

Entwurfsbeschreibung

SFM-14

Vincent Märkl
Karl Kaiser
Marc Wolff

12.04.2014

Inhaltsverzeichnis

| | |
|---|----|
| 1 Allgemeines | 2 |
| 2 Produktübersicht | 2 |
| 3 Sprachdesign | 3 |
| 3.1 Shapes | 3 |
| 3.2 Validationrules | 4 |
| 4 Serialisierung | 5 |
| 5 Grundsätzliche Struktur- und Entwurfsprinzipien | 5 |
| 6 Struktur- und Entwurfsprinzipien einzelner Pakete | 6 |
| 6.1 Shapes | 6 |
| 6.2 Validierung | 6 |
| 7 Datenmodell | 7 |
| 8 Test | 9 |
| 8.1 Komponenten | 9 |
| 8.2 Graphical User Interface | 9 |
| 9 Glossar | 10 |

1 Allgemeines

Kommt es zu einem Ausfall in der Industrie, bedeutet jede Sekunde, die ein System nicht läuft große Verluste für die Firma. Prozesse, die in Ausfallsituationen einen automatischen Plan vorgeben, welche Personen zu welchem Zeitpunkt kontaktiert werden und was zu welchem Zeitpunkt am System getan wird heißen Eskalationsprozesse.

Das Ziel unseres Projektes ist es, Experten in der Erstellung solcher Eskalationsprozessen eine Möglichkeit zu geben, eben diese von ihnen entworfenen Eskalationsprozesse in für Systeme verständliche MAINT-LA Dokumente zu übersetzen, ohne dass sie sich selbst mit der Sprache auseinander setzen müssen. Das spart zum einen Zeit und Kosten, zum anderen werden auch Fehlerquellen eliminiert, die durch mangelhafte Kenntnis der Sprache entstehen.

2 Produktübersicht

Das Projekt ist ein Stencil in Visio, das Endnutzern auf eine einfache Art und Weise ermöglicht, Eskalationsprozesse mithilfe von verschiedenen Shapes und Connectoren anschaulich zu verwirklichen. Diese werden daraufhin durch die Export-Funktion unseres Projekts in saubere, einheitlich verfasste MAINT-LA Dokumente übersetzt, die daraufhin einfach an den entsprechenden Stellen der Systeme eingefügt und verwendet werden können.

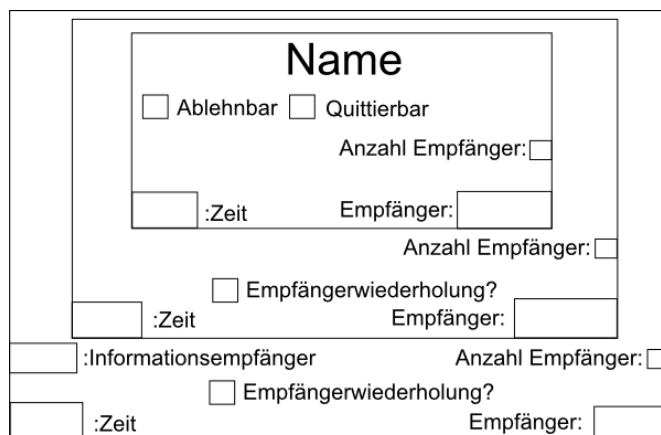
3 Sprachdesign

3.1 Shapes

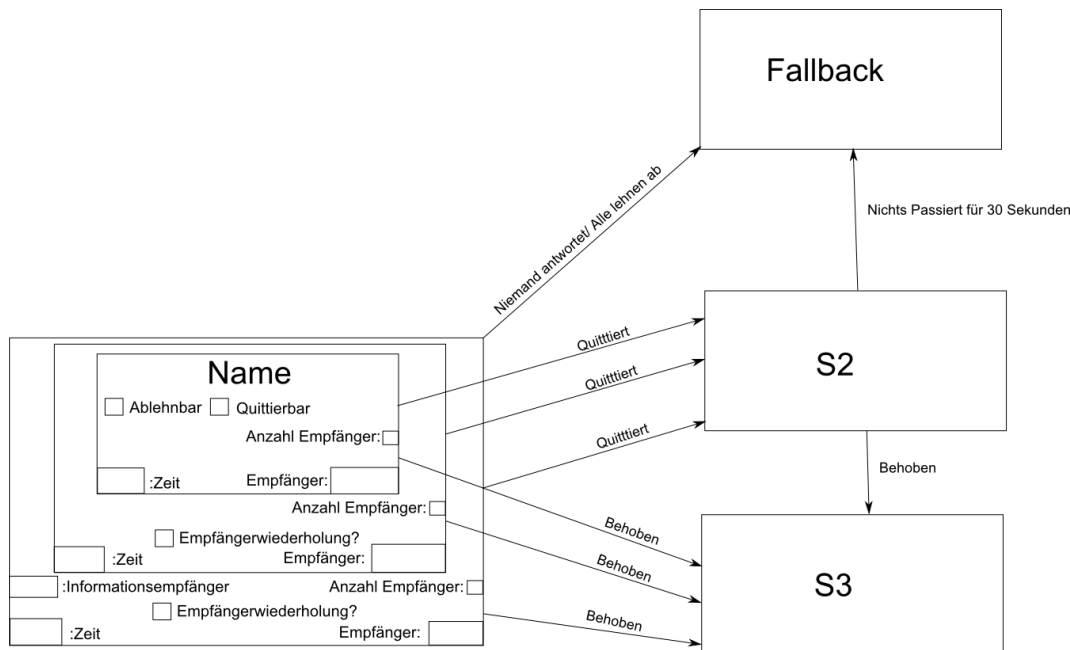
Startzustand:

- dient als Startpunkt des Eskalationsprozesses

normale Zustände:



- kann alleine auftreten oder gruppiert werden
- es gibt einen zentralen Zustand (in der Abbildung durch das in der Mitte liegende Rechteck dargestellt), gibt es nach dem Ablauf der Reviewzeit keine ausgehende Verbindung wird in den Zustand der 2. Schale über, gegebenenfalls noch einen Zustand als 3. schale
- kann die Eigenschaften:ablehnbar,quittierbar,Empfänger,Zeit,Anzahl der Empfänger haben
- kann mit weiteren Schalen weitere Eigenschaften haben

Eskalationsebenen:

- je nach dem wie der Zustand behandelt wird, geht er in die nächste Eskalationsebene über

3.2 Validationrules**Startzustand:**

- es muss genau 1 Startzustand vorhanden sein

normale Zustände:

- es muss mindestens 1 Zustand vorhanden sein
- jeder Zustand muss einen Namen, sowie einen Eventnamen haben
- jeder Zustandsname muss eindeutig sein
- jeder Zustand muss einen Wert für das Reviertiming haben
- der Wert für das Reviertiming muss größer als 0 sein
- jeder Zustand muss eine Empfängergruppe haben
- jeder Zustand muss einen Wert für die Anzahl von Empfängern haben
- der Wert für für die Anzahl von Empfängern muss größer als 0 sein

Verbindungen:

- jede Verbindung muss einen Zustand als Startpunkt haben
- jede Verbindung muss einen Zustand als Endpunkt haben
- Start- und Endpunkt einer Verbindungen dürfen nicht der selbe Zustand sein
- jede Verbindungen muss sich von allen anderen in entweder Startpunkt, Endpunkt oder Typ unterscheiden
- der Startzustand muss mindestens 1 ausgehende Verbindung des Typs "neu" haben
- der Startzustand darf nur Verbindungen des Typs "neu" haben
- der Startzustand darf keine eingehenden Verbindungen haben
- nur der Startzustand darf ausgehende Verbindungen des Typs "neu" haben
- jeder Zustand muss mindestens 1 eingehenden Verbindung haben (bei Zuständen mit mehreren Schalen muss bloß der zentrale Zustand einen eingehenden Verbindung haben)
- Zustände auf der 2. und 3. Schale eines Zustands dürfen keine eingehenden Verbindungen haben
- falls die Nachricht in einem Zustand nicht abgelehnt werden darf, darf dieser keine ausgehende Verbindung des Typ "abgelehnt" haben
- falls die Nachricht in einem Zustand nicht quittiert werden darf, darf dieser keine ausgehende Verbindung des Typ "quittiert" haben

4 Serialisierung

Es wird als Eingabe die Vektoren der einzelnen Shapes und Konnektoren verwendet. Auf diesen läuft im Sinne einer Breitensuche die Serialisierungsfunktion, dabei wird bei der Wurzel(hier der Startpunkt des Eskalationsprozesses) begonnen und die Daten in z.B eine .txt übertragen. Danach werden um doppelte Einträge zu vermeiden die Vektoren des eben bearbeiteten Shapes/Konnektors gelöscht. Als nächstes geht man zum nächsten Konnektor/Shape.

5 Grundsätzliche Struktur- und Entwurfsprinzipien

Alle Shapes die dem Benutzer zur Verfügung stehen sollen können komplett in Visios Editor als Mastershape erstellt werden und anschließend zu dem Stencil des Templates hinzugefügt werden. Visios API enthält bereits alle benötigten Funktionen zum erstellen von Regeln und dem Validieren von Dokumenten anhand dieser Regeln, daher beschränkt sich das implementieren der Validierung auf die Implementierung der Regeln welche eingehalten werden müssen um gültige Maint LA Diagramme zu erstellen.

6 Struktur- und Entwurfsprinzipien einzelner Pakete

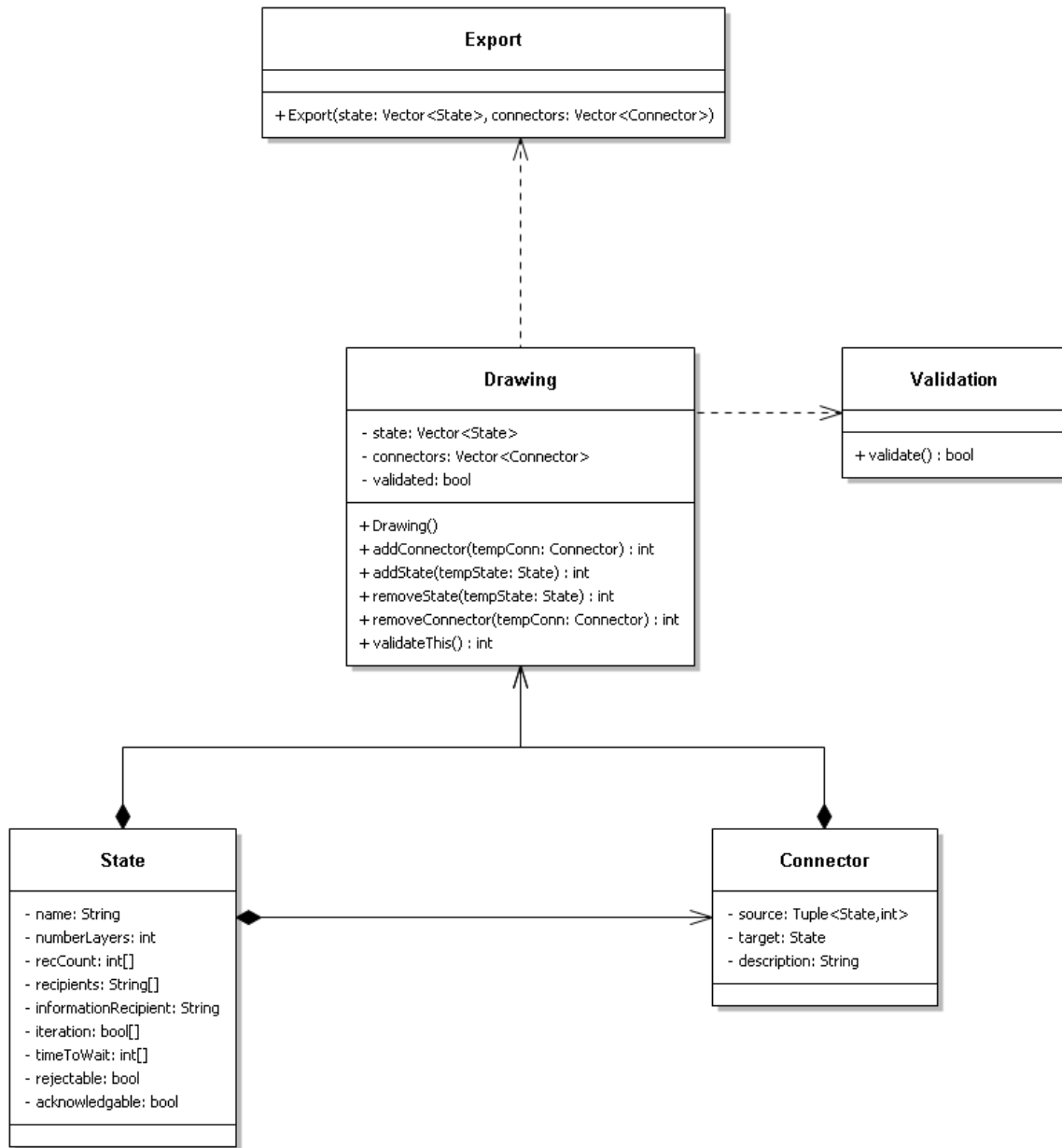
6.1 Shapes

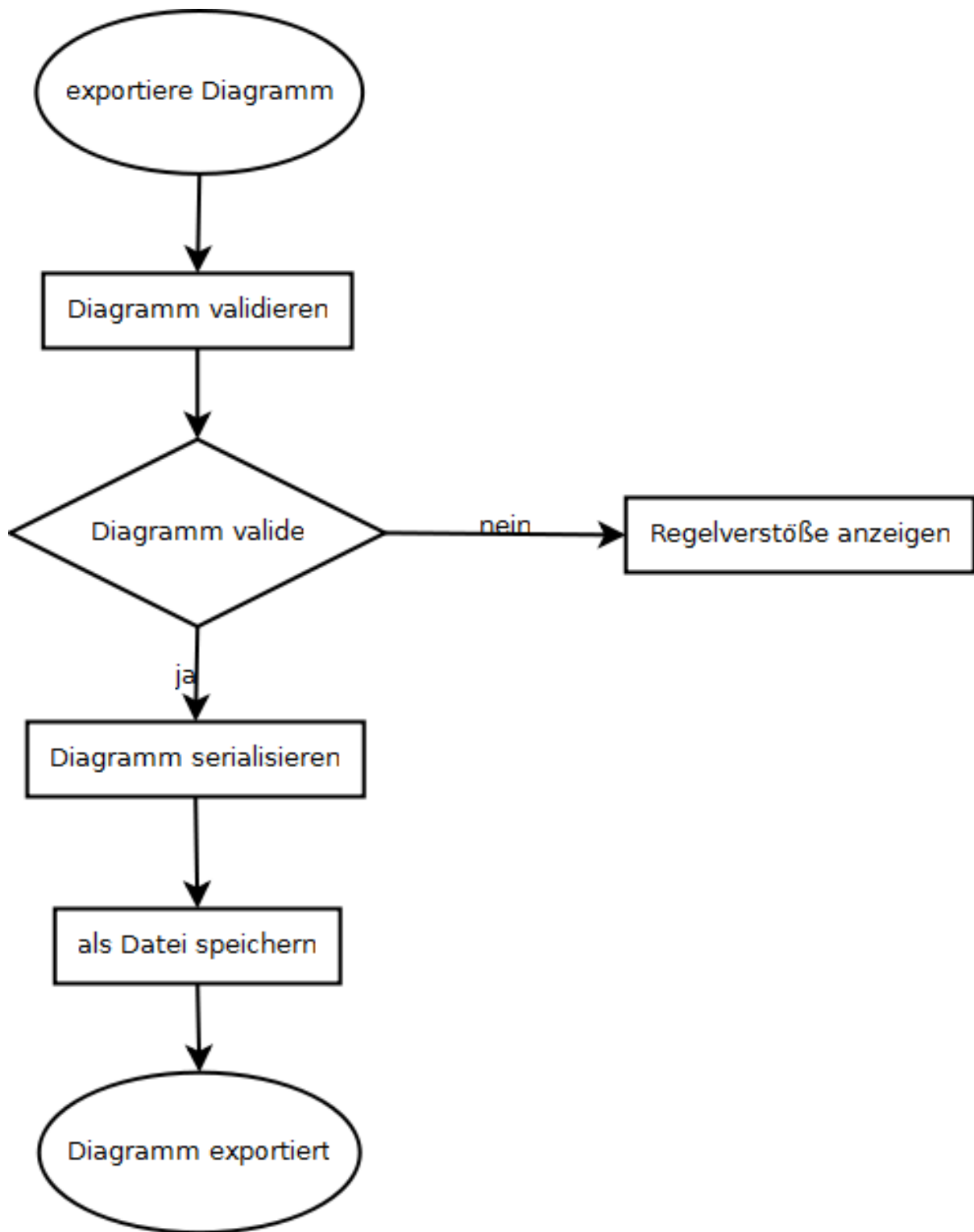
Alle Mastershapes werden durch die grafische Oberfläche des Shape Editors in Visio erstellt. Der Stencil der alle Masterhapes enthält wird als Dokumentstencil im Templatedokument das beim Öffnen des Addins geladen wird abgelegt und steht so dem Nutzer zur Verfügung.

6.2 Validierung

Die Funktionen zum Aufrufen der Validierung und zum Darstellen der Ergebnisse sind bereits durch Visio gegeben. Um eine Regel zu implementieren muss man daher diese nur erstellen und sie dem entsprechenden Ruleset hinzufügen. Die 3 wichtigen Bestandteile einer Regel sind die FilterExpression, die TestExpression und ihre Beschreibung. Die FilterExpression entscheidet auf welche Shapes die Regel die in der TestExpression definiert ist angewendet werden soll. Falls es zu einem Regelverstoß kommt wird dem Benutzer die Beschreibung dieser Regel angezeigt, daher muss diese klar verständlich und einfach zu verstehen sein. Die Ausdrücke der FilterExpression und TestExpression werden als Shapesheetfunktionen Referenz eingefügt mit der normalen Shapesheetsyntax wobei es einige spezielle Funktionen gibt die zur Definition der Regeln zur Verfügung stehen, zu finden unter Link

7 Datenmodell





8 Test

8.1 Komponenten

Zum Testen der einzelnen Klassen und Komponenten, kommt bei uns das Framework NUnit zum Einsatz. Es dient als unit-testing Framework für alle .NET Sprachen und ist somit für C# sehr gut geeignet.

Durch NUnit wird uns ein Framework geboten mit dessen Hilfe jegliche Möglichkeit einer Funktion ausgelöst und überprüft wird. Fehler können schnell zurückverfolgt werden, da genau aufgezeigt wird in welchem Teil der Funktion es zu dem Fehler kam.

Um nach Änderungen erneut und ohne Aufwand, die Funktion auf Korrektheit überprüfen zu lassen, kann mit NUnit (automatisch oder in bestimmten Zeitintervallen) eine vorher angelegte Testmethode ausgelöst werden.

Durch die Verwendung eines solchen unit-testing Frameworks wird erheblich viel Zeit, durch das Wegfallen von manuellen Tests und eine leichteren Fehlerdetektion, gespart. Es wird sichergestellt, dass nach einer Veränderung einer Funktion diese weiterhin einwandfrei funktioniert.

Weiterhin werden Funktionen die bereits in ihrer aktuellen Form getestet wurden, als solche kenntlich gemacht und dokumentiert. So ist klar erkenntlich, selbst für die Personen die vorher nichts mit der Funktion zu tun hatten, ob es Fehler in der Funktion gibt, oder ob diese einwandfrei funktionieren. Somit sind Funktion bzw. Klasse die auf getestete Funktionen zurückgreifen, vor Fehlern aus diesen Funktionen bzw. Klassen geschützt. Entstehen nun Fehler muss man sich nicht durch unzählige andere Klassen lesen. Damit wird unter anderem auch die Qualität des ganzen Projekts gewahrt und eine Erweiterung erleichtert.

8.2 Graphical User Interface

Um die graphische Oberfläche zu testen, müssen wir uns auf das Testen "per Hand", also durch einen oder mehreren Menschen, verlassen. Trotzdem muss auch hier zum Anfang überlegt werden, welche Probleme und Komplikationen auftreten können. Über den gesamten Entwicklungszeitraum sowie mit der Entwicklung neuer Shapes und Funktionalitäten muss dokumentarisch festgehalten werden, inwiefern es bei der Benutzung eben dieser zu Fehlern und Ausnahmeständen kommen kann.

Nach jeder Ergänzung des Projekts sowie in zusätzlichen festen Intervallen sollte jeder einzelne Punkt dieser Sammlung getestet und überprüft werden und Fehler sowie minder schwere Auffälligkeiten festgehalten werden, um das Programm zu verbessern und neue Punkte zu erfassen, die in Zukunft überprüft werden sollten. Die Liste der Tests sieht wie folgt aus und besteht unter anderem aus:

- dem Hinzufügen neuer Shapes auf die Arbeitsoberfläche
- dem Verbinden bestehender Shapes
- dem Löschen von Shapes von der Oberfläche
- dem Übereinanderpositionieren von gleichen und/oder verschiedenen Shapes
- dem Verhalten von Shapes bei unvorhergesehenen langen Bezeichnungen
- dem Verhalten der Shapes und des Plungins bei nicht vorgesehenen Verbindungen

9 Glossar

MS Visio

MS Visio ist ein Visualisierungsprogramm von Microsoft. Es lassen sich mit MS Visio schnell und einfach Diagramme mit den passenden Werkzeugen erzeugen.

Shape

Shapes sind zentrale Elemente einer Zeichnung in MS Visio. Sie sind meist Nachbildungen von Gegenständen oder Komponenten. Es gibt auch Verbindungsshapes die Beziehungen zwischen Shapes verdeutlichen und beschreiben. **Stencil**

Ein Stencil ist eine Sammlung von Visio Shapes. Meist erstellt man für Shapes die häufiger verwendet werden, ein eigenes Stencil.

Validationrules

Diese Regeln dienen zum Validieren des Diagramms. Es wird überprüft, ob das Diagramm sich an eine vorher festgelegte Syntax hält.

Add-in

Ein Add-in ist ein Zusatzprogramm für MS Office Programme. Es können beliebige Funktionen zu dem Programm hinzugefügt werden. Ein Add-in kann in jeder Sprache mit .NET Anbindung geschrieben werden.

Ribbon

Ein Ribbon ist ein graphisches Bedienelement in den MS Office Programmen. Es vereint die Elemente Menüsteuerung, Symbolleisten und Dialoge miteinander. Die Ribbon kann durch ein Add-in verändert werden.

Parser

Ein Parser verarbeitet eine Eingabe, zerlegt diese und setzt sie anschließend in ein neues Format um. In unserem Projekt werden wir eine Eingabe aus dem Visio Format in ein Eskalationsformat umgewandelt.

Eskalationsprozess

Mit Hilfe des Eskalationsprozesses durchläuft ein Störfall die einzelnen Eskalationsstufen, dabei wird auf schnelle und somit kostengünstige Lösung abgezielt. Am Ende des Prozesses steht die erfolgreiche Behebung des Störfalls.

MAINT LA

Maint LA ist die Programmiersprache in der das Störfallmanagement geschrieben wurde.

Connector

Ist ein Bindeglied zwischen zwei oder mehreren Shapes.

Framework

Ein Framework (englisch für Rahmenstruktur) ist ein Programmiergerüst, das in der Softwaretechnik, insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen, verwendet wird. Im allgemeineren Sinne und nicht als dezidiertes Softwareprodukt verstanden, bezeichnet man mit Framework auch einen Ordnungsrahmen.