УДК 004

Дорощенко Александр Леонидович
Амурский государственный университет
г. Благовещенск, Россия
E-mail: aldoroshchenko@mail.ru
Фомин Денис Васильевич
Амурский государственный университет
г. Благовещенск, Россия
E-mail: gefest-uni@yandex.ru
Жилиндина Ольга Викторовна
Амурский государственный университет
г. Благовещенск, Россия
E-mail: olgashest@mail.ru
Doroshchenko Alexander Leonidovich
Amur State University
Blagoveschensk, Russia
E-mail: aldoroshchenko@mail.ru
Fomin Denis Vasilyevich
Amur State University
Blagoveschensk, Russia
E-mail: gefest-uni@yandex.ru
Zhilindina Olga Viktorovna
Amur State University
Blagoveschensk, Russia
E-mail: olgashest@mail.ru

РАЗВИТИЕ ПРОГРАММНОГО КОМПЛЕКСА АВТОМАТИЧЕСКОЙ ИНВЕНТАРИЗАЦИИ ИТ-ИНФРАСТРУКТУРЫ INVOI

DEVELOPMENT OF THE INVOI SOFTWARE COMPLEX FOR AUTOMATIC INVENTORIZATION OF IT-INFRASTRUCTURE

Аннотация. Рассмотрена проблема инвентаризации компьютерной сети, обсуждены способы решения, представлены промежуточные выводы. Проанализированы наиболее популярные существующие инструменты сканирования сети, их возможности и недостатки. Учтены преимущества и недостатки Invoi 1.0. Предложены планы по разработке новой редакции программы, включающие изменения и новые возможности в архитектуре данного программного обеспечения.

Abstract. The problem of computer network inventory is considered, solutions are discussed, and intermediate conclusions are presented. The most popular existing network scanning tools, their capabilities and disadvantages are analyzed. The advantages and disadvantages of

Invoi 1.0 are taken into account. Plans are proposed for the development of a new edition of the program, including changes and new features in the architecture of this software.

Ключевые слова: компьютерные сети, комплексы программ, инвентаризация, автоматизация, системное администрирование, кибербезопасность, информационная безопасность.

Key words: computer networks, software complexes, inventory, automation, system administration, cybersecurity, information security.

Введение

У каждой современной организации есть собственная ИТ-инфраструктура, позволяющая автоматизированно решать многие задачи хозяйственной, экономической, юридической и других сторон деятельности. К сожалению, ИТ-инфраструктура уязвима и является объектом для реализации множества угроз разной природы и разного уровня сложности – от единичных мелких неисправностей и случайных ошибок пользователей до намеренной порчи и сложных скоординированных атак высококвалифицированных злоумышленников.

Для обеспечения необходимых уровней функционирования и защищенности ИТ-инфраструктуры предприятия применяют сложные дорогостоящие комплексы, включающие среди прочего SIEM-системы (СерчИнформ SIEM, MaxPatrol, KOMRAD Enterprise, Security Capsule, RUSIEM, KUMA Platform) [1-7], службы каталогов и системы централизованного управления (ActiveDirectory, Ред Адм, Samba4, OpenLDAP) [8-17]. Перечисленные программные продукты предоставляют большие возможности по сбору информации, контролю и управлению ключевым компонентом ИТ-инфраструктуры – компьютерной сетью. Также существует ряд более узко специализированных решений, предназначенных для сбора информации о техническом состоянии отдельных устройств и сети в целом.

Инвентаризация позволяет не только предотвратить возникновение серьезных неполадок, выявляя первые признаки их возникновения, но и устранить часть угроз кибербезопасности ИТ-инфраструктуры путем выявления несанкционированных изменений в программном и аппаратном обеспечении узлов сети.

Однако существующие решения, как правило, ориентированы на сети крупных предприятий, потому обходятся дорого, требовательны к ресурсам, сложны в использовании и обслуживании. Это делает актуальной разработку программных средств инвентаризации, ориентированных именно на компьютерные сети малого и среднего масштаба.

Задача инвентаризации

Инвентаризация – сбор информации о физических средствах за счет обхода и документирования. Инвентаризация сети – сбор информации о компьютерах и других физических устройствах в локальной сети и их состоянии.

Основная цель инвентаризации состоит в том, чтобы получить полное представление о топологии сети, ее конфигурации, оборудовании и программном обеспечении. При этом

существуют два основных подхода к инвентаризации сети – ручной и автоматизированный. А сам процесс инвентаризации включает четыре основных шага: 1) идентификация устройств в сети; 2) сбор данных об устройствах; 3) построение топологии сети;4) документирование результатов.

Разница между подходами выражается в способе решения каждого из этапов. Так, при ручной инвентаризации, вследствие необходимости физически обходить каждое устройство сети, процессы идентификация устройств, сбор данных и построение топологии осуществляются одновременно. Впоследствии собранная информация заносится в учетную систему, которая документирует результаты в виде отчета. Проблема такого подхода в значительных затратах времени и человеческих ресурсов, что не позволяет оперативно обнаружить проблему.

Автоматизированная инвентаризация предполагает следующее выполнение названных этапов:

1) программный комплекс сканирует сеть для идентификации активных узлов, при необходимости строит топологию;

2) для каждого активного узла собирается вся необходимая для проверки информация;

3) результат сбора данных далее можно оформить в виде отчета.

Инвентаризация с таким подходом экономит как время, так и человеческие ресурсы: необходим минимум персонала для проведения данной операции – один-два человека, использующих специальный программный комплекс.

Среди типов решений автоматизированной инвентаризации наблюдается небольшое разнообразие: приложения с интерфейсом командной строки, веб-приложения, административные скрипты, системные программы и утилиты.

В особенности стоит выделить Nmap (Network Mapper) [18]. Это решение представляет собой приложение с интерфейсом командной строки (CLI). Оно позволяет просканировать заданный пользователем диапазон IP-адресов или домен. Nmap дает возможность сканировать с использованием полуоткрытых соединений, системного вызова TCP connect, протокола UDP, уязвимостей в спецификации протоколов.

В последних версиях данной программы добавлен NSE – Nmap Scripting Engine – движок для пользовательских сценариев на языке Lua.

Существует также библиотека, реализующая возможности Nmap на платформе .NET. Она называется SaltwaterTaffy DotNmap [19].

Среди веб-приложений самым популярным для малого и среднего бизнеса является Spiceworks Inventory [20]. Оно дает базовые возможности инвентаризации и пользуется популярностью благодаря своей бесплатности, но ее функционал сильно уступает платным аналогам: например, нет собственного конструктора отчетов, а для получения детальной информации об устройствах в сети необходима установка соответствующих агентов.

Как было сказано, существует множество решений инвентаризации сети, у каждого есть свои плюсы, минусы, преимущества и неудобства. Имеет смысл на основе этих данных создать свое собственное решение – небольшую программу, обладающую определенным

набором функций и возможностей, с учетом особенностей предприятия. Такой вариант позволит оперативно реагировать на обратную связь и совершенствовать программу, а также упростит процесс приобретения предприятием решения проблемы инвентаризации сети.

Обзор планируемых возможностей Invoi 2.0

Первая версия программы Invoi позволяет сканировать сеть, получать информацию об аппаратном и программном обеспечении устройств в сети и создавать отчеты [21], то есть Invoi 1.0 обладает минимальными базовыми возможностями, позволяющими использовать его для решения намеченных задач.

Однако у первой версии данного комплекса отмечаются следующие недостатки:

1) поддерживает работу только со встраиваемой СУБД SQLite;

2) в качестве операционной системы поддерживает только семейство ОС Windows;

3) требует использования фонового агента, входящего в состав программного комплекса.

Данные проблемы должна будет решить новая версия программы – Invoi 2.0.

В новой версии Invoi планируется:

1) реализовать работу с различными СУБД;

2) избавиться от необходимости использовать программу-агент на подчиненных устройствах;

3) реализовать кроссплатформенность;

4) создать мобильное приложение для отслеживания состояния сети;

5) обновить пользовательский интерфейс.

При реализации работы с различными СУБД во внимание приняты три самые популярные из них: Microsoft SQL Server, Oracle MySQL и PostgreSQL.

При этом стоит учесть разницу в подключении к СУБД и работе с ними. Возможные подзадачи для выполнения этой задачи следующие: 1) сделать несколько видов строк подключения (является общим); 2) вручную написать запросы для создания базы данных и таблиц; 3) спроектировать классы для генерации различных запросов и обработки их результатов.

При написании запросов на создание таблиц может возникнуть сложность, связанная с разными названиями типов или свойств атрибутов. Так, свойства PRIMARY KEY и AUTO_INCREMENT заменяются на SERIAL в PostgreSQL [22].

Проектирование классов для генерации запросов уже непростая задача. Необходимо сделать гибкие методы, позволяющие конструировать запросы любой сложности, учитывая все операторы и ключевые слова языка SQL. Такие методы быстро станут очень громоздкими, если потребуются сложные запросы для отчетов.

Использование библиотек, реализующих ORM, – более простой и быстрый шаг. Такие библиотеки автоматически создают таблицы, проводят миграции, сохраняют внесенные в записях изменения. Они позволяют работать с таблицами баз данных как с классами-коллекциями, автоматически достраивая внутренние запросы при обращении к новым полям. С таким подходом значительно ускоряется построение форм, потому что тот же язык XAML

поддерживает прямую привязку к сущностям и полям баз данных [23].

Вместо программы-агента можно использовать сетевые протоколы, – например, SMB для ОС Windows, запуская нужные скрипты удаленно.

Протокол SMB – это сетевой протокол прикладного уровня для удаленного доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия. Протокол SMB можно использовать поверх протокола TCP/IP или других сетевых протоколов [24].

Реализация кроссплатформенности – важная задача для новой версии разрабатываемой программы. Кроссплатформенность может быть достигнута различными способами. За счет: создания Docker-контейнера с приложением и всеми необходимыми библиотеками; использования Wine; создания программного продукта с учетом специфики других операционных систем.

Docker-контейнер удобен для быстрого тестирования и публикации решений [25]. Однако данный вариант не подходит для использования в Invoi. Docker представляет собой изолированную виртуальную машину, что не даст приложению получить доступ ни к компьютеру, где оно запускается, ни к сети, к которой он подключен.

Использование Wine – действительно возможный вариант для запуска программы в различных операционных системах. Wine – это свободное программное обеспечение для запуска Windows-приложений на *nix-системах. В первую очередь, он не является эмулятором (Wine – рекурсивная аббревиатура от Wine Is Not an Emulator – «Wine не есть эмулятор»). Это приложение позволяет процессору выполнять все инструкции приложения без прослойки в виде транслятора: большинство инструкций выполняется одинаково независимо от платформы, в противном случае оно подменяет их на правильные системные вызовы [26].

Wine – это программное средство обеспечения кроссплатформенности приложений, без изменения их исполняемого кода. И оно отлично работает для огромного количества программ, в том числе для компьютерных игр – программного обеспечения, требовательного к производительности и взаимодействию с железом [27]. В частности, на данный момент помечено как официально поддерживаемые 5336 приложений из Windows полноценно, еще 4397 – при дополнительных настройках и внешних DLL, у 3943 программ наблюдаются ошибки, которые не мешают работе основных функций приложения [28].

Однако функционирование комплекса Invoi предполагает взаимодействие именно со специальными службами и компонентами операционной системы. Специфика каждой отдельной операционной системы не будет учтена простой подменой.

Третий вариант – написание программы для каждой операционной системы. Это не только дополнительная работа, но также интересный опыт и новые навыки, которые обязательно пригодятся в будущем. Первая проблема – выбор фреймворка для создания пользовательского интерфейса. Для платформы .NET и ОС Linux есть несколько библиотек:

1) Avalonia – кроссплатформенная версия WPF с открытым исходным кодом, 35-й диалект XAML [29];

2) MAUI (Multi-platform App UI) – форк MAUI под Linux, наследник

Xamarin.Forms, 36-й диалект XAML [30];

3) Blazor – подобие Electron, преобразующего программы в браузер с локальным сервером;

4) Gtk# – подобие Windows Forms для Linux [31];

5) Uno – кроссплатформенная версия UWP/WinUI, 37-й диалект XAML, официально поддерживается компанией Microsoft – разработчиком .NET;

6) Qml.NET – проект по адаптации QT в .NET [32].

Данный список может потребовать некоторых объяснений. Под диалектами XAML понимается различное описание одинаковых элементов интерфейса в зависимости от фреймворка.

XAML (eXtensible Application Markup Language) — это язык разметки, который упрощает создание пользовательского интерфейса для приложений .NET. С его помощью можно описать визуальные элементы интерфейса в декларативной XAML-разметке и отделить от логики выполнения, размещенной в коде, связав их с помощью разделяемых классов.

XAML позволяет напрямую создавать экземпляры объектов, используя конкретный набор типов, определенных в сборках. Это отличает его от большинства других языков разметки, которые обычно интерпретируемы и не имеют прямой связи с системой типов. XAML также поддерживает рабочий процесс, позволяющий разным специалистам разрабатывать пользовательский интерфейс и логику приложения, используя различные инструменты. [33].

Из приведенного списка самым привлекательным вариантом является Uno, он до сих пор получает обновления и активно используется в разработке [34]. Сейчас поддерживаются почти все доступные платформы: Windows, Linux, MacOS, iOS, Android и Web. Первая версия Invoi применила WPF для пользовательского интерфейса, но создать его заново будет несложно, – прибегнув к тому же языку, с теми же принципами и парадигмами. Uno является также самым быстрым из всех перечисленных фреймворком.

Стоит рассмотреть и Blazor – интерфейсную веб-платформу .NET, которая поддерживает SSR (server-side rendering – отрисовка на стороне сервера) и взаимодействие клиента в одной модели программирования [35]:

1) это разработка сложных интерактивных пользовательских интерфейсов с использованием C#;

2) объединение серверной и клиентской логики в приложениях, созданных на .NET;

3) отображение пользовательского интерфейса в виде HTML-страницы с CSS для обеспечения совместимости с различными браузерами, включая мобильные устройства;

4) создание гибридных приложений для классических и мобильных платформ с использованием .NET и Blazor.

Этот фреймворк позволяет размещать приложение на локальном сервере и запускать его как из бинарного файла, так и через браузер, обеспечивая кроссплатформенность интерфейса. Библиотека разрабатывается и поддерживается корпорацией Microsoft.

Для создания мобильного приложения можно рассмотреть два варианта: использовать мобильное приложение из проекта на Uno либо разработать самостоятельно, при помощи

одного из множества фреймворков. Его основными функциями должны быть принятие сообщений от основного приложения и показ их в виде уведомлений или отображение краткой сводки по результатам инвентаризации.

Важной задачей является обновление пользовательского интерфейса. Можно обновить его с учетом новых функций, сохраняя при этом парадигмы. Макет пользовательского интерфейса показан на рис. 1.

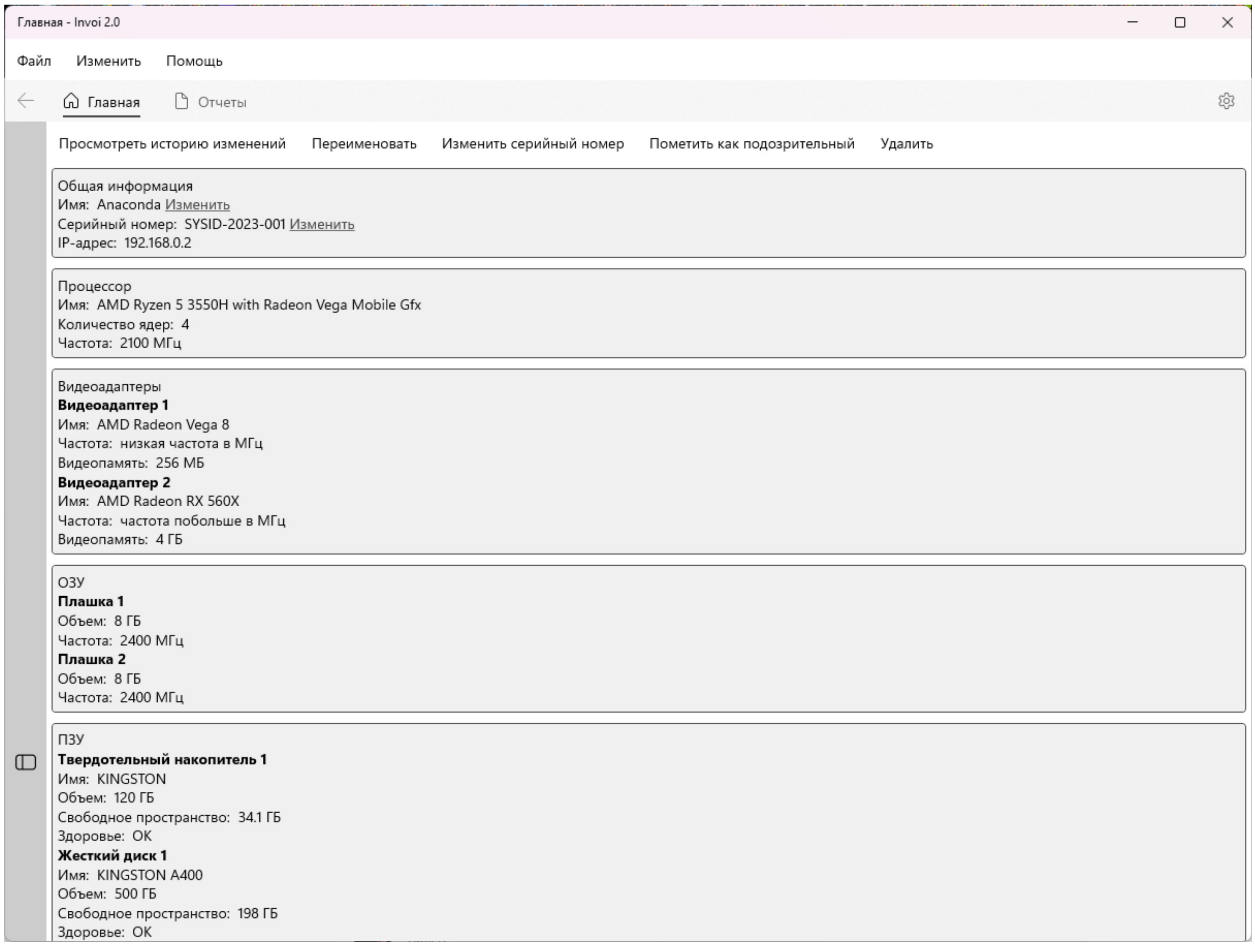

Рис. 1. Макет нового пользовательского интерфейса Invoi 2.0.

Как видно из рисунка, добавилась скрываемая боковая панель, элементы навигации стали горизонтальными и у каждого компьютера есть набор дополнительных действий: просмотреть историю изменений, переименовать (можно при помощи кнопки или нажатием на гиперссылку рядом со значением изменяемого поля), изменить серийный номер. Конечно, данный пользовательский интерфейс не финальный, обязательно будут еще некоторые косметические изменения, но уже сейчас он приблизился к более современному, с использованием лучших практик UI/UX.

Заключение

Автоматизированная инвентаризация – это решение с целью контроля состояния компьютерной сети предприятия. Но у каждого предприятия свой масштаб компьютерной сети и объем денежных средств, которые оно может потратить на содержание и поддержание работоспособности ИТ-инфраструктуры.

В предложенной статье рассмотрены важные особенности различных решений, которые могут быть применены в разработке новой версии программного комплекса Invoi 2.0, спроектированной с учетом недостатков прошлой версии, сохраняя низкие системные требования, оставаясь доступной для малых и средних предприятий. Это главное требование к данному программному пакету. Использование Invoi в компьютерных сетях повысит эффективность работы ИТ-персонала, сократит время инвентаризации и облегчит процессы учета, планирования, управления и изменения ИТ-инфраструктуры.

1. СерчИнформ SIEM [Электронный ресурс] // SearchInform. URL: https://searchinform.ru/ products/siem/ (дата обращения: 19.03.2024).

2. MaxPatrol SIEM [Электронный ресурс] // Positive Technologies. URL: https://www.ptsecurity.com/ru-ru/products/mpsiem/ (дата обращения: 19.03.2024).

3. KOMRAD Enterprise SIEM [Электронный ресурс] // Эшелон. URL: https://etecs.ru/komrad (дата обращения: 19.03.2024).

4. SECURITY CAPSULE SIEM [Электронный ресурс] // Инновационные технологии в бизнесе. URL: https://www.itb.spb.ru/products/Security_Capsule_SIEM/ (дата обращения: 19.03.2024).

5. RuSIEM [Электронный ресурс] // RuSIEM. URL: https://rusiem.com/ (дата обращения: 19.03.2024).

6. Обзор RuSIEM российской SIEM-системы [Электронный ресурс] // Издание Anti-Malware. URL: https://www.anti-malware.ru/reviews/RuSIEM (дата обращения: 19.03.2024).

7. Kaspersky Unified Monitoring and Analysis Platform [Электронный ресурс] // Крайон. URL: https://www.krayon.ru/kaspersky/kuma (дата обращения: 19.03.2024).

8. Обзор доменных служб Active Directory [Электронный ресурс] // Microsoft.Документация. URL: https://learn.microsoft.com/ru-ru/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview (дата обращения: 19.03.2024).

9. Настраиваем и автоматизируем развертывание Active Directory [Электронный ресурс] // Хабр. URL: https://habr.com/ru/companies/testo_lang/articles/525326/ (дата обращения: 19.03.2024).

10. Полное руководство по Active Directory, от установки и настройки до аудита безопасности [Электронный ресурс] // HackWare. URL: https://hackware.ru/?p=16316 (дата обращения: 19.03.2024).

11. РЕД АДМ – система централизованного управления ИТ-инфраструктурой [Электронный ресурс] // РЕД СОФТ. URL: https://redos.red-soft.ru/product/redadm/ (дата обращения: 19.03.2024).

12. Обзор РЕД АДМ и Атом.Домен: новые альтернативы службе каталогов MS Active Directory [Электронный ресурс] // Хабр. URL: https://habr.com/ru/companies/k2tech/articles/761898/ (дата обращения: 19.03.2024).

13. About Samba [Электронный ресурс] // Samba. URL: https://www.samba.org/samba/ (дата обращения: 19.03.2024).

14. Features/Samba4 [Электронный ресурс] // Red Hat. URL: https://fedoraproject.org/wiki/ Features/Samba4 (дата обращения: 19.03.2024).

15. Возможности и ограничения Samba4 как контроллера домена Active Directory [Электронный ресурс] // Хабр. URL: https://habr.com/ru/articles/272777/ (дата обращения: 19.03.2024).

16. The OpenLDAP Project Overview [Электронный ресурс] // OpenLDAP. URL: https://www.openldap.org/project/ (дата обращения: 19.03.2024).

17. Введение в службы каталогов OpenLDAP [Электронный ресурс] // OpenNET. URL: https://opennet.ru/docs/RUS/openldap_admin/intro.html (дата обращения: 19.03.2024).

18. Справочное руководство Nmap [Электронный ресурс]. URL: https://nmap.org/man/ru/ (дата обращения: 07.03.2024).

19. SaltwaterTaffy, an nmap wrapper library for .NET [Электронный ресурс]. URL: https://github.com/thomdixon/SaltwaterTaffy/ (дата обращения: 07.03.2024).

20. Inventory your network devices [Электронный ресурс] // Spiceworks. URL: https://www.spiceworks.com/freepc-network-inventory-software/ (дата обращения: 20.02.2024).

21. Программные комплексы автоматической инвентаризации ИТ-инфраструктуры // Амурский государственный университет. URL: https://vestnik.amursu.ru/archive/2023/101/programmnye-kompleksy-avtomaticheskoj-inventarizatsii-it-infrastruktury/ (дата обращения: 07.03.2024).

22. PostgreSQL Documentation [Электронный ресурс] // Postgres. URL: https://www.postgresql.org/docs/ (дата обращения: 21.02.2024).

23. Общие сведения о совместном использовании файлов с помощью протокола SMB 3 в Windows Server [Электронный ресурс] // Microsoft.Документация. URL: https://learn.microsoft.com/ru-ru/windows-server/storage/file-server/file-server-smb-overview (дата обращения: 18.02.2024).

24. Общие сведения о привязке данных (WPF .NET) [Электронный ресурс] // Microsoft. Документация. URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/data/?view=netdesktop-8.0 (дата обращения: 25.02.2024).

25. Понимая Docker [Электронный ресурс] // Хабр. URL: https://habr.com/ru/articles/253877/ (дата обращения: 21.02.2024).

26. Стабильный релиз Wine 9.0? Спустя год он все-таки появился — вместе с 7 000 изменений [Электронный ресурс] // Хабр. URL: https://habr.com/ru/companies/ru_mts/articles/787720/ (дата обращения: 22.02.2024).

27. What is Wine? [Электронный ресурс] // WineHQ. URL: https://www.winehq.org/ (дата обращения: 22.02.2024).

28. Стабильный релиз Wine 9.0 [Электронный ресурс] // OpenNET. URL: https://www.opennet.ru/opennews/art.shtml?num=60446 (дата обращения: 22.02.2024).

29. Build with Avalonia [Электронный ресурс] // Avalonia. URL: https://docs.avaloniaui.net/ru/ (дата обращения: 25.02.2024).

30. NET MAUI on Linux with Visual Studio Code [Электронный ресурс] // Microsoft.Документация. URL: https://techcommunity.microsoft.com/t5/educator-developer-blog/net-maui-on-linux-with-visual-studio-code/ ba-p/3982195 (дата обращения: 25.02.2024).

31. GtkSharp [Электронный ресурс] // Mono. URL: https://www.mono-project.com/docs/gui/gtksharp/ (дата обращения: 25.02.2024).

32. A Qt/Qml integration with .NET [Электронный ресурс] // GitHub. URL: https://github.com/qmlnet/qmlnet/blob/develop/README.md (дата обращения: 25.02.2024).

33. Обзор XAML (WPF.NET) [Электронный ресурс] // Microsoft.Документация. URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0 (дата обращения: 25.02.2024).

34. Как ориентироваться на несколько платформ с помощью приложения WinUI 3 [Электронный ресурс] // Microsoft. URL: https://learn.microsoft.com/ru-ru/windows/apps/how-tos/uno-multiplatform (дата обращения: 25.02.2024).

35. ASP.NET Core Blazor [Электронный ресурс] // Microsoft.Документация. URL: https://learn.microsoft.com/ru-ru/aspnet/core/blazor/?view=aspnetcore-8.0 (дата обращения: 25.02.2024).