Оглавление

1. Жизненный цикл ИС и его структура 2

1.1 Стадии жизненного цикла ИС 3

1.2 Стандарты жизненного цикла ИС 4

2. Модели жизненного цикла 6

2.1 Типы моделей жизненного цикла ИС 6

2.2 Достоинства и недостатки моделей жизненного цикла ИС 8

3. Процессы жизненного цикла ИС 11

3.1 Основные процессы жизненного цикла 11

3.2 Вспомогательные процессы жизненного цикла 13

3.3 Организационные процессы 14

Список использованной литературы 16

1. Жизненный цикл ИС и его структура

Жизненный цикл информационной системы — период времени, который начинается с момента принятия решения о необходимости создания информационной системы и заканчивается в момент ее полного изъятия из эксплуатации.

Понятие жизненного цикла является одним из базовых понятий методологии проектирования информационных систем.

Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла (ЖЦ) ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Полный жизненный цикл информационной системы включает в себя, как правило, стратегическое планирование, анализ, проектирование, реализацию, внедрение и эксплуатацию. В общем случае жизненный цикл можно в свою очередь разбить на ряд стадий. В принципе, это деление на стадии достаточно произвольно. Мы рассмотрим один из вариантов такого деления, предлагаемый корпорацией Rational Software – одной из ведущих фирм на рынке программного обеспечения средств разработки информационных систем (среди которых большой популярностью заслуженно пользуется универсальное CASE-средство Rational Rose).

1.1 Стадии жизненного цикла ИС

Стадия — часть процесса создания ИС, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей, программных компонентов, документации), определяемого заданными для данной стадии требованиями. Соотношение между процессами и стадиями также определяется используемой моделью жизненного цикла ИС.

Согласно методологии, предлагаемой Rational Software, жизненный цикл информационной системы подразделяется на четыре стадии.

Границы каждой стадии определены некоторыми моментами времени, в которые необходимо принимать определенные критические решения и, следовательно, достигать определенных ключевых целей.

1) Начальная стадия

На начальной стадии устанавливается область применения системы и определяются граничные условия. Для этого необходимо идентифицировать все внешние объекты, с которыми должна взаимодействовать разрабатываемая система, и определить характер этого взаимодействия на высоком уровне. На начальной стадии идентифицируются все функциональные возможности системы и производится описание наиболее существенных из них.

2) Стадия уточнения

На стадии уточнения проводится анализ прикладной области, разрабатывается архитектурная основа информационной системы.

При принятии любых решений, касающихся архитектуры системы, необходимо принимать во внимание разрабатываемую систему в целом. Это означает, что необходимо описать большинство функциональных возможностей системы и учесть взаимосвязи между отдельными ее составляющими.

В конце стадии уточнения проводится анализ архитектурных решений и способов устранения главных факторов риска в проекте.

3) Стадия конструирования

На стадии конструирования разрабатывается законченное изделие, готовое к передаче пользователю.

По окончании этой стадии определяется работоспособность разработанного программного обеспечения.

4) Стадия передачи в эксплуатацию

На стадии передачи в эксплуатацию разработанное программное обеспечение передается пользователям. При эксплуатации разработанной системы в реальных условиях часто возникают различного рода проблемы, которые требуют дополнительных работ по внесению корректив в разработанный продукт. Это, как правило, связано с обнаружением ошибок и недоработок.

В конце стадии передачи в эксплуатацию необходимо определить, достигнуты цели разработки или нет.

1.2 Стандарты жизненного цикла ИС

Современные сети разрабатываются на основе стандартов, что позволяет обеспечить, во-первых, их высокую эффективность и, во-вторых, возможность их взаимодействия между собой.

Среди наиболее известных стандартов можно выделить следующие:

ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют каскадной модели жизненного цикла.

ISO/IEC 12207(International Organization of Standardization /International Electrotechnical Commission )1995 - стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО. Стандарт не содержит описания фаз, стадий и этапов.

Rational Unified Process (RUP) предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования. Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы. Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы. Суть работы в рамках RUP - это создание и сопровождение моделей на базе UML.

Microsoft Solution Framework (MSF) сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования. MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.

Extreme Programming (XP). Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалось в 1996 году. В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС, а разработка ведется с использованием последовательно дорабатываемых прототипов.

2. Модели жизненного цикла

Модель жизненного цикла ИС — структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла. Модель жизненного цикла зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует.

Модель ЖЦ ИС включает в себя:

стадии;

результаты выполнения работ на каждой стадии;

ключевые события — точки завершения работ и принятия решений.

Модель жизненного цикла отражает различные состояния системы, начиная с момента возникновения необходимости в данной ИС и заканчивая моментом ее полного выхода из употребления.

2.1 Типы моделей жизненного цикла ИС

В настоящее время известны и используются следующие модели жизненного цикла:

Каскадная модель (рис. 2.1) предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.

Поэтапная модель с промежуточным контролем (рис. 2.2). Разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.

Спиральная модель (рис. 2.3). 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 (макетирования).

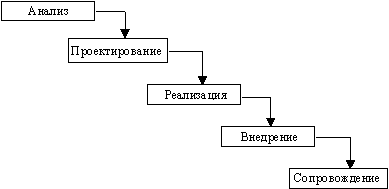


Рис. 2.1. Каскадная модель ЖЦ ИС

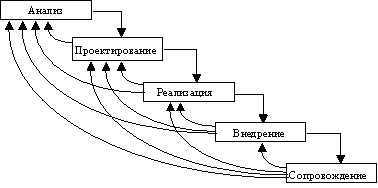


Рис. 2.2. Поэтапная модель с промежуточным контролем

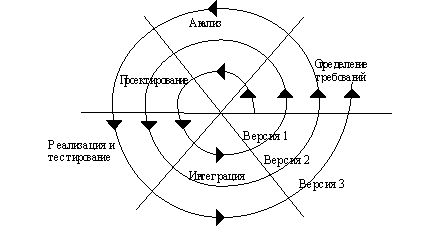


Рис. 2.3. Спиральная модель ЖЦ ИС

На практике наибольшее распространение получили две основные модели жизненного цикла:

каскадная модель (характерна для периода 1970-1985 гг.);

спиральная модель (характерна для периода после 1986.г.).

2.2 Достоинства и недостатки моделей жизненного цикла ИС

В ранних проектах достаточно простых ИС каждое приложение представляло собой единый, функционально и информационно независимый блок. Для разработки такого типа приложений эффективным оказался каскадный способ. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

Можно выделить следующие положительные стороны применения каскадного подхода:

на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;

выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Каскадный подход хорошо зарекомендовал себя при построении относительно простых ИС, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к системе. Основным недостатком этого подхода является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания ИС оказывается соответствующим поэтапной модели с промежуточным контролем.

Спиральная модель ЖЦ была предложена для преодоления перечисленных проблем. На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы. Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.

Основная проблема спирального цикла - определение момента перехода на следующий этап. Для ее решения вводятся временные ограничения на каждый из этапов жизненного цикла, и переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. Планирование производится на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

Несмотря на настойчивые рекомендации экспертов в области проектирования и разработки ИС, многие компании продолжают использовать каскадную модель вместо какого-либо варианта итерационной модели. Основные причины, по которым каскадная модель сохраняет свою популярность, следующие:

Привычка - многие ИТ-специалисты получали образование в то время, когда изучалась только каскадная модель, поэтому она используется ими и в наши дни.

Иллюзия снижения рисков участников проекта (заказчика и исполнителя). Каскадная модель предполагает разработку законченных продуктов на каждом этапе: технического задания, технического проекта, программного продукта и пользовательской документации. Разработанная документация позволяет не только определить требования к продукту следующего этапа, но и определить обязанности сторон, объем работ и сроки, при этом окончательная оценка сроков и стоимости проекта производится на начальных этапах, после завершения обследования. Очевидно, что если требования к информационной системе меняются в ходе реализации проекта, а качество документов оказывается невысоким (требования неполны и/или противоречивы), то в действительности использование каскадной модели создает лишь иллюзию определенности и на деле увеличивает риски, уменьшая лишь ответственность участников проекта.

Проблемы внедрения при использовании итерационной модели. В некоторых областях спиральная модель не может применяться, поскольку невозможно использование/тестирование продукта, обладающего неполной функциональностью (например, военные разработки, атомная энергетика и т.д.). Поэтапное итерационное внедрение информационной системы для бизнеса возможно, но сопряжено с организационными сложностями (перенос данных, интеграция систем, изменение бизнес-процессов, учетной политики, обучение пользователей). Трудозатраты при поэтапном итерационном внедрении оказываются значительно выше, а управление проектом требует настоящего искусства. Предвидя указанные сложности, заказчики выбирают каскадную модель, чтобы "внедрять систему один раз".

3. Процессы жизненного цикла ИС

Процесс определяется как совокупность взаимосвязанных действий, преобразующих входные данные в выходные. Описание каждого процесса включает в себя перечень решаемых задач, исходных данных и результатов.

В соответствии с базовым международным стандартом ISO/IEC 12207 все процессы ЖЦ ПО делятся на три группы:

3.1 Основные процессы жизненного цикла

Приобретение (действия и задачи заказчика, приобретающего ИС)

Поставка (действия и задачи поставщика, который снабжает заказчика программным продуктом или услугой)

Разработка (действия и задачи, выполняемые разработчиком: создание ПО, оформление проектной и эксплуатационной документации, подготовка тестовых и учебных материалов и т. д.)

Эксплуатация (действия и задачи оператора — организации, эксплуатирующей систему)

Сопровождение (действия и задачи, выполняемые сопровождающей организацией, то есть службой сопровождения). Сопровождение — внесений изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Среди основных процессов жизненного цикла наибольшую важность имеют три: разработка, эксплуатация и сопровождение. Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами.

Разработка

Разработка информационной системы включает в себя все работы по созданию информационного программного обеспечения и его компонентов в соответствии с заданными требованиями. Разработка информационного программного обеспечения также включает:

оформление проектной и эксплуатационной документации;

подготовку материалов, необходимых для тестирования разработанных программных продуктов;

разработку материалов, необходимых для обучения персонала.

Разработка является одним из важнейших процессов жизненного цикла информационной системы и, как правило, включает в себя стратегическое планирование, анализ, проектирование и реализацию (программирование).

Эксплуатация

Эксплуатационные работы можно подразделить на подготовительные и основные. К подготовительным относятся:

конфигурирование базы данных и рабочих мест пользователей;

обеспечение пользователей эксплуатационной документацией;

обучение персонала.

Основные эксплуатационные работы включают:

непосредственно эксплуатацию;

локализацию проблем и устранение причин их возникновения;

модификацию программного обеспечения;

подготовку предложений по совершенствованию системы;

развитие и модернизацию системы.

Сопровождение

Службы технической поддержки играют весьма заметную роль в жизни любой корпоративной информационной системы. Наличие квалифицированного технического обслуживания на этапе эксплуатации информационной системы является необходимым условием решения поставленных перед ней задач, причем ошибки обслуживающего персонала могут приводить к явным или скрытым финансовым потерям, сопоставимым со стоимостью самой информационной системы.

Основными предварительными действиями при подготовке к организации технического обслуживания информационной системы являются:

выделение наиболее ответственных узлов системы и определение для них критичности простоя (это позволит выделить наиболее критичные составляющие информационной системы и оптимизировать распределение ресурсов для технического обслуживания);

определение задач технического обслуживания и их разделение на внутренние, решаемые силами обслуживающего подразделения, и внешние, решаемые специализированными сервисными организациями (таким образом производится четкое определение круга исполняемых функций и разделение ответственности);

проведение анализа имеющихся внутренних и внешних ресурсов, необходимых для организации технического обслуживания в рамках описанных задач и разделения компетенции (основные критерии для анализа: наличие гарантии на оборудование, состояние ремонтного фонда, квалификация персонала);

подготовка плана организации технического обслуживания, в котором необходимо определить этапы исполняемых действий, сроки их исполнения, затраты на этапах, ответственность исполнителей.

3.2 Вспомогательные процессы жизненного цикла

Документирование (формализованное описание информации, созданной в течение ЖЦ ИС)

Управление конфигурацией (применение административных и технических процедур на всем протяжении ЖЦ ИС для определения состояния компонентов ИС, управления ее модификациями).

Обеспечение качества (обеспечение гарантий того, что ИС и процессы ее ЖЦ соответствуют заданным требованиям и утвержденным планам)

Верификация (определение того, что программные продукты, являющиеся результатами некоторого действия, полностью удовлетворяют требованиям или условиям, обусловленным предшествующими действиями)

Аттестация (определение полноты соответствия заданных требований и созданной системы их конкретному функциональному назначению)

Совместная оценка (оценка состояния работ по проекту: контроль планирования и управления ресурсами, персоналом, аппаратурой, инструментальными средствами)

Аудит (определение соответствия требованиям, планам и условиям договора)

Разрешение проблем (анализ и решение проблем, независимо от их происхождения или источника, которые обнаружены в ходе разработки, эксплуатации, сопровождения или других процессов)

3.3 Организационные процессы

Управление (действия и задачи, которые могут выполняться любой стороной, управляющей своими процессами)

Создание инфраструктуры (выбор и сопровождение технологии, стандартов и инструментальных средств, выбор и установка аппаратных и программных средств, используемых для разработки, эксплуатации или сопровождения ПО)

Усовершенствование (оценка, измерение, контроль и усовершенствование процессов ЖЦ)

Обучение (первоначальное обучение и последующее постоянное повышение квалификации персонала)

Управление проектом связано с вопросами планирования и организации работ, создания коллективов разработчиков и контроля за сроками и качеством выполняемых работ. Техническое и организационное обеспечение проекта включает:

выбор методов и инструментальных средств для реализации проекта;

определение методов описания промежуточных состояний разработки;

разработку методов и средств испытаний созданного программного обеспечения;

обучение персонала.

Список использованной литературы

1. Избачков С.Ю., Петров В.Н. Информационные системы–СПб.: Питер, 2008. – 655 с
2. http://ru.wikipedia.org
3. http://www.intuit.ru
4. http://www.emanual.ru