2. При разработке современных программных комплексов зачастую возникают задачи, связанные с поддержкой расширения и автоматизации программного обеспечения. Это необходимо для реализации возможности конфигурирования, настройки, исправления ошибок и устранения уязвимостей, переопределения поведения приложения конечным пользователем, а также для расширения функционала приложения и добавления каких-либо новых возможностей. Перечисленные выше особенности присущи большинству современного программного обеспечения, что доказывает актуальность этого вопроса. Меня данные вопросы заинтересовали при работе над реальным проектом, в рамках которого было необходимо портировать приложение, разработанное с использованием ныне устаревших технологиях, на современные языки и платформы. В «старом» приложении для автоматизации и расширения использовалась технология VBA, которая довольно гармонично вписалась в архитектуру и успешно и эффективно решала поставленные задачи. Более подробно о данном проекте написано в пояснительной записке.

3. Чаще всего реализация данных возможностей достигается с помощью одного из следующих способов:

* Автоматическое обновление ПО
* Поддержка расширений сторонних производителей
* Наличие SDK для разработки плагинов
* Поддержка скриптов

С точки зрения числа предоставляемых возможностей и гибкости наибольший интерес представляет последний вариант. В зависимости от специфики разрабатываемого ПО могут быть различные сценарии использования скриптов: простейшее конфигурирование приложения конечным пользователем, разработка дополнений, расширяющих возможности приложения, распространение скриптов, разработанных сторонними производителями. Именно поэтому в настоящее время многие разработчики предоставляют возможность использовать скрипты в своих приложениях. Более того, на этой идее построены некоторые платформы, претендующие на звание универсальных фреймворков для автоматизации и расширения приложений.

4. В процессе исследования были рассмотрены некоторые существующие решений, позволяющие интегрировать в разрабатываемое приложение возможность автоматизации и расширения. В работе подробно описан каждый продукт и сделаны выводы о его плюсах и минусах. На слайде вы видите некоторые из рассмотренных продуктов. В данном случае особняком стоят два решения: VBA и VSTA. При детальном анализе стало понятно, что VBA было действительно довольно универсальным решением, обладающим большим числом конкурентных преимуществ. В свою очередь VSTA была заявлена компанией Microsoft как замена VBA, и именно эта разработка была первоначально интегрирована на реальном проекте (правда, в процессе интеграции были выявлены многие недостатки, существенно затрудняющие использование VSTA и не позволяющие назвать данное решение универсальным).

5. Рассмотрим основные недостатки существующих решений:

* Решают частные случаи задачи
* Сложны в интеграции
* Имеют неоправданно большой размер и нагромождение невостребованных возможностей
* Сводятся к практически полностью ручной реализации требуемых функций

Помимо этого каждая разработка обладает своими специфичными особенностями и недостатками, однако перечисленные выше проблемы присущи большинству продуктов.

6. Проанализировав проблемы, не позволяющие использовать в рамках коммерческого проекта текущие разработки, а также интегрировав в тестовой версии продукта одну из разработок (VSTA – опыт интеграции и возникшие проблемы с подробным анализом причин подробно описаны в работе), мы решили реализовать собственную платформу. Также была поставлена цель использовать платформу на нескольких проектах (коммерческом в компании FirstLineSOftware и каким-либо Open-Source проекте). Полученные результаты необходимо было сравнить с существующими разработками.

7. На слайде представлены основные требования к платформе:

* Автоматизация и расширение как единое понятие
* Простота интеграции
* Полуавтоматическая интеграция
* Возможность отладки расширений
* Инструменты для взаимодействия с основным приложением
* Возможность повторного использования существующего кода

8. Архитектура платформы разрабатывалась с учётом опыта интеграции VSTA на реальном проекте. На слайде изображены основные компоненты платформы и способы их взаимодействия. Важной составной частью является интегрированная среда разработки, в качестве которой была выбрана среда с открытым исходным кодом SharpDevelop. Среда разработки, а также модуль интеграции, отвечающий за взаимодействие с основным приложением, работает в отдельном процессе, взаимодействуя с процессом основного приложения с помощью Windows Communication Foundation. Основное приложение взаимодействует с расширениями с помощью .NET Reflection. Более подробно детали реализации описаны в пояснительной записке.

9. Разработанная платформа отвечает сформулированным требованиям, однако помимо этого предоставляет ряд дополнительных возможностей:

* Управление расширениями
* IntelliSense с учётом специфики основного приложения
* Инструменты для полуавтоматической интеграции
* Полноценная среда разработки

10.Пришло время обсудить результаты работы Итак,

* Изучены вопросы, связанные с автоматизацией и расширением приложений
* Изучены существующие подходы в данной области
* Предпринята попытка интеграции одного из существующих решений
* Сделаны выводы о преимуществах и недостатках существующих решений и сформулированы требования к новой разрабатываемой платформе
* Разработана новая платформа
* Платформа интегрирована в существующие проекты
* Проведён анализ результатов интеграции и сравнение полученной платформы с существующими решениями

Ещё раз отмечу, что разработанная платформа была успешно интегрирована в реальный коммерческий проект, и результаты работы были высоко оценены как в компании First Line Software, так и заказчиком проекта.