# Teil 1

## Einführung

* + 1. IPA

Die IPA ist ein Projekt, welches in einem Zeitrahmen von 80h im Betrieb erledigt wird. Die IPA bedeutet «Individuelle Praktische Arbeit». Es wird dabei geprüft, ob ein Informatik Lehrling über das Wissen und die praktische Umsetzungsfähigkeit besitzt, ein ganzes Projekt alleine abzurollen. Die Informationen dazu erhielten wir vom Berufsbildner und den Experten und Expertinnen über das PkOrg.

* + 1. Zweck des Dokumentes

Der

* + 1. Ziel Publikum

## Projektauftrag gemäss PkOrg

* + 1. Ausgangslage

TsNet wird seit geraumer Zeit zur Testspezifikation und Testdurchführung innerhalb der Siemens Gebäudetechnik genutzt.  
Der Bereich Applications hat ein Excel-Sheet zur Spezifikation von Testschritten erstellt. Das Projekt TsNet V2 besteht aus   
1. einem Definitionsteil (Excel-Template), in dem die Testschritte und die erwarteten Ergebnisse definiert werden sowie  
2. einem Run-Time-Teil, der mit dem Controller kommuniziert und Testschritte vorgibt und Ergebnisse abfragt.  
  
Im Rahmen von Lehrlingsarbeiten und Praktika wird der Definitionsteil komplett neu erstellt. Dazu gibt es bereits ein abgeschlossenes Excel-Projekt, das in dieser IPA mit weiteren Funktionen ergänzt werden soll.  
Das Ziel dieser IPA ist es, die Testspezifikation mit Hilfe von Diagrammen darzustellen, um eine bessere Verifikation und Dokumentation zu erhalten.

* + 1. Detaillierte Aufgabenstellung

Als Basis für die Arbeiten gilt eine Requirement-Specification (Dokumentenpool, IPA Patrick Schöpfer)  
Neben der Spezifikation werden diverse Beispieldateien zur Verfügung gestellt. Diese dienen zur Erläuterung der Spezifikation und zum Test der Funktionen  
Aufgabenstellung: Erstellen der Diagrammfunktion für TsNet V2 gemäss Spezifikation  
- Terminplanung und Projektstatus.   
- Alle 2 Tage ist ein Statusmeeting mit dem Auftraggeber durchzuführen  
- Erstellung der Software  
- Erstellung der Softwaredokumentation. Zielgruppe: SW-Entwickler, die das TsNet-Tool warten, pflegen und weiterentwickeln  
- Erstellen der Testfälle und Durchführen von Tests.  
- Test der Eingabedialoge mit korrekten und falschen Werten  
- Test der Useability der Eingabebedialoge mit Fachperson ausserhalb vom TsNet-Projekt  
- Test aller erstellten Funktionen gegenüber der Spezifikation  
- Test aller erstellten Funktionen gegen Fehlbedienung und fehlerhafte Daten und Dateien  
- Entsprechend den IPA-Regeln ist ein Arbeitsjournal zu führen.  
  
Erwartete Lieferungen  
- Terminpläne und Projektstatus alle 2 Tage und am Ende der IPA  
- Excel-File mit der Erweiterung für die Diagrammfunktion  
- kommentierter Source-Code  
- kommentierte Eingabedialoge  
- Softwaredokumentation:   
Es kann jede grafische Darstellung des Programmablaufs und des Datenflusses verwendet werden,   
sofern sie dazu geeignet ist, einem anderen Entwickler die Weiterarbeit zu ermöglichen.   
Einfache Funktionen müssen nicht grafisch dokumentiert werden, jedoch müssen alle erstellten Funktionen  
aufgelistet und detailliert beschrieben werden  
- Testspezifikation und Dokumentation der Testergebnisse  
- Arbeitsjournal

* + 1. Mittel und Methoden

Entwicklungsumgebung:  
- Standard-PC mit Microsoft Windows 7 oder höher  
- Microsoft Excel 2007. Programmierung in VBA.  
- Erstellung der Dokumentation, der Präsentation und weiterer Dokumente mit Microsoft Office 2007  
- Firmen-Richtlinie Codierungsrichtlinie Visual Basic und weitere Vorgaben in der Spezifikation.

* + 1. Vorkenntnisse

Patrick hat bereits im Vorfeld der IPA im Projekt TsNet gearbeitet, so dass ihm die Entwicklungsumgebung und das Projektumfeld bekannt sind.

* + 1. Vorarbeiten

Mit dieser Funktion soll Patrick eine weitere Funktion zu dem bestehenden TsNet V2 hinzufügen.   
Vorarbeiten für diese Function werden keine gemacht, Patrick fängt bei IPA-Start mit dem Umsetzen an.

* + 1. Neue Lerninhalte

Darstellung von Diagrammen mit Excel. Komplettes Dokumentieren des Arbeitsprozeses und des Programms.

* + 1. Arbeiten in den letzten 6 Monaten

Arbeiten an diversen kleineren Excel VBA Projekten (ca. 2 Monate) PC-Installationen (3 Wochen) Arbeiten am Projekt TsNet v2 mit Excel VBA (ca. 3 Monate).

## Projektorganisation

* + 1. Arbeitsplatz

Ich arbeite seid dem August 2017 in der Abteilung BT CPS R&D ZG CS SAP und im Büro XXX. Mein Büro befindet sich in Zug im Zählerweg 5 auf dem 4. Stock. Mein Arbeitsplatz ist mit 2 Bildschirmen, einer von HP und einer von Fujitsu, versehen. Ausserdem habe ich einen Fujitsu Laptop mit dem ich arbeite.

* + 1. Versionierung

Ich habe alle meine Dateien auf dem Server von Siemens. Ich habe diese Methode gewählt, weil sich mein Code in einem Excel File befindet und ich jedes Modul, jeden Tag, exportieren müsste wenn ich eine andere Methode wähle. Dies wäre sehr viel Zeitaufwändiger.

Dies ist meine Ordnerstruktur:

BILD

Ich habe ausserdem alle Dateien lokal auf meinem Rechner abgespeichert, um ein eigenes Backup zu besitzen.

BILD

* + 1. Personen

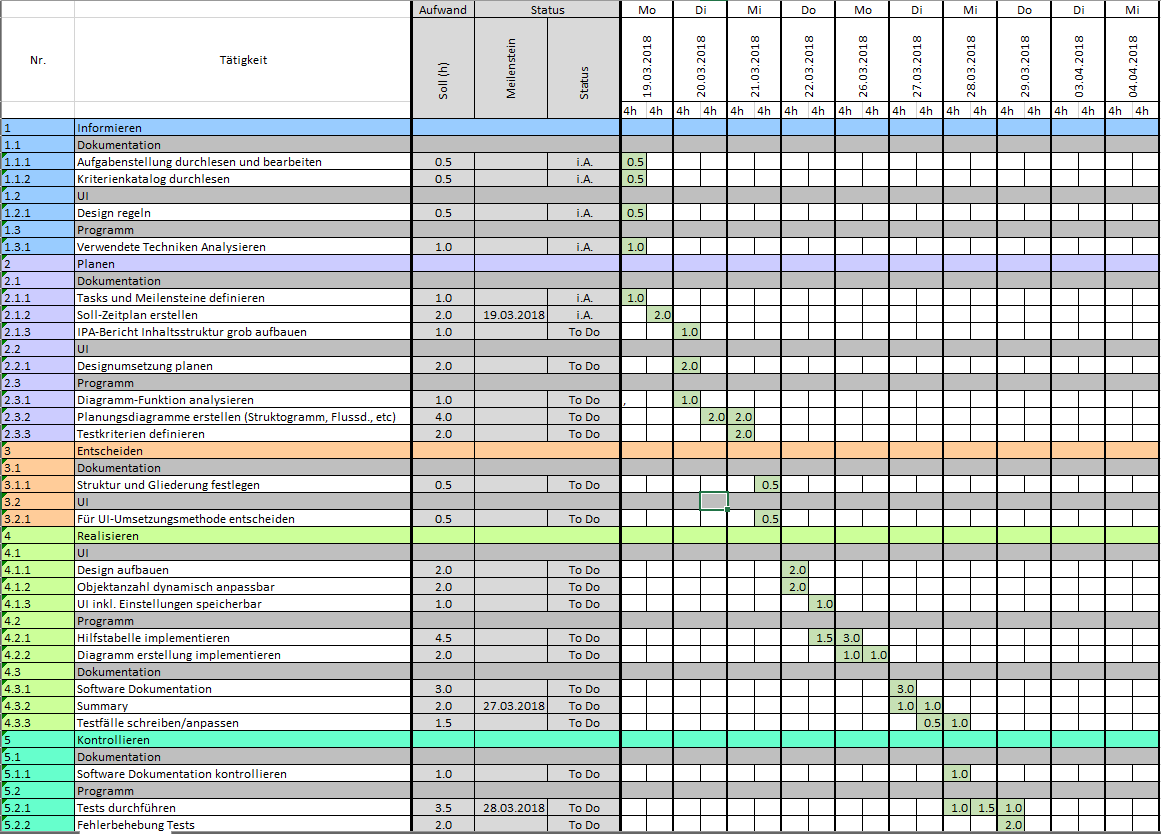
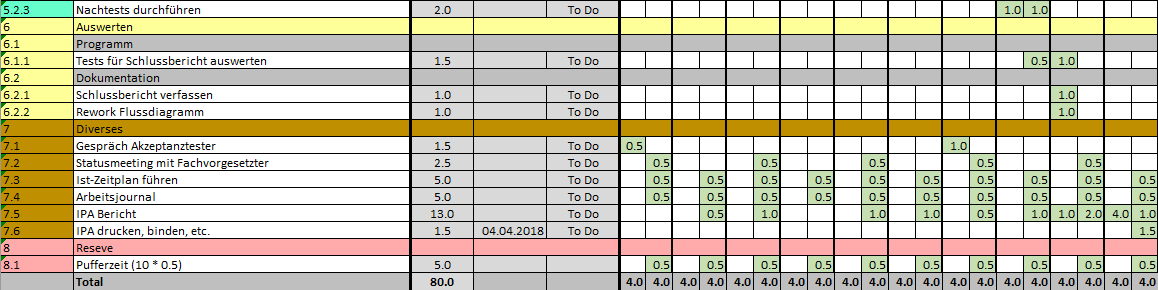
|  |  |
| --- | --- |
| Michael Speckien | Fachvorgesetzter |
| Patrick Schöpfer | Kandidat |
| Anton Kryenbühl | Akzeptanztester |
| Martin Häusler | Berufsbildung |
| Isabel Müller | Nebenexpertin |
| Martina Rakaric | Hauptexpertin |

* + 1. Abteilung BT CPS &D ZG CS SAP
    2. Projektmanagement Methode

Ich habe die…

## Planung

* + 1. Soll-Zeitplan



* + 1. Soll-Ist-Vergleich
    2. Tätigkeiten

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Tätigkeit** | **Beschreibung** |
| **1.1.1** | Aufgabenstellung durchlesen und bearbeiten | Die detailierte Aufgabenstellung online auf PkOrg genau durchlesen und verstehen. Ich werde hier handschriftliche Notizen erstellen um mir einen Überblick über die Aufgaben zu schaffen. |
| **1.1.2** | Kriterienkatalog durchlesen | Ich werde den Kriterienkatalog durchlesen um einen überblick zu bekommen, was wie viele Punke für die Bewertung gibt. Ich werde diese am Ende des Projektes nochmals genau durchlesen. |
| **1.2.1** | Design regeln | Ich werde die Designregeln nochmals nachschauen, da ich ein UI erstellen muss und die Designvorschriften einhalten möchte. |
| **1.3.1** | Verwendete Techniken Analysieren | Ich werde die Techniken die ich vergangenes Jahr gelernt habe und notiert habe nochmals kurz durchgehen und wichtiges markieren. |
| **2.1.1** | Tasks und Meilensteine definieren | Ich werde aufgrund von den gesammelten Informationen Tasks und Meilensteine definieren. Dazu werde ich mir genau überlegen für welche Tätigkeiten ich wie viel Zeit benötige. |
| **2.1.2** | Soll-Zeitplan erstellen | Ich werde die Tasks und Meilensteine in einen Zeitplan umwandeln. Ich werde eine absolute Zeitachse hinzufügen und sie in 4h Blöcke einteilen. |
| **2.1.3** | IPA-Bericht Inhaltsstruktur grob aufbauen | Ich werde planen, was alles in den IPA Bericht gehört und erstelle Titel und Überschriften |
| **2.2.1** | Designumsetzung planen | Ich plane wie ich das Design umsetzen möchte und mache mir Notizen und Prototypen. |
| **2.3.1** | Diagramm-Funktion analysieren | Ich werde planen, wie ich das Diagramm erstellen werde. Dazu schaue ich mir Beispielcodes im Internet an. |
| **2.3.2** | Planungsdiagramme erstellen (Struktogramm, Flussd., etc) | Ich werde Diagramme erstellen, die den Ablauf des Programms zeigen. Das Ziel der Diagramme ist, dass die Implementation schneller und einfacher verläuft. |
| **2.3.3** | Testkriterien definieren | Ich definiere die Testkriterien für die späteren Testfälle. |
| **3.1.1** | Struktur und Gliederung festlegen | Ich gehe in der Dokumentation nochmals die Gliederung und die Struktur durch, die ich geplant habe und entscheide mich für die endgültige Fassung. |
| **3.2.1** | Für UI-Umsetzungsmethode entscheiden | Ich werde mich für eine geeignete Umsetzungsmethode entscheiden. Ich werde hier den Fachvorgesetzten miteinbeziehen, da das Programm ja weiter verwendet wird. |
| **4.1.1** | Design aufbauen | Ich werde zuerst das User Interface implementieren. |
| **4.1.2** | Objektanzahl dynamisch anpassbar |  |
| **4.1.3** | UI inkl. Einstellungen speicherbar | Ich werde implementieren, dass man die gewählten Einstellungen abspeichern kann, damit man das Diagramm immer wieder erstellen kann. |
| **4.2.1** | Hilfstabelle implementieren | Ich werde die Hilfstabelle implementieren. Die Hilfstabelle beinhaltet alle Daten die zum Diagramm beitragen. |
| **4.2.2** | Diagramm Erstellung implementieren | Die Diagramm Erstellung beschreibt den Aufbau des automatisch generierten Diagrammes |
| **4.3.1** | Software Dokumentation | Ich werde eine Software Dokumentation schreiben. Sie ist für andere Entwickler oder z.B. neue Lehrlinge in der Abteilung gedacht, welche möglicherweise mit dieser Funktion arbeiten oder sie erweitern werden. |
| **4.3.2** | Summary | Das Summary ist eine etwa einseitige Zusammenfassung. |
| **4.3.3** | Testfälle schreiben/anpassen | Ich schreibe Testfälle und passe sie an, um mein Programm später testen zu können. |
| **5.1.1** | Software Dokumentation kontrollieren | Ich lese meine Software Dokumentation nochmals durch, überprüfe Rechtschreibfehler etc. |
| **5.2.1** | Tests durchführen | Ich führe meine Tests durch. Ich werde nach den Tests etwa sehen können, bei welchem Stand ich bin. |
| **5.2.2** | Fehlerbehebung Tests | Ich werde mir noch einmal Zeit nehmen, die fehlgeschlagenen Tests zu verbessern. |
| **5.2.3** | Nachtests durchführen | Ich werde die fehlgeschlagenen Tests nochmals durchgehen um zu schauen ob nun alles korrekt ist. |
| **6.1.1** | Tests für Schlussbericht auswerten |  |
| **6.2.1** | Schlussbericht verfassen |  |
| **6.2.2** | Rework Flussdiagramm | Ich werde die Diagramme nochmals anpassen, dass auch der nächste Entwickler ein sauberes Struktogramm und Flussdiagramm hat |
| **7.1** | Gespräch Akzeptanztester | Das sind Gespräche/Besprechungen, welche ich mit dem Akzeptanz Tester halte. |
| **7.2** | Statusmeeting mit Fachvorgesetzter | Ich werde mit dem Fachvorgesetzten jeden 2. Tag ein Statusmeeting halten, damit auch er über meinen Stand bescheid weiss. |
| **7.3** | Ist-Zeitplan führen | Der Ist-Zeitplan ist jeden Tag sauber nach zu führen. |
| **7.4** | Arbeitsjournal | Das Arbeitsjournal beschreibt meine Tätigkeiten, Schwierigkeiten und mögliche Verschiebungen im Zeitplan. |
| **7.5** | IPA Bericht | Diese Dokumentation ist der IPA Bericht |
| **7.6** | IPA drucken, binden, etc. |  |
| **8.1** | Pufferzeit (10 \* 0.5) |  |

* + 1. Meilensteine
    2. (Design)
    3. Struktogramme
    4. Flussdiagramme
    5. Testkriterien definieren
  1. Arbeitsjournal
     1. Zweck
     2. Anwendungsbereich
     3. Aufbau
     4. Arbeitsjournale

# Teil 2

* 1. IPA Kurzfassung
     1. Ausgangssituation

Das Projekt TsNet V2 besteht aus einem Definitionsteil und einem Run-Time-Teil. Das TsNetV2 ist ein Testautomationstool mit dem man Geräte Testen kann. Die Daten, welche man in solch einem Test hat, bestehen aus vielen Zahlen, Bereichen und Booleans. Diese so auszulesen ist sehr mühsam und unübersichtlich. Ziel der IPA ist es, diese Daten grafisch in einem bis mehreren Diagrammen darstellen zu lassen. Die Visualisierung soll helfen die Daten schneller auslesen und verstehen zu können.

* + 1. Umsetzung

Die Anforderungen, welche in der Ausgangssituation genannt wurden, werden im Rahmen der individuellen praktischen Arbeit umgesetzt.

Ich habe einen Zeitplan erstellt, auf dem die absolute Zeitachse, also 10 Tage, definiert ist. Der Zeitplan wurde nach der Projektmanagement Methode IPERKA aufgebaut. Das bedeutet, dass ich die Tätigkeiten in die 6 Phasen aufgeteilt habe. Zusätzlich habe ich die Aufgaben nochmals in 3 Stufen, Ui, Programm und Dokumentation aufgeteilt.

In der Informationsphase habe ich alle wichtigen Informationen gesammelt, welche ich für die Umsetzung brauchte. In der Planungsphase habe ich, wie schon erwähnt, den Zeitplan erstellt. Ich habe ausserdem viel Zeit ins UI investiert, da dies meiner Meinung nach wichtig für die spätere Umsetzung war. Ich habe Struktogramme und Flussdiagramme erstellt. Durch die Struktogramme und der genauen Aufgabenbeschreibung konnte ich später, in der Realisierungsphase alles umsetzen. Ich habe nicht nur das Programmieren selber in die Realisierung genommen, sondern ich habe auch die Software Dokumentation in dieser Phase geschrieben.

In der Kontrollphase habe ich mein Programm, mit den davor aufgeschriebenen Akzeptanztests getestet. Ein Siemens Mitarbeiter hat Usability Tests durchgeführt, woran man erkennen konnte, was am noch nicht ganz Selbstverständlich für den User ist.

* + 1. Ergebnis

F

F

F

F

F

* 1. Realisierung
     1. Entwicklungsumgebung
     2. UI
     3. Programm
     4. Hilfstabelle
     5. Software Dokumentation

//funktionen

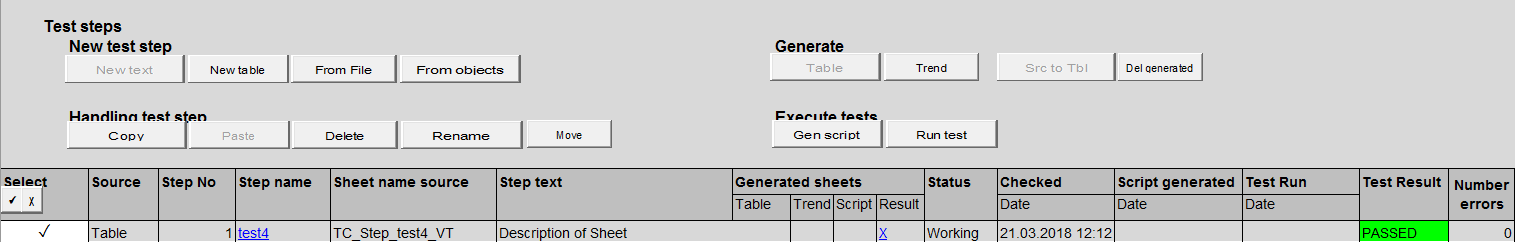
Nachfolgend beschreibe ich alle Funktionen, die ich für mein Projekt implementiert habe. Viele dieser Funktionen, sind Funktionen, die einen gewissen Wert zurückgeben. Ich werde zuerst die Funktionen beschreiben, welche den Trend an sich generieren und anzeigen und danach alle Funktionen, welche die benötigten Werte zurückgeben.

Ich beschreibe nur die wichtigsten Funktionen, da einige Funktionen durch Kommentare im Code selbsterklärend sind.

2.2.5.1

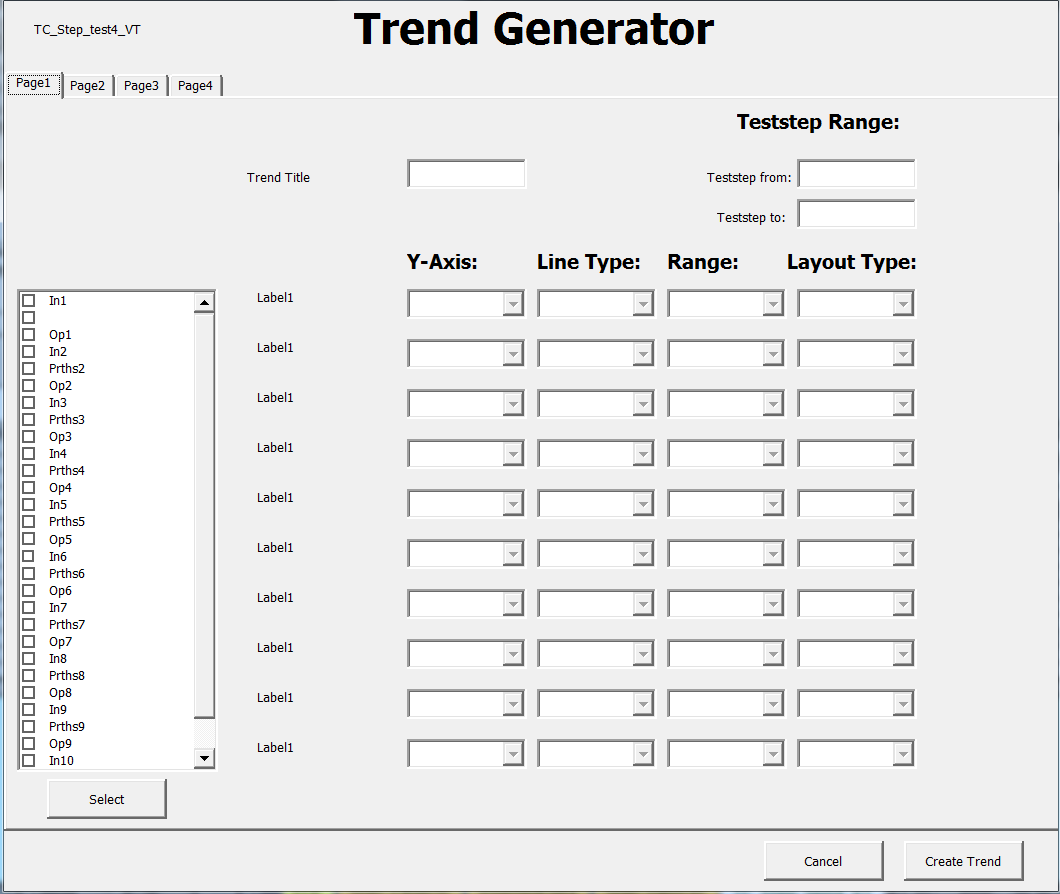
TrendCreatorLoad()

Die Funktion «TrendCreatorLoad()» wird durch das bestätigen des Buttons im Overview ausgeführt. Es wird überprüft, ob ein Test im Sheet Overview selektiert wurde. Der Name des selektierten Sheets wird auf eine Variable «strSheetName» gespeichert. Der Code wechselt zum ausgewählten Sheet und fügt jedes Objekt in die vier Listboxen des UI, bei denen der Datentyp nicht String, Struct oder Array ist. Diese drei Datentypen kann man in einem Diagramm nicht anzeigen und werden deshalb ignoriert. Das UI wird aufgerufen und die verfügbaren Objekte werden angezeigt. Der Name des selektierten Sheets wird ebenfalls eingetragen.



UserForm\_Initialize()

Diese Funktion wird aufgerufen, sobald die Objekte zu den Listboxen hinzugefügt wurden. In dieser Funktion wird das UI hauptsächlich geladen. Durch die Einstellungsmöglichkeiten kann man das Diagramm den wünschen entsprechend anpassen. Diese werden in dieser Funktion den Comboboxen zugewiesen und eingetragen. Die Comboboxen und alle anderen Steuerelemente ausser dem Select Button werden dann blockiert.



ButtonSelect10\_Click()

Diese Funktion gibt es für jede der 4 Seiten. In dieser Funktion werden die 10 ausgewählten Objekte in die rechte Spalte geschrieben. Die Comboboxen werden freigegeben, dass der User die Einstellungen pro Objekt vornehmen kann. Die Textbox für die Überschrift des Diagrammes (ist gleichzeitig der Name von Hilfstabelle) wird freigegeben. Die 2 Textboxen für die Eingabe der Teststeps werden auch freigegeben und mit einem Standard Wert aufgefüllt.

ButtonCreate\_Click()

Diese Funktion wird aufgerufen wenn der Button «Create Trend» gedrückt wird. Diese Funktion ist die Hauptfunktion in meinem Projekt. Zuerst zählt die Funktion wie viele Diagramme zu erstellen sind und schreibt alle Werte, die der User im UI angegeben hat in ein Array.

Als nächstes wird die Hilfstabelle erstellt und benannt. Die Einstellungen und die anderen User Eingaben werden durch das oben erwähnte Array eigespeichert.

Die Funktion wechselt erneut in das TestCase Sheet und sucht in der Reihe «AliasName» nach den selektierten Objekten. Wenn Sie gefunden wurden, werden die Daten, welche weiter unten im Sheet stehen ausgelesen und in ein anderes Array gespeichert. Falls ein Test nicht gefunden werden sollte, wird eine Fehlermeldung ausgegeben. Diese Daten werden in die Hilfstabelle gespeichert. Jeder Wert wird 5 mal eingetragen, da dies die Diagramm Anzeige verschönert. Bei jedem 3. Eintrag des Wertes in der 2. Spalte wird in der 1. Spalte ein «Test» hinzugefügt. Auch dies dient der Darstellung.

Nachdem die Daten hinzugefügt wurden, werden die Einstellungen gelesen und die Daten entsprechend abgeändert. Je nach Einstellung werden verschiedene Funktionen aufgerufen, die für die Änderung der Daten zuständig sind oder Werte zurückgeben, welche wir weiter verrechnen müssen.

-

-

-

-

TrendCreation()

Diese Funktion wird am Schluss des ButtonCreate\_Click() aufgerufen. In dieser Funktion geht es um die Diagramm Erstellung. Zuerst wird das Diagramm erstellt. (dafür wird ein extra Sheet gemacht, indem bis zu 4 Diagramme angezeigt werden können.) Dem Diagramm wird eine Range zugeteilt, um die Daten dafür festzulegen. Die Einstellungen «YAxis» und «LineType» werden abgefragt. Mit der YAxis Einstellung wird bestimmt, ob ein Objekt links, an der Primärachse oder rechts, an der Sekundärachse befestigt wird. Mit der LineType Einstellung kann festgelegt werden, ob die Linie für das Objekt gepunktet oder als ganze Linie angezeigt wird. Danach wird überprüft, ob sich noch irgendwelche «Datenreihen» darin im Diagramm befinden, denn die Datenreihen sollen die Objektnamen anzeigen und nicht «Datenreihen» anzeigen. Die X-Achse bekommt übernimmt die TestSteps als X-Achsen-Wert und die Range und False Daten, welche zusätzlich angezeigt werden ins Diagramm eingefügt.

//bilder

//flussdiagramme

//Information zur bennenung der Elemente

* 1. Kontrollieren
     1. Tests
  2. Auswerten
     1. Schlussbericht
     2. Rework Flussdiagramm
  3. Glossar
  4. Quellen
  5. Anhang