KPI PROCESS IMPROVEMENT

Interim Report

Anders Ingelsten 20095402

Higher Diploma in Science in Computer Science

Contents

Summary

Last Text to add

Introduction

Project Type

This is a work-based project, where I will be learning new skills in relation to developing a KPI dashboard using the knowledge acquired from the course modules. The internal company mentor will be the Chief Financial Officer. This means that I will be able to spend time on this as part of my day-to-day work.

The organisation

Company A is an engineering solutions provider operating in Ireland, the UK and Scandinavia. It provides a wide range of services from the Design & Build of Sub-stations to construction of Airside Aviation Infrastructure to Turn-key Wind & Solar Energy Solutions. The company has a turnover of approx. 30million euro and employees approx. one hundred staff.

System Background and Project Scope

Company A has several systems that hold business information. Finance, Health and Safety and Operations staff query this data on a regular basis to produce business performance reports and to generate KPI (Key Performance Indicators) metrics for the different departments. The reports are generated on a weekly or monthly basis but there are also ad hoc reports.
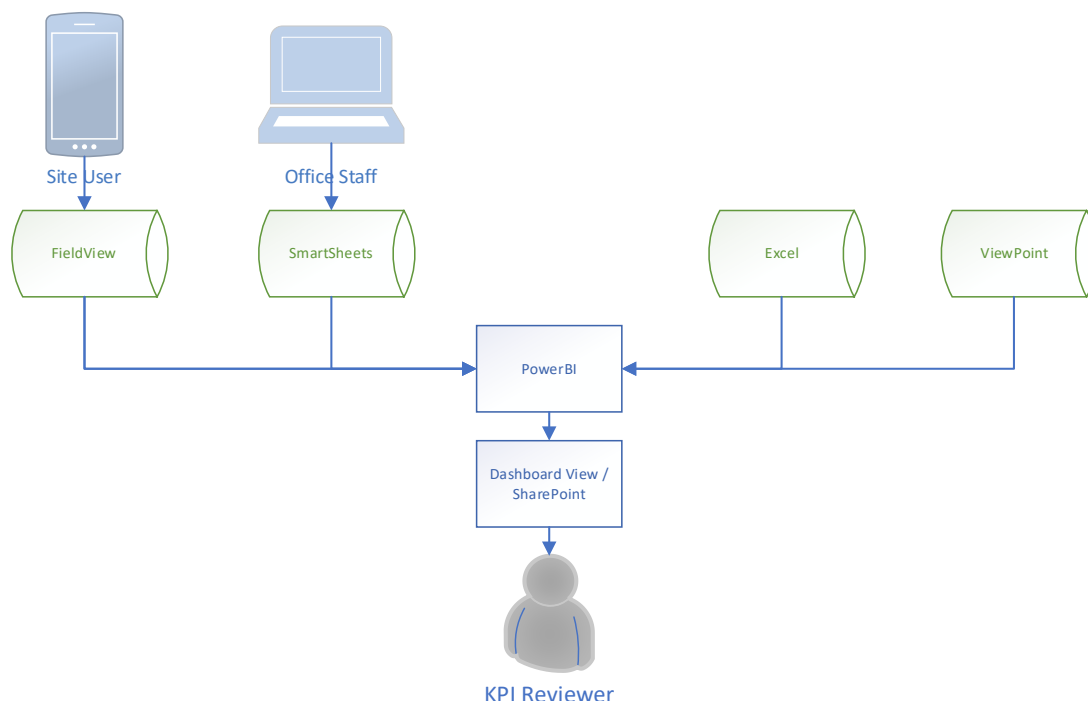

Figure 1: Overview of existing potential systems available to the project

The current systems identified for the initial project scope are:

System	Function	Location
Fieldview	Record QA/QC/Admin forms and Timesheet for site staff	Cloud
Smartsheet	Record timesheets for Salaried Staff	Cloud
Office365	SharePoint, Excel, word etc	MS Cloud

Fieldview

Fieldview is a third-party cloud-based and off-line mobile solution developed by Trimble (Floor et al., n.d.). It is used in Company A to replace paperwork on site. Users are equipped with a mobile device (phone and tablet), where the app has been installed. The users log in and use the application for snagging tasks, forms & permits, project delivery and handover. When the mobile device is synced – data is pushed to the cloud hosted database.

Figure 2: Sample view of Fieldview mobile form
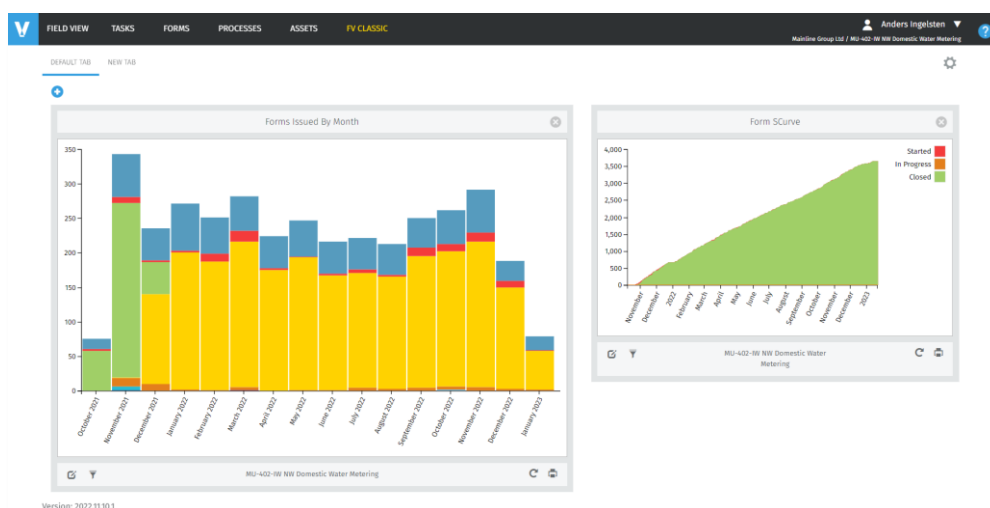
Figure 3: Sample view of the Fieldview widget reports

Smartsheet

Smartsheet is an online hosted solution that allows organisations to plan, track, automate, and report on work (Smartsheet, 2019). Company A uses the online application to track timesheets for salaried staff.

Figure 4: Sample view of Smartsheet form

Sharepoint/Office365

During the spring of 2022, the organisation has fully migrated its whole IT environment to the Sharepoint Online cloud, i.e., the full file repository of the organisation approx. 2.5 TB of files is hosted online with 24/7 365 access.

SharePoint Online is a cloud-based SAAS - "software as a service" provided by Microsoft, where organisation and users store and share information and use it collaboratively (www.microsoft.com, n.d.). All data and software are hosted on Microsoft own servers. Access, storage, and use of software is done by using a subscription model.

Current Situation

The company has over the summer of 2022, revised the KPI process and in September rolled out a new KPI process where departmental stakeholders fill in excel spreadsheet reports with various KPI data. For example, hours spent by employee per contract and other data like work site accident

frequency etc. The departmental stakeholders access several systems and from there pull-down pdf and excel reports, from where they extract data and then collate it into excel spreadsheets which is then presented to the leadership team. This process is repeated monthly.

The company is aware that this is a time-consuming process and when it's well established is looking to improve the efficiencies in this process. The company has already identified that business intelligence tools like Power BI can be used to display almost real-time dashboard views of the business data. During 2022 there has been internal talks about identifying and bringing in business intelligence consultants/developers to do this work during 2023. Within the project, I will investigate business intelligence products for example the Microsoft Power BI platform. It's worth noting that the company has fully migrated to Office365/SharePoint during 2022 and is using the SharePoint platform as Intranet and repository for company data.

What are the problems.

The two main problems are:

1. Manual input of data by staff into excel spreadsheets is time consuming and this time can be better spent on other business processes.

2. The time difference between the live situation and the compiled report by staff leads to delays in business understanding, for example if work is profitable or not.

The CFO of Company A is therefore looking for a solution that can display KPI information from systems on the internal company SharePoint internet website.

Use Case

The primary reason for developing a dashboard pulling data from sources is the time saving. For example, the Fieldview application is structed in such a way that when a user, for example a project manager, would like to access the online portal, and view and pull-down information it can ONLY be done on a project-by-project basis. The systems do not have functionality in the user interface for a user to pull down for example all vehicle checks across all projects.

This means a user must access one project, access the report view, pull the information down and then access the next project and repeat the process until completed. Currently a user can create widgets that displays summaries but again, this must be done for every user on an individual basis.

A simple calculation will show how a dashboard can provide efficiencies.

Example, a user spends 1.5 hour every week to access every project area and compile a report of the amount of vehicles report submitted by staff. 1.5 hours x 48 working weeks = 72Hours which equates to 9 working days per year.

Potential Technologies, Tools, and Languages

As this is a work-based project the preferred direction from the leadership team is to use known technologies that are currently in use by the company. The two main reasons for this are that it creates greater resilience, and any costs are known, for example the company already host a Windows 2019 server in the Azure cloud. List of potential technologies, tools and languages:

- Office365
- SharePoint
- Power BI Desktop
- Power BI Data Gateway
- Power Query Editor
- SOAP UI
- MS Planner
- Html
- PowerShell
- Sharepoint Online Management Shell
- M language and DAX

Potential Issues

As the project will have access to commercially sensitive and potentially personal data, measures will be put in place to minimise exposure of this data. This may limit the scope of which systems and data will be incorporated in the project. If it's not feasible to limit the exposure, test data will be utilised.

Objectives

This project has three main objectives:

- **Ease of access.** By connecting separate data sets, transforming, and cleaning the data into a data model and creating charts or graphs to provide visuals of the data; it will assist users to find insights, within the organisation, of the operational data generated in Fieldview.

- **Real-time information.** To give users in the organisation, an updated almost real-time view of the situation in the company. This should provide the ability to solve problems and identify issues and opportunities.

- **Process improvement.** By streamlining publication and the distribution of the data into dashboards – users interpret published data whenever the underlying dataset is updated. This is instead of compiling reports through a time-consuming process and sharing the data in emails or a shared drive for stakeholders to then review.

Methodology

Dataflow

The data is envisaged to flow in the following way.

1. Users generate data onsite in the Fieldview app.
2. By syncing the device – the data generated in the app is pushed to the Fieldview server environment (Cloud)
3. The PowerShell scripts will then run automated on a server hosted in the Company A Azure environment, pull data from the Fieldview cloud.
4. Data is then saved in CSV files by the PowerShell scripts to Sharepoint Online.
5. PowerBI is then connected to the CSV files and refreshed on a regular basis.

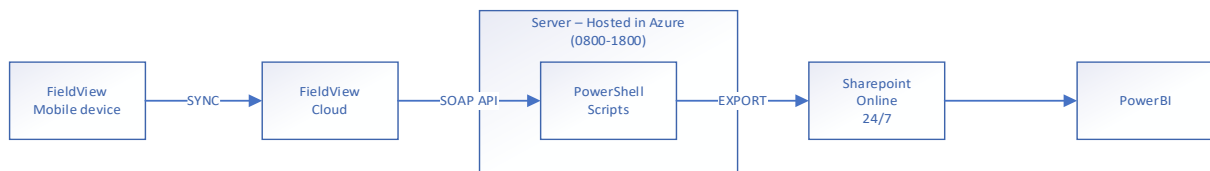*Figure 2: The Flow of Data from Fieldview to PowerBI*

Modelling

Below is a simple representation of modelling of the data in Power BI, it is envisaged that data is presented in several tables. These tables are then joined pending any queries they may relate to. These tables and queries will then be the basis for the visualisation.
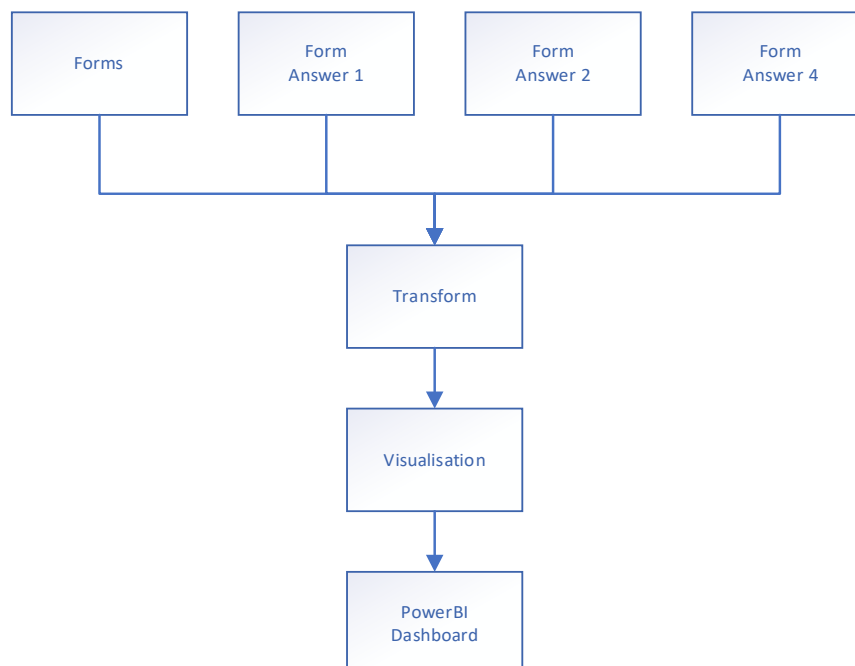

Figure 3:Sample Representation of the data model in PowerBI

Feedback Loop

The senior leadership team and the supervisor are very keen on the KPI Process improvement, its outputs, results of visualisation the data and the automation of the process.

The company is in a rapid growth spurt as several contracts was won during 2022/2023 and a current ramp up of activities are in progress. The direction was to use a feedback loop during the phased development process. Envisaged steps taken during the feedback loop:

1. Initial requirement meeting with suggestion by the supervisor of the visualisation of data.

2. Analyse, develop and display data as per initial requirement.

3. Follow up meeting to determine if the result has fit the expected requirements.

4. Publish any changes or improvements of any feedback received.

Repeat step 3 and 4 until decision to move to next visualisation feature. As the scope of the project and allocated time is controlled by the organisation supervisor's priorities; it's important to note that I will have to adhere to my supervisor's direction, which is fine and something to be expected when you work in SME environment where priorities and work allocation change on a day-to-day basis.

Technologies

The technologies used in the project was dictated by several limitation set by several factors. For example, the type of API used by Fieldview, scripting language/framework that can very efficiently and quickly generate data to PowerBI and any other technologies linked, researched or discovered during the project development process. The key drivers for deciding the scripting language were the Fieldview API, SharePoint Online and how quickly and cost effective the solution could be put into productivity. There was no direction from the internal supervisor in relation to choose made of which technology is used.

PowerBI

PowerBI will be the tool to display dashboards of KPI views to the stakeholders of the organisation. Developed by Microsoft, PowerBI is an interactive data visualization software product (Microsoft, 2022). It is a collection of software services, apps, and connectors that work together to turn unrelated sources of data into coherent, visually immersive, and interactive insights. Data can be inputted/connected by reading directly from a database, webpage, or structured files such as spreadsheets, CSV, XML, and JSON.

SOAP API

Fieldview uses SOAP API. SOAP is an acronym for Simple Object Access Protocol (AltexSoft, n.d.), which is a messaging protocol specification for exchanging structured information. I.E., allows for implementation of web services. Normally it uses XML Information Set for its message format, and

relies on application layer protocols i.e., Hypertext Transfer Protocol (HTTP). It is worth noting Fieldview uses Hypertext Transfer Protocol Secure (HTTPS) for its API.

SOAP is over two decades old and allows users to pull or push data from a range of operating systems as well as numerous clients to run web services and receive responses over a range of script language and platforms.

PowerShell

During the research process several scripting languages was considers for the project for example Node JS, PowerShell or Python.

PowerShell was selected as it was deemed it would cause the least amount of impact on any existing system or servers and could very easily be transferred between machines in organisations IT environment. It has the potential of running future API requests inside the SharePoint Online Management Shell.

PowerShell is a command-line shell and scripting language (sdwheeler, n.d.) It supports variables, functions, branching (if-then-else), loops (while do, for, and foreach) and structured error/exception handling and closures/lambda expression

Power Automate

Power Automate is part of Microsoft's Power platform, which is a low-code application environment which allows for data analytics and workflow automation. This is where power automate will be used for the project, automating the flow of getting data from for example the Fieldview API to SharePoint Online.

Power Automate has 2 environments and both will be used in the project.

- The Power Automate Cloud Based service will be used to schedule Power Automate desktop tasks to run on a required basis (powerautomate.microsoft.com, n.d.).

- The Power Automate for desktops application will be installed and used to run PowerShell scripts. The scripting feature allows the user to run blocks of code in a couple of different languages, for example PowerShell, Python, VBScript and JavaScript (georgiostrantzas, n.d.).

DAX

DAX is a formula language used in Power Bi for data analysis. DAX is an acronym for Data Analysis Expressions. DAX allows the user to formulate formulas to perform queries and calculations on data in tables stored in Power BI (Minewiskan, n.d.).

Measuers are an important component of Power BI and Dax. A measure is a dynamic calculated formula which results change depending on the content i.e., when the data was last refreshed. Measures are created by using the DAX formula bar in the model designer. (Minewiskan, n.d.).

A formula in a measure can use standard functions such as COUNT or SUM, or you can define your own formula by using the DAX formula bar. It's worth noting that measures can be passed as an argument to other measures (Minewiskan, n.d.).
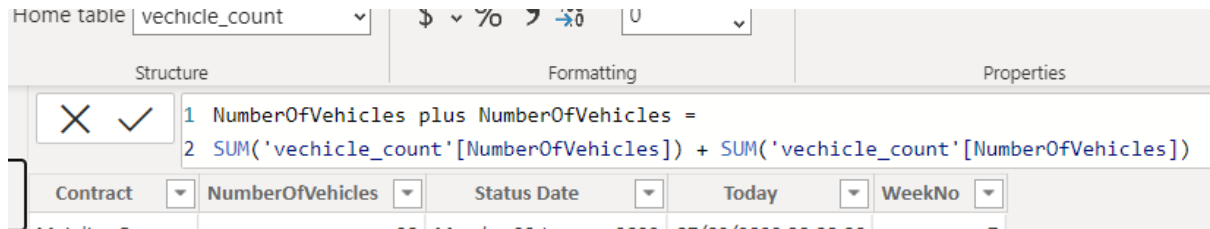

Figure 7: Sample View of DAX language in the formula bar of Power BI

Project Plan

The milestones to achieve are outlined below.

Milestone	Description	Due Date
Draft Proposal	Initial project proposal and concept	6th of November
Final Proposal	Articulate project nature and concept	4th of December 2022
Interim Report	Substantial update progress of the project	12th of February 2023
Final Project	Final project submission	2nd of April 2023

The project intends to span over the below phased development activities with preliminary due dates. Note the dates and phases may be subject to change as purpose of phase and allocated time is controlled by the organisation supervisor's priority.

Phase	Preliminary Sprint Due Dates
Research & Training	8th January 2023
Data Connector setup – Smartsheet	15th of January 2023
FV API setup and Data Persistence	29th of January 2023
PowerBI Dashboard – Fieldview Vehicle-check	12th of February 2023
PowerBI Dashboard – Fieldview Form Count	25th of February 2023
PowerBI Dashboard – SmartSheet Timesheet	19th of March 2023
Automation of Data flows and Power BI Refresh	26th of March 2023
Project Wrap Up	2nd April 2023

Project Planner

The Microsoft Planner (www.microsoft.com, n.d.) tool was used for planning and execution of the project. In planner the user can divide development phases also knowns as sprints into buckets. Inside the buckets the user would then add tasks into it.
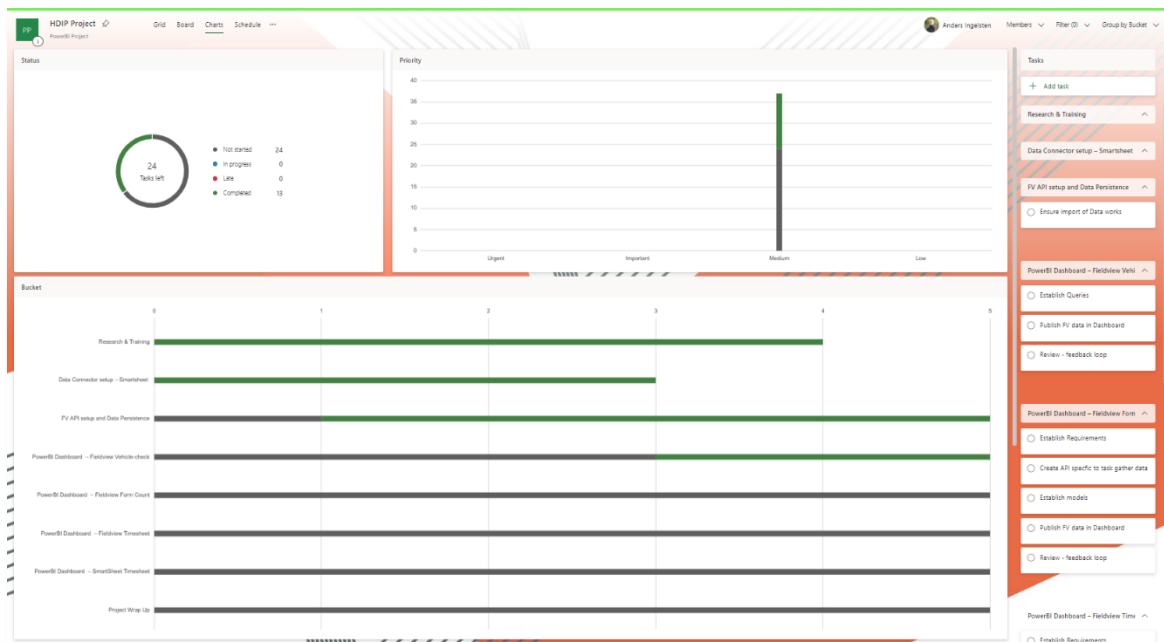
Figure 8: Microsoft Planner - Status of tasks and Buckets
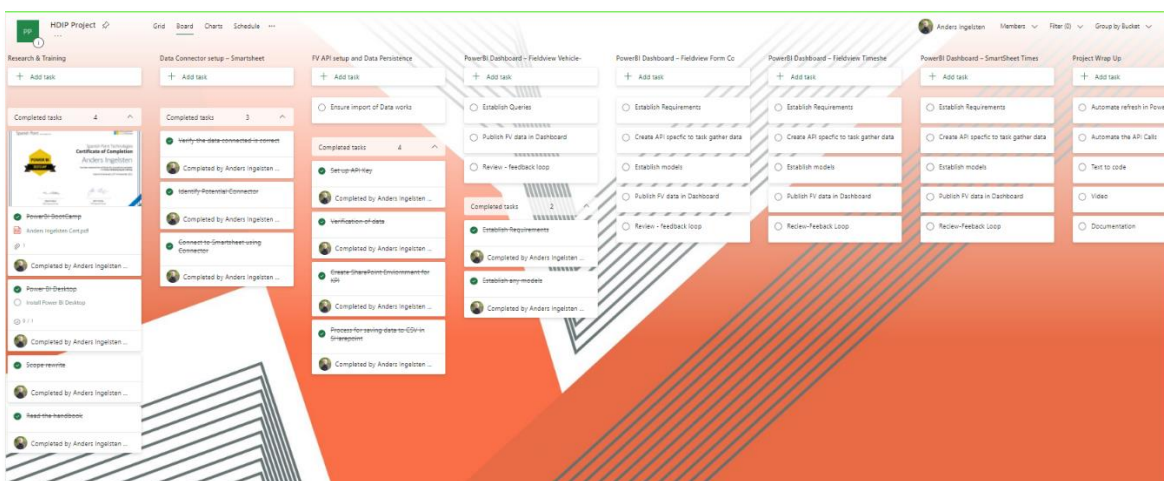
Figure 9: Microsoft Planner - Status of tasks and Buckets

Implementation

Phase 1: Research & Training

The phase of research and training covered three main areas.

The Fieldview API. The API Documentation of in the Viewpoint Help Section was investigated (help.viewpoint.com, n.d.), contact was also made to the support desk of Trimble to get direction of how the API was structured. The research revealed the following.

Fieldview has 3 Data Centre API regions/URLs

1. UK
2. North America
3. Australia New Zeeland

The organisations data is stored in the UK region data centre.

Every Region has twelve APIs in two set of six different API (help.viewpoint.com, n.d.). Grouped by XML or JSON, the APIs are:

- Configurations Services
- Forms Services
- Tasks Services
- Process Services
- Assets Services
- Project Services

2 APIs was explored for the purposes of the project

- Configurations Services – this API allowed me to identify and call project ids and associated information. For Example, the projectID is required to get Form Information
- Forms Services – this API allowed me to get form information, and individual answers.

Below follows 3 images of the formsservices API GetQuestionAnswer() command, first the parameters, a view of the SOAP API and then the returned answer information

Parameter	Type	Max Length	Required	Description
apiToken	string	20	Y	Your API security token configured in Field View.
formId	string	20	Y	The unique ID of the form.
questionAlias	string	100	Y	The question alias in the form you wish to retrieve the answer for.

Figure 10: Sample of input Parameters

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FieldViewWebServices/WebServices/XML/API_FormsServices.asmx HTTP/1.1
Host: www.priority1.uk.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "https://localhost.priority1.uk.net/Priority1WebServices/XML/GetQuestionAnswer"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetQuestionAnswer xmlns="https://localhost.priority1.uk.net/Priority1WebServices/XML">
      <apiToken>string</apiToken>
      <formId>string</formId>
      <questionAlias>string</questionAlias>
    </GetQuestionAnswer>
  </soap:Body>
</soap:Envelope>
```

Figure 11: SOAP request and response

```
<FormAnswerResponse>
    <Status>
        SUCCESS
        <Message>Success.</Message>
    </Status>
    <FormAnswerInformation>
        <FormAnswerID></FormAnswerID>
        <FormTemplateID></FormTemplateID>
        <QuestionType></QuestionType>
        <DataType></DataType>
        <Question></Question>
        <Answer></Answer>
        <AnsweredBy></AnsweredBy>
        <AnsweredDateTime></AnsweredDateTime>
        <HasActions></HasActions>
        <HasImages></HasImages>
        <HasComments></HasComments>
        <HasDocuments></HasDocuments>
    </FormAnswerInformation>
</FormdAnswerResponse>
```

Figure 12: Returned information

Time was spent understanding which API's information had to be pulled, stored, and passed on to other API's. One important feature noted in the research period was the API Call quota. The call quota is the number of points an API token can spend within a minute. An API token in Fieldview has a maximum quota of 120 points allocated. This means you are limited to the number of calls that can be made per minute. For example, the GetProjectFormsList() has a point quota of 10, so this would mean that associated token to the API can only process 12 calls in one minute.

API Token	Maximum Quota (/min)	Remaining Quota	Last Reset
AAAA-BBBB-CCCC-DDDD	120	32	2010-09-21 15:49:43.833

Figure 13: Call quota per API token

The research also showed that the Fieldview API would not easily return any data using the built-in data connectors in PowerBI and that the route of developing scripting of an API had to be taken. This is in line with the purpose of the project i.e., to showcase skills learned from the HDIP course.

It's worth noting that there exist third party apps like ZappySys ODBC Power Pack, who integrate SOAP API with PowerBI. However, this not a free software and annual subscription is approx. $650 per desktop install (ZappySys, 2018).

PowerBI. To get an understanding of what capabilities PowerBI has and its functionality I signed up for Spanish Point Technologies "Dashboard in a day" course. Spanish Point Technologies is a software company and a Gold Certified Microsoft partner. Spanish Point specialises in Azure, Microsoft 365, SharePoint, Dynamics 365, PowerApps & Power Automate, Power BI and SQL Server solutions (Technologies, n.d.). Company A has over the years used Spanish Point for various software solutions.

Dashboard in a day is a free full day workshop covering the capabilities of PowerBI through an instructor led online course.

- The goal of the course is to better understand how to:
- Connect to, import, and transform data from a variety of sources
- Define business rules and KPIs
- Explore data with powerful visualization tools
- Build stunning reports
- Share dashboards with their team and business partners, and publish them to the web

The power BI Bootcamp and training was completed on the 23rd of November 2022.

Figure 44: Certificate of Completion
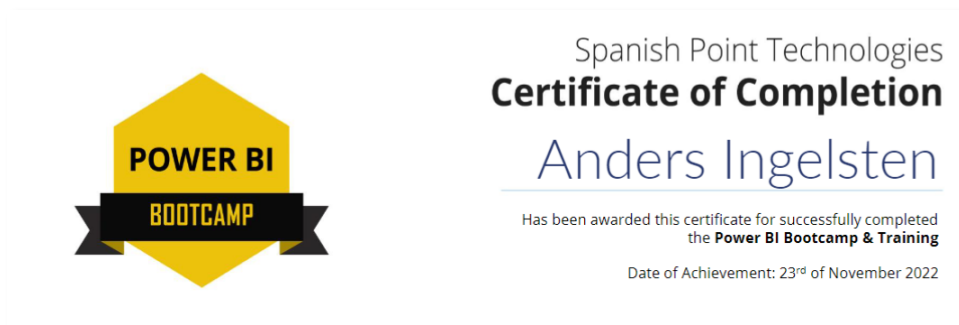
In addition to the above training, I also conducted various research online to find out how to do certain various items, anything from PowerShell Scripting to SharePoint Online solutions and Power BI queries.

Phase 2: Smartsheet Data Connector setup

Smartsheet provides a Data Connector, which is part of PowerBI built in connectors.

Following steps were completed in this Phase

1. Set up of user credentials in SmartSheet
2. Ensure user had access to relevant tables in SmartSheet
3. Connect Smartsheet to PowerBI using the built-in get Data feature in Power BI
4. Authenticate connection in PowerBI with Smartsheet user credentials.
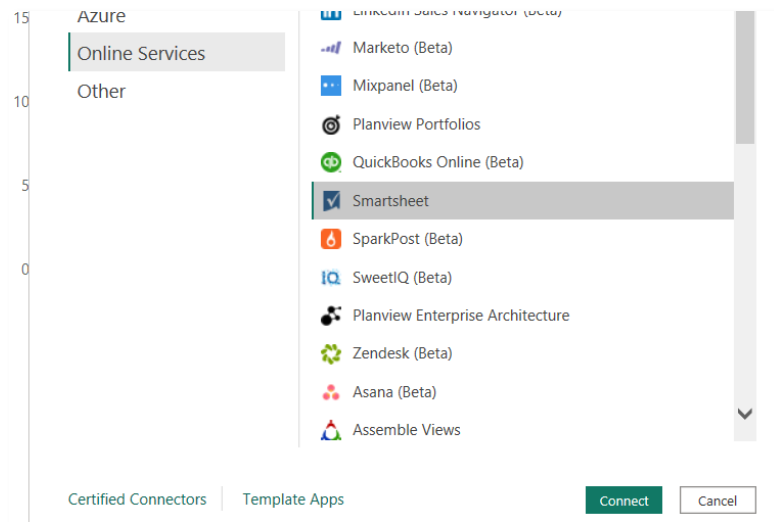5. Load and transform the relevant table from Smartsheet into PowerBI

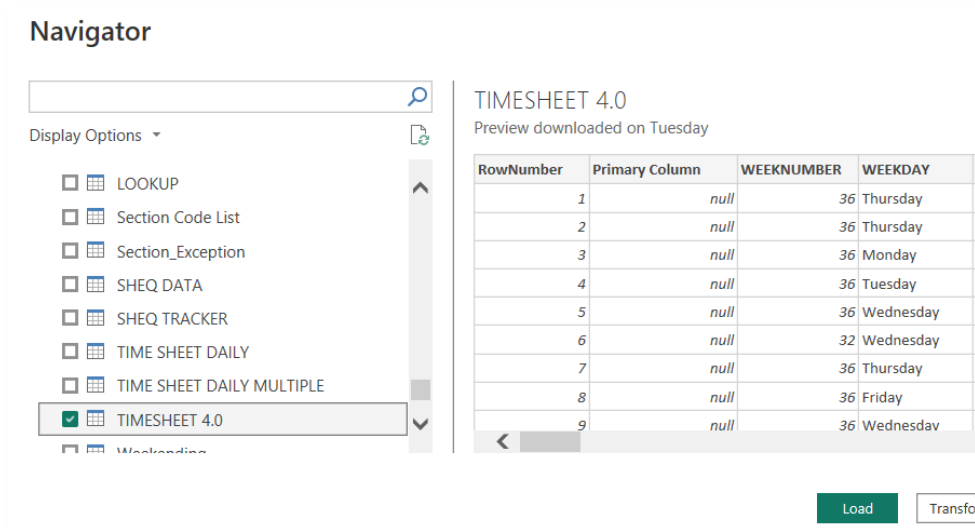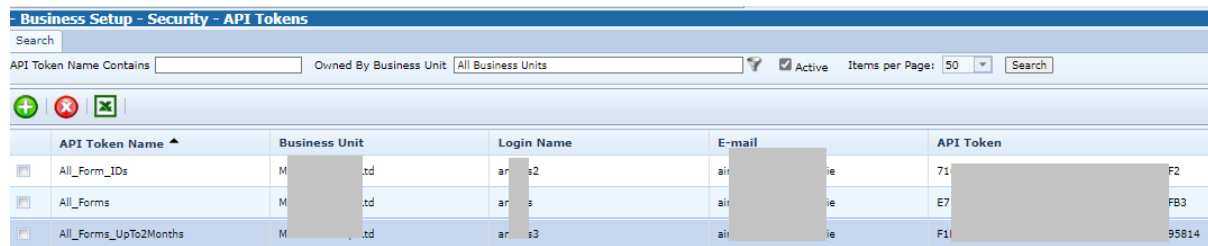*Figure 15: Get Data view in PowerBI of Online Services*

Figure 16: Preview of Connected Table ready to be loaded into PowerBI

Phase 3: FV API setup and Data Persistence

The focus of this phase was to be able to reliably pull information from the Fieldview API to the organisation's hosted Data repository i.e., SharePoint Online.

The first thing I did was to generate API keys inside Fieldview, API key can be set on Group Organisation and Company level. A Company is a child of an Organisation. All keys were set up on Group Level.

Figure 17: View of Created API Tokens in Fieldview

The SOAP protocol does not have to have any readily available plugins and add-on and the SOAP API relies on discreet calls for every interaction with the host. These calls also need to be structured inside a SOAP API envelope. I needed to be able to use a scripting language compatible with SharePoint online and which has the capabilities to execute general programming tasks, for example for each loop and if statements.

By utilising PowerShell and its module New-WebServiceProxy I managed to achieve this. The PowerShell New-WebServiceProxy will download the API's WSDL and use it to generate types for the proxy's interface, data contracts and headers. PowerShell also allows you to import variables from a file and export variables to csv files. Tasks can then be scheduled to run at certain intervals, but this will be resolved in a later phase.

```
<#
Fieldview API 1 - Lst of all froms up to 3months old.
This API pulls all project id's and pull all forms up to 3 months old by modified data
When pull is completed the data is exported to Sharepoint Online
#>

$ApiToken = Get-Content C:\Users\aingelsten\scripts\api_id.txt

Write-Output "Connecting to API"

$FVApiConfig = New-WebServiceProxy -Uri "https://www.priority1.uk.net/FieldViewWebServices/WebServices/XML/API_ConfigurationServices.asmx?WSDL"


$FVApiForms = New-WebServiceProxy -Uri "https://www.priority1.uk.net/FieldViewWebServices/WebServices/XML/API_FormsServices.asmx?WSDL"

Write-Output "Getting Project ID's"

$FVApiConfig.GetProjects($apiToken, $null, $null, 1, 0, 100).ProjectInformation.childnodes.id


$id = $FVApiConfig.GetProjects($apiToken, $null, $null, 1, 0, 100).ProjectInformation.childnodes.id
```

Figure 18: Code Snippet of the New-WebServiceProxy used to pull data from Fieldview

The New-WebServiceProxy sets up a proxy object (sdwheeler, n.d.), which allows for interaction with the Fieldview SOAP API and by utilising PowerShell capabilities of running foreach loops, if statements, I could connect and store all the project ids, and then loop them back into the next call which would in this phase retrieve form information. Every API Token has a call quota per minute (help.viewpoint.com, n.d.). Therefor I generated several API keys so I can generate multiple calls. I also added delays in the loop, to avoid exceeding the call quota.

```
foreach ($projectid in $projectids)
{

Write-Output $projectid

$FVApiForms.GetProjectFormsList($apiToken, $projectid, $null, 0, $datefrom, $dateTo, $null, $null, $null, $null, $null, $null).ProjectFormsListInformation.childnodes

$formsList = $FVApiForms.GetProjectFormsList($apiToken, $projectid, $null, 0, $datefrom, $dateTo, $null, $null, $null, $null, $null, $null).ProjectFormsListInformation.childnodes

if ($formsList -eq $null)
{
Write-Output "*****NOTHING FOUND****"
}
else
{
Write-Output  "Adding Projectid"

$formsList | Add-Member -MemberType NoteProperty -Name "ProjectId" -Value $projectid

Write-Output  "Writing to file"

$formsList | Export-Csv -Path c:\Users\aingelsten\scripts\formslist.csv -append -NoTypeInformation

Write-Output  "Data written to file"
}

Start-Sleep -Seconds 10

}
```

Figure 19: Code snippet of foreach loop with an if else statement

```
FormID                : F268753.52
FormTemplateLinkID    : 14680656
Deleted               : false
FormType              : Operations
FormName              : MP-194 Site Diary
FormTitle             : 2022-11-25
CreatedDate           : 2022-11-25T12:29:28
OwnedBy               :
OwnedByOrganisation   : Mainline Power Ltd
IssuedToOrganisation  :
Status                : Completed and signed off
StatusColour          : #009900
StatusDate            : 2022-11-25T12:48:12
Location              : MP-194 Cork Airport Substation
OpenTasks             : 0
ClosedTasks           : 0
FormExpiryDate        : FormExpiryDate
OverDue               : false
Complete              : true
Closed                : true
ParentFormID          :
LastModified          : 2022-11-25T13:07:17
LastModifiedOnServer  : 2022-11-25T13:07:17
ClosedBy              :
FormTemplateID        : 15738621
ParentProcessTaskID   :

Adding Projectid
Writing to file
Data written to file
23754
*****NOTHING FOUND****
23755
```

Figure 21: Screenshot of Sample Data from the GetProjectFormsList( ) call, returning ProjectFormsListInformation.

Data persistence

The main factor that decided which type of method for data persistence was going to be used in the project was availability and cost. The organisation has access to hosted servers but none of these are operational 24 hours a day, 7 days a week.

Increasing availability would lead to increased cost and for the data to be persistent it must write to non-volatile storage. Data persistence was achieved by saving the data to a file, after duplications was removed, it was then exported to Sharepoint Online (Rajack, 2018). By saving a file with the same name to the same location, SharePoint online just saves a new version of the file. A CSV file stored in SharePoint Online allows very easy connection to PowerBI by its built in Web connector.

```powershell
Write-Output "Exporting to SharePoint"

#Configuration of Sharepoint Variables
$SiteURL = "https://typetecmg.sharepoint.com/sites/ITMainline"
$SourceFilePath ="c:\Users\aingelsten\scripts\formslist.csv"
$DestinationPath = "Kpi_Data" #Site Relative Path of the Library


#Connect to PnP Online using Weblogin
Connect-PnPOnline -Url $SiteURL -UseWebLogin

#Powershell pnp to upload file to sharepoint online
Add-PnPFile -Path $SourceFilePath -Folder $DestinationPath

Write-Output "Process Completed"
```

Figure 22: Sample Code of writing a file to Sharepoint Online

```
InformationRightsManagementSettings      : Microsoft.SharePoint.Client.InformationRightsManagementFileSettings
IrmEnabled                               : False
Length                                   : 0
Level                                    : Published
LinkingUri                               : https://typetecmg.sharepoint.com/sites/ITMainline/Kpi_Data/formslist.csv?d=wfa70bd2094764f24956703ae372bd2a1
LinkingUrl                               : https://typetecmg.sharepoint.com/sites/ITMainline/Kpi_Data/formslist.csv?d=wfa70bd2094764f24956703ae372bd2a1
ListItemAllFields                        : Microsoft.SharePoint.Client.ListItem
LockedByUser                             : Microsoft.SharePoint.Client.User
MajorVersion                             : 1
MinorVersion                             : 0
ModifiedBy                               : Microsoft.SharePoint.Client.User
Name                                     : formslist.csv
PageRenderType                           :
Properties                               : Microsoft.SharePoint.Client.PropertyValues
ServerRedirectedUrl                      :
ServerRelativePath                       :
ServerRelativeUrl                        : /sites/ITMainline/Kpi_Data/formslist.csv
SiteId                                   :
TimeCreated                              : 22/01/2023 10:20:18
TimeLastModified                         : 22/01/2023 10:32:45
Title                                    :
UIVersion                                : 512
UIVersionLabel                           : 1.0
UniqueId                                 : fa70bd20-9476-4f24-9567-03ae372bd2a1
VersionEvents                            :
```

Figure 53: Screenshot of successfully writing the data to Sharepoint Online in PowerShell

Phase 4: PowerBI Dashboard – Fieldview Vehicle-check

This dashboard has a view visualizing Vehicle checks completed per week

Blank with 0 [Replace BLANK with Zero in Power BI Visuals Such as Card - RADACAD](#)

Starts with [Solved: DAX Starts with - Microsoft Power BI Community](#)

How to publish to web

[Publish to web from Power BI - Power BI | Microsoft Learn](#)

Phase 5: PowerBI Dashboard – Fieldview Form Count

This dashboard has a view visualizing salaried count of forms per contract

Pipedrive

[Embed a report web part in SharePoint Online - Power BI | Microsoft Learn](#)

Phase 6: PowerBI Dashboard – SmartSheet Timesheet

This dashboard has a view visualizing salaried staff breakdown per contract

Phase 7: Automation of Data flows and Power BI Refresh

Initial setup and testing of the process was done in February. Automation testing (www.youtube.com, n.d.)
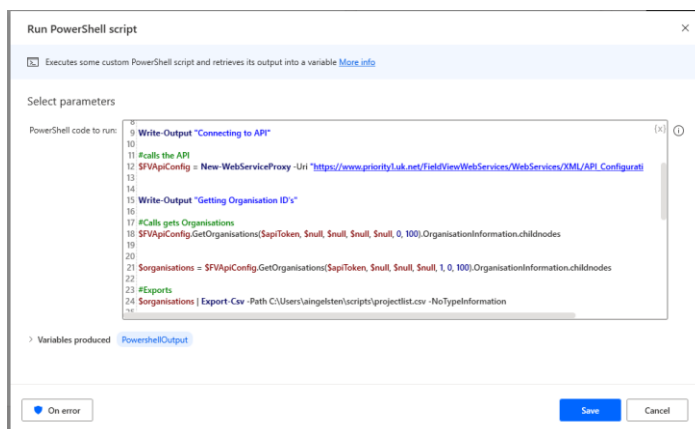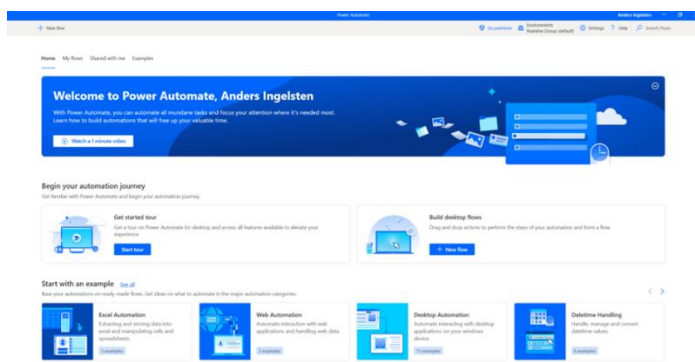
28-day run history ⓘ			Edit columns ⟳ All runs
Start	**Duration**	**Status**	
Feb 2, 04:00 PM (18 min ago)	00:00:38	Succeeded	
Feb 2, 02:59 PM (1 h ago)	00:00:42	Succeeded	
Feb 2, 02:00 PM (2 h ago)	00:00:47	Succeeded	
Feb 2, 01:00 PM (3 h ago)	00:00:50	Succeeded	
Feb 2, 12:40 PM (3 h ago)	00:00:33	Succeeded	
Feb 2, 12:38 PM (3 h ago)	00:00:44	Test succeeded	

Project Evaluation

Few pointers to expand on:

- Business logic – is it for example week 08/02/2023 – week 6 or 7
- DAX – Learning a new syntax, quite different to SQL Syntax
- Quality of the data and labelling of things like company names and are the correct person/project assigned to the correct company name.
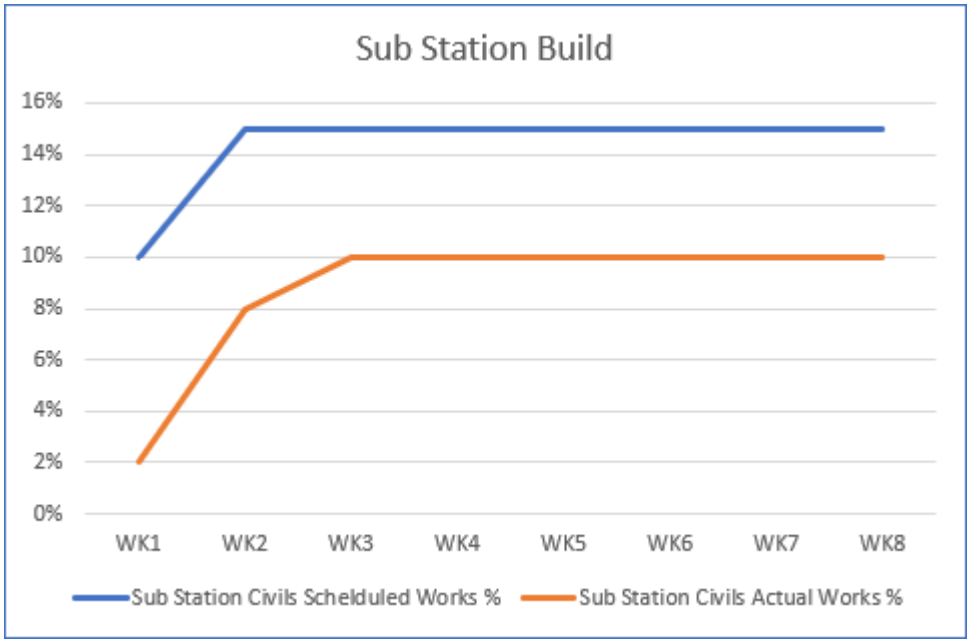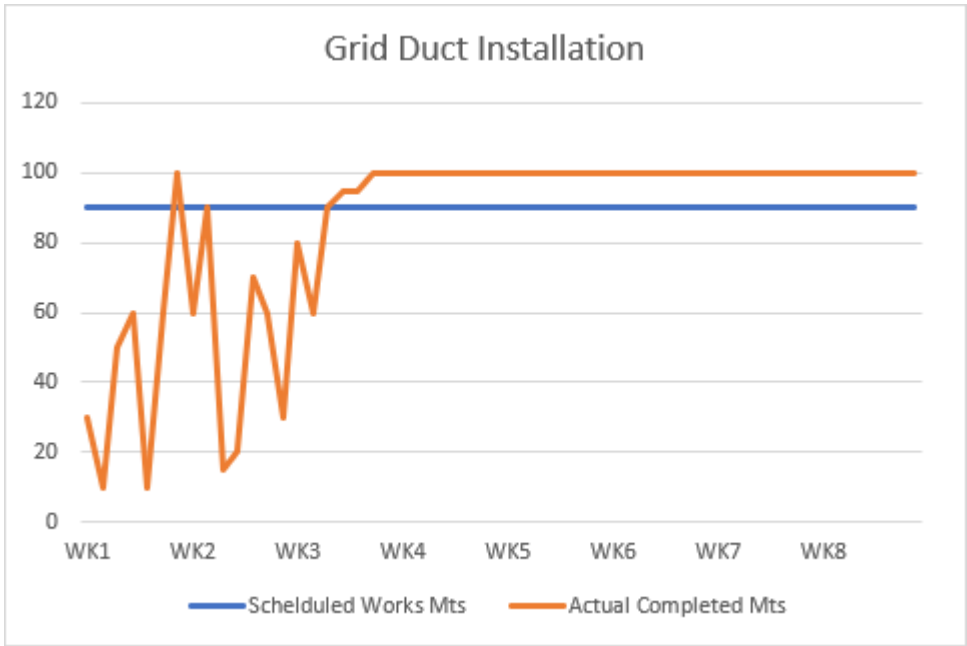
Future Work

Context

Typical request received in the process

"Hi Anders,

Not sure if it's possible, maybe it's there, I was wondering if it were possible to create a Table on View Point where a person could just fill in a Tab showing a Weekly Progress Report say on the Meterage of Grid Duct Trenching / Installation, that may run an excel file in the background which would show for instance the agreed amount per week which is a fixed point & the other the actual, Or do we have something else set up, as it would be great to track, say cable installed, joints made off etc"

Grid Duct Installation

Scheduled Works Mts — Actual Completed Mts

Sub Station Build

Sub Station Civils Scheduled Works % — Sub Station Civils Actual Works %

Chainage control per project

References

Smartsheet (2019). Smartsheet: Less Talk, More Action. [online] Smartsheet. Available at: https://www.smartsheet.com/.

Floor, T.V.U.O. 4th, Square, C., Tyne, F.S.N.U. and Kingdom +44.0800.048.8152, N. 3PJ U. (n.d.). *Cloud-Based Field Construction Software*. [online] Trimble Viewpoint. Available at: https://www.viewpoint.com/en-gb/products/viewpoint-field-view.

www.microsoft.com. (n.d.). SharePoint, Team Collaboration Software Tools. [online] Available at: https://www.microsoft.com/en-ie/microsoft-365/sharepoint/collaboration.

Microsoft (2022). Data Visualisation | Microsoft Power BI. [online] powerbi.microsoft.com. Available at: https://powerbi.microsoft.com/en-gb/.

AltexSoft. (n.d.). What is SOAP: Formats, Protocols, Message Structure, and How SOAP is Different from REST. [online] Available at: https://www.altexsoft.com/blog/engineering/what-is-soap-formats-protocols-message-structure-and-how-soap-is-different-from-rest/.

sdwheeler (n.d.). PowerShell Documentation - PowerShell. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/powershell/.

powerautomate.microsoft.com. (n.d.). Power Automate | Microsoft Power Platform. [online] Available at: https://powerautomate.microsoft.com/en-gb/.

georgiostrantzas (n.d.). Scripting actions reference - Power Automate. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/power-automate/desktop-flows/actions-reference/scripting.

Minewiskan (n.d.). Data Analysis Expressions (DAX) Reference - DAX. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/dax/.

www.microsoft.com. (n.d.). Task Management Kanban Solution for Teams | Microsoft Planner. [online] Available at: https://www.microsoft.com/en-ie/microsoft-365/business/task-management-software.

help.viewpoint.com. (n.d.). Viewpoint Help. [online] Available at: https://help.viewpoint.com/en/viewpoint-field-view/field-view/api-documentation/apis.

ZappySys (2018). Calling SOAP API in Power BI (Read XML Web Service data). [online] ZappySys Blog. Available at: https://zappysys.com/blog/call-soap-api-power-bi-read-xml-web-service-data.

Technologies, S.P. (n.d.). Spanish Point Technologies. [online] Available at: https://www.spanishpoint.ie.

Rajack, S. (2018). How to Connect to SharePoint Online using PnP PowerShell? [online] SharePoint Diary. Available at: https://www.sharepointdiary.com/2018/03/connect-to-sharepoint-online-using-pnp-powershell.html.

www.youtube.com. (n.d.). Power Automate Desktop || How to schedule Desktop Flows? [online] Available at: https://www.youtube.com/watch?v=IJ1I2737JWU.

APPENDIX X:

Declaration I declare that the work which follows is my own, and that any quotations from any sources (e.g., books, journals, the internet) are clearly identified as such by the use of 'single quotation marks', for shorter excerpt and identified italics for longer quotations. All quotations and paraphrases are accompanied by (date, author) in the text and a fuller citation is the bibliography. I have not submitted the work represented in this report in any other course of study leading to an academic award.

Student…………………………………..................................... Date …..……………................

Work Place Mentor…………………………………………… Date …………………………………