

## Übungsblatt 2 – Ausgabe am 04.05.2020

Abgabe bis 11.05.2020, 12 Uhr, per Mail

### Aufgabe 2.1: Grundlagen Konfigurationsmanagement

Beantworten Sie die folgenden Fragen so kurz wie möglich und so ausführlich wie nötig.

- Nennen und beschreiben Sie die fünf in der Vorlesung behandelten *Arbeitsgebiete* des Konfigurationsmanagements.
- Definieren Sie folgende Begriffe des Konfigurationsmanagements: *Dokument*, *Versionobjekt*, *Variante*, *Revision*, *Konfiguration*, *Baseline*, *Release*.
- Nennen Sie jeweils die Vor- und Nachteile der Speicherung von Versionshistorien von Dateien mittels *Vorwärts-* und *Rückwärtsdeltas*.
- Worin unterscheidet sich ein *pessimistisches* von einem *optimistischen Sperrkonzept*?
- Was ist im Releasemanagement der Unterschied zwischen einem *Soft Freeze* und einem *Hard Freeze*?
- Welche drei möglichen Ergebnisse gibt es beim Änderungsmanagement bei der *Evaluierung eines eingereichten Änderungswunsches*?

### Aufgabe 2.2: Buildmanagement mit make<sup>1</sup>

- Eine freie Windows-Installation von make finden Sie zum Beispiel hier:  
<http://gnuwin32.sourceforge.net/packages/make.htm>  
Für Linux gibt es GNU make unter:  
<https://www.gnu.org/software/make/>
- Außerdem benötigen Sie für diese Aufgabe die freie Software „ImageMagick“:  
<https://www.imagemagick.org/>  
Die Software magick beherrscht das Konvertieren diverser Bildformate. Der Befehl „magick input.jpg output.png“ wandelt z.B. eine .JPEG-Datei in eine .PNG Datei um. In den meisten Linux-Systemen gibt es ebenfalls ein entsprechendes „convert“-Kommando, wie es in dieser Aufgabe verwendet wird.
- Zusätzlich benötigen Sie eine zip-Installation zur Ausführung auf der Kommando-Zeile (z.B. für 7zip wäre das etwa: `7z -a archivname.zip <Dateiliste>`)
- Erstellen Sie nun in einem neuen Verzeichnis eine Datei „makefile“ mit folgendem Inhalt:

```
CONVERT = magick
ARCHIVE = zip
ARCHIVENAME = pictures.zip
OBJECTS = $(patsubst %.jpg, %.png, $(wildcard *.jpg))

DELETE = rm -f

default : $(ARCHIVENAME)

$(ARCHIVENAME) : $(OBJECTS)
```

---

<sup>1</sup> <http://www.gnu.org/software/make/manual/>

## Softwaretechnik II – Sommersemester 2020

```
$(ARCHIVE) $(ARCHIVENAME) $(OBJECTS)
echo $(ARCHIVENAME) created

%.png :    %.jpg
$(CONVERT) $< $@

cleanpng:
echo removing generated files
$(DELETE) *.png

cleanjpg:
echo removing original files
$(DELETE) *.jpg
```

- Legen Sie im gleichen Verzeichnis 2 beliebige jpg-Dateien, eine beliebige gif-Datei und eine beliebige tif-Datei ab (z.B. über Google-Bildersuche), z.B.:
  - tiger.jpg
  - puma.jpg
  - antilope.gif
  - puma.tif
  
- a) Listen Sie alle expandierten Systembefehle auf, die nach Expansion aller Makros beim Aufruf von „make“ in der Systemkonsole ausgeführt werden.
  
- b) Was passiert, wenn Sie den Befehl „make“ erneut aufrufen?
  
- c) Welche Befehle werden ausgeführt, wenn Sie danach die Datei `tiger.jpg` ändern (z.B. mit Paint) und anschließend „make“ aufrufen?
  
- d) Welche Befehle werden ausgeführt, wenn Sie danach `puma.tif` ändern und anschließend „make“ aufrufen?
  
- e) Wie müssen Sie Ihr Makefile ändern, wenn Sie eine neue Datei (z.B. `ente.jpg`) hinzufügen und in ein PNG konvertieren wollen?
  
- f) Wie müssen Sie Ihr Makefile ändern, wenn Sie eine neue Datei `leopard.bmp` hinzufügen und in ein PNG konvertieren wollen?
  
- g) Begründen Sie warum die Makros „cleanpng“ und „cleanjpg“ nicht ausgeführt werden. Wie können die beiden Makros zur Ausführung gebracht werden?