

Спиральная модель ЖЦ: достоинства и недостатки

Журавлев Николай Вадимович, студент
МГТУ им. Н.Э. Баумана

История создания

Спиральная модель была впервые описал Барри Боэм в своей статье 1986 года «Спиральная модель разработки и улучшения программного обеспечения»

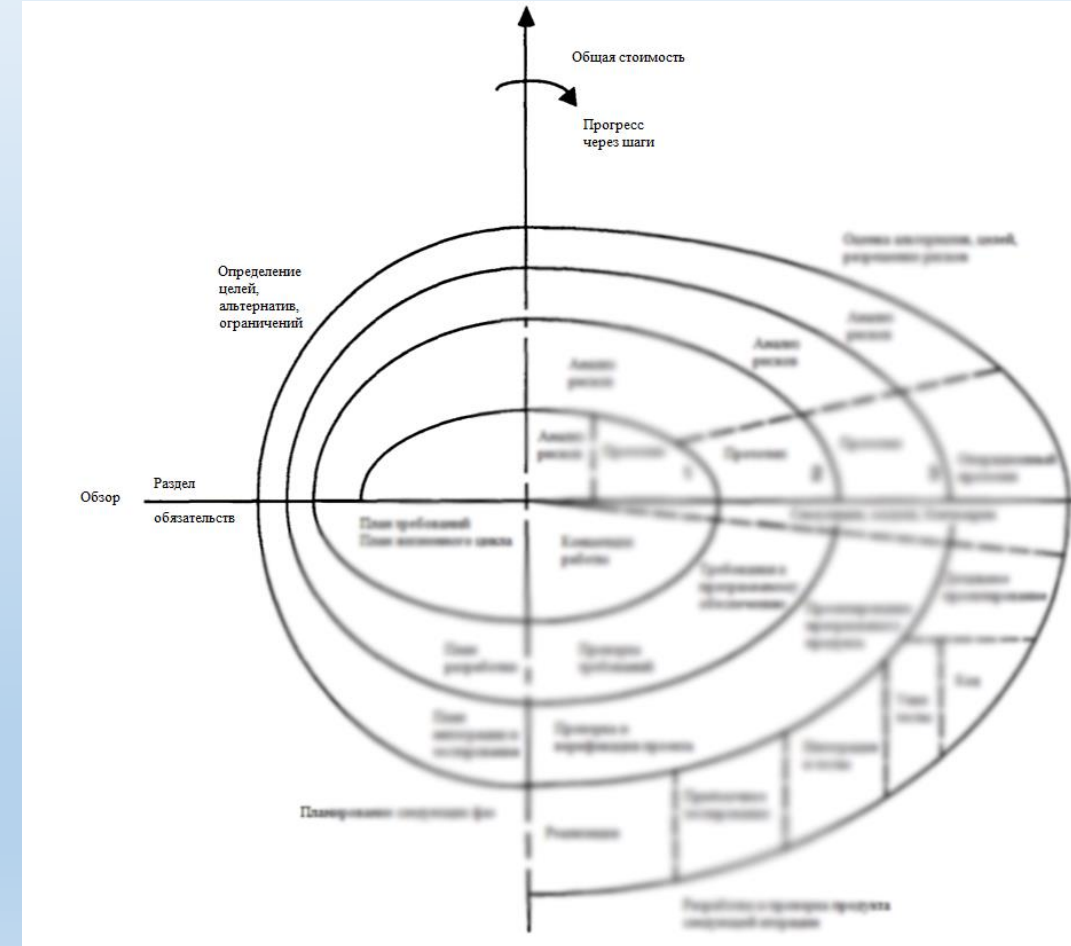
Главной отличительной чертой спиральной модели является то, что она создает подход к процессу разработки ПО, основанный на оценке рисков



Определение целей, альтернатив, ограничений

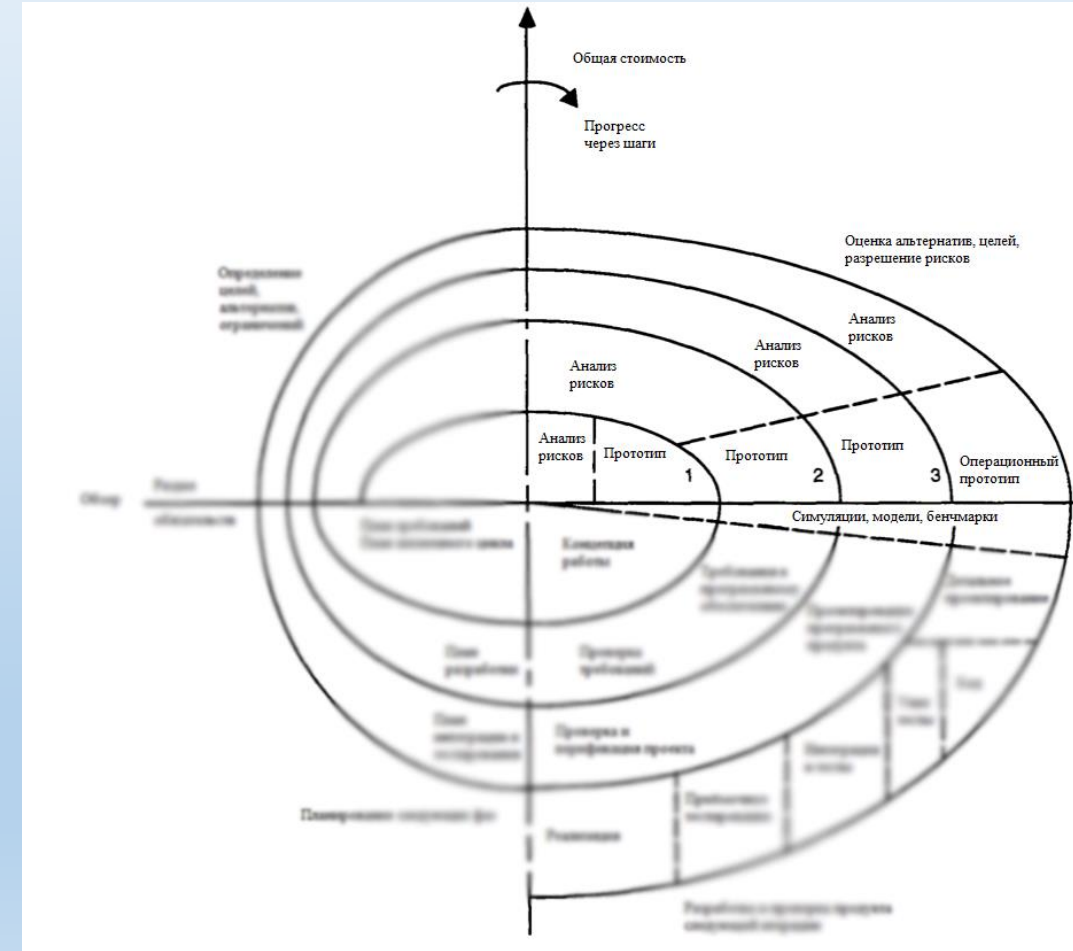
Каждый цикл спирали начинается с определения следующих пунктов:

- цель части разрабатываемого продукта
- альтернативные средства реализации этой части продукта
- ограничения, налагаемые на альтернативы



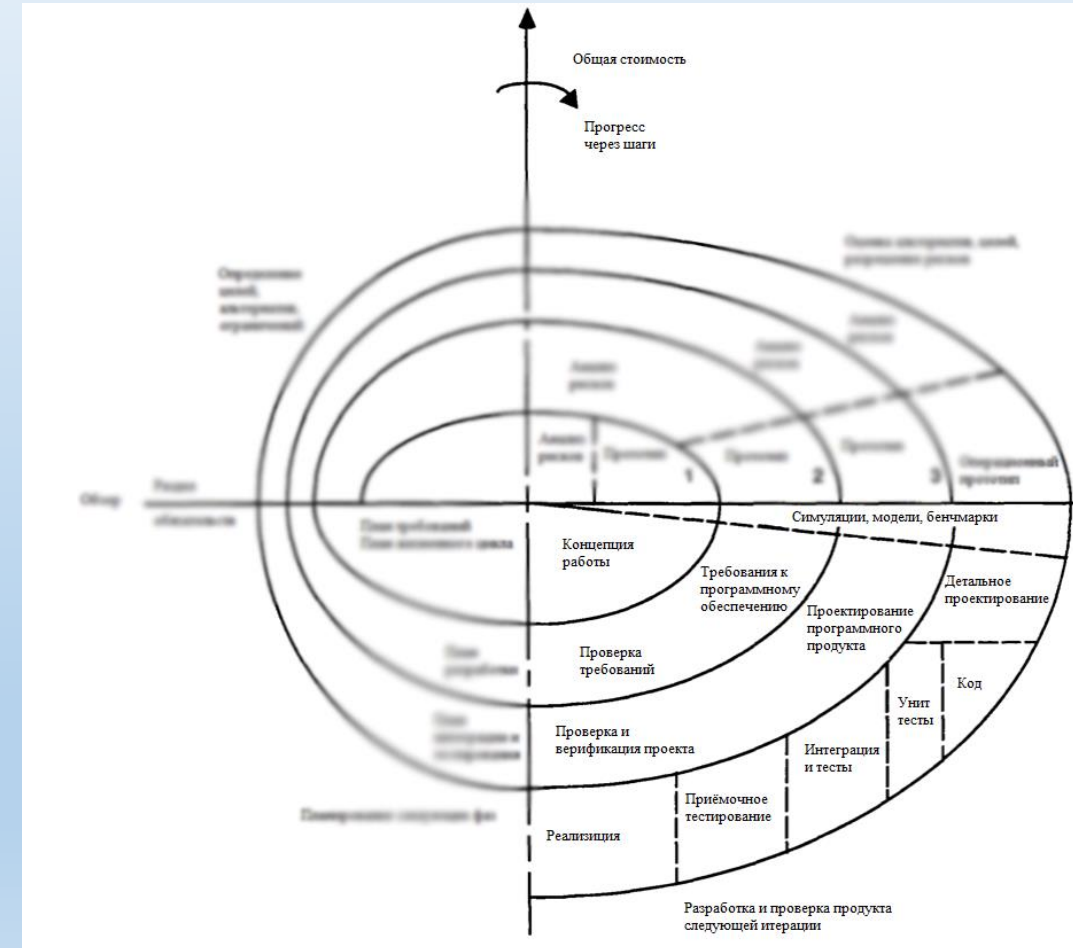
Оценка альтернатив, целей, разрешение рисков

- Эта фаза выявляет области неопределенности, которые являются значительными источниками проектного риска.
- В этой фазе результатом является создание прототипа и его оценка



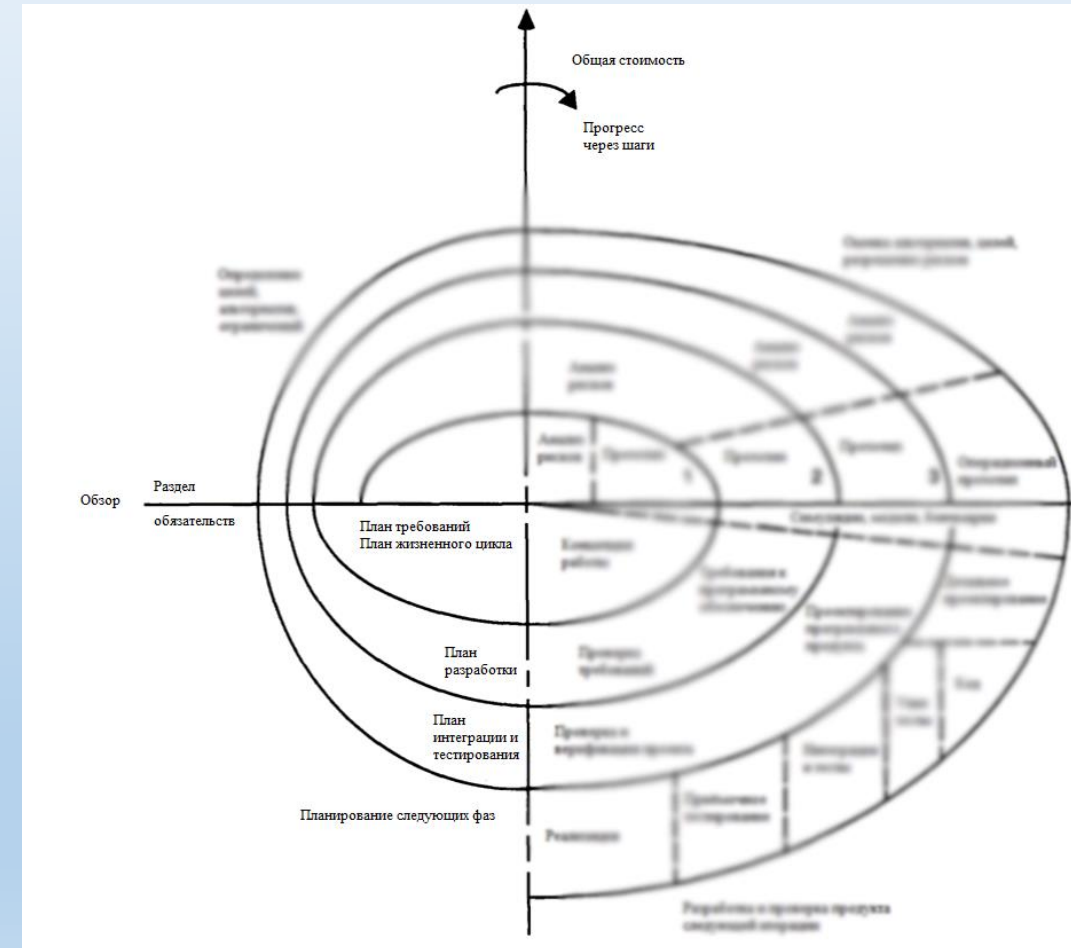
Разработка и проверка продукта следующей итерации

- В данной фазе идёт активная разработка, в результате которой будет законченный для этого этапа продукт.
- Выявленные функции разрабатываются и проверяются посредством тестирования. В конце фазы доступна следующая версия программного обеспечения.



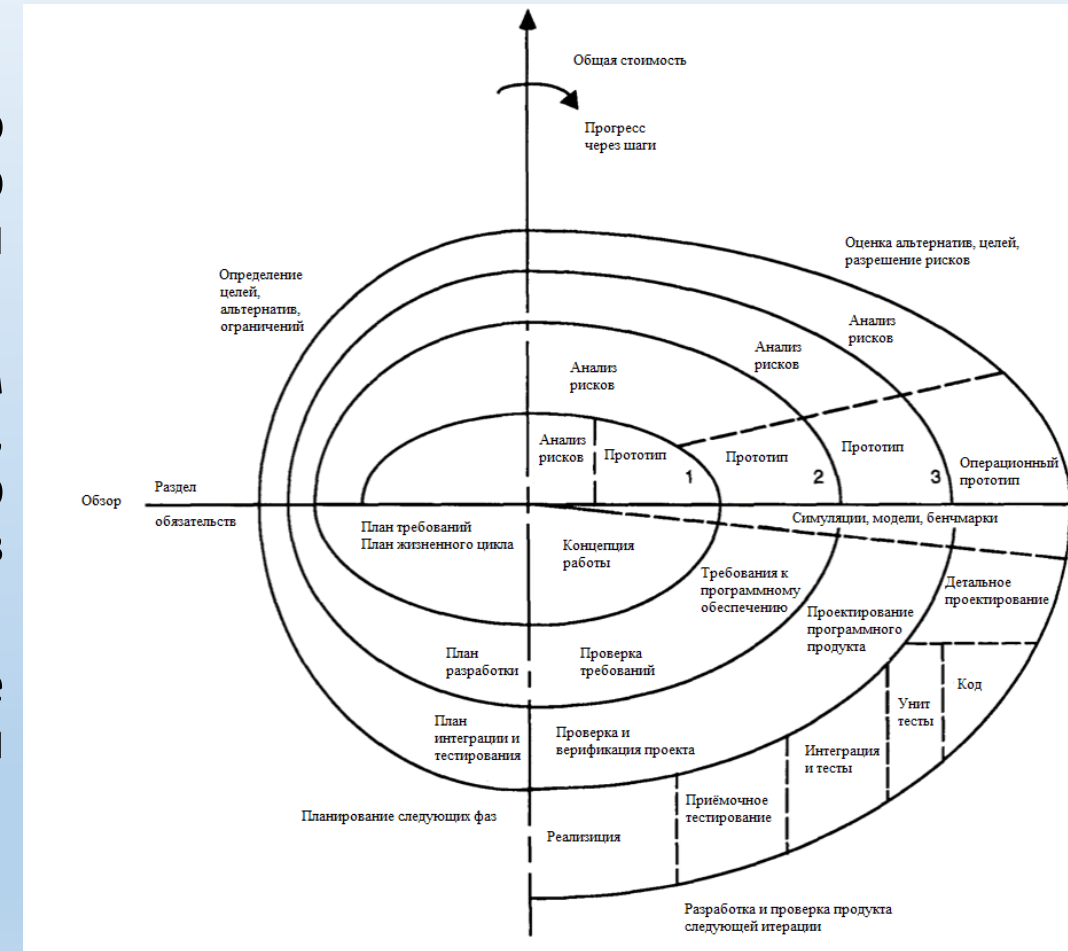
Планирование следующих фаз

- В данной фазе заказчики оценивают уже разработанную версию программного обеспечения. В конце концов, начинается планирование следующего этапа.
- Планы для последующих фаз могут также включать декомпозицию продукта для последующей разработки или компоненты



Переход между витками

- Сама спираль начинается с гипотезы о том, что конкретная операционная миссия (или набор миссий) может быть улучшена с помощью усилий по разработке программного обеспечения.
- Каждая фаза завершается обзором с участием основных людей или организаций, заинтересованных в продукте. Этот обзор охватывает все продукты, разработанные в течение предыдущего витка
- Основная цель обзора - гарантировать, что все заинтересованные стороны взаимно готовы для прохождения следующей фазы



Преимущество

- Ориентация на риски
- На ранних этапах она фокусирует внимание на вариантах, включающих повторное использование существующего программного обеспечения
- Она обеспечивает подготовку к эволюции жизненного цикла, росту и изменениям программного продукта.
- Она предоставляет механизм для включения целей качества программного обеспечения в разработку программного продукта.
- Он фокусируется на устранении ошибок и непривлекательных альтернатив на раннем этапе
- Для каждого из источников проектной деятельности и расходов ресурсов он отвечает на ключевой вопрос: «Сколько достаточно?»
- Он обеспечивает жизнеспособную основу для разработки интегрированной аппаратно-программной системы



Недостатки

- Соответствие контрактному программному обеспечению
- Спиральная модель во многом полагается на способность разработчиков программного обеспечения выявлять и управлять источниками риска проекта
- Спецификация, ориентированная на риск, также будет зависеть от людей
- В целом, шаги процесса спиральной модели требуют дальнейшей разработки, чтобы гарантировать, что все участники разработки работают в согласованном контексте



Применение

- Если проект имеет низкий риск в таких областях, как получение неправильного пользовательского интерфейса или несоответствие строгим требованиям к производительности, то эти соображения риска приводят спиральную модель к эквивалентности каскадной модели.
- Если требования к программному продукту очень, и если наличие ошибок в программном продукте представляет высокий риск для миссии, которой он служит, то эти соображения о рисках приводят спиральную модель к сходству с двухсторонней моделью точной спецификации и формальной дедуктивной разработки программы.
- Если проект имеет низкий риск в таких областях, как потеря предсказуемости и контроля бюджета и графика, и если он имеет высокий риск в таких областях, как получение неправильного пользовательского интерфейса или требований поддержки принятия решений пользователем, то эти соображения о рисках приводят спиральную модель к эквивалентности модели эволюционного развития.
- Если доступны возможности автоматизированной генерации программного обеспечения, то спиральная модель учитывает их либо как варианты быстрого прототипирования, либо как применения модели преобразования, в зависимости от связанных с этих рисков.

Список использованных источников

- Boehm B. W. A Spiral Model of Software Development and Enhancement // Computer, Volume: 21, Issue: 5, May 1988, P. 61—72.
- Спиральная модель и архитектура разработки программного обеспечения [сайт]. – URL: <https://janberg.by/spiral-model-spiralnaya-model-i-arxitektura-razrabotki-programmnogo-obespecheniya/>
- Жизненный цикл разработки ПО (SDLC). Спиральная модель [сайт]. – URL: <https://xbsoftware.ru/blog/zhiznennyj-tsykl-razrabotki-spiral/>

Спасибо за внимание!