

# Recherchebericht Gruppe Störfall-Management

Vincent Märkl

Marc Wolff

Lars Ole Lorenz

Klaus Friedrich Schulze

Karl Kaiser

10. Januar 2014

# Inhaltsverzeichnis

<b>1</b>	<b>Begriffe</b>	<b>3</b>
1.1	C++ . . . . .	3
1.2	C# . . . . .	3
1.3	IDE . . . . .	3
1.4	Framework . . . . .	3
1.5	MS Visio . . . . .	4
1.6	Plugin . . . . .	4
1.7	Stencil . . . . .	4
1.8	Shape . . . . .	4
1.9	Eskalationsbeschreibungssprache . . . . .	5
1.10	Eskalationsprozess . . . . .	5
<b>2</b>	<b>Konzepte</b>	<b>5</b>
2.1	Maint LA . . . . .	5
2.2	.NET Framework . . . . .	5
2.3	Störfallmanagementsysteme . . . . .	6
<b>3</b>	<b>Aspekte</b>	<b>6</b>
3.1	Vergleich C++/C# . . . . .	6
3.2	Vergleich IDEs . . . . .	7
3.3	Visio . . . . .	7

# 1 Begriffe

## 1.1 C++

C++ ist eine Weiterentwicklung der Programmiersprache C welche diese vor allem um Klassen erweitert und so objektorientierte Programmierung ermöglicht. Im Vergleich zu vielen moderneren Programmiersprachen zeichnet sich C++ durch maschinennähere Anwendungsmöglichkeiten aus, daher ist C++ in den Gebieten der Betriebssystem- und Treiberprogrammierung und bei allen Anwendungen bei denen Performance im Vordergrund steht (z.B. Videospiele, Simulationen, aufwändige Berechnungen, etc.) steht, weit verbreitet.

## 1.2 C#

C# (ausgesprochen C-Sharp) ist eine von Microsoft entwickelte Programmiersprache welche in der Regel zusammen mit dem .NET-Framework eingesetzt wird. Sie besitzt Funktionen für die objektorientierte Programmierung und zeichnet sich des weiteren durch automatisierte Garbage Collection und starke Typsicherheit aus.

## 1.3 IDE

IDE steht für “integrated development environment” (deutsch: “integrierte Entwicklungsumgebung”) und bezeichnet Programme welche mehrere Funktionen welche für das Programmieren und für Software Entwicklung benötigt werden anbieten. Diese beinhalten in den meisten Fällen Texteditoren, Compiler, Debugger, automatische Codevervollständigung und viele weitere. IDEs ermöglichen es so die Softwareentwicklung effizienter zu gestalten da alle elementaren Funktionen in einer Anwendung enthalten sind.

## 1.4 Framework

Ein Framework ist ein Programmiergerüst das im Rahmen der objektorientierten Softwareentwicklung verwendet wird. Ein Framework ist somit kein Programm, sondern der Rahmen innerhalb dessen ein Programmierer seine Anwendung erstellt, wobei durch das Framework insbesondere auch die Struktur der Anwendung beeinflusst wird(bsp: Model-View-Controller). Ralph

E. Johnson und Brian Foote beschrieben 1998 in ihrem Artikel “designing Reusable Classes” ein Framework wie folgt: “Ein Framework ist eine semi-vollständige Applikation. Es stellt für Applikationen eine wiederverwendbare, gemeinsame Struktur zur Verfügung. Die Entwickler bauen das Framework in ihre eigene Applikation ein, und erweitern es derart, dass es ihren spezifischen Anforderungen entspricht. Frameworks unterscheiden sich von Toolkits dahingehend, dass sie eine kohärente Struktur zur Verfügung stellen, anstatt einer einfachen Menge von Hilfsklassen.”

## 1.5 MS Visio

Ein von Microsoft hergestelltes Visualisierungsprogramm für Windows. Es dient dazu mit Hilfe verschiedener Vorlagen, Werkzeugen und Symbolen Diagramme zu erzeugen. Diese lassen sich via Drag and Drop oder als eigenständige Datei (\*.vsd) in andere Dokumente einbetten. Die Verknüpfung von einzelnen Shapes mit Daten aus beliebigen Datenbanken und Excel-Tabellen, ist die Besonderheit von Visio-Diagrammen. Mit Hilfe eines SharePoint Server können die Diagramme auch ohne installiertes Visio präsentiert werden.

## 1.6 Plugin

Ein Softwaremodul, welches in eine größere Softwareanwendungen integriert wird um deren Funktionalität zu erweitern. Es wird eine übergeordnete Plattform für die Ausführung eines Plug-ins benötigt. Welche wiederum die benötigten Schnittstellen zur Integration des Plug-Ins zur Verfügung stellt. Im Gegensatz zu Add-Ons werden eigene Bibliotheken zur Verfügung gestellt. Ein elementarer Bestandteil bei der Entwicklung von Plug-ins sind die Schnittstellen. Sie werden aus Sicht der Hersteller von Software-Produkten definiert. Das Prinzip Inversion of Control wird häufig für die Steuerung von Plug-ins eingesetzt. (Damit wird ein Umsetzungsparadigma gekennzeichnet, das mit der objektorientierten Programmierung entstanden ist, und in Zusammenhang mit Frameworks die Steuerung eines Programmteils beschreibt.) Generell sind die Schnittstellen von Plug-ins nicht vereinheitlicht.

## 1.7 Stencil

Stencil's (engl. für Schablone). Eine Schablone um beliebig oft ein bestimmtes Muster, Umriss etc. zu übertragen. Stencil's sind bearbeitbare Schablone in

Visio und enthalten mehrere verwandter Shapes.

## 1.8 Shape

Ein Shape (engl. für Form, Gestalt, Umriss) ist ein grafisches Objekt, welches auf dem Computerbildschirm dargestellt werden kann. Aus funktionstechnischer Sicht ähneln sie Sprites. Sie werden jedoch im Gegensatz zu Sprites nicht autonom von der Grafikhardware sondern von der CPU des Computers dargestellt. Die CPU kopiert den Shape an passender Stelle in die Bitmap. Bei Visio-Shapes handelt es sich um vordefinierte Objekte, die auf das Zeichenblatt gezogen werden. Sie dienen als die Bausteine des Diagramms. Zusätzlich können zu jedem Shape Daten hinzugefügt werden, außerdem gibt es Shapes mit besonderem Verhalten. Diese besondere Funktion werden mit dehnen, verschieben oder via klick der rechten Maustaste auf den gelben Steuerpunkt des Shapes abgerufen z.B. kann das Personenshape gedehnt werden, damit mehrere Personen angezeigt werden.

## 1.9 Eskalationsbeschreibungssprache

Mit Hilfe einer Eskalationsbeschreibungssprache können Eskalationsprozesse entworfen und umgesetzt werden. Es gibt verschiedene Eskalationsbeschreibungssprachen, in unserem Projekt werden wir Maint LA verwenden.

## 1.10 Eskalationsprozess

Eskalationsprozesse beschreiben wie auf einen gegebenen Störfall reagiert werden muss und wie vorgegangen wird falls er auf die vorhergesehene Weise nicht behoben werden kann (Eskalation in die nächste Ebene des Eskalationsprozess). Eskalationsprozesse bieten so eindeutige Vorgehensweisen wie mit Störfällen aller Art umgegangen werden muss.

# 2 Konzepte

## 2.1 Maint LA

Maint LA ist eine Eskalationsbeschreibungssprache. Ein Eskalationsprozess wird in Main LA mit einem "Level-System" beschrieben. Dies ermöglicht

verschiedene Level, je nach dringlichkeit und Status des derzeitigen Levels, aufzurufen. Es werden die Level gewechselt, bis die Bedingung die für den Erfolg festgelegt wurde, erfüllt wurde. In der Syntax von Main LA werden Kommentare mit einer “#” am Anfang der Zeile repräsentiert.

## 2.2 .NET Framework

.NET-Framework ist eine von Microsoft entwickelte Entwicklungsplattform, auf der unter Verwendung von C# oder Visual Basic und umfangreicher Klassenbibliotheken Programme für die meisten Microsoft-Plattformen (z.B. Windows, Windows Phone etc.) entwickelt werden können. Ausserdem stellt .NET eine als Common Language Runtime bezeichnete Laufzeitumgebung bereit, die Dienste für die Erleichterung des Entwicklungsprozesses bereitstellt, wie Speicherverwaltung, Threadverwaltung und Remoting. Weiterhin werden Versionskonflikte und Konflikte bei der Softwarebereitstellung auf ein Minimum beschränkt. Die Common Language Runtime verwaltet zum Ausführungszeitpunkt Code. Sie erzwingt strikte Typsicherheit und andere Formen der Codegenauigkeit, mit denen Sicherheit und Zuverlässigkeit unterstützt werden. Die gesamte Kommunikation der Anwendungen ist auf Industriestandards aufgebaut um eine Integration von Code, der auf .NET Framework basiert, in jeden anderen Code zu gewährleisten. Die Klassenbibliothek ist eine umfassende, objektorientierte Sammlung wiederverwendbarer Typen für die Entwicklung verschiedenster Anwendungen, von der Befehlszeile bis hin zu XML-Webdiensten.

## 2.3 Störfallmanagementsysteme

In Produktionsabläufen gibt es oft viele verschiedene Fehler die an verschiedenen Stellen auftreten können. Da Unterbrechungen die Produktivität beeinträchtigen und so den Umsatz verringern ist es wichtig Systeme bereit zu haben die mit diesen umgehen können. Störfallsysteme dienen dazu auftretende Störfälle möglichst effektiv und schnell zu beheben. Zum Beispiel kann es Mitarbeitern mitteilen wer welchen Teil der Problembehebung übernimmt, welche Bereiche der Produktion für wie lange unterbrochen werden, etc.

## 3 Aspekte

### 3.1 Vergleich C++/C#

C# besitzt eine Vielzahl von Features welche die Softwareentwicklung im Vergleich zu C++ erleichtern, unter anderem Garbage Collection, Typensicherheit, nicht vorhandene Trennung von Deklaration und Definition und somit das Wegfallen von Headerfiles, Properties und viele weitere. Die Vorteile von C++ sind vor allem der allgemein höhere Freiheitsgrad (z.B. manuelles Speichermanagement) wodurch besser Möglichkeiten für Low-Level Programmierung gegeben sind, was allerdings gleichzeitig ein Nachteil sein kann da es programmieren oft komplizierter macht, und zum anderen höhere Performance. Die Performance Vorteile kommen zum großen Teil daher das C++ üblicherweise nicht die Common Language Runtime aus dem .NET Framework benutzt. Da dieses bei diesem Projekt allerdings verwendet wird gäbe es nur einen geringen Performance Vorteil bei der Verwendung von C++ und da diese bei dem Projekt sowieso nicht im Vordergrund steht kann man diesen Vorteil von C++ in unserem Fall ignorieren. Im Endeffekt biete C# viele Features mit denen komfortabler und einfacher programmiert werden kann und die üblichen Vorteile von C++ sind in unserem Fall kaum ein Faktor.

### 3.2 Vergleich IDEs

Unter Windows bietet Micorsofts Visual Studio im allgemeinen die beste Umgebung zur Softwareentwicklung da es sämtliche wichtige Features welche man von einer IDE erwartet enthält, z.B. einen sehr guten Debugger, und generell ein benutzerfreundliches GUI besitzt. Es stehen einige Versionen zur Auswahl, wobei es jedoch für uns keinen Grund gibt nicht die neuste 2012 Version zu verwenden.

### 3.3 Visio

Wie bereits erwähnt, handelt es sich bei Visio um eine Visioalisierungssoftware von Microsoft. Wenn man Software einsetzt muss man sich immer fragen, welche Version unter den eigenen spezifischen Aspekten, die sinnvollste ist. In unserem Fall stehen die Visioversionen von 2010 und 2013 zur Auswahl. Die 2013er Version hat gegenüber der 2010er einige Verbesserungen und Neuerungen. Ausschlaggebend für das Projekt sind die Downkompatibilität und

die längere Unterstützung von Seiten des Softwareherstellers. Bei der 2013er Version schlagen zwar auf Grund der Aktualität höhere Lizenzgebühren zu Buche, die aber durch die erstmals vorhandene Downkompatibilität zu seinen Vorgängerversionen teilweise abgefangen wird. Denn es muss nur auf den Rechnern auf denen das Plug-in eingesetzt werden soll, das neue Visio installiert werden. Somit ist es allen Firmen, die Visio einsetzen, möglich das Plug-in zu nutzen. Da unter .NET entwickelt werden soll, drängt sich VSTO (Visual Studio Tools for Office) in den Vordergrund. Denn VSTO macht es möglich die Officeanwendungen mit Hilfe von Visual Basic und Visual C# zu erweitern. Beim Erstellen des Plug-ins muss neben den allgemeinen Konventionen darauf geachtet werden, dass die Shapes immer mit ihrem Identifier angesprochen werden. Denn Visio versteht die Shapes mit der Kennung Tab.X. Das X ist hierbei eine Nummer die automatisch hochgezählt wird. Später ist es immer noch möglich den Namen, aber nicht die Kennung, des Shapes zu ändern, daher ist es wichtig die Shapes über ihre Kennung anzusprechen.