в центре эллипса. Совместно с пользователем разрабатывается предварительный план проекта на основе предварительных требований. Результат начального планирования – документ, описывающий в общих чертах примерные графики и результативные данные.

Следующий уровень – создание исходного прототипа на основе быстрого анализа ,проекта база данных,пользовательского интерфейса и некоторых функций.Затем начинается итерационный цикл быстрого прототипирования.Разработчик проекта демонстрирует очередной прототип,пользователь оценивает его функционирование ,совместно определяются проблемы и пути их преодоления для перехода к следующему прототипу. Этот процесс продолжается до тех пор ,пока пользователь не согласится,что очередной прототип в точности отображает все требования. Получив одобрение пользователя ,быстрый прототип преобразуют в детальный проект и систему настраивают на производственное использование,именно на этом этапе настройки ускоренный прототип становится полностью действующей системой.

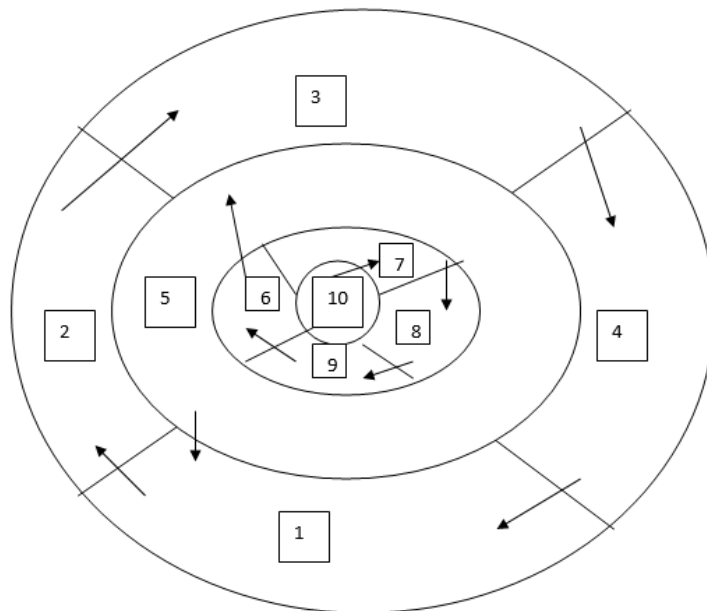
При разработке производственной версии программы может понадобиться более высокий уровень функциональных возможностей ,различные системные ресурсы,необходимые для обеспечения полной рабочей нагрузки или ограничения во времени. После этого следуют тестирование в предельных режимах, определение измерительных критериев и настройка,а затем,как обычно, функциональное сопровождение.В настоящее время широкое применение получают так называемые промышленные технологии создания программного продукта .Эти технологии были разработаны фирмами ,накопившими большой опыт создания ПО.Такие технологии,как правило, поддерживаются набором CASE-средств,охватывают все этапы ЖЦ ПО и успешно применяются для решения практических задач.Эти технологии подробно рассматриваются в курсах *Технологии разработки ПО* ,*САПР ПО*.

Рис.2.9.Схема модели быстрого прототипирования

1 – эксплуатация и сопровождение

2 – утверждение пользователем

- 3- производная разработка
- 4- подгонка
- 5- итеративное прототипирование
- 6- функции
- 7-быстрый анализ
- 8-создание базы данных
- 9-пользовательский интерфейс
- 10-план проекта



Життєвий цикл програмного забезпечення

ЖЦ ПЗ – це період часу, який починається з вирішення питання про розроблення ПЗ і закінчується припиненням використання ПЗ(4).

ЖЦ ПЗ описується декількома способами:

- моделлю
- групою стандартів

У загальному випадку життєвий цикл програмного забезпечення описується в формі методологій, що визначає комплекс робіт, задач та процесів. Найпоширенішими моделями є:

Каскадна – передбачає послідовність виконання робіт на всіх етапах проекту в строго визначеному порядку.

Спіральна – має ітераційний характер, та передбачає перехід на наступний етап, не вимагаючи завершення робіт на поточному.

Еволюційна – послідовно розробляється з блоків конструкцій, де вимоги встановлюються частково та уточнюються на подальших блоках(5).

Проте велика увага приділяється стандартам, націленим на забезпечення якості. Вважається, що стандартизація ЖЦ ПЗ є інструментом забезпечення якісної розробки, експлуатації та супроводу. Стандарти у сфері розробок ПЗ поділяють на три групи: Єдиної системи програмної документації; пов'язані з оцінкою показників якості програмного забезпечення; розроблені на базі відповідних міжнародних стандартів.

Використання процесів є найпоширенішим способом організації ЖЦ ПЗ. Зазвичай такий спосіб представлення ЖЦ містить декілька моделей, що описуються блоками, та належать до різних галузей знань і виконуються паралельно(6).

ЖЦ веб-сайта подібний до циклу розробки програмного забезпечення, проте має інші фази. Розглянемо фази стандартного веб-дизайну, які можуть бути підлаштовані розробником відповідно до меж проекту:

- Аналіз
- Характеристики
- Дизайн та розробка
- Наповнення контентом
- Кодування та тестування
- Просування
- Технічне обслуговування та модернізація(7)

Проте деякі розробники подають спрощений ЖЦ, який складається лише з чотирьох етапів:

- Проектування

- Дизайн

- Верстка

- Програмування(8)