

Beraterprofil

Name: Peter Most

Kontaktdaten: PERA Software Solutions GmbH
Ulrich-Nanshaimer-Straße 29
85652 Landsham

Internet: www.PERA-Software.com
E-Mail: 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++, Java, C#
- GUI-Entwicklung
- Refactoring, Unit Tests
- Multithreading / Multicore
- Netzwerkprogrammierung, Interprocess-Kommunikation
- Programmiersprachen-Interoperabilität
- Plattformübergreifende bzw. -unabhängige Implementierung

Betriebssysteme:

- Windows (7, XP)
- Linux ([K]Ubuntu)

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)
- Continuous Integration

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: 20. Juli 2018
→ http://www.pera-software.com/html/projects/Peter_Most_Profil.pdf

C++

Erfahrung:	Hervorragend (seit 1990)
Versionen:	<ul style="list-style-type: none">• C++ 11, C++ 14• Visual C++ 2013, Visual C++/CLI 2013• GCC C++ 6.x
IDEs:	<ul style="list-style-type: none">• Visual Studio 2013,• Qt Creator• Elipse (CDT)• gVim, CTags
Buildwerkzeuge:	<ul style="list-style-type: none">• CMake, qmake, make• Scons, BakeFile
GUI Frameworks:	<ul style="list-style-type: none">• Qt 5, MFC, wxWidgets
Unit Tests:	<ul style="list-style-type: none">• Qt Test, Boost Test, CppUnit, NUnit
Networking:	<ul style="list-style-type: none">• Boost.Asio, Qt Sockets, BSD Sockets• Boost.Serialization, TLV-Protokoll• Named Pipes• TCP/IP, IPC
Multithreading:	<ul style="list-style-type: none">• Boost Threads, Qt Threads, QProcess• POSIX Threads, Windows Threads• Parallel Patterns Library (PPL)
Logging:	<ul style="list-style-type: none">• Boost Format, NLogC
Language interoperability:	<ul style="list-style-type: none">• JNA, P/Invoke, C++ CLI
Datenbanken:	<ul style="list-style-type: none">• SQLite3 C-API, DBase
Debugger:	<ul style="list-style-type: none">• Visual Studio, GDB
Profiler:	<ul style="list-style-type: none">• VTune
Versionsverwaltungen:	<ul style="list-style-type: none">• Jazz, Git, Subversion• GitHub
Bugtracker:	<ul style="list-style-type: none">• Jazz, Trac

Sonstige:

- Boost, STL
- Qt Xml
- Boost Signals2, Boost Variant
- CppAidKit
- u.a.

Java

Erfahrung: Sehr gut (seit 2005)

Versionen:

- Java SE 8, Java EE 7

IDEs:

- Eclipse (Oxygen)

Buildwerkzeuge:

- Maven, Ant, Gant

GUI Frameworks:

- JavaFX, Swing

Unit Tests:

- JUnit, JUnit Addons
- EasyMock, EasyMock Class Extension, PowerMock
- Sikuli, FEST (Fixtures for Easy Software Testing)
- HTTP-Unit, XML-Unit, HTML-Unit

HTML:

- RenderSnake, jsoup

XML:

- Java API for XML Processing (JAXP)

Networking:

- Java NIO Sockets
- TCP/IP, IPC

Multithreading:

- Java Threads

Logging:

- Log4j, LogBack

Language interoperability:

- JNA, JNI, ComfyJ
- Jython

CommandLine Parser:

- args4j

Executable Wrapper:

- Launch4j

**Versions-
verwaltungen:**

- Jazz, Git, ClearCase
- GitHub, BitBucket

Bugtracker: • Jira, Jazz, Bugzilla

Sonstige: • Google Guava
 • Apache Commons: Lang, Net, CSV
 • JavaAidKit
 • u.a.

C#

Erfahrung: Gut (seit 2011)

Versionen: • C# 4.0, .Net 4.0

IDEs: • Visual Studio 2013

GUI Frameworks: • WPF, WinForms

Unit Tests: • NUnit
 • Moq

Networking: • C# Sockets
 • TCP/IP, IPC

Multithreading: • C# Threads
 • Task Parallel Library (TPL)

Logging: • NLog

Language interoperability: • P/Invoke, C++ CLI

Datenbanken: • SQLite3, System.Data.SQLite

Versionsverwaltungen: • Jazz

Bugtracker: • Jazz

Projekte

Projekt:

"QuickStep"

Aufgaben:

- Portieren eines Windows Programms nach Linux

Auftraggeber:

Rohde & Schwarz GmbH &
Co. KG

Funktionalität:

-

Beschreibung:

Technologien:

- Visual Studio
- Windows / Linux

Laufzeit:

09.2018 – Heute

Methoden:

-
-

Projekt:

"TeDiSPlus"

Aufgaben:

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

Auftraggeber:

BWI GmbH

Funktionalität:

- Abfragen der Oracle/Derby Datenbank mit JPA.
- Empfangen bzw. Senden der JSON Frontend-Daten.

Beschreibung:

Disponierung von
Technikern

Technologien:

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

Laufzeit:

04.2017 – 08.2018

Methoden:

- OOA, OOD, OOP
 - Agile Entwicklung (Scrum)
 - Test Driven Development (TDD)
-

Open Source Projekt:
"FritzBoxCallMonitor"

Auftraggeber:
PERA Software Solutions
GmbH

Beschreibung:
Empfangen und Anzeigen
von FritzBox Call-
Information.

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, QTest, Qt Creator, CMake
- CppAidKit
- TCP/IP, XML
- Git, GitHub.com
- Linux

Homepage:

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

Projekt:
"HomePageGenerator"

Auftraggeber:
PERA Software Solutions
GmbH

Beschreibung:
Erzeugen von statischen
HTML Seiten.

Laufzeit:
01.2015 – heute

Aufgaben:

- Design und Entwicklung eines statischen Webseitengenerators für PERA-Software.com.

Funktionalität:

- Parsen der Kommandozeile mit args4j.
- Laden aller Page-Klassen mit der Guava-ClassPath-Klasse.
- Umwandeln der Page-Klassennamen in "Pretty URLs".
- Schreiben der HTML Seiten mit HtmlCanvas, Renderable u.a.

Technologien:

- Java 8, Eclipse Neon
- RenderSnake, jsoup, HTML, CSS
- Apache Commons: Lang, Net, CSV
- Google Guava
- args4j (Kohsuke)
- JUnit 4
- JavaAidKit
- Git, Bitbucket.org
- Linux

Projekt:

"CMWmars (PCAP Dekodierung)"

Aufgaben:

- Design und Entwicklung der MCT-Tools Nachfolgeneration.
- Design und Entwicklung der PCAP-Dekodierung.

Auftraggeber:

Rohde & Schwarz GmbH & Co. KG

Funktionalität:

- Dekodieren von 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.

Beschreibung:

Analyse und Visualisierung von Mobilfunktestdaten.

Technologien:

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

Laufzeit:

01.2016 – 08.2016

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)
- Continuous Integration

Projekt:

"CMWmars (LoggingServer)"

Aufgaben:

- Design und Entwicklung der MCT-Tools Nachfolgeneration.
- Design und Entwicklung eines C++ LoggingServers.

Auftraggeber:

Rohde & Schwarz GmbH & Co. KG

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#.

Beschreibung:

Analyse und Visualisierung von Mobilfunktestdaten.

Technologien:

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

Laufzeit:

01.2011 – 12.2015

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)
- Continuous Integration

Open Source Projekt:

"Show Project
Dependencies"

Auftraggeber:

PERA Software Solutions
GmbH

Beschreibung:

Anzeigen von Visual Studio
Projektabhängigkeiten.

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

Homepage:

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

Open Source Projekt:

"CppAidKit"

Auftraggeber:

PERA Software Solutions
GmbH

Beschreibung:

Bereitstellen von
wiederverwendbaren
Klassen.

Laufzeit:

09.2014 - heute

Aufgaben:

- Design und Entwicklung von wiederverwendbaren plattformübergreifenden C++/Qt Klassen.

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
- u.a.

Technologien:

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

Homepage:

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

Open Source Projekt:
"JavaAidKit"

Auftraggeber:
PERA Software Solutions
GmbH

Beschreibung:
Bereitstellen von
wiederverwendbaren
Klassen.

Laufzeit:
05.2014 - heute

Aufgaben:

- Design und Entwicklung von wiederverwendbaren Java/JavaFX Klassen.

Funktionalität:

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

Technologien:

- Java 8, Eclipse Neon
- JUnit, JDK, JavaFX
- Git, GitHub.com
- Linux/Windows

Homepage:

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

Projekt:
"CMWmars
(LegacyFileImporter)"

Auftraggeber:
Rohde & Schwarz GmbH &
Co. KG

Beschreibung:
Analyse und Visualisierung
von Mobilfunktestdaten.

Laufzeit:
01.2011 – 12.2015

Aufgaben:

- Design und Entwicklung der MCT-Tools Nachfolgegeneration.
- 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, C# 4.0
- PPL, SQLite, Boost.Variant
- NLog, NUnit, Moq
- Visual Studio 2010, 2013
- Jazz, ClearCase
- VTune

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)
- Continuous Integration

Projekt:

"CMWmars"

Aufgaben:

- Design und Entwicklung der MCT-Tools Nachfolgegeneration.
- Analyse und Behebung von Defekten.

Auftraggeber:

Rohde & Schwarz GmbH &
Co. KG

Technologien:

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

Beschreibung:

Analyse und Visualisierung
von Mobilfunktestdaten.

Laufzeit:

01.2011 – 12.2015

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)
- Continuous Integration

Open Source Projekt:

"DeleteOldFiles"

Aufgaben:

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

Auftraggeber:

PERA Software Solutions
GmbH

Funktionalität:

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

Beschreibung:

Löschen von alten
temporären Dateien.

Technologien:

- C++ 11 (GCC)
- Qt 5.x, CppAidKit, Qt Creator
- Git, GitHub.com
- Linux/Windows

Laufzeit:

12.2013 – 05.2014

Homepage:

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

Projekt:

"MCT-Tools
(HardwareService)"

Auftraggeber:

Rohde & Schwarz GmbH &
Co. KG

Beschreibung:

Visualisierung, Verwaltung
und Ausführung von
Mobilfunktests.

Laufzeit:

02.2007 – 12.2010

Aufgaben:

- Entwicklung von Anwendungen und Benutzeroberflächen für eine Mobilfunk-Protokolltesterfamilie.
- Design und Entwicklung einer Java-Anbindung zu einer COM-Schnittstelle.

Funktionalität:

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

Technologien:

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

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)

Projekt:

"MCT-Tools
(LoggingServer)"

Auftraggeber:

Rohde & Schwarz GmbH &
Co. KG

Beschreibung:

Visualisierung, Verwaltung
und Ausführung von
Mobilfunktests.

Laufzeit:

02.2007 – 12.2010

Aufgaben:

- Entwicklung von Anwendungen und Benutzeroberflächen für eine Mobilfunk-Protokolltesterfamilie.
- 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, TCP/IP
- JUnit
- Eclipse
- Bugzilla, ClearCase

Methoden:

- OOA, OOD, OOP, UML
 - Agile Entwicklung (Scrum)
 - Test Driven Development (TDD)
-

Projekt:

"MCT-Tools"

Auftraggeber:

Rohde & Schwarz GmbH &
Co. KG

Beschreibung:

Visualisierung, Verwaltung
und Ausführung von
Mobilfunktests.

Laufzeit:

02.2007 – 12.2010

Aufgaben:

- Entwicklung von Anwendungen und Benutzeroberflächen für eine Mobilfunk-Protokolltesterfamilie.
- Analyse und Behebung von Defekten.

Technologien:

- Java (5, 6), C++
- Swing, FEST
- JUnit4, JUnit4-Extensions, Log4j
- PowerMock, EasyMock, EasyMock-Extensions
- Socket, NIO
- JNA, JNI
- CORBA, IDL, ComfyJ
- Eclipse, Ant, Launch4j
- Jython, Python
- Bugzilla, ClearQuest, ClearCase, Windows

Methoden:

- OOA, OOD, OOP, UML
- Agile Entwicklung (Scrum)
- Test Driven Development (TDD)

Open Source Projekt:

"wxWidgets"

Auftraggeber:

wxWidgets.org

Beschreibung:

Plattformübergreifende GUI-
Softwarebibliothek.

Laufzeit:

03.2008 – 06.2011

Aufgaben:

- Analyse und Behebung von Defekten.
- Design und Entwicklung des typensicheren "Bind".
- Initiale Portierung auf Qt (wxQt)

Technologien:

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

Homepage:

www.wxWidgets.org

Projekt:
"FIA"

Auftraggeber:
Vossloh IT AG

Beschreibung:
Fahrplanverwaltung

Laufzeit:
05.2006 – 01.2007

Aufgaben:

- Weiterentwicklung der „Kernprozesse“ um eine Unterstützung für das VDV-Protokoll.
- Entwicklung von C++ Unittests zur Vorbereitung von Refactoring.
- Entwicklung eines Java Validierungsprogramms für das vom „VDV-Koppler“ implementierte VDV-Protokoll.

Technologien:

- C, C++, Java, SQL
- JUnit, HTTP-Unit, HTML-Unit, XML-Unit, Boost-Test, Log4j
- PostgreSQL, Apache Tomcat, Ant
- TCP/IP, Sockets, XML
- Eclipse, KDevelop, gVim
- GNU make, Boost-Jam, CVS, Cervisia
- ClearQuest, SSH
- Windows, Linux

Projekt:
"Auslandsauskunft"

Auftraggeber:
SoftMethod GmbH

Beschreibung:
Such-Web-Service für die Auslandsauskunft der Telekom.

Laufzeit:
12.2005 – 03.2006

Aufgaben:

- Analysieren und Dokumentieren der UserGuides-Gemeinsamkeiten.
- Entwurf der Serviceschnittstelle mit Java und dem Axis-Framework (Java2WSDL).
- Entwicklung eines Serviceprototypen mit dem Axis-Framework (WSDL2Java).
- Entwicklung einer Test-GUI für den Serviceprototypen.

Funktionalität:

- Semiautomatische Erstellung der GUI mit Hilfe von Reflection.
- Generische Delegate-Klasse zur Implementierung von Listnern (ActionListener, ItemListener)

Technologien:

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

Projekt:

"Mediation Device"

Auftraggeber:

Siemens AG

Beschreibung:

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

Laufzeit:

11.2002 – 09.2005

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Entwurf und Dokumentation der TLV- und XML-Schnittstellen.
- Analysieren der Logdateien mit kleinen Python-Sripten.

Funktionalität:

- Hohe Ausfallsicherheit und Verfügbarkeit durch Multithreading.
- Empfangen und weiterleiten der Daten via TCP/IP.
- Implementierung unterschiedlicher Netzwerkprotokolle zu den Datenkollektoren (Probes).
- Implementierung der Datenkollektoren als Plugins mit dem Strategy-Designpattern.
- Implementierung der Funktionalität innerhalb der Plugins mit dem TemplateMethod-Designpattern.
- Instanziierung der Plugins mit der ProbeFactory-Template-Klasse, die auf dem AbstractFactory-Designpattern basiert.
- Entwicklung von Test-Probes zur Unterstützung von automatischen Tests.
- Pufferung der Daten mit einem abgewandelten Flyweight-Designpattern.
- Propagieren von Ereignissen zu Observern mit den Event- bzw. Functor-Template-Klassen, die auf dem Observer-Designpattern basieren.
- Asynchrone Verarbeitung der GUI-Messages von den Threads mit einer Template-Klasse, die auf dem Command-Designpattern basiert.
- Speichern von zusätzlichen Daten in einer Triggerinstanz mit einer Template-Klasse, die auf dem Property-Designpattern basiert.
- Persistente Speicherung des Programmzustands in einer INI-Datei und mit dem ChainOfResponsibility- bzw. Serialize-Designpattern.
- Garantierte Threadsynchronisierung mit der ResourceGuard-Template-Klasse, die auf dem Proxy-Designpattern basiert.

Technologien:

- C++, Python
- GUI, wxWidgets (wxWindows)
- STL, Boost, AidKit
- Design Patterns, Template-Klassen, TLV, XML
- TCP-IP, Sockets, Threads, Pipes
- TCP-Dump, Ethereal, SSH
- GCC, GDB, DDD, Insight, KDevelop
- CVS, Cervisia, Make, SCons, Doxygen
- SuSE Linux 7.1 – 9.3, FreeBSD

Projekt:

"Remote Player"

Auftraggeber:

Siemens AG

Beschreibung:

GUI-Tool zur Steuerung eines Java-Media-Players aus anderen Anwendungen (MS-Word, Notepad usw.) heraus.

Laufzeit:

05.2002 – 10.2002

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Design der dialogbasierten GUI.

Funktionalität:

- Abfangen der Player-spezifischen Tastendrücke mit einer globalen Windows Hooks DLL.
- 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:

- DLL-Erstellung
- GUI (MFC), AidKit, Globale Windows Hooks
- MS-Windows 2000
- MS-Visual C++ 6.0
- MS-SourceSafe

Projekt:

"LIRC"

Auftraggeber:

Siemens AG

Beschreibung:

Extrahieren von gesprächsbegleitenden Meta-Daten.

Laufzeit:

10.2001 – 04.2002

Aufgaben:

- Erweiterung der GUI-Dialoge um neue Einstellmöglichkeiten.
- Design und Entwicklung von protokollspezifischen Dekodern.
- Fehlerbehebung beim existierenden ASN.1-Dekodern.
- 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.
- Weiterleiten der Daten an eine Datenbankschnittstelle (COM).
- Zentralisieren der Dekoderinstanzierung mit dem AbstractFactory-Designpattern.
- Vereinheitlichen der unterschiedlichen COM-Schnittstellen mit dem Adapter-Designpattern.

Technologien:

- GUI (MFC)
 - ASN.1, COM
 - Reguläre Ausdrücke (pcre-Library)
 - Design Patterns
 - MS-Windows 2000
 - MS-Visual C++ 6.0
 - PVCS
 - BoundsChecker
-

Projekt:

"VR-Networld"

Auftraggeber:

E & M Software Solutions

Beschreibung:

Homebanking-Software

Laufzeit:

04.2001 – 05.2001

Aufgaben:

- Analyse der Quellcodequalität bezüglich Wart- und Erweiterbarkeit.
- Erstellung eines Gutachtens über die Analyseergebnisse.

Technologien:

- MS-Windows 2000
- MS-Visual C++ 6.0
- MS-SourceSafe 6.0

Projekt:

"Clicktionary, Clevertrainer"

Auftraggeber:

Cleverlearn GmbH

Beschreibung:

Übersetzungstool für Wörter und Wendungen.

Laufzeit:

04.2000 – 01.2001

Aufgaben:

- Weiterentwicklung und Anpassung der MFC-GUI an neue Anforderungen.
- Design und Entwicklung von wiederverwendbaren Basisklassen für beide Applikationen.

Funktionalität:

- Abfragen und Bearbeiten der Proxyserver-Einstellungen mit einer Browser-Klasse, die auf dem Bridge-Designpattern basiert.
- Entwickeln eines Activator Event-Mechanismus.

Technologien:

- GUI (MFC)
 - MS-Windows 2000
 - MS-Visual C++ 6.0
 - BoundsChecker
 - Perforce
-

Projekt:
"RCC"

Auftraggeber:
Siemens AG

Beschreibung:
Überwachen und
Aufzeichnen von
Telefongesprächen.

Laufzeit:
02.1998 – 03.2000

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Design und Implementierung der Multithreading-Architektur.
- Entwurf und Dokumentation der externen TLV-Schnittstellen.
- Einführung einer Versionsverwaltung (SCM).

Funktionalität:

- Aufzeichnung des Gesprächs mit einem externen Voice-Rekorder.
- Empfangen der EWSD-Ereignisse (verbunden, aufgelegt usw.) von mehreren externen Protokolladaptern (PAD) über Named Pipes.
- Instanziierung eines Threads pro externen PAD.
- Kommunikation mit den externen PADs mit asynchronen Named Pipes und TLV-Protokoll.
- Abbildung der PAD-Konfiguration in einem Gerätebaum mit dem Composite-Designpattern.
- Rekursive Bearbeitung des Gerätebaumes mit dem Visitor-Designpattern.
- Benachrichtigung der internen Klienten mit dem Observer- bzw. NullObject-Designpattern.
- Asynchrone Benachrichtigung der Threads mit dem Command-Designpattern.
- Threadsichere Speicherverwaltung mit einer ReferenceCounter basierten SmartPointer-Template-Klasse.
- Speichern von zusätzlichen Daten in einer Geräteinstanz mit dem Property-Designpattern.

Technologien:

- MFC, Named Pipes, Sockets, Threads
- Design Patterns, Template-Klassen
- BoundsChecker
- SourceSafe
- Innovator
- MS-Visual C++ 5.0, MFC
- MS-Windows 2000, NT4.0

Projekt:

"Payout-Center"

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).

Auftraggeber:

BETA-Research

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.
- Wrappen der Roboter-API mit dem Facade-Designpattern um const-Korrektheit zu erreichen.
- Entwicklung einer Konvertierungs-Template-Klasse, um zwischen Strings und Enums typensicher umzuwandeln.
- Implementierung der Tape-Klasse als Flyweight-Designpattern.

Beschreibung:

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

Laufzeit:

02.1996 – 04.1997

Technologien:

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

Projekt:

"Maskeneditor"

Aufgaben:

- Umstrukturierung der Software zur besseren Fehlerbehebung.
- Implementierung der Druckerausgabe.

Auftraggeber:

Softlab GmbH

Funktionalität:

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

Beschreibung:

Grafischer Editor zur Erstellung grafischer Masken.

Laufzeit:

01.1997 – 12.1997

Technologien:

- MS-Visual C++ 5.0
 - StarView Klassenbibliothek
 - Source Safe
 - Purify
-

Projekt:

"BodyCare"

Auftraggeber:

Oldenbourg Datensysteme
GmbH

Beschreibung:

Speicherung von
Patientendaten auf
Mikroprozessorkarten.

Laufzeit:

09.1995 – 12.1996

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Design und Entwicklung der GUI.

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.

Technologien:

- Serielle Schnittstelle (RS-232)
- MS-Visual C++ 1.52c
- MFC

Projekt:

"Card-DLL"

Auftraggeber:

Oldenbourg Datensysteme
GmbH

Beschreibung:

Kartenterminal-DLL zur
Kommunikation mit einem
multifunktionalen
Kartenterminal.

Laufzeit:

09.1996 – 10.1996

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Design einer Mikroprozessorkarten-API.

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.

Technologien:

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

Projekt:

"Lieferschein"

Auftraggeber:

Oldenbourg Datensysteme
GmbH

Beschreibung:

Lieferscheinprogramm zum
Ausfüllen und Bedrucken
von Lieferscheinen.

Laufzeit:

04.1994 – 08.1996

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Design und Entwicklung der Lieferscheinmaske.

Funktionalität:

- Positionieren des Druckers zum Bedrucken der vorgedruckten Lieferscheine.

Technologien:

- MS-DOS 5.0
- Borland C++ 3.1
- TurboVision

Projekt:
"UNIKA"

Aufgaben:

- Analyse (OOA), Design (OOD) und Implementierung (OOP).
- Simulation der C++ Polymorphie.

Auftraggeber:

Oldenbourg Datensysteme
GmbH

Funktionalität:

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

Beschreibung:

Embedded Firmware für den
Handheld Computer
"UNIKA".

Technologien:

- ANSI-C
- ASN.1

Laufzeit:

05.1993 – 03.1994

Projekt:

"EC-Kartenverwaltung"

Aufgaben:

- Analyse, Design und Implementierung.

Auftraggeber:

Oldenbourg Datensysteme
GmbH

Funktionalität:

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

Beschreibung:

Programm zum Verwalten
des EC-Kartenbestandes.

Technologien:

- MS-DOS 5.0
- Clipper 5.0 (DBase-Kompiler)

Laufzeit:

03.1996 – 06.1996
08.1992 – 10.1993

Projekt:

"Konvertierungen"

Aufgaben:

- Analyse, Design und Implementierung.
- Analysieren der Kundendaten.

Auftraggeber:

Oldenbourg Datensysteme
GmbH

Funktionalität:

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

Beschreibung:

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

Technologien:

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

Laufzeit:

08.1990 – 02.1996