# Обоснование выбора инструментов и платформы разработки

Так как росреестр использует для своей работы сервисы ArcGIS, при создании данной системы так же используются библиотеки этого сервиса. Для работы с картами разработчику .Net предоставляются наборы средств разработки Esri.ArcGISRuntime и Esri.ArcGISRuntime.WPF. Одним из явных преимуществ является наличие подробного описания API для работы с данными библиотеками. Детальное описание Esri.ArcGISRuntime и наличие примеров использования ее компонентов позволяет быстрее освоить и использовать данные механизмы в собственной разработке. Для работы с пользовательским интерфейсом ArcGIS предлагает собственные элементы управления Windows Presentation Foundation в библиотеке Esri.ArcGISRuntime.WPF, облегчающие взаимодействие пользователя с системой. Перечисленные выше факторы, наличие опыта использования и личные предпочтения сыграли наибольшую роль при выборе среды разработки и языка программирования.

Для реализации системы предпроектной проработки размещения объекта капитального строительства была выбрана среда разработки Microsoft Visual Studio.

Microsoft Visual Studio имеет следующие особенности:

* является интуитивно понятной, расширяемой, унифицированной средой для языков программирования высокого уровня;
* предоставляет разработчикам набор модернизированных взаимодействующих языков .NET Framework, соответствующих имеющимся у разработчиков навыкам;
* реализует высокопроизводительные инструментальные средства для всех этапов жизненного цикла разработки программного обеспечения (ПО) – от определения требований и планирования до его последующего сопровождения.

Microsoft Visual Studio разрабатывался таким образом, чтобы обеспечить высокую надежность и совместимость создаваемых с его использованием приложений.

Разработчики, применяющие Microsoft Visual Studio, получают в свое распоряжение интегрированный продукт, включающий различные средства, необходимые для процесса разработки, тестирования, развертывания, интеграции и управления ПО. Эти средства работать с большей отдачей и затрачивать меньше усилий на решение повторяющихся задач.

В качестве языка программирования для написания приложения был выбран язык C#.

Для клиентского приложения была выбрана система Windows Presentation Foundation (WPF). WPF предоставляет средства для создания визуального интерфейса, включая язык XAML, элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. XAML представляет собой язык декларативного описания интерфейса, основанный на XML. Также реализована модель разделения кода и дизайна, позволяющая кооперироваться программисту и дизайнеру. Кроме того, есть встроенная поддержка стилей элементов, а сами элементы легко разделить на элементы управления второго уровня, которые, в свою очередь, разделяются до уровня векторных фигур и свойств/действий. Это позволяет легко задать стиль для любого элемента.

При проектировании архитектуры приложения использовался шаблон проектирования Model-View-ViewModel (MVVM). Он используется для разделения модели и её представления, что необходимо для изменения их отдельно друг от друга. Шаблон MVVM имеет три основных компонента:

* Модель (англ. Model), представляет собой бизнес логику и фундаментальные данные, необходимые для работы приложения;
* Представление (англ. View) — это графический интерфейс, то есть окно, кнопки и т. п. Представление является подписчиком на событие изменения значений свойств или команд, предоставляемых Моделью- Представления. В случае, если в Модели-Представления изменилось какое-либо свойство, то она оповещает всех подписчиков об этом, и Представление, в свою очередь, запрашивает обновленное значение свойства из Модели-Представления. В случае, если пользователь воздействует на какой-либо элемент интерфейса, Представление вызывает соответствующую команду, предоставленную Моделью- Представления;
  + - Модель-Представления (англ. ViewModel) является, с одной стороны, абстракцией Представления, а с другой, предоставляет обёртку данных из Модели, которые подлежат связыванию. То есть, она содержит Модель, которая преобразована к Представлению, а также содержит в себе команды, которыми может пользоваться Представление, чтобы влиять на Модель.