#### PROJECT INITIATION DOCUMENT

(Version created to support DAP605)

|  |  |
| --- | --- |
| **Project name** | SSATS to LabVIEW ETL Data Transposition |

|  |  |
| --- | --- |
| **Release** | Draft/Final FIX AMERICANISMS  APPENDICES  REFERENCES  PHOTOS  FIGURES  Date: |

|  |  |
| --- | --- |
| **PRINCE2** | Based on a reduced version of the PRINCE2 PID documentary requirements |

|  |  |
| --- | --- |
| Author: | Jack Sargeant |
| Owner: | Jack Sargeant |
| Client: | Spellman UK |
| **Document Number:** | 1 |

#### Document History

|  |  |
| --- | --- |
| **Document Location** | This document is only valid on the day it was printed.  The source of the document will be found in the Control section of the Project File. |

|  |  |
| --- | --- |
| **Revision History** | Date of next revision: |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision date | Previous revision date | Summary of Changes | Changes marked |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| **Approvals** | This document requires the following approvals.  Signed approval forms are filed in the project files. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Signature | Title | Date of Issue | Version |
| David Mason |  |  |  |  |
| Matthew Ford |  |  |  |  |
|  |  |  |  |  |

|  |  |
| --- | --- |
| **Distribution** | This document has been distributed to: |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Date of Issue | Version |
|  |  |  |  |

#### Purpose

#### To define the project, to form the basis for its management and the assessment of overall success.

Contents

[Background 4](#_Toc158041388)

[Project Definition 4](#_Toc158041389)

[Project Objectives 4](#_Toc158041390)

[Business Case 4](#_Toc158041391)

[Options Analysis 4](#_Toc158041392)

[Defined Method of Approach 4](#_Toc158041393)

[Project Scope 4](#_Toc158041394)

[Project Deliverables 4](#_Toc158041395)

[Exclusions 4](#_Toc158041396)

[Constraints 4](#_Toc158041397)

[Assumptions 4](#_Toc158041398)

[Project Organisation Structure 4](#_Toc158041399)

[Communication Plan 4](#_Toc158041400)

[Project Quality Plan 4](#_Toc158041401)

[Project Controls 4](#_Toc158041402)

# Background

Spellman High Voltage manufacture high voltage power systems used across varying markets. These include but are not limited to applications in medical, research, and security sectors. The organization is based in Hauppauge, New York, and employs 2300 staff across facilitates worldwide. Locations include Mexico, China, Germany and the United Kingdom.

Historically each site has operated somewhat independently, being responsible for their own customers and practices. Recently however, corporate want to begin to centrally control practices, procedures, and data. This initiative aims to achieve a holistic view of global operations and enable the identification and sharing of best practices across the company. Ultimately, this could lead to the streamlining of manufacturing and procurement processes, empower data-driven decisions, and make the movement of product lines to other sites far less ad-hoc.

A critical manufacturing step for Spellman is the testing of products. This step ensures customers receive their goods as expected, and such operate as agreed. Testing procedures and software have been historically developed independently on a per-site per-unit basis, with each site utilizing their own approach. Corporate have instructed all sites to streamline testing practices by grandfathering current solutions and replacing such with solutions that articulate Test Stand[[1]](#footnote-1) and LabVIEW[[2]](#footnote-2).

The UK site have developed their own test platform called “SSATS”. SSATS is a C#.NET derived form-based application, that articulates excel based test specifications to initiate tests and collect resulting test data. These excel workbooks are equipped with embedded automated testing procedures, achieved by implementing a native software framework using VBA. This enables the UK site to make the testing of products almost automatic, and only require an operator to enter product details and press start. This approach has proven suitable since its inception 10 years ago; however, it is rife with problems regarding adaptability, centralized control, and platform centric limitations.

Data from SSATS is collected once a test specification is closed. A worksheet named “Database” states all the fields to be lifted from the workbook once closed. The SSATS GUI maps this information to a DTO (Data Transfer Object) in which writes its contained contents to the SSATS database. SSATS was designed to test one serial number at a time, however, through business need, the UK team needed a solution to test more than one unit at a time using a single workbook. The team developed “Squirrel”. Squirrel is an API centric service responsible for lifting results for more than one serial number from a single workbook. This means that there are two data inception points for information stored in the Mustang Database.

Whilst the UK site have started to develop test solutions using the outlined software packages, it will take some time before solutions can adequately replace tests currently carried out using SSATS. However, corporate wish to start collating test data as soon possible. Because of this, a requirement exists to copy, sanitize and reformat data from the UK’s own SSATS result database into a format that matches the globally agreed, new database schema. In doing so, UK test data can be used in global business intelligence initiatives and visualizations and enabling such data to be present in their ERP system and material movement software Scanstar[[3]](#footnote-3).

# Project Definition

## Project Aims and SMART Objectives

### **Aim**

* **Prototype a method to migrate Spellman UK's product testing data to a standardized format and prepare such for integration with the global system by June 2024.**

##### **Objectives**

* **Implement a data cleaning process and procedure that addresses identified data quality issues in UK test data extracted from the SSATS database to achieve 90% data completeness and consistency by May 15, 2024**
* **Implement a dynamic, remote and highly available distributed data integration service that can access and process test data from different test data inception points by June 2024**
* **Identify an appropriate solution for transposing test data from one database schema to another, in which 95% of all test data captured is transferrable and usable by April 2024**

## Business Case

Spellman’s new global initiative focuses on enabling organization wide collaboration, but also formalizing data collection, and ensuring a common format for all similarly shaped data collected. The collection of test derived data will be crucial for Spellman to build business continuity and security, as such can be used to understand product performance, highlight and forecast any problems, and can also be used to calculate throughput and yield, factors that affect the larger business paradigm.

Corporate have also highlighted how data like the test data will be ingested into the global data lake[[4]](#footnote-4). This centralized repository can empower deeper understanding of the organizations current and future stance through data cubes[[5]](#footnote-5), enabling multidimensional analysis across various parameters, kinds of data, locations, and timeframes.

Building on this strategic data initiative, Spellman require the UK’s test data, and other site-specific data within the global data lake. As the UK contributes a share to global operations and revenue, comprehensive representation of the UK’s data becomes critical. In collating such, Spellman can ensure data-driven initiatives include insights into all locations under their umbrella. By readily integrating UK data into the data lake, Spellman will be able to gain a clearer picture of the UK’s contribution and performance and reinforce organizational security by painting a far more accurate picture of this site’s performance.

It is evident that a business need exists for this data, and UK site should immediately start transposing test data into a format suitable for this global initiative, to begin fostering an age of improved global business intelligence.

While the outlined global initiative addresses the standardization for broader analysis, there's also a crucial need for it at a more local level in the UK. Having historically experienced challenges with the quality and consistency of data written their SSATS database, the UK team sees the data transposition requirement as a golden opportunity to cleanse, standardize, and strengthen all locally collected data. Improvements to data and associated visualizations can be used to look at product performance, identify bottlenecks and issues, and look at equipment performance. Most importantly however, such could assist the UK site in bolstering the business case for automation initiatives and stand as concrete proof to improvements made regarding performance. visualizations of this kind are Ad-Hoc currently, and the data collected is not in a fit state to be prudently used, due to the volatile nature of the UK’s test platform and data collection method(s).

## Options Analysis

Spellman UK have a few options available to them relative to their requirements. It would be important to add that **the organization shows no interest in additional investments to technologies at this time.** The requirements outlined infer the use of a remote service, responsible for moving results from one database to another, whilst remaining dynamic and automatic as specified. Some of recommendations would incur a cost, and have been stated for the interest of both the project board, and stakeholders:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Options | Benefits | Drawbacks | Feasibility | Impact | Compliance |
| Do Nothing | No cost or time spent required.  No additional risk.  Frees up development staff. | No standardization to data.  Data in no state to be moved or utilized by Spellman UK or corporate.  No means of corporate retrieving test data in an expected format. | Not very feasible. This approach very much misaligns the requirements by corporate to centralize and collect data. | Corporate will not have full scope of all data required from the UK site to create meaningful conclusions. | Would not affect or violate any compliance ruling regarding regulation, or company policy. |
| Developed in-house, self-hosted | Full control.  Tacit knowledge applicable from both developers and IT staff.  Flexibility.  Security.  Customizable SLAs with IT regarding service uptime | Time and cost on development and hosting.  Complexity and scalability regarding infrastructure management. | Completely feasible following review with IT. Developers have knowledge of practices and requirements to deliver solution. Infrastructure also exists and can support. | Would broaden the portfolio of UK IT, as would need to upkeep service.  Development time spent takes away from another project.  May require additional training. | Does not affect or violate any policies. Data collected is not personal so does not breach GDPR or similar. IT staff and developer would be responsible for adhering to relevant policies, however. |
| Developed in-house, cloud hosted | No reliance or requirement on in-house hardware/infrastructure.  Scalability | Security concerns regarding handling of sensitive data  Vendor-Lock In  Less control than self-hosted.  Potentially limited by SLAs agreed with IT.  Additional Costing | Not as feasible as all in-house. Organization have voiced there is no interest in investing with cloud or “as a service” providers. Developers and IT staff also lack knowledge to control such.  This suggestion is also the only one that would incur additional cost, outside of investments made by the organization in terms of time and allocating owned infrastructure. Appendix 1 demonstrates a quote from AWS (2024), which denotes a monthly cost of $55.40 based of current organizational activity. | Would require ongoing costs and negotiations with cloud provider. | Would likely breach any policies regarding data safeguarding under the organization’s ISO:27001 centric policies. Organization very much wants to keep data “In-House” |
| Database centric solution | Efficiency: Can leverage DBMS engine.  Ease of management.  Can be integrated into current DBMS.  Limits movement of data. | Limited functionality.  Costing for additional cores.  Limited scheduling flexibility.  Known volatility with currently deployed DBMS.  Added complexity compared to what could be possible in code. | Whilst somewhat feasible, and seeming the organization already use databases, this could work. However, the UK have experienced with varying database configurations and problems and may not wish to tie critical activities to such. | Would require load balancing and throttling to ensure DBMS can process all information.  System would also require reporting mechanisms, meaning additional development time. | Would not affect or violate any compliance ruling regarding regulation, or company policy as data is being kept in-house. |
| Attach Writing events to Excel Test Specifications | No need for a service to be hosted or controlled.  Data writing could be tied to specific tests and rolled out incrementally. | Would need to be attached to every single test specification individually.  Limited by VBA code.  Would need to be event driven.  Very much becomes tightly coupled with Office, and any updates that may hinder operation | Not too feasible. Whilst it makes to write data from the source itself, by nature all test specs remain somewhat independently controlled, meaning each would need to be manually updated and configured to move such. This would be a huge task, and one not likely to be chosen by the organization. | Would require Test Equipment staff to begin manually changing every test specification, taking them away from other responsibilities.  Also tied to Office and automatic updates that have previously hindered the organization. | Whilst it could be possible to adhere to policies set by IT, XLSM file or any VBA code written in office documents have notoriously been associated as a cyber-attack vector. An automatic write to a database on a network via VBA could raise problems. |

## Defined Method of Approach

##### **Understanding PMLC and SDLC**

Any project’s success hinges on the identification and selection of a suitable PMLC (Project management lifecycle), and as this is a software centric project, also a SDLC (Software Development Lifecycle). Choosing the right methodologies doesn't just ensure clear direction and efficient resource allocation, it directly impacts the quality, functionality, and timeliness of the project deliverables. Both PMLC’s and SDLC’s should also be concerned with dictating project control measures, generating project artefacts, risk mitigation and defining project phases.

A diagram of a process

Description automatically generatedThis project will be carried out in phases. For this project, the PMLC will very much serve as a top level, being concerned with overarching project controls and documentation, whereas the SDLC will be concerned with controlling development activities once that stage is reached. The following figure attempts to demonstrate such, where orange dictates PMLC stages, and purple SDLC stages; it attempts to show their relationship:

Figure 1- demonstrating the relationship between a SDLC and PMLC within the context of this project.

To understand both a suitable PMLC and SDLC for this project, we must understand the organizations current stance and ability to meet the project’s aim, and subsequent objectives:

* **The development team that will be assigned to this project have a solid grasp of the theory involved regarding data cleaning and mapping.**
* **Whilst the development team are technically capable of delivering a solution, it is more than likely this approach could involve technologies not found in their current stack, indicating some level of learning and prototyping will be required; this kind of project is different from what they have worked on previously.**
* **The organization have a plethora of self-owned and controlled technologies suitable for deploying a solution fit for their requirements.**

Previous sections of the document and this synopsis of the organizations current stance ultimately indicates that whilst the goal and business case for this project are clear, there may be some friction regarding delivery. This is due to potential unfamiliarity with required tooling, and subsequently need to learn, research and prototype solutions; some level of global communication is also required, something the project team have only just started doing.

A diagram of a solution

Description automatically generatedIn his book “Effective Project Management” Wysocki (2019, p.23) denotes 5 different PMLC/SDLC approaches, with each suitable for different kinds of project. These approaches are iterative, linear, adaptive, incremental and extreme. Wysocki proceeds to explain (2019, p.38) four different kinds of process groups, and how the PMLC/SDLC approaches are best mapped to such. These process groups aim to define the characteristics of a project, and what a project of that type could entail. Wysocki mapped these based on the certainty or uncertainty of a project’s goal and/or solution.

Figure 2- Wyscoki (2019, p39). This model demonstrates how different project approaches can be mapped to different kinds of project.

The process groups and corresponding PMLC/SDLC’s have been mapped as such:

* **TPM (Traditional Project Management):** Projects within this quadrant are ones in which both the end goal and solution are both certain.
* **Agile Projects:** Agile projects generally have a clear goal, but uncertainty about regarding the solution.
* **Extreme Projects:** Extreme projects do not have either a clear goal or solution.
* **Emertxe (ee-MURT-see) Project:** Emertxe projects do not have a clear goal, but do have means of delivering a possible solution, whatever it may be.

Whilst the development team will more than likely need to develop some new skills and understand new approaches, this should be something factored in to the phases of the SDLC, rather than classifying the solution as “uncertain”. Projects of this caliber, and phases within any SDLC will generally include a phase for identifying requirements, prototyping and theory crafting. Furthermore, it has also been identified that the development team are more than capable of meeting the goal stated. Because of this, this project is most likely suited to TPM.

As demonstrated with Wysocki’s classification, the selection of a TPM approach to this project highlights three different PMLC/SDLC options:

* **Linear**

This model is the most traditional approach to a project. It consists of sequential phases, in which are to be completed one after another. This approach is predictable and requires well-defined requirements to function. However, it is not adaptable to change, and can be a stubborn choice for complex projects.

A grey rectangular sign with black text

Description automatically generated

Figure 3 - The Linear PMLC (Wysocki, 2019, p34)

* **Incremental**

This model delivers functionality in smaller increments, in which build upon each other. This enables feedback to be captured early, and incorporated into increments, this mobility allows it to adapt to change. It does require good planning and co-ordination, however, as increments still need to be planned/scheduled. This approach is also quite rigid, and user feedback may not influence increments appropriately; this could lead to an undesired project trajectory.

A diagram of a system

Description automatically generated

Figure 4- The Incremental PMLC (Wysocki, 2019, p44)

* **Iterative**

The iterative approach balances elements of TPM and Agile methodologies, offering flexibility and adaptability while maintaining some aspects of structured planning. The iterative approach works around “iterations”. Each has defined goals, tasks, and control mechanisms, like TPM, but within shorter timeframes. This allows for early feedback and adjustments, lowers risk by catching errors early, keeps stakeholders engaged through frequent deliveries, and easily adapts to changing needs. However, this approach does require discipline within iterations and fostering clear communication.

A diagram of a system

Description automatically generated

Figure 5- The Iterative PMLC (Wysocki, 2019, p49)

### **Selecting a PMLC**

As the requirements for this project are so clear cut, it would be most feasible to control the top level of this project using a Linear SDLC. This is because its sequential phases, from planning to deployment, align well with a static roadmap; whilst the requirements from the organization may be verbose, in essence, the project is simple, as data effectively needs to be moved from A to B. Furthermore, this project is carried out with best practices with PRINCE2, which indicates what key documents need to be generated, and how each iteration/stage is to be controlled. With well-defined requirements upfront, each stage builds upon the last, minimizing the need for course correction along the way.

### **Selecting a SDLC**

As stated previously, the SDLC is to be contained with the PMLC, with its own unique and direct phases unique from the PMLC. Whilst the Linear approach will be effective in the greater project space, this approach will not suffice for the development stages of the project, mainly due to the inflexibility of such.

Based on the capability of the organisation to deliver the project, it would be most feasable for all development activities and the SDLC to follow an Iterative approach.

Firstly this approach would enable the development team to work on all aspects of the software solution simutaneously, slowly growing the solution into an artefact fit for purpose, one in which is directed by requirements, but also stakeholder feedback. With constraints on time and potential skill gaps, this can enable the project team to build components out as much as necessary, and come back to more troublesome areas if need be. Secondly if any turbualnce is apparent once development begins with new technologies, the Iterative approach can help remedy any issues, and place resources where neccessary.

A diagram of a diagram

Description automatically generated

Figure 6- Demonstrating the SDLC iterative model (Geek Culture, 2023)

Nuon (2023) deontes how iterations allow developers to get closer to the correct, final, and desired product, or the “prototype”, whilst constantly being able to adhere to requirements from stakeholders, correct problems, and mobilse best practices more effectively. These very much cover both conerns identifed with the project, but also reinforce the development team to deliver a pertinent solution.

## SDLC Phases

The diagram below demonstrates the composite relationship between the PMLC and SDLC. As seen in figure 1, the orange components demonstrate top-level, or “meta” project phases of the Linear PMLC; these phases have been fleshed out further in “Project Controls”, in which discusses adherence with PRINCE2 practices including key project documents, stage/phase boundaries, and stage/phase control. The purple sections of the diagram denote phases associated with the SDLC, and how the PMLC and SDLC relate to each other.

A diagram of a process

Description automatically generated

Figure 7- Demonstrating the SDLC phases associated with the project.

The figure demonstrated states what phases are to be carried out in the SDLC. Please note, the green jotted line demonstrates the iterative nature of each of the SDLC phases, and how each phase should on move on to the next once the phase is deemed complete by stakeholders/project team. Typically, phases such as Requirements Capturing, Planning, and Operations and Maintence would be included in a SDLC, however, based on the composite nature of this project approach, these crucial stages are to take part in PMLC phases, rather than directly in the SDLC; the SDLC is strictly for development activities. Furthermore, phases like Maintence would normally be included, however with the purpose of this project being a proof of concept system, this will not be included.

The phases of the SDLC are to be carried out as followed:

* **Design and Research**

Following inputs from phase 2 (Requirements, Aims, Objectives, Resources, Stakeholders etc.), development can officially begin. This phase of the SDLC should be concerned with designing the project solution, including relevant diagramming and design documentation. This phase should also be centric on understanding best practices, for whatever approach is taken for delivering the project solution. All of this is to be collated into a System Specification document, which is to serve as a guidebook through subsequent phases. This phase should only be marked as complete once the project team and stakeholders are satisfied with the findings.

* **Development and Testing**

During this phase, the project solution is implemented based on the design specifications outlined in the System Specification document. As the organization does not employ any software testers, and their developers historically test their own software, such is to be done during this development stage. Testing continues iteratively alongside development, encompassing unit testing, acceptance testing, and integration testing to identify and rectify any issues promptly. This phase is complete once stakeholders/project team are satisfied that the proof-of-concept solution meets project aims and objectives and can adequately be demonstrated.

* **Deployment and Documentation**

Following the development and testing phase, the focus shifts to deploying the prototype solution for thorough evaluation. This stage involves showcasing the proof-of-concept solution in a controlled environment, ensuring it functions as expected and aligns with the project's objectives and requirements.

Additionally, user feedback and any observed issues are gathered to refine the prototype further before completing the full SDLC. Documentation regarding HCI, and program functionality is to also be formalized at this stage; documentation should be generated for all code written and more technical aspects throughout the development phase and consist of content outlined in the System Specification document, alongside code functionality. This stage, and the SDLC will be deemed as completed once a full demonstration of all moving parts is achievable, and documentation has all been generated.

## Project Scope

The outlined project is to deliver a proof-of-concept solution. This means the delivered solution is to only function with sample data, and example test specifications.

Only a few test specifications and expected data sets are to be supported first. As the need grows, the organization can add more product lines to any data movement activities.

The solution will not be able to recalculate and sanitize all error types. Common problems will be supported; however, this feature must evolve as more tests are incorporated into movement activities, and data associated error types/problems are identified.

This project is to only be responsible for copying data from SSATS derived test only, and not support other test options that may be supported by the organization.

The solution is to adhere to all IT policies outlined by Spellman High Voltage and must be able to function on currently owned hardware/network; Spellman are a Microsoft centric organization in terms of PCs/Networking.

The solution is to be solely supported by, implemented by, monitored, and controlled by Spellman UK.

Spellman UK are to transpose test data from the SSATS database to a locally hosted database that matches the LabVIEW schema. Spellman UK are not responsible for integrating transposed test data to corporate, or integrating such into the data lake.

## Project Deliverables

### **Self-Hosted ETL data transfer service**

This is the main artefact to be delivered from this project. Spellman need means of dynamically transposing data from one database to another, effectively writing a reformatted, sanitized, and standardized set of test data to the LabVIEW database, derived from tests carried out from SSATS, in which has had its results written to the SSATS database. This solution needs to demonstrate what a final product could look like and give some kind of indication to what both the process of transposing the data looks like, but also what kind of shape the data takes once transposed.

The project sponsors have also indicated that this service needs to be both dynamic and remote, due to the use of test data used in visualizations, but also the printing of test results to send to customers. Cleaned, sanitsed data needs to be readily available. Because of this, this service needs to be readily available when a test is completed, from anywhere in production.

Based on this synopsis, this project will output an API, like the Squirrel service currently used by Spellman UK. This API will be a Controller derived solution written is ASP.NET and hosted by the organization themselves on the server “SHVUKTED”. This API will be responsible for invoking the ETL functionality.

ETL functionality is to be a C#.NET Class Library project, in which exposes all the functionality required to transpose test data. .NET has been chosen as the company widely adopts the .NET platform, and servers can be easily configured to use whatever runtimes necessary. Furthermore, being a Microsoft owned platform, satisfies security concerns from IT. The ETL aspect will be solely responsible for extracting SSATS data, cleaning such, and then writing the sanitized data to the LabVIEW database.

The ETL and API are to be developed separately, with the ETL being started first. Once in a suitable place, the API aspect will be developed, in which will drive transposition events by using DLLs built from the ETL solution. Both aspects of this project are to be written in adherence to SOLID, to promote adaptability, flexibility, and scalability.

Both parts to this solution are to be accompanied by appropriate testing processes and documentation. Calls to the API (In SSATS and Squirrel) to attach functionality are to be supported towards the end of the project. Both parts are to also be developed following their own best practices, for example the API being only accessible by key.

Further details will be provided upon agreement of this PID and will be available in the System Specification document if this project is given the green light.

### **“Transfer Receipt” Database Functionality**

Whilst movement events will be obvious based on the growing size of the LabVIEW database, there is no mechanism to account for missed transposition events or errors. This is something that needs to be monitored to build business continuity and understand what kind of data is failing to match and be written into the database. The addition of tables that almost act as a transaction receipt in the SSATS database will be essential in improving the coverage of what data is successfully written into the database.

This will be delivered alongside the ETL/API tool, feature similar nomenclature regarding naming, and be implemented with normalization in mid to the 3NF.

### **Software and Operation Documentation**

Documentation that corresponds with software design, and how the service operates is to be generated and distributed to appropriate parties upon project completion.

## Exclusions

No improvements are to be made to SSATS or Squirrel in terms of functionality. Modifications that are required to either of these programs are to strictly support the project solution only.

No historic data is to be pulled across to the LabVIEW data. Only data from new Pre-Tests (-26) and Final Tests (No Affixes) are to be moved. If historic data requires movement, this is another project entirely due to volume, volatility, and various required cleaning procedures.

The SSATS database is strictly open for addition and closed for modification; no tables are to be reorganized or spliced; however, tables can be added. The LabVIEW database is strictly closed in all cases.

No training is to be included in the delivery of this project.

This project will not be responsible for putting mechanisms in place in the event where data is not written to the database due to network issues etc. Whilst control measures are to be included that indicate write events, any mitigating action will not be covered in the delivered prototype.

## Constraints

There are some constraints this project is subject to, these include:

### **Databases**

* Neither of the databases are normalized beyond the 1NF. This means extra care is going to be required when constructing queries to read and write from either of them.
* The schema further stray away from SQL best practices by not using recommended nomenclature, tables missing primary key relationships, and the existence of plenty of transitive relationships within each table.

### **Time Allowance**

* Work on this project must end by June 21st, 2023; stage iterations are to be scheduled appropriately around this.
* The amount of time committed to the project by the team each 30 hours.
* The project sponsor is only available between 08:15 and 17:00 Monday to Thursday, and unit 12:45 Friday.

### **Budget**

* Whilst the project does not have to worry about costing, all development tooling and software packages must be open source or free of charge, limiting potential candidates.

### **Development**

* The development team must aim to adhere to and follow the SOLID methodologies. This may make projects overly verbose.

### **Implementation**

* The hosting of the completed/prototype API is at the discretion of IT, who may require agreement to SLAs, or have other priorities.

## Assumptions

Detail anything that you need to have or must be available in order for you to undertake a successful project, for example; access to specific members of staff, regular reviews of progress, quick decisions, access to the organizations intranet etc. Be specific e.g. if you need decisions or sign-offs to be turned round in 3 working days, say so.

# Project Feasibility Assessment

SWOT

# Project Organisation Structure

It is often useful to include a picture of the organization, showing the team, there roles and who your principle contacts are. This section should have a clear statement on the roles and responsibilities of each member of the team. Customer roles should also be specified.

Mendelow

# Communication Plan

*Identify who the key contacts are for the project together with their contact details. Also identify how often you intend to review and communicate your project progress and who to. It is useful to include some more formal reviews that follow PRINCE2 practice.*

*Mendelow stakeholder mapping matrix*

*MEETINGS ETC, HOW DO THEY INFLUENCE THE ITERATIONS*

# Project Quality Plan

*The project quality plan details how you intend to meet the customer’s quality expectations. You need to begin by defining these. You should also state the acceptance criteria, who is responsible for quality, any quality standards you will adopt e.g. covering documentation and verification and validation.*

# Project Tolerances

*This section identifies any agreed tolerances on the project. Typically, these would relate to time, cost, quality and functionality. So, for example, the project could be delayed by up to one week without reference to the Project Board but beyond this would have to seek approval. You may note that all essential requirements have to be delivered but there may be some tolerance re desirable requirements.*

# Project Controls

*This should document the project controls you intend to put in place. It is likely to include statements on the following: that you will be following the PRINCE2 methodology, how and where you will maintain and keep your project files, how you will manage and requests for change, configuration management controls etc.*

*Controlling Phases using PRINCE2 – following the methodology*

*File Storage and Management*

*Software Development Controls*

*Documentation*

#### Attachments

|  |
| --- |
| Initial Benefits Case *It is worth thinking through the potential benefits of your piece of work from the perspective of the customer. You have detailed the objectives but if you deliver on these, how might the business benefit? Understanding this may help you to identify your key project themes and ideas to pursue.* |
| Initial Project Plan *Ideally, this should be a Gannt chart showing the tasks to be undertaken and when you expect to do them. You should also include critical review points and milestones. It should build on the SDLC phases defined. This will evolve during the project.* |
| Initial Risk Log *You should think through any of the potential risks that might prevent you completing the project to plan and expectation. Issues may be difficulties in contacting people, arranging interviews, unplanned absence etc. Document these risks and identify any actions that you can take to try and prevent them occurring in the first place. Monitor risks on a regular basis*  *Product Description*  *Project Brief.*  *Captured Requirements* |

# References

<https://www.ni.com/en-gb/shop/product/teststand.html>

<https://www.ni.com/en/shop/labview.html#:~:text=LabVIEW%20is%20a%20graphical%20programming,and%20fully%20integrated%20user%20interfaces>.

<https://aws.amazon.com/what-is/data-lake/>

<https://www.thedataschool.co.uk/sherina-mahtani/data-cube-an-introduction-to-the-concept/>

<https://www.ibm.com/topics/rpa>

<https://www.fbi.gov/news/stories/melissa-virus-20th-anniversary-032519>

https://www.goodcore.co.uk/blog/iterative-development/

Wysocki, RK 2019, Effective Project Management : Traditional, Agile, Extreme, Hybrid, John Wiley & Sons, Incorporated, Newark. Available from: ProQuest Ebook Central. [13 February 2024].

https://medium.com/geekculture/software-development-framework-iterative-model-68584bfad773

A screenshot of a computer

Description automatically generated

1. Test Stand is test management software that enables the development, debugging, and deployment of test systems, it also provides full visibility into testing process and results. (NI, 2024) [↑](#footnote-ref-1)
2. LabVIEW is a graphical programming environment that provides test system development options, such as an intuitive approach to programming, connectivity to any instrument, and fully integrated user interfaces. (NI, 2024) [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)
4. A data lake is a centralized repository that allows for the storage of unstructured and structured data at any scale. (AWS, 2024) [↑](#footnote-ref-4)
5. A data cube refers to a multi-dimensional data structure. Data within a data cube is shown by specific dimensional values. (Mahtani, 2024) [↑](#footnote-ref-5)