"Идеальная СЭД"

Это условная система, которая соответствует лучшим практикам в создании СЭД и является своего рода эталоном, с которым будут сравниваться реальные системы в ходе исследования.

Общие сведения	**Описание**	**Оценка**
Название продукта	**Идеальная СЭД**	*СЭД, какой она могла бы быть, если собрать в одном продукте все передовые технические решения и лучшие практики, известные на текущий момент.*
Версия	0.0	*В природе не существует.*
1. Архитектура		
Тип архитектуры • Уровни (звенья) • Серверы (логические) • Общая шина • Cloud-ready *(в т.ч. multi-tenancy)*	В корпоративном сегменте пока преобладает классическая 3-х уровневая архитектура, ставшая развитием архитектуры клиент-сервер. На сегодняшний день эта архитектура не является передовой и может восприниматься лишь как стандарт де-факто. Концепция SOA на основе единой шины данных (ESB) не получила надежной практической реализации ввиду своей сложности. Однако частный случай SOA – веб-сервисная архитектура, довольно распространен. Обеспечивает взаимодействие инкапсулируемых веб-сервисами приложений путем обмена сообщениями между ними в форме XML-документов на основе общего коммуникационного протокола (например, протокола SOAP). Главное достоинство веб-сервисной архитектуры – возможность достижения высокой степени интероперабельности независимо разработанных веб-приложений. Продукты нового поколения скорее всего будут создаваться в микросервисной архитектуре, что решает проблемы с добавлением функциональности без переделки всей системы, облегчает масштабирование. Сложность управления разработкой в этом случае возрастает, но это неизбежная плата за новые преимущества. Система должна быть готова к облачному развертыванию. Лучше, если используется одна и та же редакция продукта для обычного и облачного развертывания, а не специальная версия.	**5 баллов** – система реализована в сервисной архитектуре (SOA, веб-сервисная, микросервисная). **4 балла** – система реализована в многоуровневой уровневой архитектуре, обеспечивает горизонтальное масштабирование. **3 балла** – система реализована в классической 3-х уровневой архитектуре: сервер БД, бизнес-логика, клиент. **2 балла** – система реализована в архитектуре клиент-сервер. **+0,5 балла** если поддерживается работа в облаке в режиме multi tenancy (но не специальной версией, одним продуктом).

Модульность • Как реализуется • Взаимодействие модулей • Микросервисы	Современные системы строятся по модульному принципу. Это, с одной стороны, обеспечивает гибкое наращивание функциональности. С другой – если модули расширения реализованы в виде сервисов (web- или Windows), то они могут устанавливаться на физические или виртуальные сервера, отличные от основного сервера системы, что позволяет оптимизировать нагрузку и обеспечивает масштабируемость системы. Например, почти всегда в виде отдельных модулей в СЭД реализованы модуль управления бизнес-процессами (BPM или workflow) и модуль отчетов. В рамках данного исследования не рассматриваются вопросы лицензирования модулей расширения – надо ли их приобретать или они существуют как составные части системы. Идеальным решением с точки зрения современных архитектурных подходов подставляется концепция «магазина приложений», где доступны модули различного назначения: расширяющие функциональность, изменяющие интерфейс, коннекторы, утилиты и проч. В рамках исключительно предметной области документооборота такой подход реализовать едва ли возможно – не наберется критической массы разработчиков и пользователей этих расширений. Но если СЭД является частью более широкой экосистемы – то есть, реализована на какой-то платформе, где кроме нее есть еще множество бизнес-приложений, то появление магазина по типу App Store становится более вероятным.	**5 баллов** –. есть магазин приложений, где пользователь может выбрать нужные ему модули и установить их в свою систему. Некоторые модули реализованы в виде сервисов и устанавливаться отдельно, другие – в виде плагинов и ставятся на сервер системы. **4 балла** – система содержит ядро и набор модулей, устанавливаемых дополнительно. Модули расширения также могут изменять базовую функциональность без модификации ядра. **3 балла** – система имеет ограниченный набор модулей, из которых собирается конкретная конфигурация. Разработка новых модулей – на стороне вендора или авторизованных партнеров. **2 балла** – система представляет собой монолитный продукт. Расширение функциональности возможно только путем доработки системы целиком. **+0,5 балла** за широкий ассортимент готовых модулей.
Хранилище контента • Место хранения *(БД или файловая система)* • Централизованное или распределенное • Поддержка систем хранения *(обычные СХД и специальные: EMC Centera, NetApp, и т.д.)* • Управление хранением *(контроль емкости, перемещение данных)*	Лучшим вариантом считается, когда репозиторий СЭД выполнен в гибридной архитектуре. То есть, метаданные (атрибуты) документов хранятся в СУБД, а контент (файлы документов) – в виде файлов в специально отведенной области в файловой системе. Это обеспечивает более высокую масштабируемость и гибкость в управлении хранением. Хранение контента (файлов документов) внутри БД может поддерживаться как опция, но подобный подход сегодня не является мейнстримом. Ведущие ECM-вендоры, такие как Open Text, Alfresco и др. этот подход не используют.	**5 баллов** – метаданные лежат в БД, файлы (контент) на файловой системе. Поддерживается управление хранением и перемещение между носителями. Поддерживается распределенная схема файлового хранилища. **4 балла** – метаданные в БД, файлы (контент) на файловой системе. Есть инструменты управления хранилищем. **3 балла** – метаданные в БД, файлы (контент) в БД – как основной вариант,

• Иерархическое хранение *(перенос в оффлайн или на медленные носители)* • Защита хранилища	Весьма желательна поддержка специализированных СХД, обеспечивающих дополнительную надежность хранения и функциональность однократной записи, как, например, EMC Centera. Средства администрирования системы должны включать управление размещением файлов и контролировать их перемещение и неизменяемость. С удешевлением стоимости дисков потребность в иерархическом хранении стала менее актуальной. Однако, для больших массивов документов, например, при создании электронных архивов, это может быть востребовано. Для дополнительной защиты хранилища контента, кроме штатных средств ОС, могут использоваться средства шифрования, встроенные в систему Есть встроенные средства или отдельная утилита для управления размещением файлов, их переносом на другие носители и контролем целостности.	есть опция хранения на файловой системе. **2 балла** – весь контент хранится в БД. Это выглядит проще, но при росте объема быстро вызывает проблемы масштабирования. **+0,5 балла** за интеграцию со специализированными СХД, библиотеками оптических дисков, ленточными библиотеками и пр.
Распределенность *Как обеспечивается работа в распределенной среде в рамках одного предприятия или шире*	Все крупные заказчики требуют поддержки распределенной архитектуры. С точки зрения функционального и процессного проектирования (то есть, с точки зрения бизнес-аналитика и конечного пользователя) удобнее, когда на логическом уровне система выглядит централизованной, а ее техническая сложность скрыта. Например, когда вы заходите в свой аккаунт Google, вы не размышляете о том, к какому конкретно серверу нужно подключиться. Это было бы решение уровня Web-scale. Текущие внедрения СЭД пока ограничиваются корпоративными рамками, максимум порядка 100 тыс. пользователей – это далеко еще не Web-scale, где счет может идти на сотни миллионов. Однако, тенденция к укрупнению систем прослеживается (например, госуслуги) и поэтому подобные архитектурные подходы с неограниченной масштабируемостью могут оказаться продуктивны. Также, эта архитектура нужна для облачного развертывания. В более практическом ключе, СЭД можно уподобить корпоративному порталу: единая точка входа для всех сотрудников организации. логически система выглядит централизованной, но на техническом уровне есть горизонтальное масштабирование сервисов, балансировка нагрузки и	**5 баллов** – архитектура системы позволяет развернуть глобальное решение (Web-scale) или публичное облако. **4 балла** – единая точка входа в документооборот для всех сотрудников организации (по аналогии с корпоративным порталом). Техническая и физическая сложность скрыта. **3 балла** – реализуется посредством сети взаимодействующих серверов СЭД и сервиса обмена документами. Сервера СЭД на уровне филиалов или предприятий холдинга. Возможны сквозные бизнес-процессы. **2 балла** – реализуется посредством сети серверов СЭД, которые могут обмениваться документами между собой в режиме «входящий-исходящий».

	кэширование; полномочия администратора могут делегироваться на уровень филиала или предприятия в холдинге. Исторически более ранняя схема поддержки работы в распределенной среде подразумевала установку самостоятельных серверов СЭД в структурных единицах и организацию обмена документами между ними посредством специального сервиса. В этом случае также возможна поддержка сквозных бизнес-процессов, когда передача документов между серверами осуществляется прозрачным для пользователя образом. Более примитивная (и устаревшая) схема подразумевает ведение общих справочников, но передачу документов между структурными единицами в режиме «входящий-исходящий», то есть в режиме эмуляции бумажного документооборота.	**0 баллов** – работа в распределенной среде не обеспечивается.
Стандарты и спецификации *(Полное соответствие подтверждается сертификатом. При отсутствии сертификата разработчик может декларировать соответствие стандарту от своего имени.)*	Система должна быть построена в соответствии с известными стандартами и спецификациями. Это облегчает реализацию политик управления документами по ГОСТ ИСО 15489 и если не гарантирует, то создает предпосылки для функциональной полноты решения. Национальная «прописка» стандарта с функциональной или архитектурной точки зрения не является ограничением для следования ему в России – так самолеты летают по законам аэродинамики, а не по директивам чиновников. Наиболее важными стандартами в области ECM/СЭД являются: • MoReq – в части полноты функциональности; • CMIS – в части обеспечения интероперабельности и интеграции. Регуляторные документы, стандарты оформления документов, соответствие требованиям информационной безопасности (ИБ) -- за рамками данного исследования.	*Соревноваться по количеству поддерживаемых стандартов было бы бессмысленно. ИТ-систему в принципе нельзя построить без стандартов.* *Но стоит выделить соответствие наиболее важным стандартам в предметной области ECM:* **+1 балл** за соответствие MoReq. **+1 балл** за соответствие CMIS.
Протоколы *(WebDAV и др., имеющие отношение к предметной области управления документами и процессами)*	Система должна поддерживать известные протоколы, применяемые в области управления контентом. Это нужно для упрощения интеграции с инфраструктурными сервисами (служба каталогов, электронная почта), с другими сервисами управления контентом и для взаимодействия как между компонентами своего приложения, так и с внешними модулями. Использование тех или иных протоколов во многом обусловлено выбором базовой платформы СЭД, поэтому прямое сравнение было бы некорректным.	*Справочная информация. Оценка не выставляется.*

Нотации описания процессов	Автоматизация бизнес-процессов является одним из основных мотивов внедрения СЭД. Методологически область BPM очень хорошо проработана, есть общепринятые нотации описания процессов, так что «изобретение велосипедов» здесь будет смотреться весьма странно. Кроме чисто процессоного подхода также актуальна концепция кейс-менеджмента, поддержка специализированной нотации описания работы с кейсами была бы плюсом. • BPMN • CMMN • BPEL • XPDL • jPDL • EPC (ARIS)	**5 баллов** – Система имеет собственный редактор бизнес-процессов, который поддерживает несколько популярных нотаций и совместим с инструментами моделирования бизнес-процессов (ARIS, Business Studio, Bizagi или аналогичные). **4 балла** – Система использует одну из популярных нотаций и собственный редактор бизнес-процессов. **3 балла** – Система использует собственную нотацию и редактор бизнес-процессов. **2 балла** – Процессы жестко запрограммированы. **0 баллов** – Система не поддерживает концепцию бизнес-процессов. **+0,5 балла** за нотацию кейс-менеджмент (CMMN).
2. Инфраструктура		
Сервер • Возможные варианты ОС • Кросплатформенность? *(Один продукт под все ОС или разные реализации?)*	Система должна работать на Windows и Linux. (Из практических соображений более экзотические варианты ОС не рассматриваются.) Предпочтительнее иметь одну редакцию системы под обе платформы. Это экономит ресурсы разработчика, упрощает поддержку и гарантирует совместимость. Кроссплатформенность обеспечивается благодаря использованию высокоуровневых языков программирования, сред разработки и выполнения, поддерживающих условную компиляцию, компоновку и выполнение кода для различных платформ. Наличие разных редакций под разные платформы – это фактически наличие разных продуктов, каждый из которых нужно оценивать отдельно.	**5 баллов** – Кроссплатформенность на уровне среды выполнения. Единый код и единая сборка продукта под платформы Windows и Linux. **4 балла** – Кроссплатформенность на уровне исходного кода с использованием условной компиляции. Отдельные сборки под разные платформы. **3 балла** – Система работает на Windows и Linux, но это две отдельные разработки, схожие функционально. **2 балла** – Система работает только на одной платформе.

СУБД • Поддерживаемые СУБД • Как организована работа с БД • Использование средств БД *(триггеры, хранимые процедуры, представления)* • Независимость от СУБД?	В идеальном случае, СЭД должна быть абсолютно нейтральна по отношению к СУБД, то есть, одинаково работать с любой. На практике система должна поддерживать популярные проприетарные и свободные реляционные СУБД. Как минимум, MS SQL, Oracle, Postgres. Взаимодействие системы с базой данных желательно организовать через слой ORM (Object-Relational Mapping). Для приложений СЭД, где используются сложные объекты, но совершается мало транзакций (по сравнению с банковскими системами, например) использование ORM будет наиболее эффективным решением. Это облегчает жизнь разработчиков, но не становится узким горлом в достижении производительности. Если не используется ORM, то как минимум, следует избегать использования в коде продукта специфических возможностей конкретных СУБД, поскольку это ухудшает переносимость. (Теоретически можно рассмотреть архитектуры СЭД, не использующие реляционные СУБД, например, Hadoop и другие NoSQL базы данных, но это выходит за рамки практического исследования.)	**5 баллов** – Система работает с любой СУБД, как минимум, с MS SQL, Oracle, Postgres, используя какой-либо ORM-фреймворк. Тип СУБД выбирается на этапе конфигурации. **4 балла** – Система поддерживает работу с разными СУБД на уровне исходного кода (условная компиляция). Тип СУБД выбирается на этапе конфигурации. **3 балла** – Существуют разные редакции системы для работы с разными СУБД. **2 балла** – Система жестко привязана к единственной СУБД.
Сервер приложений *(Если используется)*	Сервер приложений – достаточно широкая концепция. Аналитики и разработчики дают разные определения. Мы будем придерживаться определения Gartner: «Сервер приложений – это современная форма платформенного ПО промежуточного слоя (middleware), которое находится между операционной системой (ОС) с одной стороны, внешними ресурсами (такими как СУБД, коммуникационными службами и интернет-сервисами) с другой стороны и приложениями пользователей на третьей стороне.» Сервер приложений служит хостом или контейнером для бизнес-логики приложения и обеспечивает доступ к нему клиентов, устойчивость к сбоям и балансировку нагрузки. Это позволяет разработчикам сфокусироваться на предметной области и не тратить ресурсы на программирование инфраструктурных сервисов. Web Server vs. Application Server. Раньше их функции четко разделялись – веб-сервер отвечал за обработку HTTP-запросов и выдавал клиентам содержимое веб-страниц; сервер приложений обращался к базе данных, выполнял бизнес-логику и передавал данные для визуализации страниц на	**5 баллов** – Система использует один из промышленных серверов приложений (соответственно выбранному стеку технологий – C++, Go, Java, JavaScript, .Net, Python, PHP, Perl). **4 балла** – в качестве сервера приложений используется Microsoft IIS (возникает платформенное ограничение – только Windows). **3 балла** – используется кросс-платформенный сервер приложений собственной разработки. **2 балла** – используется сервер приложений собственной разработки, жестко привязанный к одной платформе (как правило, Windows).

	веб-сервер. В современных продуктах эти границы стали достаточно условными, это приходится учитывать. В рейтингах наиболее популярные сервера приложений располагаются в следующем порядке: WebLogic, WebSphere, Tomcat, JBoss, GlassFish, IIS, TIBCO и т.д. Предпочтительно использование промышленных серверов приложений. Формально можно вынести слой бизнес-логики в отдельный модуль и назвать его сервером приложений, но разрабатывать собственное решение, сопоставимое по возможностям с промышленным – задача слишком объемная, чтобы ей заниматься на должном уровне только ради СЭД. Поэтому собственный сервер приложений получает более низкий балл.	**0 баллов** – сервер приложений не используется.
Клиент *(десктоп)* • Возможные варианты ОС • Кросплатформенность • Наличие толстого клиента • Устанавливаемые компоненты продукта на компьютере пользователя	В идеальном случае на компьютер пользователя не должно устанавливаться никаких локальных компонент СЭД. Даже функции администрирования должны выполняться из тонкого клиента. Такой подход значительно упрощает развертывание и эксплуатацию системы, а также экономит ресурсы разработчика и настоящее время стал стандартом де-факто в разработке корпоративных приложений. Исторически, толстый клиент, специфичный для конкретной ОС, позволял обеспечить более богатую функциональность по сравнению с веб-клиентом. Однако, такая точка зрения устарела – веб-клиенты сегодня позволяют реализовать любую функциональность и возможности интерфейса. Тем не менее, наличие толстого клиента можно рассматривать как дополнительный бонус. (Хотя его отсутствие при выборе системы не должно считаться недостатком.) Варианты реализации: • Windows-приложение; • Java-приложение (кроссплатформенность)	**3 балла** – есть полнофункциональный десктоп клиент Windows или Java. **0 баллов** – десктоп клиента нет.
Офисные пакеты • Какие поддерживаются? • Механизм интеграции	Необходимость бесшовной интеграции с офисными пакетами, пожалуй, единственный аргумент в защиту установки локальных компонент. Облачные приложения, такие, как Google Docs в корпоративном секторе пока не прижились. В качестве примера идеальной интеграции можно взять работу MS Office 365 с хранилищем OneDrive – документы прямо из офисных приложений сохраняются в репозиторий без каких-либо дополнительных действий.	**5 баллов** – поддержка всех популярных офисных пакетов (MS Office, LibreOffice, Мой Офис). Открытие и сохранение документов не требует ручных операций выгрузки-загрузки. **4 балла** – поддержка MS Office и LibreOffice.

	Дополнительно для СЭД было бы необходимо еще заполнение или редактирование карточки документа в рамках единого сценария работы. В порядке приоритетности должны поддерживаться следующие пакеты: • MS Office в разнообразных редакциях • LibreOffice/OpenOffice • Мой Офис • Google Docs и другие Поскольку большинство текстовых документов в СЭД довольно небольшие – 1-2 страницы и простые по верстке, то использовать для их редактирования MS Word в принципе избыточно. Поэтому может быть продуктивной идея использования встроенного непосредственно в СЭД веб-редактора документов.	**3 балла** – поддержка только MS Office. **2 балла** – работа с документами через выгрузку-возврат. **+0,5 балла** за наличие встроенного (веб) редактора для работы с простыми бизнес-документами без вызова профессионального офисного пакета.
Веб-клиент • Поддерживаемые браузеры • Технология *(напр., GWT)* • Стиль *(напр. Single page)*	На сегодня веб-клиент является стандартом де-факто для корпоративных систем. Причины этого очевидны – значительное упрощение поддержки, независимость от операционных систем (для многих пользователей стала принципиально важна работа на MacOS, Linux на рабочем столе скорее редкость). Технологии веб-разработки достигли высокого уровня зрелости и могут предоставить в среде браузера пользовательский опыт (UX), вполне сопоставимый с Windows-средой. Веб-клиент не может более считаться ограниченной по функциональности, легкой версией Windows-клиента. Самый передовой на сегодня подход – Progressive Web Application (PWA) – это название группы приложений, которые используют стек Web технологий (JS + HTML + CSS) и позволяют соединить простоту использования Web сайта со специфичными для нативных приложений операционной системы UX и техническими возможностями. Текущий стандарт де-факто в веб-разработке -Single Page Application, (SPA) — это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, обычно посредством AJAX. Rich Internet Application, (RIA) — это веб-приложение, загружаемое пользователем через интернет, предназначенное для выполнения функций традиционных настольных приложений и работающее на устройстве пользователя (не на сервере).	**5 баллов** – Progressive Web Application (PWA) **4 балла** – Single-page Application (SPA) **3 балла** – Rich Internet Application (RIA) **2 балла** – простой HTTP Web-клиент. **0 баллов** – нет Web-клиента. **+0,5 балла** – Администрирование полностью через веб-интерфейс.

	Полнота реализации функций системы в среде веб рассматривается как преимущество. Даже для администратора наличие толстого клиента не является более необходимым. • Internet Explorer и Chrome в качестве основных браузеров. • Дополнительно – FireFox, Safari, Opera, остальное – экзотика.	
Мобильный клиент • Платформы *(iOS, Android)* • Форм-фактор *(смартфон, планшет, смарт-часы)* • Технология *(нативное приложение, HTML5?)* • Функциональность *(полная, сокращенная)*	Во-первых, мобильный клиент должен быть демократичным – то есть, для всех пользователей, а не только для руководства. Для многих сотрудников мобильное устройство стало основным, поэтому для пользователя нужна полная функциональность, которая должна быть идентична функциональности веб-клиента (кроме, разве что, функций администрирования). Работа в offline перестает быть критически важным фактором, с развитием сетей 4G/5G и повышением качества связи практически всегда есть возможность быть online, даже в самолете многие авиакомпании предоставляют такой сервис. Обязательно поддержка iOS и Android, но вектор сместился с разработки отдельных нативных приложений под каждую платформу в сторону гибридных кросс-платформенных решений на основе современных фреймворков типа React Native от Facebook, Cordova или Xamarin. Параллельная разработка двух и более нативных приложений под разные платформы становится экономически нерентабельной. Планшеты в области корпоративной мобильности уступили лидерство новым смартфонам с большим экраном. Это значит, что приложения исключительно для планшетов устарели. Нужны приложения с адаптивными интерфейсами, способными работать на мобильных устройства разного форм-фактора. Поддержка умных часов – показатель продвинутости, но пока не мейнстрим. Разумеется, речь не идет о полноценной работе при помощи смарт-часов. Но уведомления о важных событиях или простые согласования, не требующие рассмотрения документов – вполне реализуемые функции.	**5 баллов** – кросс-платформенное гибридное (iOS/Android) мобильное приложение с полной функциональностью (не хуже, чем в веб) для исполнителей и руководителей. **4 балла** – полнофункциональные нативные приложения для iOS и Android, на смартфонах и планшетах. **3 балла** –адаптивные веб-приложения с полной функциональностью на смартфонах и планшетах. **2 балла** – нативные или веб приложение с сокращенной функциональностью («планшет руководителя»). **0 баллов** – нет мобильного клиента **+0,5 балла** за умные часы.
Полнотекстовый поиск • Используемый движок • Лингвистические сервисы • Другое *(напр., поиск изображений)*	Полнотекстовый поиск не просто одна из «фич» СЭД. Это возможность вовлечь неструктурированные данные – массив документов СЭД – в контур принятия решений, в аналитические системы. Это возможность извлекать знания и факты из текста.	**4 балла** – в СЭД интегрирован промышленный поисковый движок из числа лидеров, на основе которого возможна реализация концепции

	Разумеется, должен быть поиск по карточкам и по содержанию документов; конечно, сканированные документы должны распознаваться и храниться в виде двуслойного PDF, это уже 20 лет как известное решение. Неизвестны случаи, когда штатные средства ОС или СУБД использовались для создания продвинутых поисковых решений, включая текстовую аналитику и управление знаниями. Поэтому такой вариант реализации считается менее передовым, хотя позволяет обеспечить поисковую функциональность внутри СЭД. Отсутствие поиска по содержанию документов воспринимается как критический недостаток системы.	Enterprise Search (возможно расширение поисковых источников). **3 балла** – используются поисковые движки второго эшелона. **2 балла** – используются штатные средства поиска ОС и/или СУБД. **0 баллов** – нет единого полнотекстового поиска по карточкам и содержанию документов. **+0,5 балла** за расширенные лингвистические сервисы (учет морфологии русского языка – это уже стандарт де-факто).
Масштабируемость • Какие компоненты • Балансировка нагрузки • Кэширование контента • Нагрузочное тестирование • Реальные кейсы • Готовность к web-scale	Масштабируемость в целом определяется архитектурой. Основные параметры, перечисленные в данном пункте, уже рассматривались выше. Поэтому выставление отдельной оценки по масштабируемости привело бы к двойному начислению баллов. С практической точки зрения потенциальным заказчикам могут быть интересны конкретные кейсы использования систем под высокой нагрузкой в реальных проектах или в ходе тестирования. Однако, следует учитывать, что реальные и тестовые нагрузки могут быть далеки от теоретического предела нагрузочной способности систем. Также бывает, что высокие показатели масштабируемости в конкретном проекте достигаются за счет глубокой кастомизации и недоступны в типовой конфигурации. На этом основании принято решение не выставлять оценку по критерию масштабируемости. Информация, предоставленная участниками исследования может использоваться как справочная.	*В рамках данного исследования оценка не выставляется.* *Справочная информация.*
3. Использование		
Лицензирование *(Краткое описание лицензионной политики)*	Вообще говоря, лицензионная политика вендора не относится к числу технологических критериев. Однако, принятые правила лицензирования оказывают влияние на удобство эксплуатации системы.	**4 балла** – ясная и четкая система лицензирования. Все технологические компоненты включены в базовый комплект, отдельно доступны модули интеграции или отраслевые решения.

	Если оглядываться на мировой опыт, то уже довольно давно вендоры ECM отошли от продажи серверных лицензий и предоставляют только пакеты лицензий для клиентов. Чрезмерная детализация лицензирования отдельных функций или видов доступа к системе в результате становится проблемой для заказчика – ведь количество пользователей и их функциональные обязанности постоянно меняются, а оперативно проводить отдельные мелкие закупки часто нет возможности. Особенно, если организация живет по 44-ФЗ. Крайне неудобно, когда для участия в каком-то бизнес-процессе пользователю может потребоваться специальная лицензия. Например, возьмем управление договорами – при изменении регуляторной среды, оргструктуры или внутренней учетной политики организации, может вдруг оказаться, что в процесс договорной работы стало вовлечено больше человек, чем планировалось при закупке системы. Наличие разных редакций в зависимости от масштаба бизнеса следует признать обоснованным. При этом будет логично в серверную лицензию Enterprise сразу включить все технологические компоненты, которые могут понадобиться. Отдельно имеет смысл поставлять только специфические модули интеграции. Стоит считать best practice предоставление инструментов администрирования и разработки в составе басового комплекта, не требуя приобретения отдельных лицензий. У заказчика должен быть выбор, какого типа лицензии использовать для клиентов – именованные (фиксированные) или конкурентные. Для СЭД важна функция сканирования, которая обычно реализуется с использованием OEM-продуктов. Поскольку здесь возникают лицензионные платежи в сторону партнера, то наличие отдельной лицензии представляется оправданным. Дискриминация пользователей по типу интерфейса – десктоп, веб или мобильный – не должна существовать. Веб-доступ давно стал стандартом де-факто для корпоративных систем и взимание дополнительной платы за это выглядит архаичным. См., например, MS Outlook – можно пользоваться Windows-приложением и Web. Мобильность сегодня входит в фазу «must have» - это уже не эксклюзивная фича, а просто канал доступа к корпоративным данным. Наличие отдельной лицензии на мобильные приложения СЭД сдерживает	Сервер не лицензируется, единая клиентская лицензия для всех видов доступа. Тип лицензии – по выбору заказчика. **3 балла** – достаточно простая политика лицензирования, есть разные лицензии по виду доступа, есть лицензии на некоторые функции. **2 балла** – сложная система лицензирования, большое число серверных и клиентских компонент, очень мелкая детализация функций СЭД, отдельные лицензии для администраторов и разработчиков. **+0,5 балла** – есть лицензирование по модели SaaS. **+0,5 балла** – есть Unlimited-лицензия для крупных клиентов.

	распространение мобильности. Взгляните хотя бы на Microsoft – Office для iOS и Android можно использовать бесплатно. Если говорить об облаках, то необходимо предусмотреть пользование продуктом по модели SaaS. Крупные заказчики любят лицензии Unlimited. Не только ради статуса – просто ради экономии своих усилий по учету использования лицензий.	
Внедрение • Собственная методология • Начальная загрузка данных • Самостоятельное внедрение заказчиком возможно? • Обучение пользователей	Наличие фирменной методологии свидетельствует о зрелости продукта и обобщении накопленного опыта внедрения, это снижает риски проекта. Сегодня редко когда СЭД внедряется с нуля. Обычно происходит миграция с устаревшей системы на новую и в этом случае важно иметь удобные инструменты загрузки данных, а не делать это вручную или через написание скриптов. Сложность СЭД не запредельна, многие специалисты на стороне заказчика уже сталкивались с подобными системами, поэтому нормальной должна считаться ситуация самостоятельного внедрения СЭД силами заказчика. Обучение пользователей работе в СЭД может происходить в очной форме (занятия в классе, в т.ч. удаленно) или в виде онлайн-курсов, которые пользователи проходят самостоятельно. Весьма желательно прохождение формального экзамена на умение работать в системе, это дисциплинирует и мотивирует пользователей.	**+1 балл** – есть фирменная методология внедрения **+1 балл** – есть утилиты начальной загрузки данных. **+1 балл** – возможно самостоятельное внедрение силами заказчика (есть прецеденты). **+0,5 балла** – есть собственный учебный центр, возможно очное и онлайн обучение. **+0,5 балла** – есть формальное тестирование пользователей, прошедших обучение.
Эксплуатация • Обслуживающий персонал *(на 1000 пользователей)* • Трудоемкость сопровождения *(часов в месяц)* • Обновления *(как часто выходят, как их устанавливать)* • Риск несовместимости кастомных доработок с обновлениями *(Как противодействовать)* • Выведение из эксплуатации *(Возможность выгрузки данных и миграции на другую СЭД)*	Расходы на эксплуатацию в значительной мере влияют на общую стоимость владения системой. Однако, дать точную оценку вне контекста конкретного проекта было бы затруднительно. Поэтому в рамках данного исследования ограничимся качественными показателями – чем меньше затраты ресурсов на эксплуатацию, тем лучше. Также будем учитывать частоту обновлений – это помогает поддерживать систему в актуальном состоянии и быстрее доводить до пользователей новые функции. Всем СЭД, представленным в исследовании далеко до желаемой скорости обновлений, продекларированной Германом Грефом в рамках подхода Continuous delivery, но к этому надо стремиться.	**5 баллов** – минимальные ресурсы на эксплуатацию (1-место среди рассматриваемых систем) **4 балла** – 2-е место. **3 балла** – 3-е место. **2 балла** – ниже 3-го места. **+0,5 балла** – обновления не реже, чем раз в квартал.

Юзабилити • Стандарты, гайдлайны *(Есть ли оценки UI/UX внешними экспертами?)*	Основной фокус будет обращен на UX – на то, насколько удобно реализованы функции и сценарии работы с системой. При это должно быть удобно не только канцелярии, но и всем другим пользователям – исполнителям, руководителям, службе контроля, договорникам и т.д. Вторая составляющая оценки – современность и стильность интерфейса.	**5 баллов** – Высокое качество UX подтверждено независимой внешней оценкой. Удовлетворяет запросам разных категорий пользователей. Внешний вид соответствует современным представлениям. **4 балла** – UX на современном уровне, учитываются требования разных категорий пользователей. **3 балла** – UX проработан недостаточно, за основу взяты пожелания канцелярии как функционального заказчика. Внешний вид устаревший. **2 балла** – Ригидный интерфейс, сделанный программистами по своему разумению. Мнение пользователей не учитывалось.
Потребность в ресурсах *(Для сравнительного анализа условно рассматривается конфигурация на 1000 пользователей.)*	Чем менее требовательная система к ресурсам, тем лучше. Однако, трудно ввести какие-то абсолютные критерии для оценки данного параметра, поэтому проведем сравнение представленных систем между собой. Поскольку системы различаются по конфигурации и назначению серверов, определим условную необходимую вычислительную мощность – подсчитаем суммарное количество процессорных ядер на серверную группу и объем оперативной памяти. Будем учитывать только основные компоненты архитектуры СЭД – сервер приложений, сервер СУБД, отдельный веб-сервер (если есть) и некоторые другие сервисы, заявленные разработчиком. Общесистемные компоненты, такие как Active Directory, не учитываются. Объем жестких дисков определяется количеством хранимых документов и не зависит от характеристик конкретной системы. Соответственно, исключим из общего подсчета файл-серверы, обеспечивающие работу хранилища документов. Требования к клиентам можно считать условно одинаковыми, потому что для нормальной работы пользователям обычно нужен MS Office, а его аппетиты к ресурсам превышают потребности клиента любой СЭД.	**5 баллов** – 1-е место среди представленных систем. (То есть, система предъявляет минимальные требования к ресурсам на 1000 пользователей). **4 балла** – 2-е место. **3 балла** – 3-е место. **2 балла** – 4-е место и ниже.
4. Развитие		

Языки разработки • Сервер • Middleware • Клиенты	Язык разработки должен быть широко популярным, стабильным, принятым в корпоративной среде и оставаться актуальным. Например, COBOL не отвечает этому критерию, хотя унаследованные приложения на нем все еще эксплуатируются. Язык Go – современный и перспективный, но еще не воспринят в сегменте enterprise. Предпочтение должно отдаваться классическим языкам разработки. Использование экзотических или проприетарных языков будет считаться фактором риска.	**5 баллов** – Используются популярные языки (Python, Java, JavaScript, C++, C#. PHP) **4 балла** – Используются очень новые языки (риск нестабильности работы и недостаточного количества разработчиков, например, Groovy, Rust, Elixir, Go.) **3 балла** – Используются устаревшие языки (риск утраты актуальности и сокращения базы разработчиков, например, Lotus Notes, C, Delpi) **2 балла** – Используются собственные или какие-то редкие языки.
Средства разработки *(На чем разработан сам продукт, что используется для доработок/кастомизации.)*	Должны использоваться известные и общепринятые средства разработки, с которыми знакомо множество программистов. Лучше, когда средства разработки доступны под свободной лицензией – кроме финансовой выгоды важен еще один момент: больше аудитория программистов, поскольку они могут себе позволить изучать такой инструмент.	**5 баллов** – используются открытые общедоступные и популярные средства разработки. **4 балла** – используются популярные проприетарные средства разработки. **3 балла** – используются средства разработки, теряющие популярность. **2 балла** – используются собственные проприетарные средства разработки. Когда используются разные средства разработки для серверной и клиентской части, то берется средняя оценка.
Наличие API, его документированность и тип *(Например, REST)*	Закрытых систем, вовсе не имеющих API, на рынке не осталось. По данным сайта ProgrammableWeb, сегодня доступно более 18 тысяч API для самых разнообразных систем и приложений. Но кроме факта наличия API, нужно смотреть на его качество и соответствие современным архитектурам и паттернам разработки.	*Оценка не проводилось.* *Не удалось выработать единой шкалы оценки для сравнения API на разных платформах.*

	Поскольку приоритетом для корпоративных приложений, в том числе СЭД, является веб-клиент, то в первую очередь должны быть представлены API для разработки веб-сервисов: REST, SOAP, XML-RPC и/или JSON-RPC. Все представленные в исследовании системы обладают богатыми и документированными API. Однако задача оценки качества API требует гораздо более глубокого изучения, выходящего за рамки данного исследования. Поэтому было принято решение оценку по данному критерию не проводить.	
Интеграция *(С какими системами имеется готовая интеграция)*	Должна быть готовая интеграция с популярными ERP-системами, системами потокового ввода, электронной почтой, службой каталогов и средством электронной подписи. Примеры: • 1C, SAP • ABBYY, Kofax, и др. • MS Exchange • Active Directory • Крипто ПРО	**5 баллов** – более 7 интеграций. **4 балла** – 5-7 интеграций. **3 балла** – 3-4 интеграции. **2 балла** – 1-2 интеграции. **0 баллов** – нет готовой интеграции. *При подсчете баллов интеграции с разными версиями/редакциями одной системы учитываются однократно.*
Открытость *(Является ли код продукта открытым? Если нет, может ли заказчик получить доступ к коду и на каких условиях?)*	В современном мире явно прослеживается тенденция перехода к ПО с открытым исходным кодом (open source). Это не обязательно предусматривает бесплатность. Тем не менее, корпоративный мир пока продолжает широко использовать проприетарное ПО, поэтому если оставаться в рамках реальности, то при выборе системы критерий открытости следует рассматривать как важный, но не определяющий фактор. (Если только ваша нынешняя ИТ-стратегия не предусматривает переход на СПО.) Однако, в среднесрочной перспективе (3-7) лет, СПО имеет шансы стать мейнстримом. Это связано отнюдь не с экономией на лицензиях, а с возможностями на порядок более быстрого внесения изменений в решения на основе СПО. Фактически, переход на продукты Open Source влечет за собой переход на Agile и внедрение практики DevOps. (Что сегодня уже можно наблюдать на примере Сбербанка.)	**4 балла** – продукт является СПО (возможно наличие корпоративной платной поддержки) **3 балла** – код продукта открыт, но приобретается лицензий. Клиент имеет право и возможность изменения исходного кода. **2 балла** – код открыт частично – решения, настройки и проч., но не ядро системы. **0 баллов** – код закрытый, проприетарный.
5. Внешняя среда		

Кадровые ресурсы • Требования к квалификации разработчиков • Их доступность на рынке	На рынке должно быть много квалифицированных и начинающих разработчиков по технологиям, используемым в данной СЭД. Доступность разработчиков снижает риски выполнения крупных проектов, обеспечивает стабильность компании. Есть возможность хантить звезд, чтобы усилить собственную команду. Технология должна быть широко популярной и использоваться для создания самых разнообразных систем – область СЭД а-приори слишком узка, чтобы по этому направлению готовилось массовое количество разработчиков. (См., например, с 1С.) Популярность – явление временное. Если пик расцвета технологии миновал, то у молодежи исчезает стимул изучать ее и начинается старение контингента разработчиков. Например, средний возраст разработчиков на Cobol или FoxPro составляет примерно 60 лет, а разработчиков Java – 25. Этот фактор также надо учитывать. Использование уникальных и малоизвестных технологий создает риск в привлечении человеческих ресурсов.	**5 баллов** – на рынке есть достаточное количество разработчиков разной квалификации и опыта. Есть широкая и независимая от поставщика СЭД система подготовки разработчиков. Технология очень популярна. **4 балла** – технологии, используемые в СЭД, миновали пик популярности, но опытных разработчиков на рынке еще много. **3 балла** – используются устаревшие технологии, число разработчиков сокращается. **2 балла** – используются собственные инструменты и языки разработки, разработчиков нужно учить с нуля. **-0,5 балла** – есть дефицит разработчиков по данной технологии.
Импортозамещение • Продукт в реестре МКС • Полнота стека технологий *(Возможность работать без использования продуктов, не входящих в Реестр)* • Средства разработки	Импортозамещение – политический фактор, влияние которого нужно учитывать. С формальной стороны можно ограничиться внесением продукта, разработанного на проприетарном импортном стеке в реестр отечественного софта, но это решает вопрос импортозамещения лишь частично. Должна быть возможность развернуть решение на инфраструктуре, полностью отвечающей критериям импортозамещения, либо, как минимум, построенной на СПО, включая также средства разработки. Также следует различать решения, построенные на основе западных ECM-платформ, пусть даже с открытым кодом и действительно российские разработки в экосистеме СПО. В первом случае ядро системы все равно создается не в России и у локального разработчика есть лишь ограниченные возможности влиять на развитие продукта. Возможность взять открытый код и продолжить разработку самостоятельно – это иллюзия, потому что просто не хватит ресурсов. То есть эта ситуация аналогична партнерству с любым западным вендором ECM.	**5 баллов** – российская разработка на основе СПО с поддержкой полного стека технологий начиная с ОС, заканчивая прикладным уровнем. Среда разработки также СПО или российская. **4 балла** – популярный СПО-продукт класса ECM с кастомизацией для российской специфики. Полный стек технологий отвечает критерию импортозамещения. **3 балла** – российская разработка на импортном стеке технологий с поддержкой импортонезависимой среды (ОС, СУБД). **+0,5 балла** - СУБД из реестра, а ОС только импортная.

	Второй вариант – когда ядро платформы разрабатывается в России, также, как и прикладные решения на его основе. Только тогда можно говорить о реальном импортозамещении. В качестве примера можно привести Acronis и Росплатформу.	**+0,5 балла** - клиентская часто способна работать в импортонезависимой среде, серверная часть только на импортном стеке. **2 балла** – российская разработка на импортном стеке, импортная СУБД и ОС. **0 баллов** – импортный продукт на импортном стеке технологий.
Экспортный потенциал *Пригоден ли продукт или его модификации для использования за пределами РФ и СНГ? Есть ли подобный опыт?*	Ориентация только на внутренний рынок есть серьезное ограничение – более тепличные условия дают меньше стимулов к развитию продукта. Разумеется, для российских заказчиков актуальна поддержка национальных особенностей документооборота и соответствие требованиям регуляторов. Однако, эти вещи не должны быть жестко зашиты в код, это должно решаться настройками и кастомизацией. По мере значимости фактора: • Есть пользователи в дальнем зарубежье; • Есть пользователи в ближнем зарубежье; • Есть возможность экспорта; • Продукт никому не нужен за рубежом.	**5 баллов** – российская компания глобального уровня с развитой сетью представительств и партнеров (например, ABBYY) **4 балла** – стабильные продажи за в дальнее зарубежье по отдельным странам и регионам. **3 балла** – есть успешные прецеденты проектов за рубежом. **2 балла** – есть продажи в ближнем зарубежье. **0 баллов** – нет международной активности.
Долгосрочная живучесть • Собственное коммьюнити *Есть ли независимые разработчики (не реселлеры), их влияние на развитие продукта, возможность создания производных продуктов* • Участие в экосистемах *«Социальный граф» продукта – насколько сильны связи с сообществами, какую роль продукт в них играет. Например, в экосистема SharePoint.*	Следует различать партнерскую сеть и коммьюнити. При том, что в обоих случаях это выглядит как клуб любителей какого-то продукта, механизм партнерства базируется на коммерческой составляющей -—партнер должен выполнять план продаж, тогда как члены коммьюнити не связаны подобными обязательствами. СЭД – слишком узкая предметная область, чтобы сформировать жизнеспособное коммьюнити разработчиков вокруг одного продукта. Более реальная ситуация, когда СЭД является органичной частью более широкой экосистемы – ведь в большинстве решений для бизнеса так или иначе востребованы сервисы управления документами и задачами. (Например, экосистема 1С.)	**5 баллов** – есть развитое международное коммьюнити разработчиков, сама платформа также является частью более широкой открытой экосистемы. **4 балла** – открытое коммьюнити в начальной стадии развития. **3 балла** – есть развитая сеть партнеров, есть магазин приложений на платформе, партнеры используют решения друг друга.

	Более мощные экосистемы привлекают больше пользователей и часто становятся основой корпоративной архитектуры, что предопределяет выбор системы.	**2 балла** – есть сеть партнеров, есть решения-надстройки и расширения, разработанные партнерами. **0 баллов** – разработкой занимается только сама компания.