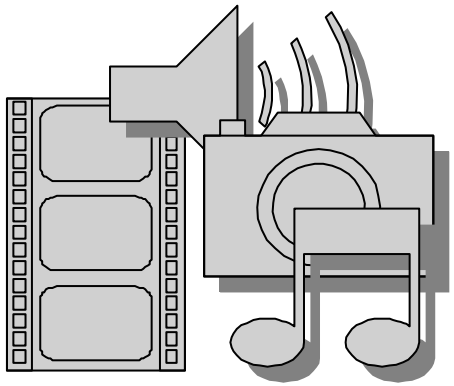
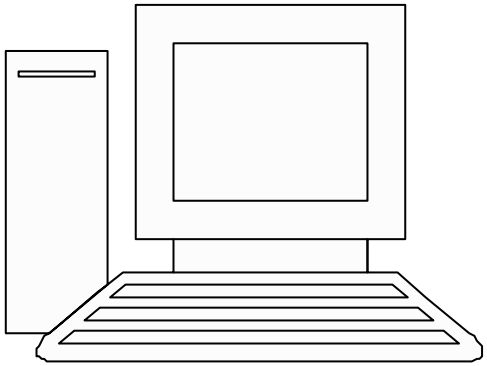
Produktentwicklung unter Linux -

von der Entscheidungsfindung bis zur Marktreife



Gliederung

- NetCam Solutions
- Das Produkt
- Ausgangssituation
- Entscheidung für Linux
- Werkzeuge
- Zwei Jahre danach
- Ausblick
- Quellen
- Noch Fragen ?



- Videolösungen
  - Eventübertragung
  - Überwachung
- Beratung
  - Möglichkeiten
  - Technische Umsetzung
- Technische Detaillösungen
  - Spezial Applikationen

- Direkte Steuerung der Kamera
- LIVE !
- Bezahlbare Lösungen für kleine und mittlere Unternehmen
- Schlank und schnell installierbar
- Optimale Lösungen auch bei schmalbandiger Anbindung
- Fernwartungs- und Analysemöglichkeiten
- Unkomplizierte Updatemöglichkeit
- Skalierbarkeit vom PC zum Embedded-System

- Geringe Latenzzeiten
  - Effektives Multitasking
- Überwachung des Datentransfers
  - Abfrage der Verbindungspuffer
- Gute Hardwareausnutzung
  - GUI nicht zwingend nötig
- Portierbarkeit auf unterschiedliche Hardware
  - System muß auf allen Platformen gleich sein
- (Logging zur Fehleranalyse)

- Kamerasteuerung
  - Übermitteln der Steuerdaten
- Universell anwendbar
- Einfache Handhabung
  - Konfiguration über den Client
- Schnelle und einfache Installation
  - Batch Installation
  - Pakete
  - Remote Installation

- Single Threaded
  - Sequentielle Abarbeitung : Empfangen / Decodieren / Anzeigen
- 1 Client / 1 Server
- Lead Tools als Transportschicht
  - Über ActiveX - Wrapper
  - Geringe Kontrollmöglichkeiten über den Datenversand
- Keine Objektorientierte Programmierung
- Server & Client in Visual Basic(TM)

- Zuviel Overhead bei den Lead Tools
  - Timing nicht zeitnah genug
  - Keine weiteren Kontrollmöglichkeiten über die Verbindung
- Zu geringe Ausführungsgeschwindigkeit
  - Hohe Grundlast des Systems
  - Starke Abhängigkeit der System-Gesamtlast
- Preview des Servers muß sichtbar sein
  - Feature des ActiveX Capture Controls
- Probleme mit einfachen Datentypen
  - Übergabe von komprimierten Bildern nur über das Dateisystem
- VB / Windows muß vorhanden sein

- Programmiersprache auf allen Plattformen vorhanden
- Multithreading
  - Minimieren der Latenzzeiten
  - Schnellere Reaktion auf verschiedene Events
- Objektorientierung
  - Einfachere Programmwartung
  - Bessere Strukturierungsmöglichkeiten
- Geringe Anlaufkosten
- IDE
  - Grafischer Editor / Debugger
- Tools
  - Betriebssystemtools zur Anzeige von Verbindungen, Speicher etc.

- Offene Quellen
  - Nachvollziehen der Vorgänge auf Sourcecodeebene
  - Entwicklerkommentare
- Systemnahe Programmierung möglich
  - Einfache Abfrage von Systemzuständen (ioctl)
- Posix - Unterstützung
- Kostengünstige Tools
  - Einfacher Test möglich, geringe Anlaufzeit
- Plattformübergreifendes GUI mit QT
  - Unterstützung verschiedener Plattformen
  - 1 Source - Prinzip

- Alle Aufrufe dokumentiert
  - Man - pages zu Systemaufrufen
  - Dokumentation im Kerneltree
  - Wenn keine Man - Page vorhanden -> siehe unten
- Quellcode vorhanden (zusätzliche Hinweise)
  - Übergabeparameter
  - Rückgabewerte nachvollziehbar
  - Entwicklerkommentare
- Einfache Beispiele im Web
  - Bequeme suche z.B. über Google etc.
  - Homepages der einzelnen Programme
- Dokumentation auf das wesentliche beschränkt
  - (Manchmal zu) Kurz und prägnant zum Ziel

- Modularer Kernel
  - Eigene Module, wo direkter Hardwarezugriff nötig
- Skalierbar auf verschiedene Systeme
  - Vom Embedded System bis zum PC
- Stabilität
  - Wenn Probleme, dann bei Hardware oder dem Programm suchen
- Ausreichende Treiberunterstützung
  - Hardwareauswahl einfacher
- Rechteverwaltung
  - Schutz und Sicherheit des Systems
- Alle nötigen Tools vorhanden
- Fernwartungsmöglichkeit

- Geringe Startkosten
  - Schneller Start, da alle Programme und Dokumentationen verfügbar
- Keine Lizenzkosten
  - Beliebig viele Entwicklerrechner
  - Keine zusätzlichen Kosten bei Testsystemen
  - Keine zusätzlichen Kosten bei Produktionssystemen
- Einfache Replizierbarkeit
  - Schnellerer Produktionsprozess
  - Geringere Produktionskosten

- Kdevelop und gcc
  - Grafische IDE
  - CVS -/ Makefile Unterstützung
  - Integrierter Grafischer Debugger
  - Integration von QT
- STL
  - Allgemein verfügbar
  - Gute Dokumentation
- QT
  - Lizenzkosten bedenken
  - Permanente Weiterentwicklung, neue OS-Unterstützung
- Visual C++
  - Lizenzkosten bedenken
  - Gute Windowsintegration

- QT - Lizenz für Linux / Windows / Mac
  - Separate Lizenz für jedes System
  - Pro Entwicklerarbeitsplatz
- Visual C++ Lizenz
  - Minimalversion genügt normalerweise
- Bücher zum Thema
  - Die C++ Standardbibliothek
  - KDE- und QT-Programmierung
  - Linux Kernelprogrammierung
  - MFC Programmierung
  - Unix Device Drivers
  - ...
- Sonstiges (Hardware, OS ...)

- Neue Features für alle Systeme planen
  - Grundschema erstellen
  - Literatur wälzen
  - Gründlich planen
- Nur plattformübergreifende Bibliotheken verwenden
  - Verzicht auf ATL, MFC und sonstige exklusive Bibliotheken
- Basisentwicklung unter Linux(TM)
  - Entwicklung vom offenen System zum geschlossenen
- Java - Entwicklung unter Windows(TM)
  - Schnellere Virtual Machine / Mehr Browser
  - Die meisten Clients laufen unter Windows
- Anschließende Portierung auf Windows(TM)
- Performance Tuning

- Namespaces
  - Immer angeben
- Systemspezifische Headerdateien
  - Präprozessor
- Stdafx
  - Präprozessor
- Threads
  - QT - Thread und/oder Wrapper Class

- Wrapper für Systemspezifische Funktionen
  - Kleinste Basis, viel Erweiterungsmöglichkeiten
  - Schlank halten
  - Wenig Hirarchiestufen (Performance)
- QT für das GUI
  - Nicht GUI Funktionen wenn möglich / sinnvoll extra implementieren
- Nur auf allen Systemen verfügbare Bibliotheken verwenden
  - Man kann es nicht oft genug sagen !

- Linux : www.linux.org
- Windows : www.microsoft.com
- KDevelop : www.kdevelop.org
- Kde : www.kde.org
- Linux-Kernel : ftp.kernel.org
- QT : www.trolltech.com
- Embedded Systeme : www.embedded-linux.de

- C++ Einführung und Leitfaden : ISBN 3-89319-375-8
- Effektiv C++ programmieren : ISBN 3-89319-816-4
- Die C++ - Standardbibliothek : ISBN 3-8273-1023-7

- KDE- und Qt-Programmierung : ISBN 3-8273-1477-1
- Linux Kernelprogrammierung : ISBN 3-8273-1476-3
- Linux Multimedia Guide : ISBN 1-56592-219-0
- Netzwerkprogrammierung unter Unix/Linux : ISBN 3-446-21093-8
- Unix Device Driver : ISBN 3-89319-229-8

- DirectX-Programmierung mit Visual C++ : ISBN 3-8273-1389-9
- DirectX, RDX, RSX and MMX Technology : ISBN 0-201-30944-0
  (Erweiterungen im Web lesen !)

Ansonsten und Jederzeit:

michael.veigel@netcam-solutions.de
http://www.netcam-solutions.de