1. Постановка задачи
   1. VBA
   2. Терминология
2. Основные требования к требуемому программному продукту
3. Возможность создания расширений для конечных пользователей.
4. Простота использования.  
   Не требует SDK и другого ПО для создания расширений. Вся работа происходит в среде разработки (IDE).
5. Возможность работы как под x86, так и под x64 архитектурами.
6. Возможность отладки расширения.
7. Доступ расширения к объектам расширяемого приложения, реакция на его события.  
   При изменении удалении или добавлении объекта в хост приложении, расширение должно автоматически обновлять информацию о своем окружении.
8. Доступ к библиотеке классов .NET Framework.
9. Поддержка большого числа расширений и взаимодействие их друг с другом.
10. Возможность разрешения зависимостей между расширениями.
11. Удобные инструменты для написания кода (Аналогично IntelliSense).
12. Возможность сохранения расширений по усмотрению пользователя в базу данных, архив, папку и т. д.
13. Динамическая загрузка и выгрузка расширений из адресного пространства хост приложения.
14. Компиляция и перезагрузка расширения «на лету» (Не требуется перезапуск хост приложения)
15. …
16. Обзор средств для построения расширяемых приложений на платформе .NET
    1. Плагины

[http://en.wikipedia.org/wiki/Plug-in\_(computing)#Purpose\_and\_examples](http://en.wikipedia.org/wiki/Plug-in_(computing)" \l "Purpose_and_examples)

* + 1. Существующие разработки для .NET

Крупные и средние проекты

1. AL Platform (<http://www.agile-sys.com/productsA.php>)
2. MEF (<http://mef.codeplex.com/>)
3. Plux.NET (<http://ase.jku.at/plux/index.html>)
4. Mono.Addins (<http://monoaddins.codeplex.com/>)
5. Compact Plugs & Compact Injection (<http://compactplugs.codeplex.com/>)

Маленькие, учебные проекты, тестовые примеры

1. .NET Based Add-in/Plug-in Framework with Dynamic Toolbars and Menus (<http://www.codeproject.com/KB/macros/Net_AddinProjFrmwork.aspx>)
2. AddIn Enabled Applications (<http://www.codeproject.com/KB/dotnet/AddInModel.aspx>)

Отдельно стоит упомянуть

1. SDA (<http://laputa.sharpdevelop.net/AnnouncingSharpDevelopForApplicationsSDA.aspx>)  
   SDA (SharpDevelop for applications), Набор инструментов, позволяющий построить собственное приложение на основе ядра SharpDevelop. Это приложение может получить «в наследство» многие возможности самого SharpDevelop, в том числе и возможность написания плагинов. (SharpDevelop Addins)

Все вышеперечисленные инструменты, платформы и библиотеки не решают поставленную задачу, а именно предоставление создания расширений *конечному пользователю* приложения. Их основная цель – дать разработчику ПО возможность сделать свое приложение более гибким, но в то же время не перегруженным функционалом, который для многих пользователей окажется попросту не нужным. Яркий пример подобных приложений – интернет браузеры Mozilla Firefox и Google Chrome. Изначально довольно аскетичные, они могут при помощи плагинов получать функционал, удовлетворяющий даже самого требовательного пользователя.

* 1. Разработки, позволяющие встраивать в приложение возможность расширения базового функционала конечными пользователями

1. VSTA ([http://www.microsoft.com/download/en/details.aspx?id=5804#overview](http://www.microsoft.com/download/en/details.aspx?id=5804" \l "overview))  
   Visual Studio Tools for Applications – мощный набор инструментов, пришедший на замену VBA и имеющий схожий набор возможностей.
2. VSTO ([http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24263#overview](http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24263" \l "overview))  
   То же, что и VSTA, используется как замена VBA в поздних версиях Microsoft Office.

Рассмотрим эти решения подробнее.

* + 1. VSTO

Внутренняя разработка компании Microsoft. Ее использование в сторонних приложениях невозможно. Для этих целей существует второй проект – VSTA. Подробно рассматривать возможности VSTO в рамках этой работы не имеет смысла.

* + 1. VSTA

Поначалу именно VSTA была выбрана, как замена VBA.

Была разработана методика внедрения инструментов VSTA в существующее приложение. В рамках работы были решены проблемы взаимодействия VSTA IDE и хост приложения, такие как программное управление, реакция на события, реализация отладки расширения. Так же была решены проблема использования механизмов VSTA без изменения объектной модели и архитектуры хост приложения.

В конце концов, Microsoft отказалась лицензировать VSTA, и закрыло этот проект для сторонних фирм-разработчиков ПО.

* 1. Технологии, библиотеки и инструменты, позволяющие реализовать расширяемое приложение

1. System.Addin
2. .NET Reflection / Remoting
3. Mono Cecil (?)
4. Iron Python (?)