



## **Camsource 0.7.1:**

Software-Anpassung nach eigenen Vorgaben.

An einem Beispiel einer WebCam Anwendung soll exemplarisch gezeigt werden, wie man nach Definition des Leistungsumfanges einer Software eine geeignete Basis-Open-Source Software auswählt und diese dann nach den eigenen Vorgaben ergänzt und anpasst. Weiterhin wird auch ein Weg gezeigt, die validierte Version einer größeren Gemeinschaft zur Verfügung zu stellen.

Hier soll gezeigt werden, wie der Quellcode aufgebaut ist, welche Werkzeuge benötigt werden und wie man formale Fehler erkennt und beseitigt und ein lauffähiges Programm erzeugen kann.



## **Camsource 0.7.1:**

### **Intension:**

- Eine bestehende Software (Windows) soll durch Linux-Pendant ersetzt werden. In diesem Fall mussten die Schnittstellen (USB, HTML-Server) gleich bleiben.
- Eine Recherche im Internet ergab keine Linux-Software, die die Anforderungen zu 100% erfüllte.
- Der Einsatz von Windows-Emulatoren wie WINE scheiterte, da die USB-Schnittstelle nicht unterstützt wurde.
- Camsource kam den Anforderungen am nächsten: Modular aufgebaut und über eine Konfigurationsdatei steuerbar.
- In der Quellen angegebenen Autor angemailt: => Keine Antwort erhalten !.



## Camsource 0.7.1:

### Intension (Teil 2):

- Kernel Quellen, „make“ und der Compiler „gcc“ müssen noch aus den Repositories nachinstalliert werden.
- Das Camsource-Verzeichnis samt Unterverzeichnisse im Home Verzeichnis ablegen.
- Im Camsource-Hautverzeichnis „./configure“ aufrufen. Dort wird das „Makefile“ an die aktuelle Verzeichnisstruktur angepasst und auf notwendigen Bibliotheken geprüft. Bei Bedarf sind noch fehlende nachzuinstallieren.
- Zum ersten Test der Original-Dateien diese über „make“ neu kompilieren. Wenn keine Fehlermeldung erscheint, sind die Quelldateien in Ordnung. Wenn nicht, dann nicht gut. Die Quell-Software ist dann wohl defekt oder es ist ein Fehler im „Makefile“. Für einen Laien wird es dann schwierig.



## Camsource 0.7.1:

### Intension (Teil 3):

- Mit „make install“ wird die Software installiert. Nach Aufruf den bisherigen Funktionsumfang testen. Wenn alles in Ordnung ist, kann die eigentliche händische Anpassung erfolgen. Aufräumen mit „make clean“: Damit werden die vom Compiler erstellten Objekt und Hilfsfiles gelöscht.
- Dateien im /src Verzeichnis anpassen/ändern und speichern.
- Programm debuggen bis gewünschter Funktionsumfang verfügbar ist.
- Programm testen, testen und testen. Wenn in alles Ordnung (keine Fehlfunktionen oder Abstürze / hängenbleiben), dann Weitergabe an die Community.
- Hier in diesem Fall erfolgte die Weitergabe an openSUSE, einige Alternativen werden in den folgenden Vorträgen näher behandelt.



## **Camsource 0.7.1:**

**Nun die praktische Präsentation:**

Bei Fragen bitte melden. Vielen Dank.



## **Camsource 0.7.1:**

**Ablage im Build von openSUSE:**

**Bearbeitungsstand/Status:**

<https://build.opensuse.org/package/show/openSUSE:Factory/camsource>

**Downloadmöglichkeit der Quellen:**

<http://download.opensuse.org/source/distribution/leap/42.1/repo/oss/>

-



## **Camsource 0.7.1:**

**Vielen Dank für Ihre Aufmerksamkeit**