Модели ЖЦ ПО

Подготовила студентка 3-2П9
Пономарева Дарья

Программное обеспечение

Програ́ммное обеспе́чение (ПО) — программа или множество программ, используемых для управления компьютером.

Жизненный цикл ПО

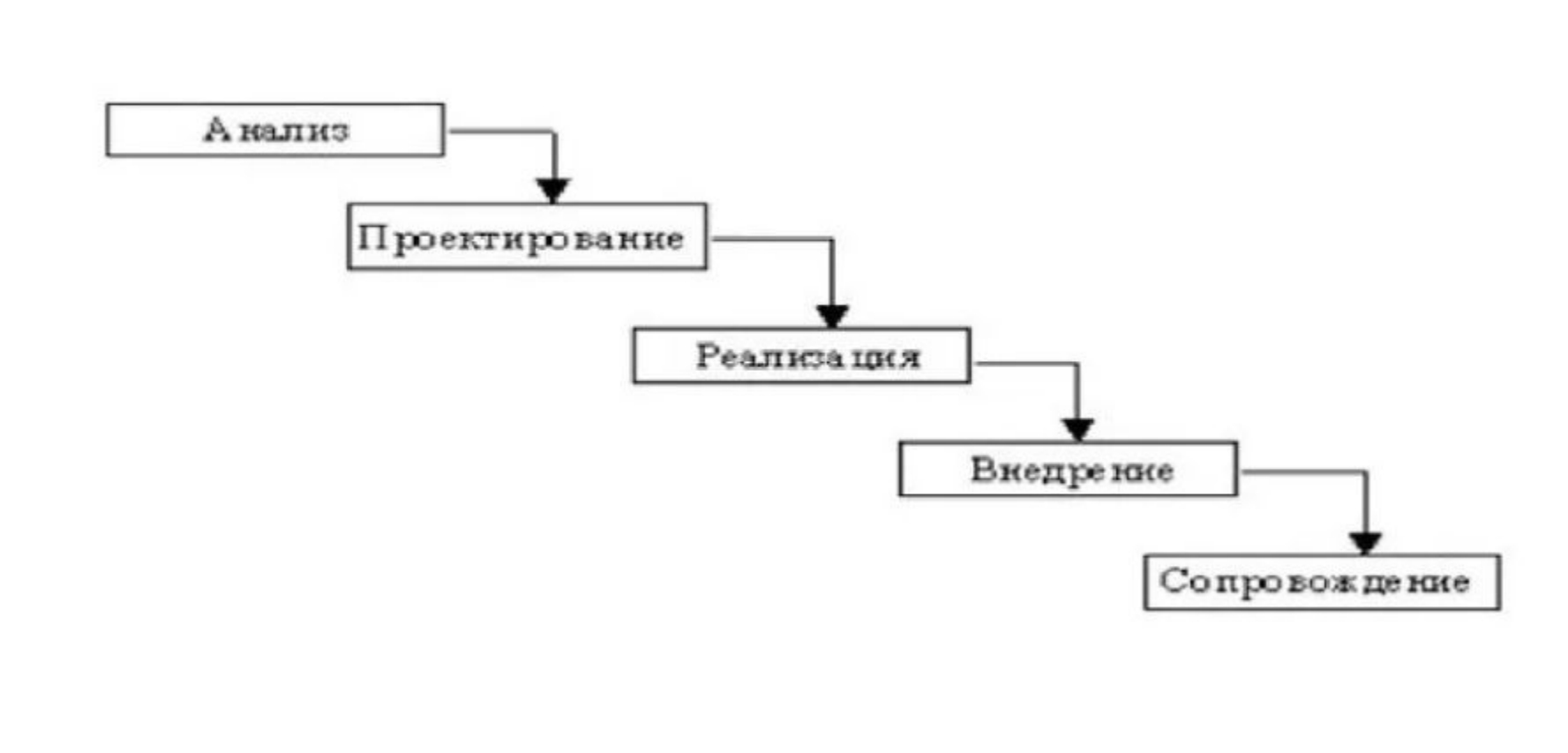
Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Модель жизненного цикла ПО

Модель жизненного цикла ПО — структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

Каскадная модель (водопад):

- завершение каждого этапа проверкой полученных результатов с целью устранить как можно большее число проблем, связанных с разработкой изделия;
- циклическое повторение пройденных этапов (как в классической модели).



Плюсы

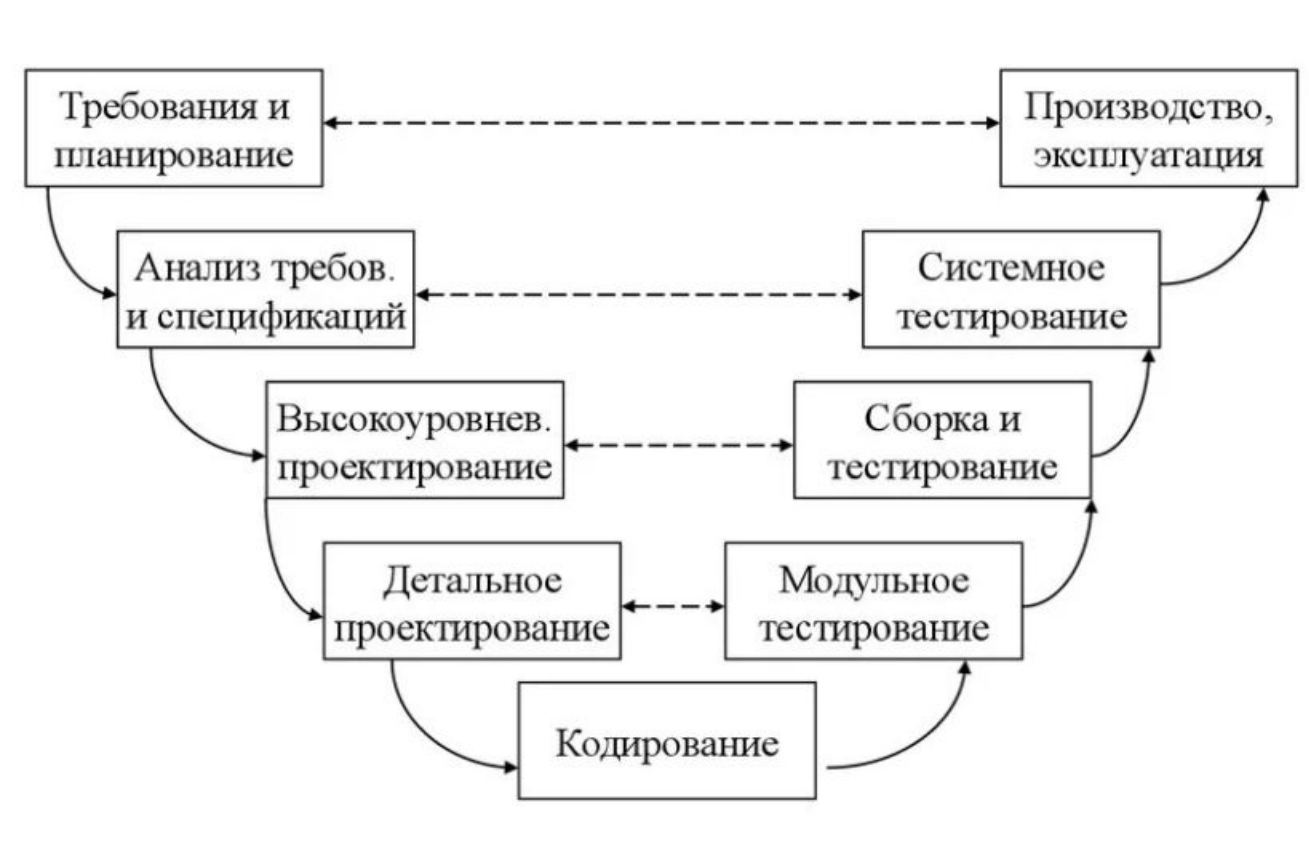
- все стадии проекта выполняются в строгой последовательности;
- строгость этапов позволяет планировать сроки завершения всех работ и соответствующие ресурсы (денежные и человеческие);
- требования остаются неизменными в течение всего цикла.

Минусы

- сложности при формулировке четких требований и невозможность их изменения;
- тестирование начинается только с середины развития проекта;
- до завершения процесса разработки пользователи не могут убедиться, качествен ли разрабатываемый продукт.

V-образная модель

В этой модели особое значение придается действиям, направленным на **верификацию** и **аттестацию** продукта. Она демонстрирует, что тестирование продукта обсуждается, проектируется и планируется на ранних этапах жизненного цикла разработки.



Плюсы

- строгая этапизация;
- минимизация рисков и устранение потенциальных проблем за счет того, что тестирование появляется на самых ранних стадиях;
- усовершенствованный тайм-менеджмент

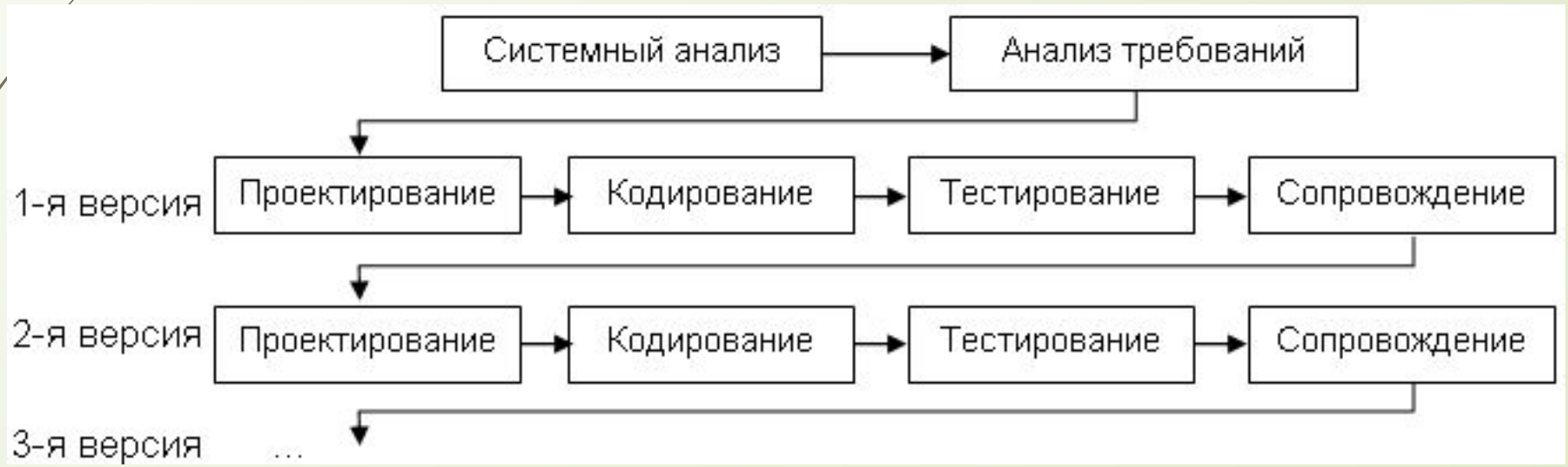
Минусы

- невозможность адаптироваться к измененным требованиям заказчика;
- длительное время разработки приводит к тому, что продукт может быть уже не нужен заказчику, поскольку его потребности меняются;
- нет действий, направленных на анализ рисков.

Инкрементная модель

ПО разрабатывается с линейной последовательностью стадий, но в несколько инкрементов (версий). Таким образом улучшение продукта проходит запланированно все время, пока жизненный цикл разработки ПО не завершится.

Требования к системе определяются в самом начале работы, после чего процесс разработки проводится в виде последовательности версий, каждая из которых является законченным и работоспособным продуктом.



Плюсы

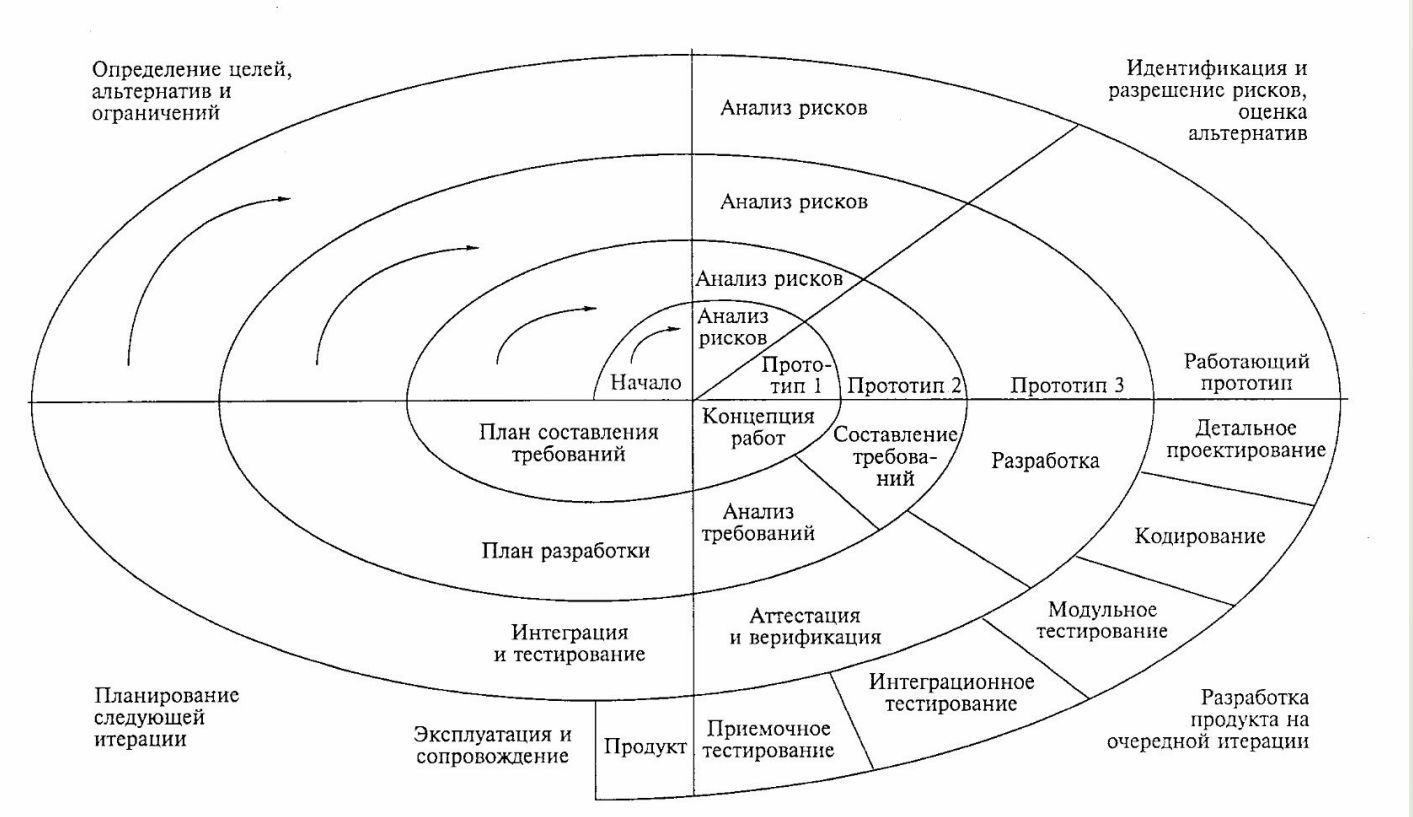
- заказчик может дать свой отзыв касательно каждой версии продукта;
- есть возможность пересмотреть риски, которые связаны с затратами и соблюдением графика;
- привыкание заказчика к новой технологии происходит постепенно.

Минусы

- функциональная система должна быть полностью определена в начале жизненного цикла для выделения итераций;
- при постоянных изменениях структура системы может быть нарушена;
- сроки сдачи системы могут быть затянуты из-за ограниченности ресурсов (исполнители, финансы).

Спиральная модель

Весь процесс создания конечного продукта представлен в виде условной плоскости, разбитой на 4 сектора, каждый из которых представляет отдельные этапы его разработки. На выходе из очередного витка мы должны получить готовый протестированный прототип. Прототип, удовлетворяющий всем требованиям – готов к релизу. Концентрация на возможных рисках.



Плюсы

- управлению рисками уделяется особое внимание;
- дополнительные функции могут быть добавлены на поздних этапах;
- есть возможность гибкого проектирования.

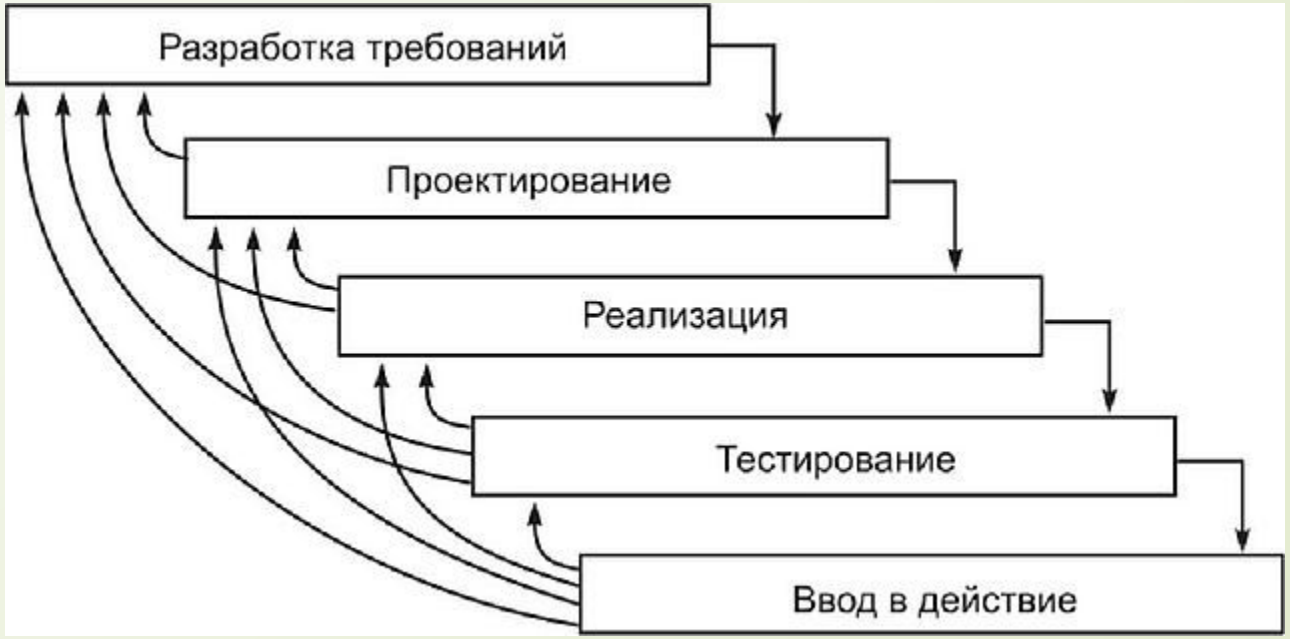
Минусы

- оценка рисков на каждом этапе является довольно затратной;
- постоянные отзывы и реакция заказчика может провоцировать все новые и новые итерации, которые могут приводить к временному затягиванию разработки продукта;
- более применима для больших проектов.

Итерационная модель

Итерационная модель предполагает разбиение проекта на части и прохождение этапов жизненного цикла на каждом их них. Каждый этап является законченным сам по себе, совокупность этапов формирует конечный результат.

С каждым этапом разработка приближается к конечному желаемому результату или уточняются требования к результату по ходу разработки, и соответственно в любой момент текущая итерация может оказаться последней или очередной на пути к завершению.



Плюсы

- позволяет бороться с неопределенностью, снимая ее этап за этапом, и проверять правильность технического, маркетингового или любого другого решения на ранних стадиях;
- снижает риски глобального провала и растраты всего бюджета, получение несинхронизированных ожиданий и ошибочного понимания процессов;
- дает возможность завершения разработки в конце любой итерации.

Минусы

- целостное понимание возможностей и ограничений проекта очень долгое время отсутствует;
- при итерациях приходится отбрасывать часть сделанной ранее работы.

Модель быстрой разработки RAD

Представляет собой инкрементную модель, в которой множество разработок маленьких кусков выбираются и развиваются одновременно для достижения большей картины. Эти куски затем разрабатываются индивидуально.

Плюсы

- быстрое развитие продукта;
- разработка многоразовых мелких компонентов;
- повторный обзор в процессе разработки;
- интеграция повторно используемых компонентов на начальном уровне, следовательно, экономит усилия, несмотря на то, что не добавляются более крупные модули;
- конструктивная реакция.

Минусы

- требуется много усилий для сбора всех требований на начальном этапе.
- навыки моделирования имеют много зависимостей.
- не подходит для малобюджетного проекта.