



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Лекция №7

Состав документации при проектировании информационно-аналитических систем

Методы и средства проектирования информационно-аналитических систем

	<i>(наименование дисциплины (модуля) в соответствии с учебным планом)</i>	
Уровень	специалитет	
	<i>(бакалавриат, магистратура, специалитет)</i>	
Форма обучения	очная	
	<i>(очная, очно-заочная, заочная)</i>	
Направление(-я) подготовки	10.05.04 «Информационно-аналитические системы безопасности»	
	<i>(код(-ы) и наименование(-я))</i>	
Институт	Институт кибербезопасности и цифровых технологий (ИКБ)	
	<i>(полное и краткое наименование)</i>	
Кафедра	Информационно-аналитические системы кибербезопасности (КБ-2)	
	<i>(полное и краткое наименование кафедры, реализующей дисциплину (модуль))</i>	
Используются в данной редакции с учебного года	2023/24	
	<i>(учебный год цифрами)</i>	
Проверено и согласовано «___» _____ 20__ г.		
	<i>(подпись директора Института/Филиала с расшифровкой)</i>	

Москва 2024 г.

Учебные вопросы:

1. Комплекс технических документов.
2. Российские стандарты на техническую документацию.
3. Международные стандарты на техническую документацию.
4. Системы сопровождения документации.

1. Комплекс технических документов

Техническая документация является составляющей проекта по созданию, внедрению, сопровождению, модернизации и ликвидации информационной системы на всем протяжении жизненного цикла.

Комплекс технических документов, который регламентирует деятельность разработчиков, называется нормативно-методическим обеспечением (НМО). В данный комплекс входят:

- стандарты;
- руководящие документы;
- методики и положения;
- инструкции и т. д.

НМО регламентирует порядок разработки, общие требования к составу и качеству программного обеспечения (ПО), связям между компонентами, определяет содержание проектной и программной документации.

Основным назначением технической документации является обеспечение эффективных процедур разработки и использования информационной системы как программного продукта, а также организация обмена между разработчиками и пользователями ИС.

Таким образом, можно выделить следующие функции технической документации:

- дает описание возможностей системы;
- обеспечивает фиксацию принятых и реализованных проектных решений;
- определяет условия функционирования ИС;
- предоставляет информацию об эксплуатации и обслуживании ИС;
- регламентирует процедуру защиты информации, регулирует права различных групп пользователей;
- определяет возможности модернизации системы.

Перед составлением технической документации необходимо иметь ответы на следующие вопросы:

- что и зачем должно быть документировано?

- для кого предназначен тот или иной документ?
- какие ошибки может допустить пользователь и что нужно сделать для их устранения?
- как и в каких условиях будет использоваться документ?
- каковы сроки разработки документа?
- как будет обновляться и поддерживаться документация, каковы механизмы и сроки внесения изменений и пересмотра документов и кто ответственен за реализацию этих действий, а также за хранение, неизменность и контроль за исполнением?
- кто будет оценивать документ и как он соотносится с отраслевыми или ведомственными требованиями на сертификацию разработки?

Ответы на эти вопросы должны быть получены на ранних стадиях разработки информационной системы и входить в состав разрабатываемой в рамках проекта документации.

Как правило, к технической документации предъявляются следующие основные требования:

- документы должны быть точными, полными и, по возможности, краткими, иметь четкое и однозначное толкование;
- документация должна создаваться параллельно с разработкой самой информационной системы;
- обязанности по документированию системы лежат на ее разработчике;
- для повышения эффективности работы с документами должны использоваться стандарты, регламентирующие форму и содержание документов.

Исходя из последнего требования к документации, необходимо рассмотреть основные стандарты, которые используются в области информационных систем на территории Российской Федерации.

2. Российские стандарты на техническую документацию

В настоящее время существует несколько классификаций стандартов на проектирование и разработку информационных (автоматизированных) систем. Классический способ классификации группирует стандарты по двум признакам.

По объекту стандартизации:

- стандарты на продукты и услуги;
- стандарты на процессы и технологии.

По предмету стандартизации:

- функциональные стандарты (стандарты на языки программирования, протоколы, интерфейсы);
- стандарты на организацию жизненного цикла (ЖЦ) автоматизированных систем и программного обеспечения.

Альтернативная классификация группирует стандарты по статусу:

- официальные стандарты;
- стандарты «де-факто».

В свою очередь официальные стандарты подразделяются на:

- международные стандарты (ISO, ANSI, IDEF0/1);
- стандарты Российской Федерации (ГОСТ);
- отраслевые стандарты;
- ведомственные стандарты.

Стандартами «де-факто» являются официально никем не утвержденные, но фактически действующие стандарты.

Как правило, в каждую из этих групп входят документы, существенно разные по степени обязательности для организаций, конкретности и детализации содержащихся требований, открытости и гибкости, а также адаптируемости к конкретным условиям.

Отдельно выделяют корпоративные стандарты.

Для большинства сложных проектов приходится создавать свои комплексы нормативных и методических документов, регламентирующих процессы, этапы, работы и документы конкретных программных продуктов. Такие стандарты называют корпоративными и представляют собой соглашение о единых правилах организации технологии или управления в организации. К таким стандартам относятся:

- стандарты проектирования;
- стандарты оформления проектной документации;
- стандарты пользовательского интерфейса.

Стандарт проектирования должен устанавливать:

- набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации;
- правила именования объектов, оформления диаграмм, включая требования к форме и размерам объектов и т. д.;
- требования к конфигурации рабочих мест разработчиков, включая настройки операционной системы;
- правила интеграции подсистем проекта, правила поддержания проекта в

одинаковом для всех разработчиков состоянии, правила проверки проектных решений на непротиворечивость.

Стандарт оформления проектной документации должен устанавливать:

- комплектность, состав и структуру документации на каждой стадии проектирования;
- требования к ее оформлению, включая требования к содержанию разделов, подразделов, пунктов, таблиц и т. д.
- правила подготовки, рассмотрения, согласования и утверждения документации с указанием предельных сроков для каждой стадии;
- требования к настройке издательской системы, используемой в качестве встроенного средства подготовки документации;
- требования к настройке CASE-средств для обеспечения подготовки документации в соответствии с установленными требованиями.

Стандарт интерфейса пользователя должен устанавливать:

- правила оформления экранов (шрифты и цветовая палитра), состав и расположение окон и элементов управления;
- правила использования клавиатуры и мыши;
- правила оформления текстов помощи;
- перечень стандартных сообщений;
- правила обработки реакции пользователя.

За основу корпоративных стандартов могут приниматься отраслевые, национальные или международные стандарты. Сюда могут относиться различные методические материалы ведущих фирм-разработчиков ПО, научных центров, фирм-консультантов, консорциумов по стандартизации.

Отечественными стандартами являются стандарты ЕСПД (Единой Системы Программной Документации) серии ГОСТ 19.XXX и комплекс стандартов на автоматизированные системы серии ГОСТ 34.XXX, созданные в 80-90-е годы двадцатого века. Кроме того, существуют более современные стандарты на программное обеспечение.

Перечень стандартов ГОСТ 19.XXX Единая Система Программной Документации

- ГОСТ 19.001-77 Общие положения
- ГОСТ 19.101-77 Виды программ и программных документов
- ГОСТ 19.102-77 Стадии разработки
- ГОСТ 19.103-77 Обозначения программ и программных документов
- ГОСТ 19.104-78 Основные надписи

- ГОСТ 19.105-78 Общие требования к программным документам
- ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом
- ГОСТ 19.201-78 Техническое задание, требования к содержанию и оформлению
- ГОСТ 19.202-78 Спецификация. Требования к содержанию и оформлению
- ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению
- ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению
- ГОСТ 19.402-78 Описание программы
- ГОСТ 19.403-79 Ведомость держателей подлинников
- ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению
- ГОСТ 19.501-78 Формуляр. Требования к содержанию и оформлению
- ГОСТ 19.502-78 Описание применения. Требования к содержанию и оформлению
- ГОСТ 19.503-79 Руководство системного программиста. Требования к содержанию и оформлению
- ГОСТ 19.504-79 Руководство программиста. Требования к содержанию и оформлению
- ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению
- ГОСТ 19.506-79 Описание языка. Требования к содержанию и оформлению
- ГОСТ 19.507-79 Ведомость эксплуатационных документов
- ГОСТ 19.508-79 Руководство по техническом обслуживанию. Требования к содержанию и оформлению
- ГОСТ 19.601-78 Общие правила дублирования, учета и хранения
- ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом
- ГОСТ 19.603-78 Общие правила внесения изменений
- ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненных печатным способом

Стандарты ЕСПД практически не имеют содержательной составляющей и дают формальные требования к составу, содержанию и оформлению документов, описывающих программу на разных стадиях ее жизненного цикла.

Комплекс ГОСТ 34 задумывался как всеобъемлющий комплекс взаимоувязанных

межотраслевых документов и рассчитанный на взаимодействие заказчика и разработчика. Он должен был разрешить проблему «вавилонской башни», при которой в различных отраслях и областях деятельности использовалась плохо согласованная или несогласованная нормативно-техническая документация. Объектами стандартизации являются автоматизированные системы различных видов и все виды их компонентов, а не только программное обеспечение и базы данных. Комплекс рассчитан на взаимодействие заказчика и разработчика, при этом в нем предусмотрено, что заказчик может разрабатывать систему для себя сам.

Перечень стандартов ГОСТ 34.XXX Стандарты информационной технологии

- ГОСТ 34.003-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения
- ГОСТ 34.201-89 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначения документов при создании автоматизированных систем
- ГОСТ 34.320-96 Информационные технологии (ИТ). Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы
- ГОСТ 34.321-96 Информационные технологии (ИТ). Система стандартов по базам данных. Эталонная модель управления данными
- ГОСТ 34.601-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания
- ГОСТ 34.602-89 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы
- ГОСТ 34.603-92 Информационная технология (ИТ). Виды испытаний автоматизированных систем
- РД 50-34.698-90 Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов

Многие авторы считают эти стандарты морально устаревшими, однако ими продолжают активно пользоваться при оформлении проектной документации. Если разрабатывается документация на программу (систему), созданную под конкретную организацию, следует воспользоваться требованиями ГОСТов 34. Если разрабатывается документация на программу массового применения, то следует использовать ГОСТы серии 19.

Если говорить о более поздних отечественных стандартах, следует выделить ГОСТ Р

51904-2002 Программное обеспечение встроенных систем. Общие требования к разработке и документированию, который был разработан Государственным научно-исследовательским институтом авиационных систем с участием Научно-исследовательского института стандартизации и унификации. Данный стандарт распространяется на процессы разработки и документирования программного обеспечения встроенных систем реального времени и все действия, имеющие отношение к разработке программного обеспечения. Стандарт подготовлен в развитие ГОСТ Р ИСО/МЭК 12207 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств, о котором пойдет речь в п. 2.3.

Отдельно стандарт ГОСТ Р 51904-2002 в части подготовки документов будет рассмотрен в главе 9 «Программное обеспечение».

Международные стандарты применяются для разработки документации международного уровня. Как правило, они не бесплатные, так как разрабатываются не государственными организациями, но, в отличие от отечественных, разработаны недавно.

3. Международные стандарты на техническую документацию

В основе практически всех современных промышленных технологий создания программных средств лежит международный стандарт ISO/IEC 12207 Information technology. System and software engineering. Software life cycle processes (**ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств**). Первая редакция стандарта ISO/IEC 12207 была опубликована в августе 1995 г. и явилась первым международным стандартом, содержащим представительный набор процессов жизненного цикла в отношении программного обеспечения, которое рассматривалось как часть большой системы, а также применительно к программным продуктам и услугам. Стандарт определяет процессы, виды деятельности и задачи, которые используются при приобретении программного продукта или услуги, а также при поставке, разработке, применении по назначению, сопровождении и прекращении применения программных продуктов.

Основными характеристиками данного стандарта являются:

- динамичность: один процесс при необходимости вызывает другой или его часть, что позволяет реализовать любую модель жизненного цикла;
- адаптивность: стандарт предусматривает исключение процессов, видов деятельности и задач, неприменимых в конкретном проекте.

Кроме того, существуют международные стандарты (на английском языке), которые направлены на написание документации:

1. IEEE Std 1063-2001 «IEEE Standard for Software User Documentation» – стандарт для написания руководства пользователя. В документе обозначены требования к структуре, содержанию и формату инструкций пользователя.

2. IEEE Std 1016-1998 «IEEE Recommended Practice for Software Design Descriptions» – стандарт для написания технического описания программы. Представлены рекомендации к документам, описывающим архитектуру программного обеспечения.

3. ISO/IEC FDIS 18019:2004 «Guidelines for the design and preparation of user documentation for application software» – стандарт для написания руководства пользователя. В данном документе есть большое количество примеров. Также в приложениях есть чек-листы «что не забыть сделать в процессе разработки документации» и «что должно быть».

Документ особенно полезен начинающим специалистам.

4. ISO/IEC 26514:2008 «Requirements for designers and developers of user documentation» – стандарт для дизайнеров и разработчиков пользователей документации.

Международных стандартов довольно много, в каждой стране они свои, так как один и тот же стандарт не всегда может подойти и европейским, и азиатским компаниям.

Международный стандарт ISO/IEC 12207: 1995-08-01 Указанный стандарт является базовым и определяет жизненный цикл программного обеспечения. Стандарт определяет стратегию и общий порядок в создании и эксплуатации программного обеспечения, охватывая жизненный цикл от концептуальной идеи до завершения проекта. Стандарт ISO/IEC 12207 состоит из крупных обобщенных процессов: приобретение, поставка, разработка и т.д. В стандарте ISO/IEC 12207 описаны пять основных процессов жизненного цикла программного обеспечения: 1) процесс приобретения определяет действия предприятия - покупателя информационной системы, программного продукта или службы программного обеспечения; 2) процесс поставки определяет действия предприятия-поставщика по снабжению покупателя информационной системой, программным продуктом или службы программного обеспечения; 3) процесс разработки определяет действия предприятия-разработчика, который разрабатывает принципы построения программного изделия и собственно программный продукт; 4) процесс функционирования определяет действия предприятия-оператора, обслуживающего систему в целом. Сюда входят консультация пользователей, получение обратной связи и т.д.; 5) процесс сопровождения определяет действия персонала, обеспечивающего сопровождение программного продукта, т.е. управление модификацией программного продукта, поддержку текущего состояния и функциональной пригодности, установку и удаление. Кроме пяти основных процессов, ISO/IEC 12207 оговаривает восемь вспомогательных процессов, которые являются неотъемлемой частью всего жизненного цикла системы: 1) процесс решения проблем; 2) процесс документирования; 3) процесс управления конфигурацией; 4) процесс обеспечения

качества; 5) процесс верификации; 6) процесс аттестации; 7) процесс совместной оценки; 8) процесс аудита. В стандарте ISO/IEC 12207 также определяются четыре организационных процесса: 1) процесс управления; 2) процесс создания инфраструктуры; 3) процесс совершенствования; 4) процесс обучения. В стандарте ISO/IEC 12207 имеется дополнительный процесс, позволяющий адаптировать стандарт к условиям конкретного проекта. Рассмотрим особенности стандарта ISO/IEC 12207. Стандарт имеет динамический и адаптивный характер. Это обусловлено способом определения последовательности выполнения процессов и задач, при котором один процесс вызывает при необходимости другой или его часть либо вообще его исключает. Стандарт принципиально не содержит описания конкретных методов действий, а тем более заготовок решений или документации. Стандарт позволяет обеспечить качество с разной степенью организационной независимости контролирующей деятельности. Стандарт обеспечивает гибкую степень обязательности. Достаточно указать небольшое количество процессов и задач, соответствующих данному стандарту. Стандарт содержит предельно мало описаний, направленных на проектирование баз данных (возможны ситуации, когда базы данных и не используются). Ценность стандарта ISO/IEC 12207 заключается в том, что он дает набор задач, характеристик качества, критериев оценки, охватывающих все проектные ситуации. Например, для характеристики требования к программному обеспечению предусмотрено 10 классов характеристик качества: 1) функциональные и возможные спецификации, включая исполнение, физические характеристики и условия среды эксплуатации, при которых единица программного обеспечения должна быть выполнена; 2) внешние связи (интерфейс) с единицей программного обеспечения; 3) требования квалификации (набор критериев, которые должны быть выполнены, чтобы признать продукт готовым к использованию в целевой окружающей среде); 4) спецификации надежности, включая спецификации, связанные с методами функционирования и сопровождения, воздействия окружающей среды и травм персонала; 5) спецификации защищенности информации; 6) человеческие факторы (по эргономике и инженерной психологии); 7) определение данных и требований к базам данных; 8) установочные и приемочные требования поставляемого программного продукта в местах эксплуатации; 9) документация пользователя; 10) требования сервиса пользователя. При использовании стандарта стороны-участники ответственны: 1) за выбор модели жизненного цикла для разрабатываемого проекта; 2) адаптацию процессов и задач к этой модели; 3) выбор и применение методов разработки программного обеспечения; 4) выполнение действий и задач, подходящих для проекта программного обеспечения. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. 5.2.2.2 Специальные процессы программных средств 5.2.2.2.1

Процессы реализации программных средств Процессы реализации программных средств используются для создания конкретного элемента системы (составной части), выполненного в виде программного средства. Эти процессы преобразуют заданные характеристики поведения, интерфейсы и ограничения на реализацию в действия, результатом которых становится системный элемент, удовлетворяющий требованиям, вытекающим из системных требований. Специальным процессом является процесс реализации программных средств, выражающий специфически программную особенность процесса реализации. Процесс реализации программных средств включает в себя несколько специальных процессов более низкого уровня: а) процесс анализа требований к программным средствам; б) процесс проектирования архитектуры программных средств; в) процесс детального проектирования программных средств; г) процесс конструирования программных средств; д) процесс комплексирования программных средств; е) процесс квалификационного тестирования программных средств.

5.2.2.2.2 Процессы поддержки программных средств Процессы поддержки программных средств предусматривают специально сфокусированную совокупность действий, направленных на выполнение специализированного программного процесса. Любой поддерживающий процесс помогает процессу реализации программных средств как единое целое с обособленной целью, внося вклад в успех и качество программного проекта. Существует восемь таких процессов: а) процесс менеджмента документации программных средств; б) процесс менеджмента конфигурации программных средств; в) процесс обеспечения гарантии качества программных средств; г) процесс верификации программных средств; д) процесс валидации программных средств; е) процесс ревизии программных средств; ж) процесс аудита программных средств; з) процесс решения проблем в программных средствах.

5.2.2.2.3 Процессы повторного применения программных средств Группа процессов повторного применения программных средств состоит из трех процессов, которые поддерживают возможности организации использовать повторно составные части программных средств за границами проекта. Эти процессы уникальны, поскольку, в соответствии с их природой, они используются вне границ какого-либо конкретного проекта. Процессами повторного применения программных средств являются: а) процесс проектирования доменов; б) процесс менеджмента повторного применения активов; в) процесс менеджмента повторного применения программ.

4. Системы сопровождения документации.

Какой должна быть аналитическая документация успешного проекта? Разберем причины, элементы, шаги. В мире информационных технологий, где разработка программного

обеспечения является ключевой деятельностью, хорошая документация - необходимое условие успешного проекта.

Причины, по которым необходима аналитическая документация на проекте:

Понимание требований

Документация помогает уточнить и формализовать требования к проекту. Это позволяет убедиться в том, что все участники проекта понимают одинаково, что нужно разработать, и что результаты будут соответствовать ожиданиям заказчика.

Минимизация рисков

Документация позволяет выявить потенциальные риски и проблемы на ранних стадиях разработки проекта. Это помогает команде разработчиков предотвратить возможные ошибки и ускорить процесс разработки.

Улучшение коммуникации

Документация – это инструмент коммуникации между участниками проекта. Она позволяет четко определить требования и ожидания заказчика и участников проекта, что улучшает взаимопонимание и помогает ускорить процесс разработки.

Улучшение процесса разработки

Документация помогает улучшить процесс разработки, определив этапы, которые необходимо пройти и задачи, которые нужно выполнить. Она также помогает определить способы проверки результатов разработки и улучшить качество продукта.

Повторное использование информации

Документация – это ценный ресурс, который может быть использован в будущем для других проектов. Она позволяет повторно использовать информацию о требованиях и процессах разработки, что ускоряет разработку новых продуктов и повышает эффективность работы команды.

Как написать хороший документ? Первое, что необходимо учитывать при написании аналитической документации, - это ее цель.

Она должна быть понятной и полезной для всех членов команды проекта, включая разработчиков, тестировщиков и менеджеров проекта. Документация должна содержать всю необходимую информацию о проекте, включая его цели, требования, функциональные возможности, ограничения и т.д.

Кроме того, документация должна быть четкой и конкретной. Используйте понятный язык и структурируйте документ таким образом, чтобы было легко найти нужную информацию.

Используйте графики, таблицы, диаграммы и другие визуальные средства для иллюстрации ключевых решений. Это даст больше понимания команде как работает продукт.

Важно также убедиться, что документация актуальна и своевременна. Периодически

обновляйте ее в соответствии с изменениями в проекте и его требованиями. Кроме того, следует убедиться, что все члены команды проекта имеют доступ к актуальной версии документации.

Следует помнить, что документация не является конечной целью проекта, а скорее инструментом для достижения этой цели. Она должна быть написана с учетом конечного пользователя и технических требований проекта, а не только с точки зрения автора документации.

Документация включает в себя следующие элементы:

Спецификация требований. Этот документ описывает требования к продукту. В нем перечислены все функциональные и нефункциональные требования к продукту, его характеристики, а также ограничения. Спецификация требований является основным документом, который определяет, что должен делать продукт.

Анализ требований. Этот документ описывает, как будут удовлетворены требования к продукту. Он включает в себя описание архитектуры продукта, анализ рисков и возможностей, а также принятых решений.

Спецификация проекта. Этот документ описывает детали проектирования продукта. В нем перечислены все компоненты продукта, их свойства, отношения и функции. Спецификация проектирования является основным документом, который определяет, как должен работать продукт.

Технические требования. Этот документ описывает, как будут реализованы компоненты продукта. Он включает в себя описание алгоритмов, структуры данных и других элементов, которые необходимы для реализации продукта.

Теперь рассмотрим основные шаги, необходимые для составления аналитической документации:

Шаг 1: Определите цели и задачи проекта. Определите, какие результаты вы хотите получить от проекта и какие задачи необходимо решить для достижения этих результатов.

Шаг 2: Составьте спецификацию требований. Определите функциональные и нефункциональные требования к продукту. Для этого необходимо провести анализ бизнес-процессов, выявить потребности пользователей и определить требования к системе.

Шаг 3: Проанализируйте требования. Определите, как будут удовлетворены требования к продукту. Разработайте архитектуру системы, проведите анализ рисков и возможностей, определите принятые решения.

Шаг 4: Составьте спецификацию проекта. Определите детали проектирования продукта. Определите компоненты системы, их свойства, отношения и функции. Убедитесь, что спецификация проектирования соответствует спецификации требований.

Шаг 5: Составьте технические требования. Определите, как будут реализованы

компоненты продукта. Определите алгоритмы, структуры данных и другие элементы, которые необходимы для реализации продукта.

Шаг 6: Оформите документацию в соответствии с требованиями. Документация должна быть понятной и легко доступной для всех участников проекта. Убедитесь, что вся документация соответствует требованиям и стандартам, принятым в вашей компании (команде).

Некоторые полезные советы по написанию документации:

Использование блок-схем и диаграмм, чтобы визуализировать взаимодействие компонентов и элементов пользовательского интерфейса.

Использование нотаций, таких как UML, BPMN и т.д., чтобы обеспечить понимание и согласованность между всеми членами команды.

Использование примеров и сценариев использования (USE CASE) для лучшего понимания того, как продукт должен работать в реальной среде.

Шаг 7: Обновляйте документацию в процессе разработки. В ходе работы над проектом могут возникать изменения, поэтому необходимо регулярно обновлять документацию, чтобы она соответствовала текущей версии продукта.

Шаг 8: Обеспечьте доступность документации. Документация должна быть доступна всем участникам проекта. Разместите ее на общей платформе или в облачном хранилище. Хорошим решением будет Confluence, один из лучших инструментов для ведения проектной документации.

Заменить Confluence? Просто. Актуальный список российских аналогов «конфы», их преимущества и разбиение на три группы в зависимости от функциональности. В результате ухода с рынка иностранных компаний началась миграция компаний с Confluence на российские решения. Atlassian покинула рынок, лицензии оплатить не получается, а долгосрочные контракты с вендором заканчиваются.

Логично, что российские компании отказываться от наработанных баз корпоративного контента не хотят. Так, спрос рождает предложение и появление альтернатив Confluence не заставило себя ждать.

Российский рынок отреагировал очень динамично. Доступны десятки отечественных систем класса KMS (система управления знаниями). С помощью этих IT-инструментов можно автоматизировать процесс сбора, хранения и распространения корпоративных знаний, а также аккумулировать их внутри одной системы и открыть доступ для сотрудников или партнеров в любое время.

На деле Confluence тоже является KMS. Но интеграция с Jira, дополнительные плагины и копившаяся годами информация значительно усложняют «переезд» на другую систему.

Однако ситуация быстро меняется.

Отечественные разработчики понимают ценность Confluence и стараются приблизиться к ней по функциональности. Предлагаемые ими инструменты совместной работы уже ничем не уступают «конфе», а возможности поиска вообще ушли далеко вперед. К слову, у некоторых даже есть специальный мигратор, который перенесет данные из Confluence за пару часов.

Обзор решений

1. **Minerva Knowledge** Универсальное решение

Лаконичный и удобный интерфейс подкупит тех, кому требуется настроенное решение «из коробки». При этом структурно и функционально придаться сложно – всё на своих местах. Выстраивать иерархию и делить базу знаний на пространства удобно, а ролевую модель можно настроить вплоть до одной статьи. Редактор статей довольно обычный. Есть пара продвинутых фишек в виде поддержки сторонних инструментов по типу Draw.io и PlantUML. Совместное редактирование контента — на месте. Также присутствует контроль версий и режим сравнения контента.

Поиск — это основа любой KMS-системы. Здесь он восприимчив к опечаткам, умеет искать информацию внутри документов и выделяет необходимые отрывки сразу под поисковой строкой, без надобности заходить в статью. Minerva хорошо интегрируется с другими инструментами: task-трекеры, CRM, программы для контакт-центров. Поддерживается интеграция с Яндекс.Картами, YouTube, Google.Docs и Miro. В сторонний софт можно встроить их виджет. Кроме того, Minerva Knowledge поддерживает макросы и добавление плагинов.

Task-трекер — это калька с английского task tracker, что переводится как «отслеживание задач». Это то место, куда стекаются и где распределяются все задачи между всеми программистами. Там же может храниться вспомогательная информация по каждой задаче, вестись обсуждение, плестись интриги, разворачиваться семейные драмы.

CRM (Customer Relationship Management) — система управления взаимоотношениями с клиентами, которая помогает выстроить отношения с клиентами и следить за совершаемыми сделками.

В арсенале Minerva есть и система управления обучением, которую они активно продвигают в связке с базой знаний. Хороший инструмент для онбординга новых сотрудников и организации обучения.

Решение отлично подходит для клиентского сервиса, IT-подразделений, коммерции, HR-отделов и бэк-офиса за счет скорости, простоты использования и моментального ответа в поиске. Развернуть можно в облаке или на сервере вашей компании. Недавно разработчик представил мигратор контента из Confluence, позволяющий перенести все данные, в т.ч. комментарии и оценки статей, в Minerva Knowledge.

2. **Teamly** Для совместной работы

Teamly привлекает своим универсальным и красивым редактором с грамотной реализацией совместного редактирования статей. Эта функция будет полезна для материалов, где требуется «коллективное творчество». В статьи можно добавлять блоки с контентом из

YouTube, Google Docs и Draw.io. Интерфейс простой, но современный и функциональный. Интересная реализация разделения на пространства со статьями и файловое хранилище, которые взаимодействуют между собой.

Поиск здесь уже отличается от предыдущих решений. Он простой, без учета опечаток и синонимов. Но есть поиск внутри файлов — этим не обделили. Имеется встроенный в базовое решение модуль обучения для создания тестов и опросов, что будет полезно для HR-ов. Также в разработке находится таск-трекер для ведения проектов внутри базы знаний. Экспорт и импорт статей с офисными форматами тоже дружит.

Teamly — хороший вариант для ведения проектной деятельности в маленьких и средних командах. Не самое шустрое решение для контактного центра, но с основными задачами справится неплохо. До 7 пользователей в облаке Teamly можно попробовать бесплатно, для пробы самое то.

3. Плюс7 Маяк Для продаж и внешних коммуникаций

Российская система управления знаниями от разработчиков израильской KMS Lighthouse. Она на первый взгляд обладает не самым современным дизайном, но в удобстве использования и расположении элементов не уступает другим решениям. Визуальный редактор статей полнофункциональный, а основной его фичей выступает массовое редактирование материалов через выгрузку или интеграцию со сторонней системой.

Поиск сходен с предыдущим решением: есть фильтры, подсказки и исправления. Плюс7 Маяк интегрируется по API с большинством классов систем, которые могут использоваться в компании, а также с почтовыми и картографическими сервисами, голосовыми помощниками и чат-ботами. Мобильного приложения, к сожалению, нет.

Лучше всего Плюс7 Маяк раскроет себя в контакт-центре и отделе продаж из-за нацеленности на повышение эффективности внешних коммуникаций. Есть как в облаке, так и on-premise. Проведут демо, дадут доступ к демо-стенду — с этим проблем нет.

On-Premise (или серверное решение) — это пакет программного обеспечения, который разворачивается клиентом самостоятельно на собственном серверном оборудовании, поддерживается и администрируется силами самого заказчика. Заказчик приобретает только программу (лицензия, право на использование), и всё, что требуется для её полноценной работы, обеспечивает самостоятельно.

4. L2U InKnowledge Для тех, кто хочет кастомизировать

Первое, что бросается в глаза — гибкий UI, позволяющий настроить большинство элементов под себя и выбрать дизайн системы из нескольких шаблонов. В структуре можно создавать нечто похожее на сайты — отдельные области со своими пользователями и контентом. Редактор не обладает инструментами совместной работы, но хотя бы можно комментировать отдельные блоки текста в статье.

Поиск хороший, есть учет синонимов и опечаток. L2U может интегрироваться с разными решениями, а данные из системы послужат базой для чат-бота или FAQ-раздела.

Дополнительным плюсом выступает возможность создать портал самообслуживания для внешних партнёров.

Гибкая настройка позволяет применять решение в разных сферах, но я бы посоветовал использовать его в проектных командах, благодаря наличию функционала управления задачами внутри базы знаний. В то же время L2U очень не хватает режима совместного редактирования текста, инструментов обучения и мобильного приложения.

5. **EvaWiki (Kaiten)** Для проектных команд

Разработчик максимально пытался приблизить EvaWiki к Confluence, и это заметно. Использование отличных друг от друга по назначению модулей может образовать площадку для работы разных отделов в компании. Дизайн системы приятный и современный, присутствует иерархия документов по папкам и статьям. Внутри базы знаний можно создавать доски и списки задач — задел на командную работу.

Редактор обладает широким функционалом и не уступает другим решениям, но для использования некоторых базовых функций приходится заходить в дополнительные меню, что замедляет работу. У статей присутствует контроль версий и режимы сравнений. Но поиск по системе простой, без определения текста внутри файлов. Из плюсов — есть поиск сразу по всей базе знаний, а не только внутри пространства. Интеграции с внутренними продуктами Eva есть, а из сторонних он поддерживается Active Directory и Draw.io.

EvaWiki довольно универсальная платформа. Основной её недостаток — слабый поиск. Главное преимущество — широкий функционал по работе с документами (статьями), который сделает переезд с Confluence почти безболезненным. У системы есть 14-дневная демоверсия (Kaiten 30 дней), которую, в отличие от некоторых конкурентов, можно поставить без особых заморочек.

6. **CraftTalk KMS** Для контактных центров

Компания CraftTalk. Вся линейка её продуктов заточена под клиентский сервис и контактные центры. Новая KMS – не исключение. Интерфейс прост и удобен в эксплуатации, все его элементы рассчитаны на быстрый доступ к самым полезным функциям. Редактор поддерживает всё, что нужно, в том числе рисунки. Версионность дает возможность отследить изменения в статье и, при необходимости, откатиться назад.

На фоне конкурентов решение выделяется применением AI и интеграцией базы знаний в чат-боты. В поиске AI помогает находить информацию с учетом ошибок и делает это, действительно, быстро. CraftTalk также позволяет использовать базу знаний из мессенджеров, что по скорости делает его эффективнее мобильного приложения. Кстати, стоит отметить, что очевидный упор на клиентский сервис не делает решение исключительно моносферным.

7. **Битрикс 24** Для тех, кто хочет все и сразу

Система управления знаниями в Битрикс 24 — это лишь часть функционала большого программного комплекса. Из этого выходят определенные нюансы использования, но всё же её можно применять как полноценную KMS. Здесь присутствует wiki-структура, есть неплохой поиск в рамках отдельного пространства в базе знаний и настройка ролевой модели.

Редактор статей очень необычный. Он построен на блоковой основе, а не на пустом полотне текста. Уже внутри блоков можно вносить изменения и редактировать текст. Привыкнуть несложно, а результат многим покажется очень даже интересным. Интеграции тут завязаны на внутренние сервисы Битрикса, потому что решение предполагается как исчерпывающее. При желании можно подключить сторонние интеграции.

В качестве системы управления знаниями использовать Битрикс 24 стоит в том случае, если ваша организация уже пользуется продуктами Битрикса или хочет на них перейти. В иных сценариях лучше обратить своё внимание на более продвинутые базы знаний. Система разворачивается как в облаке, так и на сервере компании. Попробовать продукт можно бесплатно.

8. Атлас 24 Для тех, кто ценит простоту

Атлас24 лаконичен во всём. Его дизайн воспринимается очень интуитивно и придётся по вкусу любителям минимализма. Редактор прост, но симпатичную статью написать вполне реально. Под разные отделы компании можно создать своё пространство, которое никак не будет связано с общим.

Поиск работает только в рамках одного пространства и даёт чёткий результат слово в слово по запросу. «Интегрируйте с Битрикс24 и AmoCRM» гласит слоган на странице сайта. На этом интеграции заканчиваются, но зато с ними всё будет работать идеально. В системе также есть возможность создавать тесты для сотрудников и для внешних пользователей.

В теории решение не имеет какой-то конкретной специализации. Важное замечание: Атлас 24 существует только в облачном варианте, поэтому подходит, скорее, для тех компаний, которые не сильно беспокоятся о конфиденциальности данных.

Как видим, заменить Confluence на отечественное решение, отвечающее запросу бизнеса, не так уж сложно. Да, в некоторых аспектах (дизайн, функциональность, экосистема и эргономика) есть, над чем работать. Однако у российских платформ полно собственных преимуществ. Например, отличный поиск.

Более того, российский рынок KMS находится вразвиге, поэтому все упомянутые решения будут продолжать совершенствоваться. При этом, миграция из системы в систему уже не представляется такой страшной.

Исходя из направлений работы, которые наиболее приоритетны для определенного рода деятельности, все решения можно разделить на 3 группы:

1. Тем, кому важны скорость использования, быстрый поиск и свобода в организации знаний, подойдут уже набравшиеся опыта решения: Minerva Knowledge, Плюс7 Маяк, L2U InKnowledge и CraftTalk KMS.

2. Если в приоритете сервисы для совместной работы, то советуем обратить внимание на EvaWiki, Minerva Knowledge и Teamly.

3. В случае, когда требуется суперкомбайн со всем и сразу, лучший вариант — Битрикс

24. Ну а для тем, кому нужно что-то «попроще» можно порекомендовать Атлас 24.