

## Beraterprofil

---

**Name:** Peter Most

---

**Kontakt Daten:** PERA Software Solutions GmbH  
Ulrich-Nanshaimer-Straße 29  
85652 Landsham

Internet: [www.PERA-Software.com](http://www.PERA-Software.com)  
E-Mail: [PMost@PERA-Software.com](mailto:PMost@PERA-Software.com)  
Telefon: 089/909 692 48  
Mobil: 0176/240 254 82

---

**Beruf:** Senior Softwareentwickler

---

**Bevorzugter Einsatzort:** Großraum München

---



**Ausbildung:**

- Softwareentwickler (Selbststudium)
- Bauzeichner
- Mittlere Reife

---

**Sprachen:**

- Deutsch (Muttersprache)
- Englisch (sehr gut in Wort und Schrift)

---

**Schwerpunkte:**

- C++17, Java SE 14, C# 7.0
- Systemprogrammierung (Windows/Linux)
- Anwendungsentwicklung (Windows/Linux)
- Unit Tests
- Refactoring
- Test Driven Development (TDD)
- Multithread-/Multicore-Programmierung
- Netzwerkprogrammierung, Interprocess-Kommunikation
- Programmiersprachen-Interoperabilität
- Plattformübergreifende Implementierung

---

**Betriebssysteme:**

- Windows 10
- Linux 19.10

---

**Methoden:**

- Agile Entwicklung (Scrum)
- Objektorientierte Analyse-, Design- und Programmierung (OOA, OOD, OOP)
- Test Driven Development (TDD)
- Continuous integration (CI)
- Trunk based development

---

**Git Repositories:** <https://github.com/petermost>

---

**Sonstiges:** Ich bin begeisterter Softwareentwickler, der sich durch Fachliteratur ständig weiterbildet.  
→ <http://www.pera-software.com/html/books/>

**Stand des Profils:** Januar 2021  
→ [http://www.pera-software.com/html/projects/Peter\\_Most\\_Profil.pdf](http://www.pera-software.com/html/projects/Peter_Most_Profil.pdf)

## Technologien

	C++	Java	C#
<b>Erfahrung:</b>	Hervorragend (1990)	Sehr gut (2005)	Gut (2011)
<b>Versionen:</b>	<ul style="list-style-type: none"><li>• C++ 17</li><li>• GCC C++ 8.x</li><li>• Visual C++ 2017/CLI</li></ul>	<ul style="list-style-type: none"><li>• Java SE 14</li><li>• Java EE 7</li></ul>	<ul style="list-style-type: none"><li>• C# 7.0</li></ul>
<b>Language interoperability:</b>	<ul style="list-style-type: none"><li>• JNA</li><li>• P/Invoke</li><li>• C++/CLI</li></ul>	<ul style="list-style-type: none"><li>• JNA</li><li>• ComfyJ</li><li>• Jython</li></ul>	<ul style="list-style-type: none"><li>• P/Invoke</li><li>• C++/CLI</li></ul>
<b>Standard Libraries</b>	<ul style="list-style-type: none"><li>• Boost 1.67.0</li><li>• STL</li></ul>	<ul style="list-style-type: none"><li>• JDK 14</li></ul>	<ul style="list-style-type: none"><li>• .NET Framework 4.7.2</li><li>• .NET Core 2.2</li></ul>
<b>GUI Frameworks:</b>	<ul style="list-style-type: none"><li>• Qt 5</li><li>• MFC</li><li>• wxWidgets</li></ul>	<ul style="list-style-type: none"><li>• Swing</li><li>• JavaFX</li></ul>	<ul style="list-style-type: none"><li>• WPF</li><li>• WinForms</li></ul>
<b>Unit Tests:</b>	<ul style="list-style-type: none"><li>• Google Test</li><li>• Google Mock</li><li>• Qt Test</li><li>• Boost.Test</li><li>• CppUnit</li><li>• NUnit</li></ul>	<ul style="list-style-type: none"><li>• JUnit</li><li>• Mockito</li><li>• EasyMock</li><li>• PowerMock</li></ul>	<ul style="list-style-type: none"><li>• xUnit</li><li>• NUnit</li><li>• Moq</li></ul>
<b>Networking:</b>	<ul style="list-style-type: none"><li>• Boost.Asio</li><li>• Qt Sockets</li><li>• BSD Sockets</li><li>• ZeroMQ</li><li>• AMQP-CPP</li><li>• Named Pipes</li></ul>	<ul style="list-style-type: none"><li>• java.nio</li></ul>	<ul style="list-style-type: none"><li>• System.Net.Sockets</li><li>• RabbitMQ</li></ul>

## Technologien

	C++	Java	C#
<b>Multithreading / Multi-Core:</b>	<ul style="list-style-type: none"> <li>• Boost.Threads</li> <li>• Parallel Patterns Library (PPL)</li> <li>• Qt Threads/Process</li> <li>• POSIX Threads</li> <li>• Windows Threads</li> </ul>	<ul style="list-style-type: none"> <li>• Java Threads</li> </ul>	<ul style="list-style-type: none"> <li>• System.Threading</li> <li>• Task Parallel Library (TPL)</li> </ul>
<b>Logging:</b>	<ul style="list-style-type: none"> <li>• Boost.Log</li> <li>• Boost.Format</li> <li>• NLogC</li> </ul>	<ul style="list-style-type: none"> <li>• Log4j</li> <li>• LogBack</li> </ul>	<ul style="list-style-type: none"> <li>• NLog</li> </ul>
<b>Sonstige:</b>	<ul style="list-style-type: none"> <li>• Boost.Signals</li> <li>• Boost.Variant</li> <li>• CppAidKit</li> </ul>	<ul style="list-style-type: none"> <li>• RenderSnake, jsoup</li> <li>• Launch4j</li> <li>• Google Guava</li> <li>• Apache.Commons.Lang</li> <li>• Apache.Commons.Net</li> <li>• Apache.Common.CSV</li> <li>• JavaAidKit</li> </ul>	
<b>Datenbanken:</b>	<ul style="list-style-type: none"> <li>• SQLite3</li> <li>• Qt SQL</li> </ul>	<ul style="list-style-type: none"> <li>• Derby, Oracle</li> <li>• JPA</li> </ul>	<ul style="list-style-type: none"> <li>• SQLite3</li> <li>• System.Data.SQLite</li> </ul>
<b>XML:</b>	<ul style="list-style-type: none"> <li>• Qt XML</li> </ul>	<ul style="list-style-type: none"> <li>• JAXP</li> </ul>	
<b>Buildtools:</b>	<ul style="list-style-type: none"> <li>• CMake</li> <li>• Conan</li> <li>• MSBuild</li> <li>• Jenkins</li> <li>• Makefile</li> </ul>	<ul style="list-style-type: none"> <li>• Maven</li> <li>• Eclipse</li> </ul>	<ul style="list-style-type: none"> <li>• NuGet</li> <li>• MSBuild</li> <li>• dotnet build</li> <li>• CMake</li> <li>• Jenkins</li> </ul>
<b>Versionsverwaltungen:</b>	<ul style="list-style-type: none"> <li>• Git, Jazz, ClearCase</li> <li>• GitHub, GitLab</li> <li>• Visual Studio Code</li> </ul>	<ul style="list-style-type: none"> <li>• Git, Jazz, ClearCase</li> <li>• GitHub, BitBucket</li> <li>• Visual Studio Code</li> <li>• Mercurial</li> </ul>	<ul style="list-style-type: none"> <li>• Git, Jazz, ClearCase</li> <li>• GitLab</li> <li>• Visual Studio Code</li> </ul>
<b>Issuetracker:</b>	<ul style="list-style-type: none"> <li>• Jira</li> <li>• Jazz</li> <li>• Trac</li> </ul>	<ul style="list-style-type: none"> <li>• Jira</li> <li>• Jazz</li> <li>• Bugzilla</li> </ul>	<ul style="list-style-type: none"> <li>• Jira</li> <li>• Jazz</li> </ul>
<b>IDEs:</b>	<ul style="list-style-type: none"> <li>• Visual Studio 2019</li> <li>• Resharper</li> <li>• Visual Assist</li> <li>• Qt Creator</li> </ul>	<ul style="list-style-type: none"> <li>• Eclipse</li> <li>• NetBeans</li> <li>• IntelliJ</li> </ul>	<ul style="list-style-type: none"> <li>• Visual Studio 2017</li> <li>• Resharper</li> </ul>

## Technologien

---

	<b>C++</b>	<b>Java</b>	<b>C#</b>
<b>Debugger:</b>	<ul style="list-style-type: none"><li>• Visual Studio</li><li>• Qt Creator</li><li>• GDB (TUI)</li></ul>	<ul style="list-style-type: none"><li>• Eclipse</li></ul>	<ul style="list-style-type: none"><li>• Visual Studio</li><li>• Visual Studio Code</li></ul>
<b>Code Coverage</b>		<ul style="list-style-type: none"><li>• EclEmma</li></ul>	<ul style="list-style-type: none"><li>• dotCover</li></ul>
<b>Profiler:</b>	<ul style="list-style-type: none"><li>• VTune</li></ul>		<ul style="list-style-type: none"><li>• dotTrace</li></ul>

---

## Projekte

---

**Projekt:**

"GUARDUS"  
(Engine)

**Beschreibung:**

Software für Analyse und  
Steuerung in der Produktion.

**Kunde:**

Carl Zeiss MES Solutions  
GmbH

**Laufzeit:**

09.2020 – heute

**Aufgaben:**

- Modernisieren und Refaktorisieren einer C++ Legacy-Software.
- Umstellung des Buildsystems von Visual Studio auf CMake.
- Einführen von UnitTests mit Google Test.
- Mergen von 'Copy & Paste' Sourcecode.

**Technologien:**

- C++17, Google Test/Mock
- CMake, Visual Studio 2019, Sourcetrail
- Git
- Windows 10

---

**Open Source Projekt:**

"FritzBoxCallMonitor"

**Beschreibung:**

Empfangen und Anzeigen  
von FritzBox Call-  
Information.

**Kunde:**

PERA Software Solutions  
GmbH

**Laufzeit:**

11.2016 – heute

**Aufgaben:**

- Design und Entwicklung eines C++/Qt Programms.

**Funktionalität:**

- Empfangen und Parsen der Call-Information via TCP/IP.
- Parsen eines exportierten XML-Telefonbuchs.
- Anzeigen des Anrufernamens als System Tray Notification.
- Anzeigen von Statusmeldungen in einer Liste.
- Testen der Funktionalität mit einem „FRITZ!Box-Simulator“.

**Technologien:**

- C++17, Qt 5, Qt Sockets, Qt XML, Google Test, CppAidKit
- CMake, Qt Creator, Sourcetrail
- Git, GitHub.com
- Linux

**Homepage:**

<http://www.pera-software.com/html/software/fritzbox-callmonitor/fritzbox-callmonitor.html>

---

---

**Projekt:**

"Common Sequencer"

**Beschreibung:**

Automatisierung von  
Produktionstests

**Kunde:**

Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**

10.2019 – 05.2020

**Aufgaben:**

- Weiterentwicklung eines C++ "ExecutionEnvironments" zum Laden und Starten von C# "BuildingBlock-Klassen".
- Erweiterung des C# Kodegenerators mit T4/C# Templates.

**Funktionalität:**

- .NET Core Hosting mit C++
- Kodengenerierung mit C#/T4

**Technologien:**

- C++17, Boost, Google Test/Mock
- CMake, Conan, Visual Studio 2017, Sourcetrail
- C# 7.0, .NET Core 2.2, xUnit
- NuGet
- Git, Jira
- Windows 10, Linux (Kubuntu 18.04)

---

**Projekt:**

"Server Based Testing"  
(Data Acquisition Service)

**Beschreibung:**

Microservicebasiertes Testen  
von Mobilgeräten

**Kunde:**

Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**

04.2019 – 09.2019

**Aufgaben:**

- Design und Entwicklung eines C++ Highspeed-Datenakquirierungsdienstes.
- Vereinfachen bzw. modernisieren der CMake-Skripte.
- Erweitern von Python-Skripten zur Performanzmessung.

**Funktionalität:**

- Empfangen und parsen der JSON-basierten "JobTickets".
- Extrahieren des Dateipfades und cachen der Binärdaten.
- Versenden der gecachten Binärdaten mit ZeroMQ.

**Technologien:**

- C++17, Boost, Google Test/Mock
- RabbitMQ, AMQP-CPP, ZeroMQ, libcurl, Docker
- CMake, Qt Creator, Sourcetrail
- Git, Jira
- Linux (Kubuntu 18.04), Debian

---

**Projekt:**

"QuickStep"  
(TestConsoleRunner)

**Beschreibung:**

Ausführen von Google- bzw.  
NUnit Tests

**Kunde:**

Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**

03.2019

**Aufgaben:**

- Design und Entwicklung eines C# Crossplatform "TestConsoleRunners".

**Funktionalität:**

- Erkennen anhand der Dateisignatur ob ein Google Test bzw. NUnit Test vorliegt.
- Ausführen des Tests und Erstellen einer XML-Ergebnisdatei zur weiteren Verarbeitung in Jenkins.

**Technologien:**

- C# 7.0, .NET Core 2.2, NUnit
- Jenkins, Visual Studio 2017
- Git, Jira
- Windows 10, Linux (Kubuntu 18.10)

---

**Projekt:**

"QuickStep"  
(Linuxportierung)

**Beschreibung:**

Portierung eines Windows  
Programms nach Linux

**Kunde:**

Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**

09.2018 – 02.2019

**Aufgaben:**

- Analyse des Portierungsaufwandes für den C++ Kode.
- Einführen von UnitTests mit Google Test.
- Ersetzen der nicht standardkonformen Visual C++ Sprachkonstrukte.

**Technologien:**

- C++14, Boost, Google Test/Mock
- CMake, Visual Studio 2017, Qt Creator, Sourcetrail
- Git, ClearCase, Jira
- Windows 10, Linux (Kubuntu 18.04)

---

**Open Source Projekt:**

"HomePageGenerator"

**Beschreibung:**

Erzeugen von statischen  
HTML Seiten.

**Kunde:**

PERA Software Solutions  
GmbH

**Laufzeit:**

01.2015 – heute

**Aufgaben:**

- Design und Entwicklung eines Java Webseitengenerators für PERA-Software.com.
- Design und Entwicklung eines Java FTP-Uploaders.

**Funktionalität:**

- Laden aller Page-Klassen mit der Guava-ClassPath-Klasse.
- Umwandeln der Page-Klassennamen in "Pretty URLs".
- Erzeugen der HTML Seiten mit HtmlCanvas, Renderable u.a.
- Upload der generierten Seiten mit FTP.

**Technologien:**

- Java 14, RenderSnake, FreeMarker, jsoup, JUnit 5, Apache Commons (Lang, Net, CSV), Google Guava, JavaAidKit
- HTML, CSS
- Maven, Eclipse, Sourcetrail
- Git, Bitbucket.org
- Linux

---

**Projekt:**

"TeDiSPlus"

**Beschreibung:**

Disponierung von  
Technikern

**Kunde:**

BWI GmbH

**Laufzeit:**

04.2017 – 08.2018

**Aufgaben:**

- Weiterentwicklung einer Java EE Backendanwendung.
- Erstellen und Erweitern von Unit-Tests.

**Funktionalität:**

- Erstellen von SQL-Abfragen mit JPA.
- Empfangen bzw. Senden der JSON Frontend-Daten.

**Technologien:**

- Java 8, Java EE 7, JUnit 4, JUnit 5, Mockito
- GlassFish (Payara), Derby- bzw. Oracle-Datenbank
- Maven, IntelliJ, Eclipse, NetBeans, Sourcetrail
- Mercurial, Jira
- Windows 7

---

**Projekt:**

"CMWmars"  
(PCAP Dekodierung)

**Aufgaben:**

- Design und Entwicklung eines C++/C# Programms zur PCAP-Dekodierung.

**Beschreibung:**

Analyse und Visualisierung von Mobilfunktestdaten.

**Funktionalität:**

- Dekodieren der PCAP-Daten mit TShark (Wireshark).
- Starten des TShark Programms mit QProcess.
- Senden der PCAP Daten mit Pipes/Channels.
- Empfangen der PDML Daten mit Pipes/Channels.
- Parsen der PDML Daten mit QXmlStreamReader.

**Kunde:**

Rohde & Schwarz GmbH & Co. KG

**Technologien:**

**Laufzeit:**

01.2016 – 08.2016

- C++ 11, C++/CLI, Qt 5, Qt Core, Qt XML, Boost, STL
- C# 4.0, NUnit
- Visual Studio 2013
- Jazz

---

**Projekt:**

"CMWmars"  
(LoggingServer)

**Aufgaben:**

- Design und Entwicklung eines C++ LoggingServers.
- Design und Entwicklung einer C++ DLL mit einer Java bzw. C# kompatiblen API.

**Beschreibung:**

Analyse und Visualisierung von Mobilfunktestdaten.

**Funktionalität:**

- Asynchrone Socketkommunikation mit Boost.Asio und Boost.Thread.
- Senden der Live-Daten im TLV/Boost.Serialization Format.
- Interne Entkopplung/Kommunikation mit Slot/Signals.
- Marshallen der C++ Daten nach C#.

**Kunde:**

Rohde & Schwarz GmbH & Co. KG

**Technologien:**

**Laufzeit:**

01.2011 – 12.2015

- C++ 11, C++/CLI (Bridge), Boost.Asio, Boost.Thread, Boost.Serialization, Boost.Signals2
- C# 4.0, NLog, NUnit, Moq, P/Invoke, Mixed Mode Assembly
- Java 7, JNA
- Visual Studio 2010, 2013, Eclipse Mars
- Jazz, ClearCase



---

**Open Source Projekt:**

"Show Project  
Dependencies"

**Beschreibung:**

Anzeigen von Visual Studio  
Projektabhängigkeiten.

**Kunde:**

PERA Software Solutions  
GmbH

**Laufzeit:**

02.2015 – 03.2015

**Aufgaben:**

- Design und Entwicklung eines Java/JavaFX Programms.

**Funktionalität:**

- Parsen der Visual Studio Solution- und Projektdateien.
- Anzeigen der Abhängigkeiten in einer JavaFX TreeView.
- Aktualisieren der GUI mittels einer selbst entwickelten Signal/Slot Implementierung.

**Technologien:**

- Java 8, Java Xml, JavaFX, JavaAidKit
- Maven, Eclipse Mars
- Git, GitHub.com
- Windows / Linux

**Homepage:**

<http://www.pera-software.com/html/software/show-project-dependencies/show-project-dependencies.html>

---

**Open Source Projekt:**

"CppAidKit"

**Beschreibung:**

Entwicklung von  
wiederverwendbaren  
Klassen.

**Kunde:**

PERA Software Solutions  
GmbH

**Laufzeit:**

09.2014 - heute

**Aufgaben:**

- Design und Entwicklung einer C++/Qt Klassenbibliothek.

**Funktionalität:**

- data\_mutex: Compile time save thread synchronisation.
- file: RAII file handling Klasse.
- enum\_class: Typensichere Enums mit Integer/String Konvertierung.
- enum\_flags: Typensichere Flags mit Enums.
- Qt: IntegerSpinBox, DirectoryGuide
- uvm.

**Technologien:**

- C++17, Qt 5, Boost, STL, Google Test
- CMake, Qt Creator
- Git, GitHub.com
- Linux / Windows

**Homepage:**

<https://github.com/petermost/CppAidKit>

**Open Source Projekt:**  
"JavaAidKit"

**Beschreibung:**  
Entwicklung von  
wiederverwendbaren  
Klassen.

**Kunde:**  
PERA Software Solutions  
GmbH

**Laufzeit:**  
05.2014 - heute

**Aufgaben:**

- Design und Entwicklung einer Java Klassenbibliothek.

**Funktionalität:**

- CyclicList: Implementierung einer Liste als Ring.
- DataMutex: Compile time save thread synchronisation.
- Out<>, Ref<>: Out bzw. Ref-Parameter für Java.
- Signal/Slot Implementierung
- uvm.

**Technologien:**

- Java 14, JDK, JUnit 5
- Maven, Eclipse
- Git, GitHub.com
- Linux/Windows

**Homepage:**

<https://github.com/petermost/JavaAidKit>

**Projekt:**  
"CMWmars"  
(LegacyFileImporter)

**Beschreibung:**  
Analyse und Visualisierung  
von Mobilfunktestdaten.

**Kunde:**  
Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**  
01.2011 – 12.2015

**Aufgaben:**

- Design und Entwicklung eines C++ LegacyFileImporters.

**Funktionalität:**

- Einlesen einer proprietären binären Logdatei.
- Speichern der Daten in einer SQLite-Datenbank.
- Erstellen eines RAII C++-Wrappers für die SQLite-C-API.
- Paralleles Einlesen und Speichern mit der Parallel Patterns Library (PPL).
- Kommunikation der beiden PPL-Tasks über eine BoundedQueue und Boost.Variant.

**Technologien:**

- C++ 11, C++/CLI, PPL, SQLite, Boost.Variant
- C# 4.0, NLog, NUnit, Moq
- VTune
- Visual Studio 2010, 2013
- Jazz, ClearCase

**Projekt:**  
"CMWmars"

**Beschreibung:**  
Analyse und Visualisierung  
von Mobilfunktestdaten.

**Kunde:**  
Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**  
01.2011 – 12.2015

**Aufgaben:**

- Design und Entwicklung einer C#/WPF Applikation.
- Analyse und Behebung von Defekten.

**Technologien:**

- C# 4.0, System.Data.SQLite, WPF, NLog, NUnit, Moq,
- Java 8, Sikuli
- SQLite
- Visual Studio 2010, 2013, Eclipse
- Jazz, ClearCase

**Open Source Projekt:**  
"DeleteOldFiles"

**Beschreibung:**  
Löschen von alten  
temporären Dateien.

**Kunde:**  
PERA Software Solutions  
GmbH

**Laufzeit:**  
12.2013 – 05.2014

**Aufgaben:**

- Design und Entwicklung eines C++/Qt Programms.

**Funktionalität:**

- Löschen der Dateien in einem eigenen Thread.
- Aktualisieren der GUI mittels Signalen.

**Technologien:**

- C++ 11, Qt 5, CppAidKit
- CMake, Qt Creator
- Git, GitHub.com
- Linux/Windows

**Homepage:**

<http://www.pera-software.com/html/software/delete-old-files/delete-old-files.html>

**Projekt:**  
"MCT-Tools"  
(HardwareService)

**Beschreibung:**  
Visualisierung, Verwaltung  
und Ausführung von  
Mobilfunktests.

**Kunde:**  
Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**  
02.2007 – 12.2010

**Aufgaben:**

- Design und Entwicklung eines Java Wrappers für eine COM-Schnittstelle.

**Funktionalität:**

- Ansteuern einer COM-Schnittstelle in Java
- Marshallen von C/C++ Daten nach Java

**Technologien:**

- Java 5, ComfyJ, JUnit
- Eclipse
- Bugzilla, ClearCase

**Projekt:**  
"MCT-Tools"  
(LoggingServer)

**Beschreibung:**  
Visualisierung, Verwaltung  
und Ausführung von  
Mobilfunktests.

**Kunde:**  
Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**  
02.2007 – 12.2010

**Aufgaben:**

- Design und Entwicklung eines Java LoggingServers.

**Funktionalität:**

- Empfangen von TLV-Daten mit Asynchronen Socket Channels.
- Speichern der Daten in einer proprietären binären Logdatei.

**Technologien:**

- Java 5, NIO, JUnit
- Eclipse
- Bugzilla, ClearCase

---

**Projekt:**

"MCT-Tools"

**Beschreibung:**

Visualisierung, Verwaltung  
und Ausführung von  
Mobilfunktests.

**Kunde:**

Rohde & Schwarz GmbH &  
Co. KG

**Laufzeit:**

02.2007 – 12.2010

**Aufgaben:**

- Weiterentwicklung einer Java/Swing Applikation.
- Analyse und Behebung von Defekten.

**Technologien:**

- Java 5, Swing, CORBA, Log4j, FEST, JUnit 4, PowerMock, EasyMock
- Jython, Python
- Ant, Eclipse, Launch4j
- Bugzilla, ClearQuest, ClearCase
- Windows

---

**Open Source Projekt:**

"wxWidgets"

**Beschreibung:**

Plattformübergreifende GUI-  
Klassenbibliothek.

**Kunde:**

wxWidgets.org

**Laufzeit:**

03.2008 – 06.2011

**Aufgaben:**

- Analyse und Behebung von Defekten in einer C++ Crossplatform-Klassenbibliothek.
- Design und Entwicklung des typensicheren "Bind".
- Initiale Portierung auf Qt (wxQt).

**Technologien:**

- C++ 98, Qt 5, CppUnit
- Bakefile, Elipse (CDT), gVim, CTags, Qt Creator
- Subversion, Trak
- Linux

**Homepage:**

[www.wxWidgets.org](http://www.wxWidgets.org)

---

**Projekt:**

"FIA"

**Beschreibung:**

Fahrplanverwaltung

**Kunde:**

Vossloh IT AG

**Laufzeit:**

05.2006 – 01.2007

**Aufgaben:**

- Weiterentwicklung der in C geschriebenen "Kernprozesse" um eine Unterstützung für das VDV-Protokoll.
- Einführung von C++ Unittests.
- Entwicklung eines Java Validierungsprogramms für das VDV-Protokoll.

**Technologien:**

- C, C++, SQL, PostgreSQL, Boost.Test
  - make, Boost.Jam
  - KDevelop, gVim
  - Java, Log4j, JUnit, HTTP-Unit, HTML-Unit, XML-Unit
  - Apache Tomcat
  - Ant, Eclipse
  - CVS, ClearQuest
  - Windows, Linux
-

---

**Projekt:**

"Auslandsauskunft"

**Beschreibung:**

Such-Web-Service für die Auslandsauskunft der Telekom.

**Kunde:**

SoftMethod GmbH

**Laufzeit:**

12.2005 – 04.2006

**Aufgaben:**

- Konsolidieren der UserGuides.
- Entwurf der Java WSDL-Serviceschnittstelle.
- Design und Entwicklung eines Java/Swing Prototypen für die WSDL-Serviceschnittstelle.

**Funktionalität:**

- Semiautomatische Erstellung der GUI mit Hilfe von Reflection.
- Generische Delegate-Klasse zur Implementierung von Listnern.

**Technologien:**

- Java, Swing, Log4j, TableLayout
- E115, SOAP, ASN.1, Apache Tomcat, Apache Axis, WSDL
- Ant, Eclipse
- Sun Solaris

---

**Projekt:**

"Mediation Device"

**Beschreibung:**

Aufzeichnen und weiterleiten von Internetsitzungen an einen Bedarfsträger (IP-Monitoring).

**Kunde:**

Siemens AG

**Laufzeit:**

11.2002 – 11.2005

**Aufgaben:**

- Design und Entwicklung eines Hochleistungs C++ Datenbrokers.
- Design und Entwicklung eines C++ Wrappers für BSD Sockets.
- Design und Entwicklung eines C++ Wrappers für POSIX Threads.
- Design und Dokumentation der TLV/XML-Schnittstellen.
- Analysieren der Logdateien mit kleinen Python-Sripten.

**Funktionalität:**

- Paralleles Empfangen und Senden via Multithreading.
- Empfangen und Senden der Daten via TCP/IP.
- Implementierung verschiedener Protokolle zu den Probes.
- Entwicklung von Mock-Probes zu Testzwecken.
- Propagieren von Ereignissen mit dem Observer- und Command-Designpattern.
- Persistente Speicherung des Programmzustands mit dem Chain-of-responsibility- und Serialize-Designpattern.
- Garantierte Threadsynchronisierung mit der ResourceGuard-Template-Klasse.

**Technologien:**

- C++, STL, Boost, BSD Sockets, Pipes, POSIX Threads, wxWidgets, AidKit
- TLV, XML
- SCons, GDB, DDD, Insight, KDevelop
- SuSE Linux 7.1 – 9.3, FreeBSD

---

**Projekt:**

"Remote Player"

**Beschreibung:**

GUI-Tool zur Steuerung eines Java-Media-Players aus anderen Anwendungen heraus.

**Kunde:**

Siemens AG

**Laufzeit:**

05.2002 – 10.2002

**Aufgaben:**

- Design und Entwicklung einer C++/MFC Applikation.
- Design und Entwicklung einer "Global Windows Hook" DLL.

**Funktionalität:**

- Abfangen der Player-spezifischen Tastendrücke.
- Verschicken der abgefangenen Tastendrücke mit einer global registrierten Windows Message.
- Steuern des Mediaplayers durch simulierte Tastendrücke.
- Umschalten der GUI-Sprache (Englisch und Arabisch) zur Startzeit.

**Technologien:**

- Visual C++ 6.0, MFC, Windows API, AidKit
- Globaler Windows Hook
- Visual Studio 6.0
- SourceSafe
- Windows 2000

---

**Projekt:**

"LIRC"

**Beschreibung:**

Extrahieren von gesprächsbegleitenden Meta-Daten.

**Kunde:**

Siemens AG

**Laufzeit:**

10.2001 – 04.2002

**Aufgaben:**

- Design und Weiterentwicklung einer C++/MFC Applikation.
- Design und Entwicklung von protokollspezifischen Dekodern.
- Fehlerbehebung beim existierenden ASN.1-Dekoder.
- Design und Neuentwicklung eines ASN.1-Parsers.
- Analyse und Beseitigung von Speicherlecks.

**Funktionalität:**

- Extrahieren von gesprächsbegleitenden Meta-Daten aus ASN.1- und Textdateien.
- Extrahieren der Informationen mit Regulären Ausdrücken.
- Speichern der Daten in einer Datenbank mittels einer COM-Schnittstelle.
- Vereinfachung der Dekoderinstanzierung mit dem AbstractFactory-Designpattern.
- Vereinheitlichen der unterschiedlichen COM-Schnittstellen mit dem Adapter-Designpattern.

**Technologien:**

- Visual C++ 6.0, MFC, COM, PCRE Library
- ASN.1, Reguläre Ausdrücke
- BoundsChecker
- PVCS
- Visual Studio 6.0
- Windows 2000

---

<b>Projekt:</b> "VR-Networld"	<b>Aufgaben:</b> <ul style="list-style-type: none"><li>• Analyse des C++ Quellcodes auf Wartbarkeit und Erweiterbarkeit.</li><li>• Erstellung eines Gutachtens über die Analyseergebnisse.</li></ul>
<b>Beschreibung:</b> Homebanking-Software	
<b>Kunde:</b> E & M Software Solutions	<b>Technologien:</b> <ul style="list-style-type: none"><li>• Visual C++ 6.0</li><li>• Visual Studio 6.0</li><li>• SourceSafe 6.0</li><li>• Windows 2000</li></ul>
<b>Laufzeit:</b> 04.2001 – 05.2001	

---

<b>Projekt:</b> "Clicktionary, Clevertrainer"	<b>Aufgaben:</b> <ul style="list-style-type: none"><li>• Design und Weiterentwicklung einer C++/MFC Applikation.</li><li>• Design und Entwicklung von wiederverwendbaren Basisklassen für beide Applikationen.</li></ul>
<b>Beschreibung:</b> Übersetzungstool für Wörter und Wendungen.	<b>Funktionalität:</b> <ul style="list-style-type: none"><li>• Abfragen und Bearbeiten der Proxyserver-Einstellungen mit einer Browser-Klasse, die auf dem Bridge-Designpattern basiert.</li><li>• Entwickeln eines Activator Event-Mechanismus.</li></ul>
<b>Kunde:</b> Cleverlearn GmbH	<b>Technologien:</b> <ul style="list-style-type: none"><li>• Visual C++ 6.0, MFC</li><li>• BoundsChecker</li><li>• Perforce</li><li>• Windows 2000</li></ul>
<b>Laufzeit:</b> 04.2000 – 01.2001	

---

<b>Projekt:</b> "RCC"	<b>Aufgaben:</b> <ul style="list-style-type: none"><li>• Design und Entwicklung einer C++/MFC Applikation.</li><li>• Design und Entwicklung der Multithreading-Architektur.</li><li>• Entwurf und Dokumentation der externen TLV-Schnittstellen.</li><li>• Einführung einer Versionsverwaltung (SCM).</li></ul>
<b>Beschreibung:</b> Überwachen und Aufzeichnen von Telefongesprächen.	<b>Funktionalität:</b> <ul style="list-style-type: none"><li>• Aufzeichnung der Gespräche mit externen Voice-Rekordern.</li><li>• Empfangen der EWSD-Ereignisse (verbunden, aufgelegt usw.) von mehreren externen Protokolladaptern (PAD) über Named Pipes.</li><li>• Anzeigen der PAD-Konfiguration in einer MFC-GUI.</li><li>• Kommunikation der Threads mit dem Observer- und Command-Designpattern.</li><li>• Thread sichere Speicherverwaltung mit einer ReferenceCounter basierten SmartPointer-Template-Klasse.</li></ul>
<b>Kunde:</b> Siemens AG	<b>Technologien:</b> <ul style="list-style-type: none"><li>• Visual C++ 5.0, MFC, Named Pipes, Sockets, Threads</li><li>• BoundsChecker</li><li>• SourceSafe</li><li>• Visual Studio 97 (5.0), Innovator</li><li>• Windows 2000, NT4.0</li></ul>
<b>Laufzeit:</b> 02.1998 – 03.2000	

---

---

**Projekt:**

"Payout-Center"

**Aufgaben:**

- Design und Entwicklung eines C++ Kommandozeilenprogramms.

**Beschreibung:**

Ansteuerung eines Archivroboters in einem Pay-TV Payout-Center.

**Funktionalität:**

- Abfrage der Datenbank auf den nächsten geplanten Film.
- Abfrage der Position des Films im Archivturm.
- Feststellen welches Bandabspielgerät frei ist.
- Anweisen des Roboters, den Film aus dem Archivturm zu holen und in das Abspielgerät einzusetzen.
- C++ Wrapper um die Roboter-API um const-Korrektheit zu erreichen.
- Entwicklung einer Konvertierungs-Template-Klasse, um zwischen Strings und Enums typensicher umzuwandeln.
- Implementierung der Tape-Klasse als Flyweight-Designpattern.

**Kunde:**

BETA-Research

**Laufzeit:**

02.1996 – 04.1997

**Technologien:**

- C++, DBTools++
- Make, Borland C++
- Sun Solaris

---

**Projekt:**

"Maskeneditor"

**Aufgaben:**

- Refaktorisierung der C++/StarView Applikation zur besseren Fehlerbehebung.
- Design und Entwicklung der Druckerausgabe.

**Beschreibung:**

Grafischer Editor zur Erstellung grafischer Masken.

**Funktionalität:**

- Rekursive Bearbeitung der Maskenelemente mit dem Guide- und Visitor-Designpattern.

**Kunde:**

Softlab GmbH

**Laufzeit:**

01.1997 – 12.1997

**Technologien:**

- Visual C++ 5.0, StarView
- Purify
- Source Safe
- Visual Studio 97 (5.0)



**Projekt:**  
"BodyCare"

**Aufgaben:**  
• Design und Entwicklung einer C++/MFC Applikation.

**Beschreibung:**  
Speicherung von  
Patientendaten auf  
Mikroprozessorkarten.

**Funktionalität:**  
• Visualisieren und Editieren von Patientendaten.  
• Auslesen und Beschreiben der Daten auf Mikroprozessorkarten mit verschiedenen Kartenlesegeräten.  
• Simulieren unterschiedlicher Kartenlesegeräte mit dem Proxy-Designpattern.  
• Kommunikation mit der Karte bzw. dem Lesegerät im Übertragungsprotokoll T1.  
• Sprachenumschaltung zur Laufzeit durch rekursives Übersetzen der GUI-Elemente mit dem Iterator-Designpattern.

**Kunde:**  
Oldenbourg Datensysteme  
GmbH

**Laufzeit:**  
09.1995 – 12.1996

**Technologien:**  
• Visual C++ 1.52c, MFC  
• Serielle Schnittstelle (RS-232)

**Projekt:**  
"Card-DLL"

**Aufgaben:**  
• Design und Entwicklung einer C++ Klassenbibliothek.  
• Design einer Mikroprozessorkarten-API.

**Beschreibung:**  
Kartenterminal-DLL zur  
Kommunikation mit einem  
multifunktionalen  
Kartenterminal.

**Funktionalität:**  
• Kommunikation über die serielle Schnittstelle mit einem angeschlossenen Kartenterminal.  
• Isolierung der Bibliothek-Interna durch Insulation-Design.  
• Abbildung der C++ Klassenbibliothek auf eine C-Schnittstelle.

**Kunde:**  
Oldenbourg Datensysteme  
GmbH

**Laufzeit:**  
09.1996 – 10.1996

**Technologien:**  
• Visual C++ 1.52c, MFC  
• Serielle Schnittstelle, T1-Protokoll (DIN 7816-3, DIN 7816-4)  
• Windows 3.X

**Projekt:**  
"Lieferschein"

**Aufgaben:**  
• Design und Entwicklung einer C++/TurboVision Application.  
• Design und Entwicklung der Lieferscheinmaske.

**Beschreibung:**  
Lieferscheinprogramm zum  
Ausfüllen und Bedrucken  
von Lieferscheinen.

**Funktionalität:**  
• Positionieren des Druckers zum Bedrucken der vorgedruckten Lieferscheine.

**Kunde:**  
Oldenbourg Datensysteme  
GmbH

**Laufzeit:**  
04.1994 – 08.1996

**Technologien:**  
• Borland C++ 3.1, TurboVision  
• DOS 5.0

**Projekt:**  
"UNIKA"

**Beschreibung:**

Embedded Firmware für den  
Handheld Computer  
"UNIKA".

**Kunde:**

Oldenbourg Datensysteme  
GmbH

**Laufzeit:**

05.1993 – 03.1994

**Aufgaben:**

- Design und Entwicklung einer Embedded C Applikation.
- Simulation der C++ Polymorphie.

**Funktionalität:**

- Lesen und Schreiben von Krankenkassenkarten.
- Editieren der Krankenkassendaten.
- Gültigkeit der Krankenkassendaten prüfen.
- Simulieren nicht vorhandener Hardware in der PC-Umgebung.
- Modellierung durch abstrakte Datentypen (ADT).

**Technologien:**

- ANSI-C
- ASN.1

**Projekt:**

"EC-Kartenverwaltung"

**Beschreibung:**

Programm zum Verwalten  
des EC-Kartenbestandes.

**Kunde:**

Oldenbourg Datensysteme  
GmbH

**Laufzeit:**

03.1996 – 06.1996  
08.1992 – 10.1993

**Aufgaben:**

- Design und Entwicklung einer Clipper Applikation.

**Funktionalität:**

- Verwalten des EC-Kartenbestandes.
- Erstellen von Lagerbestandslisten.
- Bedrucken von Lieferscheinen.

**Technologien:**

- Clipper 5.0 (DBase-Kompiler)
- DOS 5.0

**Projekt:**

"Konvertierungen"

**Beschreibung:**

Programme zum Überprüfen  
und konvertieren von  
kundenspezifischen Daten  
für die  
Kartenpersonalisierung.

**Kunde:**

Oldenbourg Datensysteme  
GmbH

**Laufzeit:**

08.1990 – 02.1996

**Aufgaben:**

- Design und Entwicklung von kleineren C/C++ Programmen.
- Analysieren der Kundendaten.

**Funktionalität:**

- Auswerten der Kommandozeile.
- Umwandeln der Daten in spezielle Maschinenformate.

**Technologien:**

- Turbo-C, Borland-C++
- DOS