**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

**высшего профессионального образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**филиал в г. Пятигорске.**

***РЕФЕРАТ***

**ПО ДИСЦИПЛИНЕ:** ПЛАНОВАЯ НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА

**НА ТЕМУ: «**Новые технологии проектирования и анализа систем. Достоинства и недостатки их использования»

***Выполнила***:

*Беканова А.А.*

***Проверила:***

*Битюцкая Н.И.*

Пятигорск 2014 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc403747177)

[1. НОВЫЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ И АНАЛИЗА СИСТЕМ. 5](#_Toc403747178)

[1.1 Технология Rational Unified Process (RUP). Достоинства и недостатки ее использования. 14](#_Toc403747179)

[1.2 Технология Microsoft Solutions Framework (MSF). Достоинства и недостатки ее использования. 21](#_Toc403747180)

[ЗАКЛЮЧЕНИЕ 37](#_Toc403747181)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 38](#_Toc403747182)

# ВВЕДЕНИЕ

Информация является одним из ценнейших ресурсов общества наряду с такими традиционными материальными видами ресурсов, как нефть, газ, полезные ископаемые, а значит, процесс ее переработки по аналогии с процессами переработки материальных ресурсов можно воспринимать как технологию.

Основной тенденцией общественного производства в XXI веке является приоритетное развитие наукоемких, высоких технологий, широкое внедрение научных достижений в реальную жизнь, информатизация общества. В течение жизни одного поколения коренным образом изменяется технология производства, образ жизни. Быстроразвивающаяся научно-техническая революция, основанная на процессе глобальной информатизации всех сфер общественной жизни, требует постоянного развития информационных технологий.

Актуальность моей работы заключается в том, что современные тенденции возрастания сложности и наукоемкости технических систем, обеспечивающих автоматизацию технологических процессов и производств, обусловливают необходимость непрерывного совершенствования технологий проектирования и исследования таких систем. В связи с этим, работы в указанной области приобретают первостепенное значение. В то же время реализация новых технологий возможна только на базе специализированных инструментальных средств, поддерживающих выполнение полного набора проектных процедур и операций в режиме интерактивного взаимодействия с  вычислительной  системой. Развитие новых технологий проектирования позволит России с ее колоссальными интеллектуальными ресурсами решить проблему низкой эффективности российской экономики, создать не только новые рабочие места, но и инфраструктуру для развития и эффективного применения инноваций, получить доступ к инвестициям и интегрироваться в мировое информационное и экономическое пространство в качестве сильного и равноправного партнера.

Целью данной работы является рассмотрение и изучение новых технологий проектирования и анализа систем.

Для достижения поставленной цели необходимо решить ряд задач:

1. Исследовать современные технологии проектирования и анализа систем;
2. Определить существенные достоинства и недостатки их использования;
3. Рассмотреть возможность дальнейшего развития этих технологий и областей их применения.

Объектом исследования являются новые технологии проектирования и анализа систем. Предмет исследования - сравнение методов проектирования и анализа систем.

# НОВЫЕ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ И АНАЛИЗА СИСТЕМ.

Проектирование информационной системы (ИС) задается регламентированной последовательностью технологических операций, выполняемых в процессе создания проекта, на основе того или иного метода, в результате чего стало бы ясно не только что должно быть сделано для создания проекта, но и как, кому и в какой последовательности это должно быть сделано.

Технология проектирования информационной системы – это совокупность методологий и средств проектирования системы, а также методов и средств организации проектирования (см. рисунок 1.1).

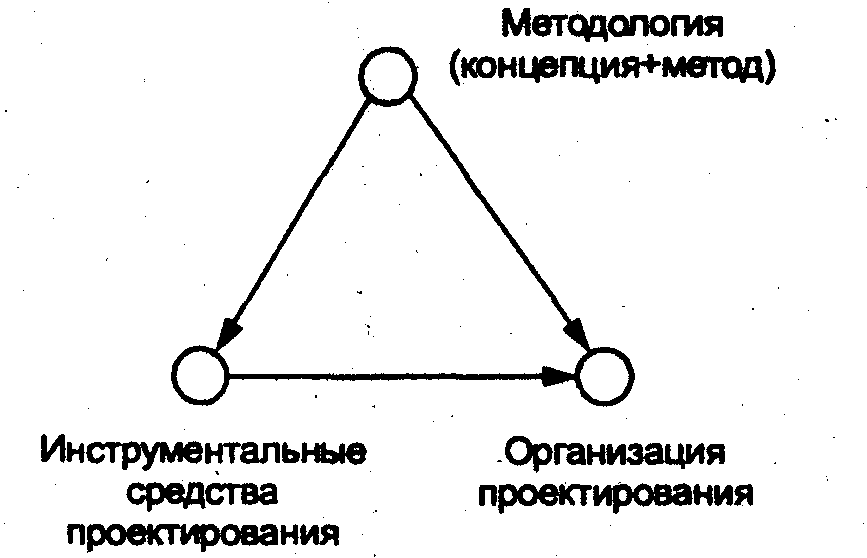


Рисунок 1.1 - Состав компонентов технологии проектирования

Методология проектирования системы предполагает наличие некоторых концепций, принципов проектирования, реализуемых набором методов проектирования, которые в свою очередь должны поддерживаться некоторыми средствами проектировании системы.

Методы проектирования системы подразумевают использование определённых программных и аппаратных средств, составляющих инструментальные средства программирования ИС.

В информационных системах методы реализуются через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства, которые обеспечивают выполнение процессов жизненного цикла системы. Методологии, технологии и инструментальные средства проектирования (CASE-средства) составляют основу проекта любой информационной системы.

Технология проектирования определяется как совокупность трех составляющих:

* пошаговой процедуры, определяющей последовательность технологических операций проектирования (см. рисунок 1.2);
* критериев и правил, используемых для оценки результатов выполнения этих операций;
* нотаций (графических и текстовых средств), используемых для описания проектируемой системы.

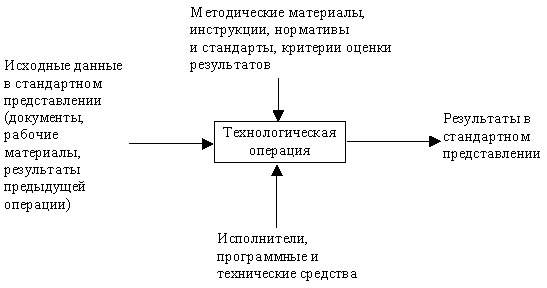


Рисунок 1.2- Представление технологической операции проектирования

Технологические инструкции, составляющие основное содержание технологии, должны состоять из описания последовательности технологических операций, условий, в зависимости от которых выполняется та или иная операция, и описаний самих операций.

Требования, предъявляемые к технологии проектирования информационных систем:

* Созданный проект должен отвечать требованиям заказчика;
* Максимальное отражение всех этапов жизненного цикла;
* Обеспечение минимальных затрат на проектирование и сопровождение проекта;
* Технология должна быть основой связи между проектированием и сопровождением;
* Технология должна способствовать росту производительности труда проектировщика;
* Технология должна обеспечивать надежность процесса проектирования и эксплуатации проекта;
* Технология должна способствовать простому ведению проектной документации.

Одним из базовых понятий методологии проектирования ИС является понятие жизненного цикла (ЖЦ). ЖЦ - это непрерывный процесс, который начинается с момента принятия решения о необходимости его создания и заканчивается в момент его полного изъятия из эксплуатации.

Методология проектирования информационных систем описывает жизненный цикл (модель ЖЦ) как некоторую последовательность стадий и выполняемых на них процессов.

Модель жизненного цикла - структура, содержащая процессы, действия и задачи, которые осуществляются с системой. Различают три основные модели ЖЦ:

* каскадную модель (водопадную);
* спиральную модель;
* итеративную и инкрементальную модель.

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

Ее основной характеристикой является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем (см. рисунок 1.3). Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

Этапы проекта в соответствии с каскадной моделью:

* Формирование требований;
* Проектирование;
* Реализация;
* Тестирование;
* Внедрение;
* Эксплуатация и сопровождение.

Положительные стороны применения каскадного подхода заключаются в следующем:

* Полная и согласованная документация на каждом этапе;
* Легко определить сроки и затраты на проект.

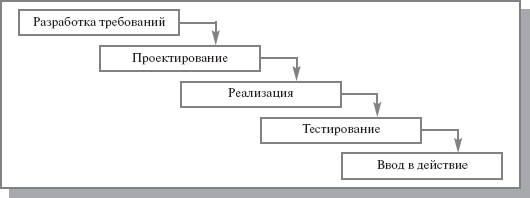


Рисунок 1.3 - Каскадная модель ЖЦ ИС

В процессе использования этого подхода обнаружился ряд его недостатков, вызванных прежде всего тем, что реальный процесс создания проекта никогда полностью не укладывался в такую жесткую схему. В процессе создания проекта постоянно возникала потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания программного обеспечения (ПО) принимал следующий вид (см. рисунок 1.4):

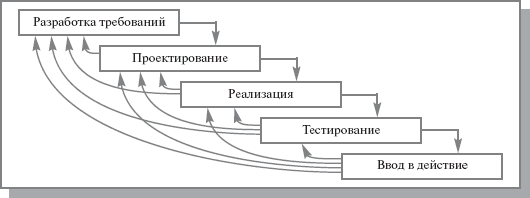


Рисунок 1.4 - Реальный процесс разработки по каскадной схеме

Основным недостатком каскадного подхода является существенное запаздывание с получением результатов. Согласование результатов с пользователями производится только в точках, планируемых после завершения каждого этапа работ, требования к ИС "заморожены" в виде технического задания на все время ее создания. Таким образом, пользователи могут внести свои замечания только после того, как работа над системой будет полностью завершена. В случае неточного изложения требований или их изменения в течение длительного периода создания программного обеспечения, пользователи получают систему, не удовлетворяющую их потребностям. Модели автоматизируемого объекта могут устареть одновременно с их утверждением. Таким образом, каскадная модель для крупных проектов мало реалистична и может быть эффективно использована только для создания небольших систем.

Для преодоления перечисленных проблем была предложена спиральная модель ЖЦ (см. рисунок 1.5), делающая упор на начальные этапы ЖЦ: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПО, на ней уточняются цели и характеристики проекта, определяется его качество и планируются работы следующей итерации. Таким образом, углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который доводится до реализации.

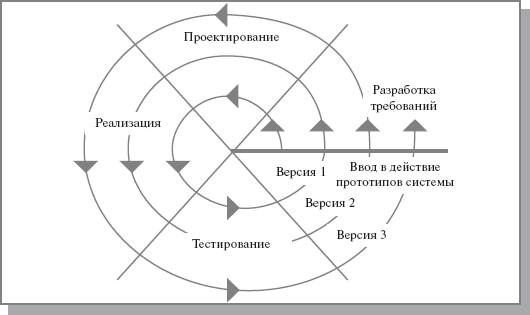


Рисунок 1.5 - Спиральная модель ЖЦ

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. При итеративном способе разработки недостающую работу можно будет выполнить на следующей итерации. Главная же задача - как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

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

Естественное развитие каскадной и спиральной моделей привело к их сближению и появлению современного итерационного подхода, который представляет рациональное сочетание этих моделей.

Альтернативой каскадной модели является модель итеративной и инкрементальной разработки (англ. iterative and incremental development, IID). Также эту модель называют итеративной и инкрементальной моделью.

Итеративная и инкрементальная модель предполагает разбиение жизненного цикла проекта на последовательность итераций, каждая из которых напоминает «мини-проект», включая все процессы разработки в применении к созданию меньших фрагментов функциональности, по сравнению с проектом в целом. Цель каждой итерации – получение работающей версии программной системы, включающей функциональность, определенную интегрированным содержанием всех предыдущих и текущей итерации. Результат финальной итерации содержит всю требуемую функциональность продукта. Таким образом, с завершением каждой итерации продукт получает приращение – инкремент – к его возможностям, которые, следовательно, развиваются эволюционно.

Шансы успешного создания сложной системы будут максимальными, если она реализуется в серии небольших шагов и если каждый шаг заключает в себе четко определенный успех, а также возможность «отката» к предыдущему успешному этапу в случае неудачи. Перед тем, как пустить в дело все ресурсы, предназначенные для создания системы, разработчик имеет возможность получать из реального мира сигналы обратной связи и исправлять возможные ошибки в проекте».

Этот подход имеет и свои отрицательные стороны:

* целостное понимание возможностей и ограничений проекта очень долгое время отсутствует;
* при итерациях приходится отбрасывать часть сделанной ранее работы;
* добросовестность специалистов при выполнении работ все же снижается, что психологически объяснимо, ведь над ними постоянно довлеет ощущение, что «все равно все можно будет переделать и улучшить позже».

Итак, итеративные или инкрементальные модели предполагают разбиение создаваемой системы на набор кусков, которые разрабатываются с помощью нескольких последовательных проходов всех работ или их части.

На первой итерации разрабатывается кусок системы, не зависящий от других. При этом большая часть или даже полный цикл работ проходится на нем, затем оцениваются результаты и на следующей итерации либо первый кусок переделывается, либо разрабатывается следующий, который может зависеть от первого, либо как-то совмещается доработка первого куска с добавлением новых функций. В результате на каждой итерации можно анализировать промежуточные результаты работ и реакцию на них всех заинтересованных лиц, включая пользователей, и вносить корректирующие изменения на следующих итерациях. Каждая итерация может содержать полный набор видов деятельности – от анализа требований до ввода в эксплуатацию очередной части ПО (см. рисунок 1.6).

Вместе с гибкостью и возможностью быстро реагировать на изменения, итеративные модели привносят дополнительные сложности в управление проектом и отслеживание его хода. При использовании итеративного подхода значительно сложнее становится адекватно оценить текущее состояние проекта и спланировать долгосрочное развитие событий, а также предсказать сроки и ресурсы, необходимые для обеспечения определенного качества результата.

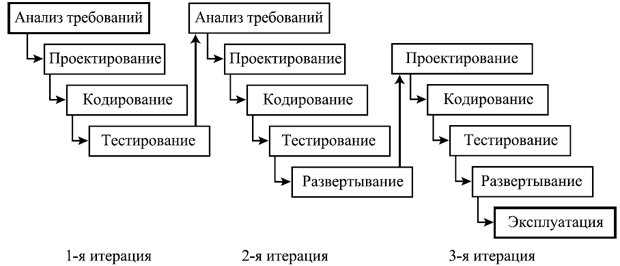


Рисунок 1.6 -Возможный ход работ по итеративной модели

Различные варианты итерационного подхода реализованы в большинстве современных технологий и методов: Rational Unified Process (RUP) и Microsoft Solutions Framework (MSF). Рассмотрим эти технологии.

## 1.1 Технология Rational Unified Process (RUP). Достоинства и недостатки ее использования.

Rational Unified Process (RUP) - программный продукт, разработанный компанией Rational Software, является одним из наиболее совершенных технологий, претендующих на роль мирового корпоративного стандарта. RUP в настоящее время входит в состав IBM.

RUP - это технологическая процедура, отражающая структуру жизненного цикла проекта, роли и ответственности отдельных исполнителей, выполняемые ими задачи и используемые в процессе разработки модели.

RUP - это конфигурируемый процесс, пригодный как для маленьких групп разработчиков, так и для крупных организаций, занимающихся созданием программного обеспечения. Основой RUP является простая архитектура процесса, обеспечивающая общность для целого семейства процессов.

RUP позволяет обеспечить четкий подход к распределению задач и ответственности внутри организации-разработчика. Его суть заключается в создании качественного программного обеспечения в установленные сроки и в рамках бюджета, отвечающего запросам конечных пользователей. Языком моделирования RUP является Unified Modeling Language (UML). UML – стандартный язык моделирования, используемый всеми членами группы, делает понятными для всех описания требований, проектирование и архитектуру системы.

Отличительной особенностью RUP является то, что вместо создания громадного количества бумажных документов, опирается на разработку и развитие семантически обогащенных моделей, представляющих разрабатываемую систему.

RUP соответствует стандартам и нормативным документам, связанным с процессами жизненного цикла программного обеспечения и оценкой технологической зрелости организаций-разработчиков.

Основные принципы RUP:

1. Итерационный и инкрементный подход к созданию программного обеспечения.
2. Планирование и управление проектом на основе функциональных требований к системе - вариантов использования.
3. Построение системы на базе архитектуры программного обеспечения.

В соответствии с первым принципом, разработка системы выполняется в виде нескольких краткосрочных мини-проектов фиксированной длительности (от 2 до 6 недель), называемых итерациями. В каждой итерации есть свои собственные этапы анализа требований, проектирования, реализации, тестирования, интеграции и завершением является создание системы.

Итерационный цикл основывается на постоянном расширении и дополнении системы в процессе нескольких итераций с периодической обратной связью и адаптацией добавляемых модулей к существующему ядру системы. В связи с тем, что система постоянно разрастается шаг за шагом, такой подход называют итерационным и инкрементным. Этот подход исключает слишком быстрое написание кода, чрезмерно продолжительный этап детального проектирования и построения моделей без обратной связи (см. рисунок 1.7).

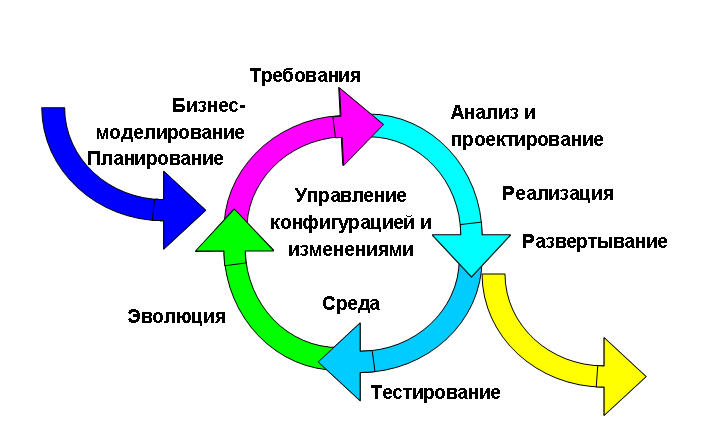


Рисунок 1.7 – Итеративная разработка в RUP

Итерационный и инкрементный подход позволяет гибко учитывать новые требования и совершать тактические изменения в деловых целях, а также выявлять проблемы на самых ранних этапах разработки и разрешать их, что экономит затраты.

Структуру жизненного цикла проекта, выполняемого по технологии RUP можно рассмотреть на координатной плоскости. Горизонтальное измерение – это динамическая структура или временное измерение процесса, которое  показывает, как процесс, выраженный в форме циклов, фаз, итераций и вех, раскрывается в ходе жизненного цикла проекта. Вертикальное измерение – это статическая структура процесса, описывающая, как элементы процесса (задачи, дисциплины, артефакты и роли) логически группируются в дисциплины или рабочие процессы.

На рисунке 1.8 показано общее представление RUP в двух измерениях.

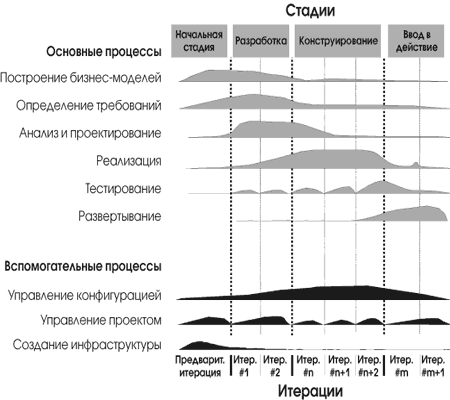


Рисунок 1.8 - Общее представление RUP

Модель RUP состоит из четырех фаз:

* начальная фаза;
* фаза разработки;
* фаза конструирования;
* фаза внедрения.

Фазы делятся на итерации. Последовательность этих фаз фиксирована, но при этом число итераций, необходимых для завершения каждой фазы, определяется индивидуально для каждого конкретного проекта. Переход с фазы на фазу производится только после выполнения задач фазы и представляет собой контрольную точку процесса. В этот период должны достигаться важные результаты и приниматься критически важные решения о дальнейшей разработке.

Начальная фаза может иметь разные формы. Для больших проектов начальная фаза может заключаться во всестороннем изучении всех возможностей реализации проекта, которое займет месяцы. На этом этапе определяются основные цели проекта, бюджет проекта, основные средства его выполнения – технологии, инструменты, ключевой персонал, составляются предварительные планы проекта, определяются также границы проекта, и выполняется некоторый начальный анализ для оценки величин проекта.

Основная цель начальной фазы – достижение компромисса между всеми заинтересованными лицами относительно задач проекта.

На фазе разработки выявляются более детальные требования к системе, выполняется высокоуровневый анализ предметной области и проектирование для построения базовой архитектуры системы, создается план конструирования и устраняются наиболее рискованные элементы проекта. Основной целью этой фазы является разработка архитектуры продукта, которая позволяет решать поставленные перед системой задачи и далее используется как основа разработки системы.

Самым важным результатом фазы разработки является описание базовой архитектуры будущей системы, включающей в себя:

* модель предметной области, отражающая понимание бизнеса и служит началом для формирования основных классов предметной области;
* технологическую платформу, определяющую основные элементы технологии реализации системы и их взаимодействие.

Базовая архитектура является основой всей разработки. В дальнейшем возможны незначительные изменения в деталях архитектуры, однако, серьезные изменения маловероятны. Фаза разработки занимает около пятой части общей продолжительности проекта.

Суть конструирования - определение последовательности итераций конструирования и вариантов использования, реализуемых на каждой итерации. Итерации на стадии конструирования являются одновременно инкрементными и повторяющимися.

Основной цель этой фазы является детальное прояснение требований и разработка системы, удовлетворяющей им, на основе спроектированной ранее архитектуры.

Результатом фазы конструирования является продукт, готовый к передаче конечным пользователям, включающий в себя программное обеспечение, руководства пользователя, описание текущей реализации.

Фаза внедрения представляет собой процесс передачи готового продукта от разработчика к заказчику.

Цель данной фазы – сделать систему полностью доступной конечным пользователям. На данном этапе происходит окончательное развертывание системы в ее рабочей среде, подгонка мелких деталей под нужды заказчиков.

На фазе внедрения продукт не дополняется новой функциональностью, а только выявляются ошибки. Если качество не соответствует ожиданиям пользователей или критериям, установленным в начальной фазе, фаза внедрения повторяется снова. Выполнение всех целей означает достижение вехи готового продукта и завершение полного цикла разработки.

Статический аспект RUP представлен следующими четырьмя основными элементами:

* роли;
* виды деятельности;
* рабочие продукты;
* дисциплины.

Основные процессы RUP определены шестью основными и тремя вспомогательными дисциплинами.

Основные дисциплины:

* 1. Построение бизнес-моделей.

Цели – понять бизнес – контекст, в котором будет работать система и убедиться, что все заинтересованные лица понимают их одинаково, понять возможные проблемы, оценить возможные их решения и их последствия для бизнеса организации, в которой будет работать система.

* 1. Определение требований.

Цели – понять, что должна делать система, определить границы системы и основу для планирования проекта и оценок ресурсозатрат в нем.

* 1. Анализ и проектирование.

Выработка архитектуры системы на основе ключевых требований, создание проектной модели, представленной в виде диаграмм UML, описывающих продукт с различных точек зрения.

* 1. Реализация.

Разработка исходного кода, компонент системы, тестирование и интегрирование компонент.

* 1. Тестирование.

Общая оценка дефектов продукта, его качества в целом, оценка степени соответствия исходным требованиям.

* 1. Развертывание.

Цели – развернуть систему в ее рабочем окружении и оценить ее работоспособность.

Вспомогательные дисциплины:

1. Управление конфигурацией и изменениями.

Определение элементов, подлежащих хранению и правил построения из них согласованных конфигураций, поддержание целостности текущего состояния системы, проверка согласованности вносимых изменений;

1. Управление проектом.

Включает планирование, управление персоналом, обеспечения связей с другими заинтересованными лицами, управление рисками, отслеживание текущего состояния проекта;

1. Создание инфраструктуры.

Настройка процесса под конкретный проект, выбор и смена технологий и инструментов, используемых в проекте.

Достоинства RUP:

– большой диапазон решений в части формализации процесса разработки;

– использование как основы для водопадного стиля разработки, так и в качестве гибкого процесса;

– снижение основных рисков заказчика и разработчика;

– экономия ресурсов за счет автоматизации регрессионного тестирования;

– улучшение качества программного обеспечения за счет многократных проверок изменений;

–улучшение качества тестирования за счет использования современных технологий.

Недостаток RUP - недостаточный уровень формализма, который может привести к несогласованности решений, принимаемых участниками проекта, к непродуктивным затратам ресурсов на переработку кода и на повторное решение типовых проблем.

## 1.2 Технология Microsoft Solutions Framework (MSF). Достоинства и недостатки ее использования.

Microsoft Solutions Framework (MSF) – это сбалансированный набор методик организации процесса разработки, который может быть адаптирован под потребности практически любого коллектива разработчиков. MSF не только содержит рекомендации общего характера, но и предлагает адаптируемую модель коллектива разработчиков, определяющую взаимоотношения внутри коллектива, гибкую модель проектного планирования, основанного на управлении проектными группами.

Благодаря своей гибкости, масштабируемости и отсутствию жестких инструкций [MSF](http://ru.wikipedia.org/wiki/MSF) способен удовлетворить нужды организации или проектной группы любого размера. Методология [MSF](http://ru.wikipedia.org/wiki/MSF) состоит из принципов, моделей и дисциплин по управлению персоналом, процессами, технологическими элементами и связанными со всеми этими факторами вопросами, характерными для большинства проектов.

MSF состоит из двух моделей:

* модель проектной группы;
* модель процессов,

и трех дисциплин:

* дисциплина управление проектами;
* дисциплина управление рисками;
* дисциплина управление подготовкой.

Модель проектной группы MSF описывает подход Microsoft к организации работающего над проектом персонала и его деятельности в целях максимизации успешности проекта.

Модель проектной группы основана на:

* + 6 принципах;
  + 6 концепциях;
  + 7 ролевых кластерах.

Основные принципы модели проектной группы:

* Единое видение проекта. Для успешной коллективной работы над проектом необходимо единое видение и понимание членами проектной группы целей и задач проекта.
* Гибкость – готовность к переменам. MSF основывается на принципе непрерывной изменяемости условий проекта при неизменной эффективности управленческой деятельности.
* Концентрация на бизнес-приоритетах. Продукт должен соответствовать определенным нуждам потребителей и приносить выгоду или отдачу.
* Поощрение свободного общения. Модель процессов MSF предполагает открытый и честный обмен информацией как внутри команды, так и с ключевыми заинтересованными лицами.

Ключевые концепции модели проектной группы:

1. Проектная группа – команда соратников.
2. Сфокусированность на нуждах заказчика.
3. Нацеленность на конечный результат.
4. Установка на отсутствие дефектов.
5. Стремление к самосовершенствованию.
6. Заинтересованные команды работают эффективно.

Модель проектной группы MSF определяет 7 ролевых кластеров (см. рисунок 1.9).

Рисунок 1.9 – Ролевые кластеры MSF

Ролевые кластеры ответственны за конкретные области компетенции и связанные с ними цели и задачи. Ролевые кластеры можно назвать ролями. Одна роль может быть представлена одним или несколькими сотрудниками, в соответствии с размерами проекта и его сложности, а также профессиональных навыков, требуемых для реализации всех областей компетенции кластера. Так как каждая из целей одинаково необходима для успешности проекта, все роли находятся в равноправных партнерских взаимоотношениях с равной значимостью при принятии решений. Чаще всего роли распределяются среди различных подразделений одной организации, но иногда часть их отводится сообществу потребителей или внешним по отношению к организации консультантам и партнерам. Ключевым моментом является четкое определение работников, ответственных за каждый ролевой кластер, их функций, ответственности и ожидаемого вклада в конечный результат.

Основной задачей ролевого кластера «Управление продуктом» является обеспечение соответствие результатов выполнения проекта нуждам заказчика.

Ролевой кластер «Управление программой» обеспечивает управленческие функции – отслеживание планов и их выполнение, ответственность за бюджет, ресурсы проекта, разрешение проблем и трудностей процесса.

Ролевой кластер «Разработка» занимается программированием программного обеспечения.

Ролевой кластер «Тестирование» - отвечает за тестирование программного обеспечения. Одобрение выпуска продукта производится после выявления и устранения всех дефектов.

Ролевой кластер «Удовлетворение потребителя» - дизайн пользовательского интерфейса и обеспечение удобства эксплуатации программного обеспечения, обучение пользователей работе с программным обеспечением, создание пользовательской документации.

Ролевой кластер «Управление выпуском» ответственен за беспрепятственное внедрение проекта и его функционирование, берет на себя связь между разработкой решения, его внедрением и последующим сопровождением.

Ролевой кластер «Архитектура» отвечает за организацию и выполнение высокоуровневого проектирования решения, создание функциональной спецификации программного обеспечения и управление этой спецификацией в процессе разработки, определение рамок проекта и ключевых компромиссных решений.

Модель проектной группы MSF предлагает разбиение больших команд (более 10 человек) на малые многопрофильные группы направлений. Эти малые коллективы работают параллельно, регулярно синхронизируя свои усилия.

* В одном ролевом кластере может быть много людей;
* Один человек может взять на себя несколько ролей;
* Большие коллективы:

- создаем группы направлений;

- создаем функциональные группы;

* Малые коллективы:

- смотрим таблицу совместимости ролей (см. рисунок 1.10). Из таблицы можно сделать вывод, что минимальный размер команды – 3 человека.

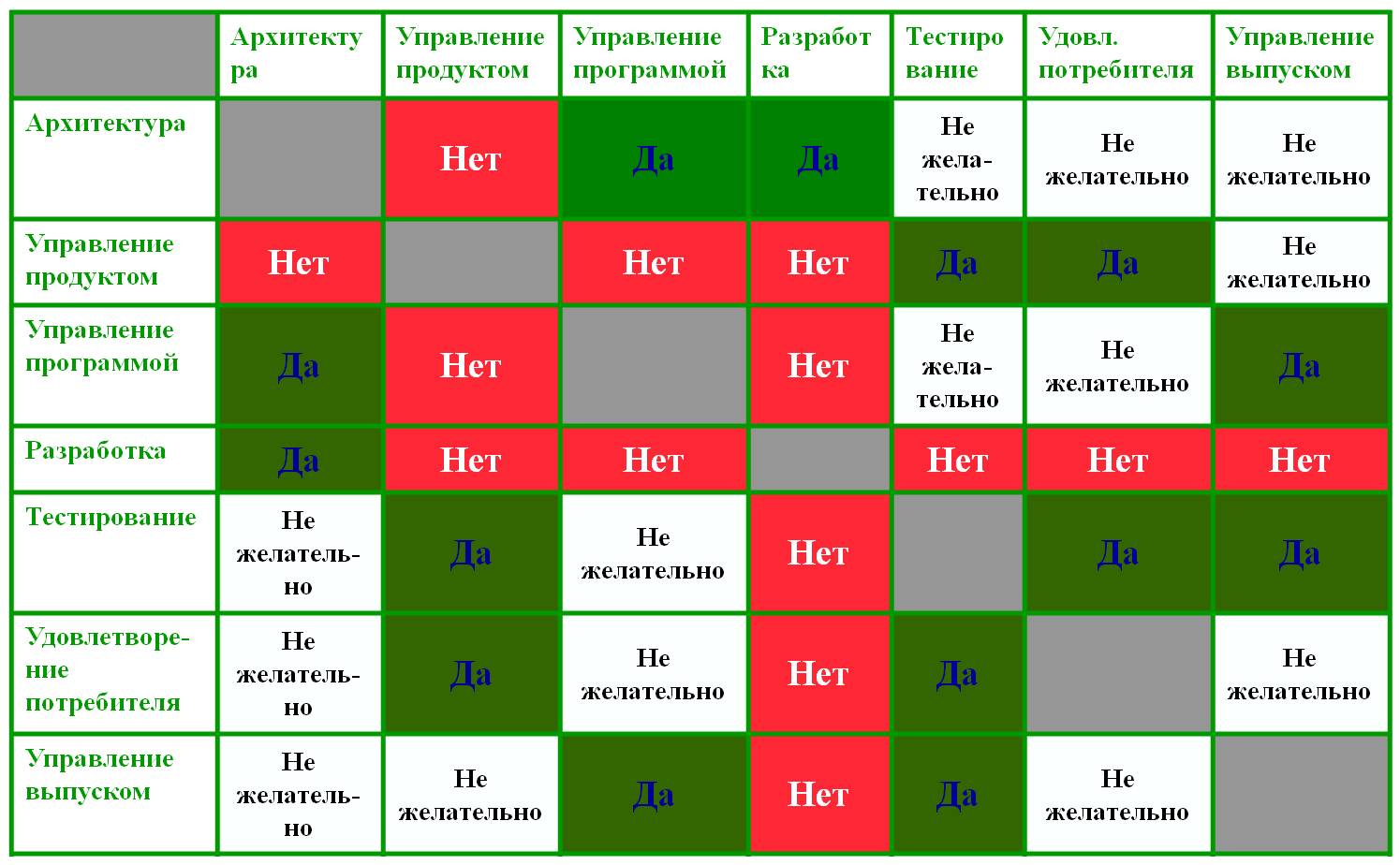


Рисунок 1.10 – Таблица совместимости ролей

Модель процессов MSF представляет общую методологию разработки и внедрения ПО, а именно описывает последовательность действий, осуществляемых в ходе реализации проекта. Модель процессов MSF объединяет в себе упорядоченность каскадной модели с гибкостью спиральной: проект реализуется поэтапно, с наличием соответствующих контрольных точек, а сама последовательность этапов может повторяться по спирали (см. рисунок 1.11).

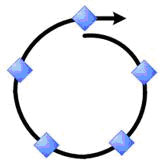


Рисунок 1.11 – Модель процессов MSF

В данном случае, от водопадной модели берется простота планирования, от классической спиральной - легкость модификаций.

Процесс MSF ориентирован на "вехи". Вехи – ключевые точки процесса разработки, характеризующие достижение какого–либо существенного результата.

В модели MSF выделяют два типа вех: главные и промежуточные.

Главные вехи служат точками перехода от одной фазы к другой. Они также определяют изменения в текущих задачах ролевых кластеров.

Промежуточные вехи показывают достижение в ходе проекта определенного прогресса и расчленяют большие сегменты работы на меньшие, обозримые участки (см. рисунок 1.12). Промежуточные вехи могут варьироваться от проекта к проекту. MSF рекомендует использовать определенный набор промежуточных вех, но на практике проектная группа может сама устанавливать их в соответствии с особенностями своей работы.

Благодаря промежуточным контрольным точкам и обратной спирали верификации облегчается взаимодействие с заказчиком.

Рисунок 1.12 – Модель процессов с промежуточными вехами MSF

Модель процессов MSF покрывает процесс создания решения с самого его начала и до момента окончательного внедрения. Каждая фаза заканчивается главной вехой, результаты которой становятся видимыми за пределами проектной команды (см. рисунок 1.13).

Рисунок 1.13 – Модель процессов MSF

На фазе выработки концепции достигается одна из основных основ успеха проекта – создание и сплочение проектной группы на основе выработки единого видения. Проектная группа должна четко сформулировать свою цель таким образом, чтобы максимально мотивировать и заказчика, и проектную команду.

На данном этапе можно выделить две промежуточные контрольные точки:

* Ядро проектной группы сформировано.
* Черновой вариант концепции проекта составлен.

Этап завершается контрольной точкой "Концепция утверждена".

На этапе планирования производится основная работа по составлению планов проекта, включающая в себя разработку дизайнов, подготовку рабочих планов, оценку стоимости и сроков получения результатов.

На данном этапе можно выделить 3 стадии: концептуальное, логическое и физическое проектирование.

На стадии концептуальное проектирование задача рассматривается с точки зрения пользовательских и бизнес - требований и заканчивается определением набора сценариев использования системы.

При логическом проектировании задача рассматривается с точки зрения проектной команды, решение представляется в виде набора сервисов.

И уже на стадии физического проектирования задача рассматривается с точки зрения программистов, уточняются используемые технологии и интерфейсы.

Результаты данного этапа используются для принятия компромиссных решений в дальнейшем.

На этапе выделяются пять промежуточных контрольных точек:

* Верификация технологий;
* Базовая версия функциональной спецификации создана;
* Базовая версия сводного плана проекта создана;
* Базовая версия сводного календарного графика проекта создана;
* Среды разработки и тестирования развернуты.

Этап завершается контрольной точкой " Планы проекта утверждены".

На этапе разработки создается решение, а также пишется и документируется код. В начале данного этапа команда проверяет выполнение всех задач, характерных для предыдущих этапов. На данной стадии все функции продукта готовы и прошли тестирование в рамках своего модуля, после чего продукт готов к внешнему тестированию и стабилизации. Участники проекта, заказчики и пользователи, сотрудники службы поддержки и сопровождения могут предварительно оценить продукт и указать все недостатки, которые нужно устранить до его поставки.

На этом этапе выделяются четыре промежуточных контрольных точек:

* Концепция подтверждена;
* Промежуточная версия 1 завершена;
* Промежуточная версия 1 завершена;
* Промежуточная версия n завершена.

Основная контрольная точка этапа - "Разработка завершена".

Во время фазы стабилизации производится тестирование разработанного решения. При этом особое внимание уделяется его эксплуатации в реалистичной модели производственной среды. Проектная группа занимается приоритезацией и устранением ошибок, а также подготовкой решения к выпуску.

Главным показателем этапа стабилизации является количество обнаруженных ошибок. Тенденция к уменьшению данной величины - признак завершения работ над продуктом. Появление версии, в которой не обнаружено ни одной ошибки - важнейшая промежуточная контрольная точка. Далее следуют выпуски кандидат-релизов продукта для их исследования в условиях пилотной эксплуатации. Основная контрольная точка этапа - "Готовность решения утверждена".

На этапе внедрения проектная группа внедряет технологии и компоненты решения, стабилизирует внедренное решение, передает работу персоналу поддержки и сопровождения и получает со стороны заказчика окончательное одобрение результатов проекта.

На этом этапе выделяются три промежуточных контрольных точек:

* Ключевые компоненты развернуты;
* Внедрение на местах завершено;
* Внедренное решение стабилизировано.

Основная контрольная точка этапа - "Внедрение завершено".

Дисциплина управления рисками предлагает принципы, идеи и рекомендации для успешного активного управления рисками.

Рассмотрим процесс управления рисками MSF, представленный на рисунке 1.14. Он определяет шесть логических шагов, посредством которых проектная группа управляет текущими рисками, разрабатывает и исполняет стратегии управления рисками и извлекает уроки из своего опыта для использования на уровне всего предприятия.

Рисунок 1.14 – Процесс управления рисками

1. Выявление рисков – это фаза, которая позволяет вынести на обсуждение всей проектной группы факты наличия рисков.

2. Анализ рисков – это фаза преобразования полученных во время предыдущего шага оценок и данных в форму, позволяющую осуществить приоритезацию рисков. Приоритезация рисков позволяет проектной группе управлять наиболее важными из них, выделяя для этого необходимые ресурсы.

3. Планирование рисков выполняется на основе информации, полученной на этапе их анализа, и ставит целью выработку стратегий, планов и конкретных шагов.

4. Мониторинг рисков выполняется для наблюдения за конкретными рисками и прогрессом в осуществлении составленных планов.

5. Корректирование ситуации - процесс исполнения принятых в отношении рисков планов и контроля за ходом их исполнения.

6. Извлечение уроков формализует процесс усвоения накопленного за время работы над проектом опыта в форме, доступной для использования как внутри проектной группы, так и на уровне всего предприятия.

Заметим, что описанные фазы не обязательно должны следовать друг за другом в строгом хронологическом порядке. Проектные группы могут циклически повторять шаги выявления-анализа-планирования по мере обнаружения дополнительных факторов, влияющих на проект. При этом извлечение уроков может выполняться лишь время от времени на уровне всего предприятия.

Дисциплина управления проектами.

В методологии MSF нет должности менеджера проекта. Типовые управленческие обязанности распределяются между лидерами ролевых кластеров, а носителем профессиональных управленческих навыков и организатором работы команды является ролевой кластер "Управление программой".

 Подходы MSF к управлению проектом:

1. Ответственность за управление проектом распределена между лидерами ролевых групп.
2. Профессиональные менеджеры являются консультантами и наставниками команды и в то же время не контролируют ее.

Данная дисциплина не предусматривает конкретных рецептов управления проектами и не содержит объяснений многочисленных методов работы, применяемых опытными менеджерами. На рисунке 1.15 показано распределение ответственности по управлению проектом среди лидеров групп.

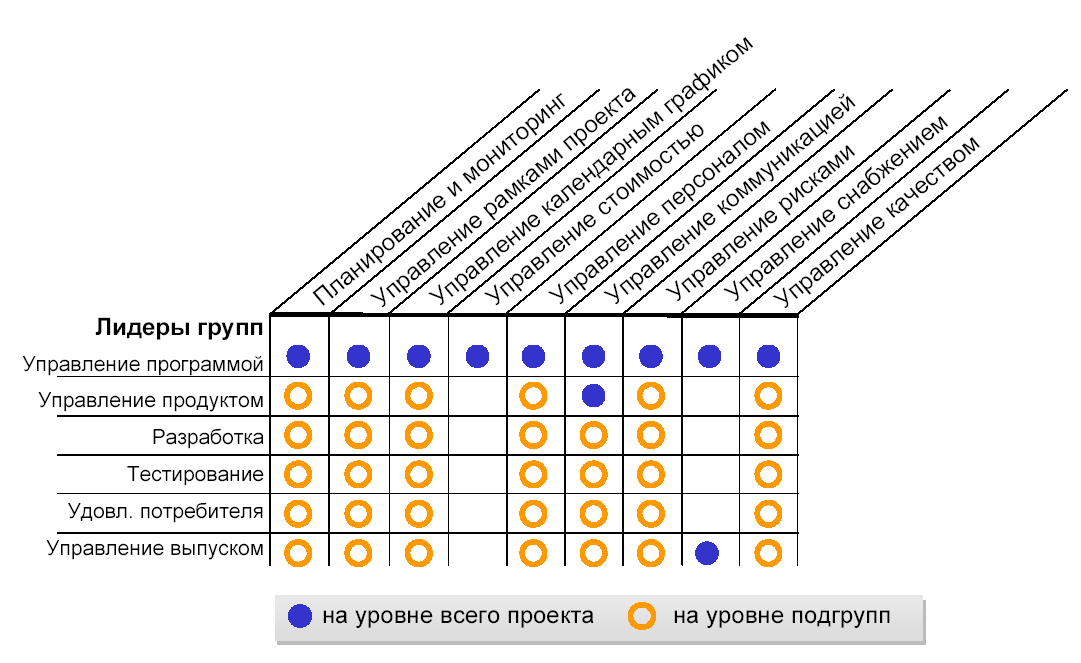


Рисунок 1.15 – Распределение ответственности по управлению проектом среди лидеров групп

При управлении проектом необходимо четко ставить цель, которую нужно достичь в результате, с учетом ограничений, накладываемых на проект. Все виды ограничений могут быть отнесены к одному из трех видов: ограничения ресурсов, ограничения времени и ограничения возможностей. Эти три вида ограничений и приоритетность задач по их преодолению образуют треугольник приоритетов в MSF (см. рисунок 1.16).

Рисунок 1.16 – Треугольник приоритетов в MSF

Матрицы компромиссов – это заранее утвержденные представления о том, какие аспекты процесса разработки будут четко заданы, а какие будут согласовываться или приниматься как есть. Основой для матрицы компромиссов является треугольник приоритетов  (см. рисунок 1.17).

Рисунок 1.17 – Матрица компромиссов в MSF

Дисциплина управления подготовкой посвящена управлению знаниями, профессиональными умениями и способностями, необходимыми для планирования, создания и сопровождения успешных решений. Она дает рекомендации по применению превентивного подхода к управлению знаниями на протяжении всего жизненного цикла информационных технологий. Этот процесс состоит из четырех шагов (см. рисунок 1.18). Каждый шаг включает в себя серию задач, помогающих достичь следующей вехи.

Рисунок 1.18 – Процесс управления подготовкой

Во время планирования архитектуры предприятия организация производит соотнесение своих бизнес целей и формирует единое видение своего будущего. Естественно, частью этого процесса является приобретение знаний и навыков, которые нужны сотрудникам для осуществления проектов, реализующих сформированное видение организации. Это первый шаг процесса управления подготовкой MSF. Он называется "Определение" и состоит из выявления и описания проектных сценариев, квалификационных требований к членам проектных групп и ожидаемых от них профессиональных навыков, необходимых для успешного планирования, создания и сопровождения решений. В зависимости от должности, сотруднику могут потребоваться профессиональные навыки, удовлетворяющие квалификационным требованиям одной или нескольких групп.

Шаг "Оценивание" процесса управления подготовкой служит для определения того, каким квалификационным требованиям удовлетворяют (или не удовлетворяют) в текущий момент сотрудники организации. На этом шаге начинается анализ квалификационных требований различных рабочих ролей, что позволяет определить необходимые профессиональные навыки сотрудников. Затем желаемая квалификация сотрудников сопоставляется с существующей на данный момент. Это указывает направления совершенствования профессионального уровня, что позволяет начать разработку планов обучения.

Шаг "Осмысление" процесса управления подготовкой MSF служит для определения эффективности реализованных планов обучения и успешности применения усвоенных работниками знаний на практике.

На этом шаге проверяется, достигнуто ли в результате проведения шага "Корректировка" описанное на шаге "Определение" желаемое состояние подготовленности. В дополнение к этому, на данном этапе производится интеграция приобретенных знаний на уровне всей организации в целях обеспечения более успешной работы над последующими проектами.

Шаг "Осмысление" мог бы считаться заключительным в процессе управления подготовкой, но так как обучение является постоянным условием достижения успеха, данный шаг рассматривается в качестве перехода к следующей итерации этого процесса. Возможны либо повторение выявления нужд подготовки путем осуществления нового шага "Определение", либо же возврат на шаг "Оценивание" для определения потребности в дальнейшем корректировании.

Достоинствами MSF являются:

– систематизация и структуризация информации в форме базы знаний;

– нестандартные подходы к организационной структуре, распределению ответственности и принципам взаимодействия внутри команды;

– MSF легко масштабируется – минимальный размер проектной группы в MSF-проекте – 3 человека, но применять данную методологию можно для коллективов и в десятки, сотни, тысячи человек;

– MSF не навязывает использование каких-либо конкретных инструментов и программных средств.

Недостатки: MSF не описывает детально важнейшие роли заказчика и пользователя; не рассматриваются методы управления группой проектов.

Microsoft Solutions Framework является наиболее сбалансированной технологией, ориентированной на проектные группы малых и средних размеров. MSF не накладывает никаких ограничений на используемый инструментарий и содержит рекомендации общего характера. Эти рекомендации могут быть использованы для построения конкретного процесса, соответствующего потребностям коллектива разработчиков.

# ЗАКЛЮЧЕНИЕ

В заключение проделанной работы хотелось бы подвести некоторые итоги.

Мы выяснили, что современные технологии проектирования и анализа систем, использующиеся в реальных проектах, очень разнообразны. Каждый из них имеет свои преимущества, проявляющиеся в конкретных условиях. Каждая технология обладает рядом характеристик, ограничивающих область его эффективного использования. Проблема обоснованного выбора и эффективного применения технологии в крупномасштабных проектах остается актуальной. Невозможно достичь удовлетворительных результатов от применения даже самых совершенных технологий, если они применяются бессистемно, разработчики не обладают необходимой квалификацией для работы с ними, и сам проект выполняется и управляется хаотически. Систематический, обоснованный подход к выбору и применению технологии может сократить время и повысить качество разработки программного обеспечения, обеспечить высокую степень его независимости от конкретных разработчиков, а также снизить затраты на разработку и сопровождение программного обеспечения.

Обобщая всё выше сказанное, отметим, что изучение новых технологий проектирования и анализа систем обладает широким потенциалом для дальнейших исследований и практических изысканий.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Ф. Крачтен -- Введение в Rational Unified Process.: Пер. с англ. -- М.: Вильямс, 2010 г.
2. Кролл, П. Rational Unified Process – это легко. Руководство по RUP для практиков / П. Кролл, Ф. Крачтен. – М. : КУДИЦ-ОБРАЗ, 2012 г. – 432 с.
3. Якунин, Ю. Ю. Технологии разработки программного обеспечения: организац.-метод. указания / Ю. Ю. Якунин. – Красноярск: ИПК СФУ, 2010 – (Технологии разработки программного обеспечения: УМКД № 183-2010 / рук. творч. коллектива Ю. Ю. Якунин).
4. Грекул В.И., Коровкина Н.Л., Денищенко Г.А. Проектирование ИС. Курс лекций. Интернет-университет, М., 2009 г.
5. Андрей Колесов. Введение в методологию Microsoft Solutions Framework http://www.bytemag.ru/Article.asp?ID=2866
6. Microsoft Solutions Framework. Методология создания программных решений (<http://www.microsoft.com/Rus/Msdn/msf/Default.mspx>)
7. http://www.gings.ru/article16.htm - что такое MSF.
8. Rational Unified Process. Методология и технология . Материалы компании Interface Ltd . ( http :// www . interface . ru / home . a sp ? artId =779)
9. Вендров А.М. Современные технологии анализа и проектирования информационных систем (Электронный ресурс)// Центр Информационных Технологий: Режим доступа: [WWW.CITMGU.RU](http://WWW.CITMGU.RU).
10. Рахимбердиев А. Современные процессы разработки программного обеспечения [Текст] / А. Рахимбердиев //RSDN Magazine. − 2012 − № 4. − C. 3–10.
11. Keeton M. Microsoft Solutions Framework (MSF) : A Pocket Guide (Paperback) [Текст] / Keeton M. – Wan Haren Publishing, 2009. – 112 p.
12. Turner M. S. V. Microsoft Solutions Framework Essentials: Building Successful Technology Solutions [Текст] /Turner M. S. V. – Microsoft Press 1 ed., − 2010. – 336 p.