Начнем с того, что в своей диссертации я решаю в первую очередь производственную задачу.

Практически каждый наш объект не обходится без подпорных стен, которые являются типовым и рутинным элементом строительства. Этот элемент может претерпеть изменения на любом этапе производства работ от согласований и вплоть до начала строительства. Каждое такое изменение влечет за собой пересмотр большого количества файлов и документов.

**Основной целью** диссертации является автоматизация процесса выпуска документов. Она включает в себя систематизацию исходных данных, создание ведомости объемов работ, опалубочных чертежей и BIM-модели. Самая главная задача - это исключение несоответствий между этими разделами из различных программных комплексов.

Для решения данной проблемы было рассмотрено 3 варианта.

- макросы на VBA

- проект в среде визуального программирования Grasshopper

- программа на Python

В конечном итоге я остановилась на написании приложения на Python с использованием различных библиотек.

На данный момент всё выполняется **вручную**, у каждого проектировщика есть **свои наработки**. Разумеется, с ними бывает трудоемко разобраться человеку, присоединившемуся к проекту в процессе.

Альтернативным решением является написание **макросов на VBA**. К преимуществам можно отнести то, что VBA уже встроен в Excel. Недостатком – отсутствие развития и устаревание языка.

Другим решением является создание алгоритма в среде визуального программирования **Grasshopper**. Существенным преимуществом такого метода является наглядность и интерактивность среды, низкий порог вхождения в среду. Недостаток – невозможность реализации сложной логики, низкая скорость работы. Grasshopper может пригодиться на этапе разработки концепции приложения, в нем можно сделать быстрый набросок алгоритма, а потом уже реализовать его в виде кода.

Ну и третий вариант автоматизации – это написание приложения на **Python**. Это активно развивающийся язык программирования с большим сообществом. С его помощью можно создавать проекты с более высокой производительностью и масштабируемостью. Есть поддержка системы версий и различные библиотеки для интеграции с другими системами. Да, порог входа в проект на Python выше, разработка дольше, но и возможности его гораздо шире.

Ввиду указанных преимуществ было принято решение разрабатывать приложение на Python. Однако на первом этапе для него необходимо **подготовить исходные данные**, которые будут использоваться и для создания ВОР, и эти же данные будут входными для отрисовки чертежей и создания БИМ-модели.

Как было отмечено в начале, **данные** по параметрам подпорных стен **могут неоднократно меняться** в процессе работы. Использование программы осуществляется на том этапе проектирования, когда расчет подпорных стен уже произведен и в данном продукте никак не затрагивается.

Для того, чтобы быстро и качественно сделать документацию по изменяемым параметрам подпорных стенок пришлось провести **систематизацию исходных данных**. Исходные данные заносятся в файл Data, на его основе заполняется файл с армированием и ведомость объемов работ.

В файле Data зеленым отмечены ячейки, которые необходимо заполнить пользователю. Задается геометрия секции стенки, ее положение в пространстве и дополнительные работы, требующие внесения в ВОР. Эти данные отправляются в файл с расчетом массы армирования, выделенный в самостоятельный блок ввиду своей объемности. Результатом его работы является масса всех арматурных стержней. Сведения о массе отправляются обратно в файл исходных данных для того, чтобы их дополнить. После это становится возможно сформировать **ведомость объемов работ в виде самостоятельного документа**, файл которого хранится отдельно и не требует никакого вмешательства пользователя.

Разобравшись с исходными данными и ведомостью объемов работ переходим к следующему этапу – **созданию опалубочного чертежа** каждой секции сооружения в AutoCAD. Что входит в состав опалубочного чертежа в данной ситуации?

(4 вида, размеры, отметки высот, спецификация и подписи)

Все указанное формируется моей программой **автоматически** на основании исходных данных из файла Data. Вот такие **модели секций** создаются в пространстве чертежа, все данные указаны в предназначенных для них слоях, а сам чертеж концептуально построен так, чтобы был легко **совместим** с другими проектами моей организации.

Программа состоит из 5 модулей.

Первый модуль – исходные данные в excel файле

Головной модуль - осуществляет считывание данных, координацию работы модулей между собой и взаимодействие с пользователем.

Вспомогательный модуль с классами, которые хранят в себе информацию по подпорным стенам и методы, доступные при работе с ними.

Модуль взаимодействия с AutoCAD.

У него есть служебный модуль, так как для построения чертежа помимо исходных данных в чистом виде требуются и вспомогательные данные, например, координаты точки для простановки названия вида).

Модуль взаимодействия с Tekla Struktures – подключается к API программы с помощью DLL файлов.

На слайде представлен интерфейс взаимодействия с программой. Она предоставляет выбор среды разработки и запрашивает точку вставки в пространстве модели AutoCAD. Таким образом, вставка чертежа производится в указанном пользователем месте, но с выставлением на правильной отметке в Балтийской системе высот согласно исходных данных.

В настоящее время в строительной отрасли взят курс на внедрение BIM-технологий. В частности, требование о наличии BIM-моделей распространяется все объекты Москвы. В целом наблюдается тенденция к структурному переходу от проектирования в AutoCAD к BIM-проектированию. Преимуществом нового подхода является комплексное проектирование всего объекта целиком, централизованная согласованная информационная модель. Частью такой модели готова стать и замоделированная моей программой подпорная стенка.

Подводя итог проделанной работы следует отметить, что цели автоматизации процесса выпуска документов были достигнуты – их формирование осуществляется за 1 клик. А выбранный способ реализации позволяет дальнейшее развитие данной программы и расширение функционала.