

# Цикл жизни программного обеспечения

Мухортова Н.Н.

# Цель

Цель - познакомиться с разными моделями жизненного цикла программного обеспечения.

# Жизненный цикл программного обеспечения

**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

# Каскадная модель

Каскадная модель (англ. waterfall model) — модель процесса разработки программного обеспечения, жизненный цикл которой выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.

# Основные этапы

1. Анализ требований,
2. Проектирование,
3. Кодирование (программирование),
4. Тестирование и отладка,
5. Эксплуатация и сопровождение.

# Каскадная модель



# Область применения Каскадной модели

- при разработке проектов с четкими, неизменяемыми в течение жизненного цикла требованиями, понятными реализацией и техническими методиками;
- при разработке проекта, ориентированного на построение системы или продукта такого же типа, как уже разрабатывались разработчиками ранее;
- при разработке проекта, связанного с созданием и выпуском новой версии уже существующего продукта или системы;
- при разработке проекта, связанного с переносом уже существующего продукта или системы на новую платформу;
- при выполнении больших проектов, в которых задействовано несколько больших команд разработчиков.

# Инкрементная модель

Инкрементная модель (англ. increment — увеличение, приращение) подразумевает разработку программного обеспечения с линейной последовательностью стадий, но в несколько инкрементов (версий), т.е. с запланированным улучшением продукта за все время пока Жизненный цикл разработки ПО не подойдет к окончанию.



# Инкрементная модель



# Достоинства

затраты, которые получаются в связи с изменением требований пользователей, уменьшаются, повторный анализ и совокупность документации значительно сокращаются по сравнению с каскадной моделью;

легче получить отзывы от клиента о проделанной работе — клиенты могут озвучить свои комментарии в отношении готовых частей и могут видеть, что уже сделано. Т.к. первые части системы являются прототипом системы в целом.

у клиента есть возможность быстро получить и освоить программное обеспечение — клиенты могут получить реальные преимущества от системы раньше, чем это было бы возможно с каскадной моделью.

# Недостатки

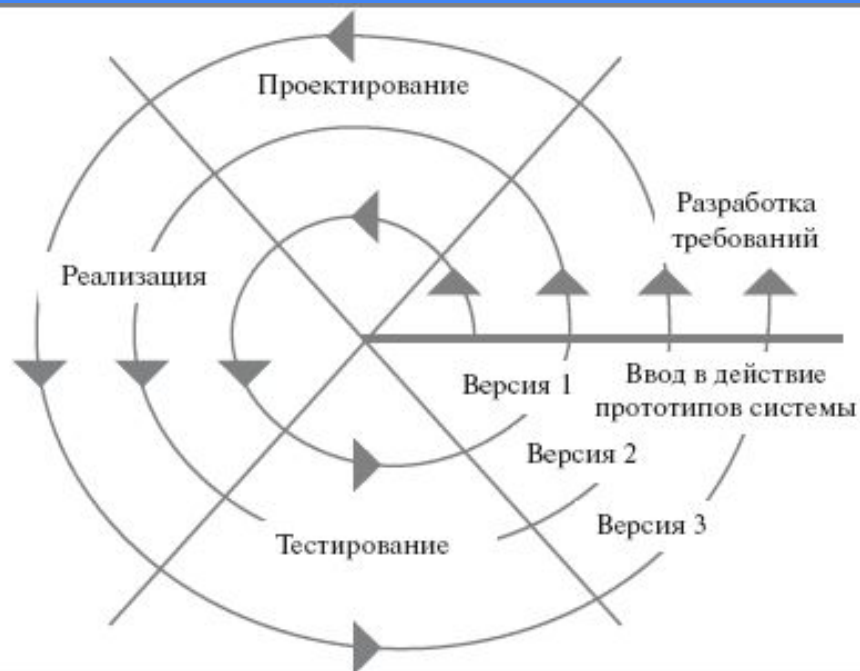
менеджеры должны постоянно измерять прогресс процесса. в случае быстрой разработки не стоит создавать документы для каждого минимального изменения версии;

структура системы имеет тенденцию к ухудшению при добавлении новых компонентов — постоянные изменения нарушают структуру системы. Чтобы избежать этого требуется дополнительное время и деньги на рефакторинг. Плохая структура делает программное обеспечение сложным и дорогостоящим для последующих изменений. А прерванный Жизненный цикл ПО приводит еще к большим потерям.

# Спиральная модель

Спиральная модель: Жизненный цикл — на каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки — анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов.

# Спиральная модель



# Спиральная модель

Данная модель представляет собой процесс разработки программного обеспечения, сочетающий в себе как проектирование, так и поэтапное прототипирование с целью сочетания преимуществ восходящей и нисходящей концепции, делая упор на начальные этапы жизненного цикла: анализ и проектирование. Отличительной особенностью этой модели является специальное внимание рискам, влияющим на организацию жизненного цикла.

# Достоинства

позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;

допускает изменение требований при разработке программного обеспечения, что характерно для большинства разработок, в том числе и типовых;

в модели предусмотрена возможность гибкого проектирования, поскольку в ней воплощены преимущества каскадной модели, и в то же время разрешены итерации по всем фазам этой же модели;

позволяет получить более надежную и устойчивую систему. По мере развития программного обеспечения ошибки и слабые места обнаруживаются и исправляются на каждой итерации;

эта модель разрешает пользователям активно принимать участие при планировании, анализе рисков, разработке, а также при выполнении оценочных действий;

уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта;

обратная связь по направлению от пользователей к разработчикам выполняется с высокой частотой и на ранних этапах модели, что обеспечивает создание нужного продукта высокого качества.

# Недостатки

если проект имеет низкую степень риска или небольшие размеры, модель может оказаться дорогостоящей. Оценка рисков после прохождения каждой спирали связана с большими затратами;

Жизненный цикл модели имеет усложненную структуру, поэтому может быть затруднено её применение разработчиками, менеджерами и заказчиками;

спираль может продолжаться до бесконечности, поскольку каждая ответная реакция заказчика на созданную версию может породить новый цикл, что отдалает окончание работы над проектом;

большое количество промежуточных циклов может привести к необходимости в обработке дополнительной документации;

использование модели может оказаться дорогостоящим и даже недопустимым по средствам, т.к. время, затраченное на планирование, повторное определение целей, выполнение анализа рисков и прототипирование, может быть чрезмерным;

могут возникнуть затруднения при определении целей и стадий, указывающих на готовность продолжать процесс разработки на следующей итерации



# Вывод

Жизненный цикл можно представить в виде моделей. В настоящее время наиболее распространенными являются: каскадная, инкрементная (поэтапная модель с промежуточным контролем) и спиральная модели жизненного цикла.