

54. Werkzeuge für das Bau-Management

1

Prof. Dr. rer. nat. Uwe Aßmann
Institut für Software- und
Multimediatechnik
Lehrstuhl Softwaretechnologie
Fakultät für Informatik
TU Dresden
<http://st.inf.tu-dresden.de>
Version 12-0.1, 31.01.13

- 1) Baumanagement
- 2) Das Baumanagement-System von GNU
- 3) Aufgabenmanagement (Fehler, Anforderungen)

In 2012/13 weggelassen
nur zur Info

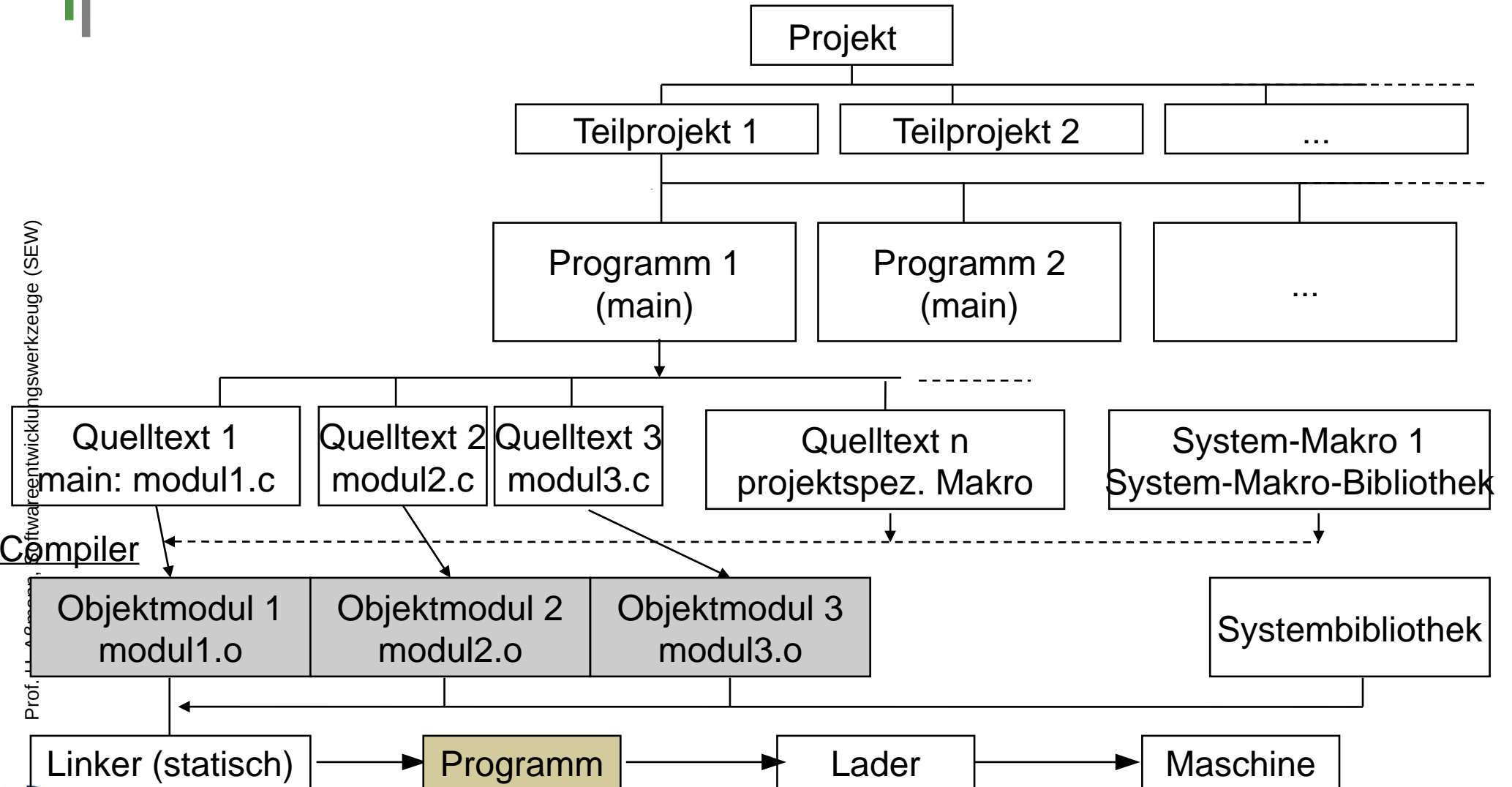
54.1 Bau-Management-Werkzeuge

2

... Kommando:
info automake

Aufbau realer Projekte mit statischem Linken (Technikraum C/C++)

3



Softwareentwicklungswerkzeuge (SEW)
Prof.



- ▶ *Make* ist ein Bauprogramm, das *abhängige Zieldateien (targets)* aus *Quelldateien (sources, Originaldateien)* durch *Ausführen eines Kommandos* erzeugt.
 - Stuart Feldman, 1977, als Teil von UNIX
 - <http://de.wikipedia.org/wiki/Make>
- ▶ Abhängigkeiten werden in **Bau-Spezifikationen (makefiles)** spezifiziert mit Hilfe von *Abhängigkeiten* und *Aktionen*:
 - `B.o: B.c C.h D.h // A ist abhängig von B, C und D`
 - `gcc -c B.c // Aktion, um B.o aus B.c zu generieren`
- ▶ Generische Regel:
 - Falls (Zieldatei älter als eine der Quelldateien)
 - führe die Aktion aus, d.h. regeneriere
- ▶ Im Wesentlichen ist *Make* ein Regelinterpreter mit lazy-Evaluation für die Regeln (träge Auswertung)
 - lazy funktionale Sprache bieten die Funktionalität umsonst an

Makefile für Generierung von pdfs aus tex

5

```
LATEX=latex
```

```
MAIN=document.tex
```

```
SOURCES=$(MAIN) figure.tex
```

```
PDFLATEX=pdflatex
```

```
G=gen
```

```
tex_it: $(SOURCES)
```

```
$(LATEX) $(MAIN) && cp $(MAIN).dvi $(MAIN).dvi~
```

```
dvips -Ppdf -o $G/$(MAIN).ps $(MAIN)
```

```
ps2pdf $G/$(MAIN).ps
```

Generische Regeln

6

```
$G/%.ps: %.pdf
```

```
pdftops $< $@
```

```
psnup -2 $@ > ${@:.ps=}-2p.ps
```

```
psselect -r ${@:.ps=}-2p.ps | psnup -2 | psselect -r > ${@:.ps=}-4p.ps
```

Operatoren (wildcard, basename...)

7

```
BIBS=$(wildcard *.bib)
```

```
MAIN=$(basename $(shell grep -l documentclass *.tex))
```

```
# ##### Replacement Operatoren: #####
```

```
BIBHTMLS=${BIBS:%.bib=$G/%-bib.html}
```

Schleifen

8

architectures:

```
for j in $(ARCHITECTURES) ; do mkdir $$j ; done;
```

links:

```
for j in $(ARCHITECTURES) ; do cd $$j; for i in $(SOURCES) ; do ln -s ../$$i  
.; done; cd ..; done;
```

cleanlinks:

```
for j in $(ARCHITECTURES) ; do cd $$j; for i in $(SOURCES) ; do mv $$i $  
(HOME)/backup; done; cd ..; done;
```

```
for j in $(ARCHITECTURES) ; do cd $$j; rm -f *.c *.cc *.C; cd ..; done;
```


Ant – Bauwerkzeug für Java

9

Zur Build-Beschreibung zusammenhängender Komponenten werden anstatt *Make-Files* durch *Ant* Beschreibungen für *Tasks* eingesetzt, die ein bestimmtes Ziel realisieren.

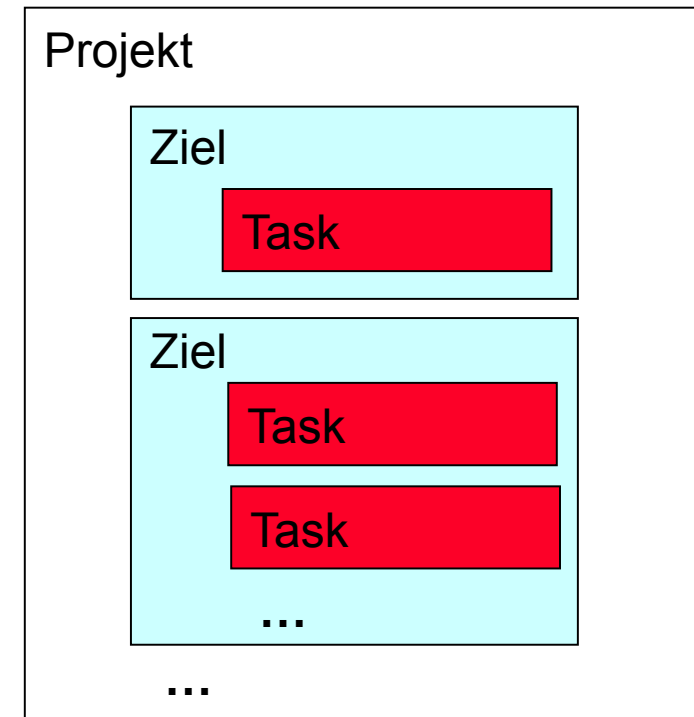
Ant (*Another neat tool*) ist ein **Framework** zur Automatisierung des Build-Prozesses aus Komponenten.

Ein **Ant-Buildfile** build.xml enthält

- genau ein **Projekt**,
- ein bis mehrere **Ziele (Targets)**, die Aspekte des Projektes (z.B. Dokumentation erstellen, Release- oder Debug-Version erstellen...) beschreiben sowie
- **Tasks** meist in Java zur Umsetzung der Ziele.

Es unterstützt Wildcards und Filesets.

Ant stellt „nur“ das Framework zur Organisation und Verwaltung der Projekte und Ziele dar, die weitergehende Logik (z.B. inkrementelles Compilieren) steht in der Verantwortung der Tasks.



Ant (2)

10

Es werden zahlreiche vorgefertigte Tasks mitgeliefert, z.B. für

- Übersetzen (javac, JspC)
- Archivieren (jar, zip, rpm)
- Dokumentation (Javadoc)
- Dateipf ege (Checksum, Copy, Delete, Move...)
- Test (JUnit)

ANT kann mit selbst geschriebenen Tasks (Java-Klassen) erweitert werden.

Aufruf: **ant [options] [target]**

`target` bestimmt dabei, welches im Buildfile aufgeführte Ziel realisiert werden soll, falls der Parameter weggelassen wird, wird das default-target verwendet. Beispielaufrufe:

ant compile Ziel „compile“ umsetzen

ant clean compile Ziele „clean“ und „compile“ umsetzen

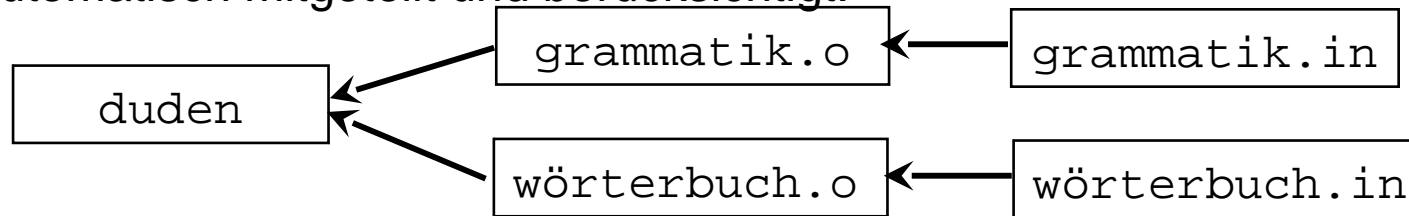
ant -buildfile test.xml Buildfile test.xml verwenden, default-Ziel

Buildfiles werden im XML-Format erstellt, damit einfache Weiterverarbeitung durch Werkzeuge möglich.

Ant XML Tasks (3)

11

Ableitungsbeziehungen der **Komponenten** untereinander bilden einen *Abhängigkeitsgraphen*. Pfeil A <-- B bedeutet „A ist abgeleitet aus B“. Abhängigkeiten werden Task **Javac** automatisch mitgeteilt und berücksichtigt.



```
<project name=„duden“ default=„compile“ basedir=„../..“> Projekt
  <target name="init"> Ziel
    <mkdir dir="{output}"/> Task
    <mkdir dir="{report}"/> Task
  </target>

  <target name="compile" depends="init"> Ziel
    <javac srcdir="{src}/duden" destdir="{output}"/> Task
  </target>
  ...
</project>
```



54.2 Das Baumanagementsystem von GNU

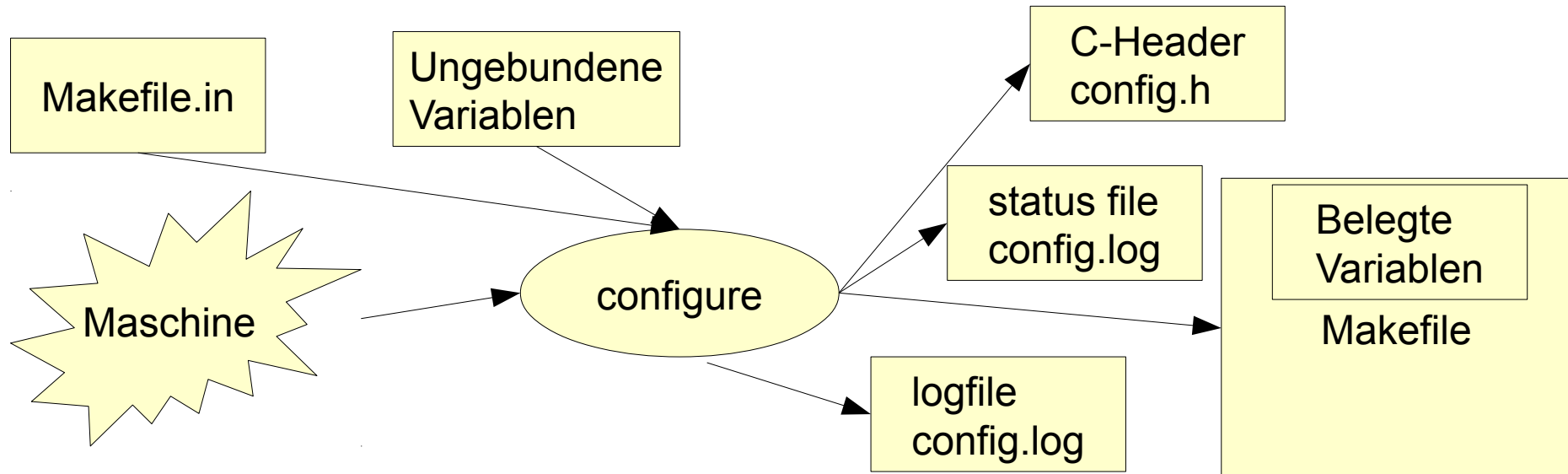
13

... Kommando:
info automake

configure

14

- ▶ <http://www.gnu.org/software/autoconf/>
- ▶ configure ist ein Shellskript, ein Generator für Bau-Spezifikationen (Makefiles)
 - untersucht die Maschine, ob bestimmte Bibliotheken vorhanden sind
 - in welcher Variante sie vorhanden sind
 - Erzeugt parametrisierte Aufrufe an C-Übersetzer und andere Werkzeuge
 - durch das Setzen von Variablen (compilation switches), die Plattformabhängigkeiten ausdrücken (für Makefiles *und* C/C++-Programme)
- ▶ configure sorgt für Portabilität der GNU-Programme

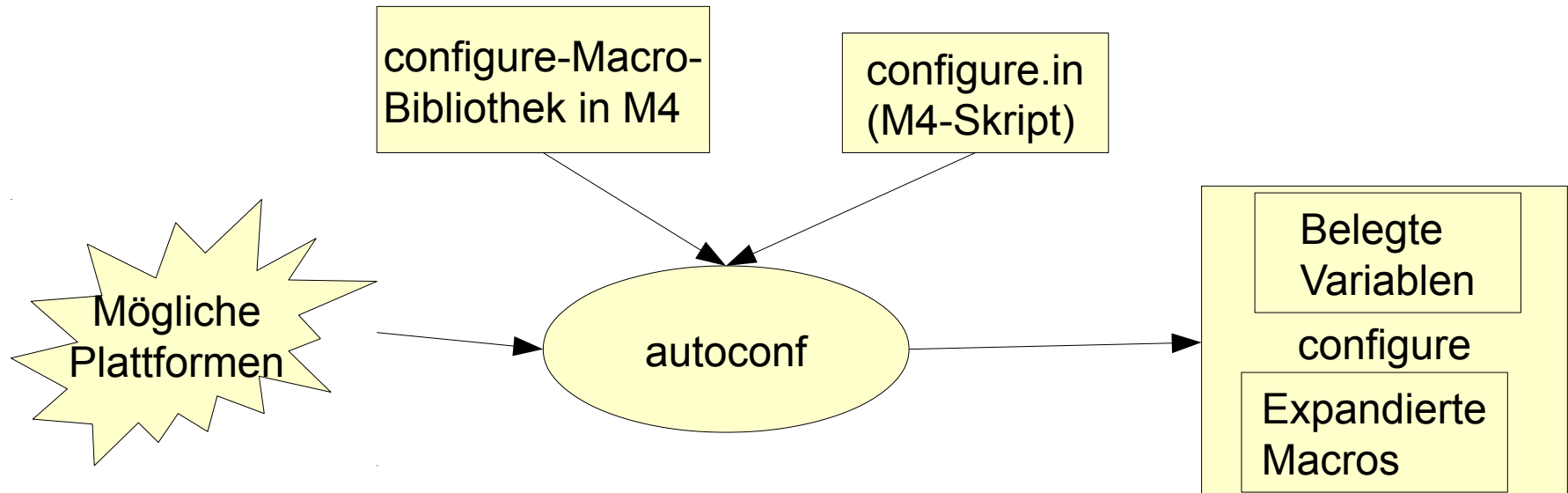


Installation mit configure und make

15

- ▶ unzip <package>
- ▶ ./configure
 - // Untersucht Maschine. Hinterlässt Makefile, config.log, config.status, config.h
- ▶ ./make
 - // baut das System
- ▶ ./make check
 - // Prüft, ob alles gutging
- ▶ ./make install
 - // installiert das Programm vom lokalen Dir in ein globales
- ▶ ./make uninstall
 - // deinstalliert das Programm

- ▶ <http://www.gnu.org/software/autoconf/>
- ▶ Autoconf generiert configure-Skripte aus M4-Präprozessorskripten, d.h. M4-Makros (wie Funktionen aufzurufen)
 - die Makros erzeugen Abschnitte des configure-Skriptes
 - Autoconf Makroarchiv: <http://ac-archive.sourceforge.net/>



M4-Makros für Java-Entwicklung

17

- ▶ <http://ac-archive.sourceforge.net> ist ein Projekt, das M4-Makros für die Erzeugung von configure-Skripten sammelt
- ▶ Zum Beispiel
http://ac-archive.sourceforge.net/ac-archive/ac_try_run_javac.html:
 - ▶ `ac_check_class (ac-archive) // prüft, ob Klasse vorhanden`
 - ▶ `ac_check_junit (ac-archive) // prüft, ob junit vorhanden`
 - ▶ `ac_check_rqrd_class (ac-archive) // etc.`
 - ▶ `ac_java_options (ac-archive)`
 - ▶ `ac_prog_jar (ac-archive)`
 - ▶ `ac_prog_java (ac-archive)`
 - ▶ `ac_prog_java_cc (ac-archive)`
 - ▶ `ac_prog_java_works (ac-archive)`
 - ▶ `ac_prog_javac (ac-archive)`
 - ▶ `ac_prog_javac_works (ac-archive)`
 - ▶ `ac_prog_javadoc (ac-archive)`
 - ▶ `ac_prog_javah (ac-archive)`
 - ▶ `ac_try_compile_java (ac-archive)`
 - ▶ `ac_try_run_javac (ac-archive)`

Autoproject

18

- ▶ <http://packages.debian.org/unstable/devel/autopproject>
- ▶ autopproject initialisiert in einem leeren Directory ein GNU-konformantes Projekt
 - installierbar auf vielen Plattformen (deployable) mit Hilfe des Werkzeuges *configure*
 - varianten-konfigurierbar
 - erzeugt ein Baumanagement-Sytem
 - das mit *autoconf* configure-skripten erzeugen kann
 - *und automake*

54.3 Aufgabenmanagement (Fehler- und Änderungsmanagement)

19

Es beinhaltet die Erfassung, Registratur, Speicherung und Verfolgung von

- Fehlermeldungen
- Änderungsanforderungen

und steuert alle dazu notwendigen Vorgänge:

- **Bewertung** der Änderung: Nach der Notwendigkeit der Auswirkungen, wobei jede Änderung einer Version zu einer neuen Version führt
- **Planung und Entscheid** des Änderungsvorgehens: Vorgehen nach einer definierten Änderungsprozedur mit Durchlauf geforderter Zustände(V-Modell).
- Einleitung und Überwachung der **Änderungsdurchführung**: Darstellung des Änderungsgeschehens in einem Logbuch, dessen Eintragungen sich immer auf definierte Versionen von Komponenten beziehen. Daraus sollen Änderungsstatistiken auf Abruf generierbar sein.
- Abschluß und **Auswertung der Änderung**:
 - Alle Änderungen sollen nachvollziehbar und rekonstruierbar sein.
 - Es ist eine Historie zu führen, die alle Änderungsdaten einschließlich eines ausführlichen Kommentars enthält.
 - Als Vorgehensbaustein „Problem- und Änderungsmanagement“ im V-Modell XT realisiert.

Aufgabenmanagement-Tools

21

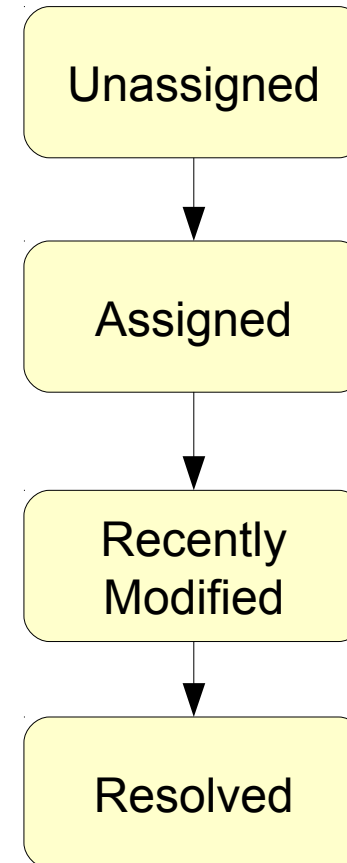
Bugzilla	Mozilla (OSS)	www.bugzilla.org
Mantis	OSS	http://www.mantisbt.org/

http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems

Aufgabenmanagement (Fehler und Änderungen) mit Mantis

22

- ▶ Mantis ist ein webbasiertes Aufgabenmanagement-System (issue tracking system)
 - Zustandsmaschine für Fehler und Änderungswünsche
 - Generierung von emails über Statusänderungen
 - Visualisierung von Zuständen und Bearbeitern



Überblick über Aufgaben [www.mantisbt.org]

23

Prof. U. Alßmann, Softwareentwicklungswerkzeuge (SEW)

The screenshot shows the MantisBT web interface. At the top, there is a navigation bar with the Mantis logo and the text "MANTIS". Below this, there are links for "Anonymous", "Login", and "Signup for a new account". The date and time are displayed as "2009-01-28 14:23 EST". The project is set to "Demo".

The main content area displays a list of issues. The table has the following columns: P, ID, US\$, #, @, Category, Severity, Status, Updated, and Summary. The issues are listed in descending order of ID.

P	ID	US\$	#	@	Category	Severity	Status	Updated	Summary
	0005069		3		GUI	minor	resolved (MacGyver)	2009-01-28	Invalid Password
	0005095				Other	minor	assigned (mow)	2009-01-28	error prueba
	0005091				Other	trivial	confirmed (ossgevalt)	2009-01-28	foo bar baz
	0004707				Other	minor	resolved (000willberty)	2009-01-28	relationships I
	0005093				GUI	minor	assigned (enzyme)	2009-01-28	Test fichier attache
	0005082				Other	minor	assigned (121212)	2009-01-28	test
	0005094				GUI	minor	new	2009-01-28	Test fichier attache
	0005092				GUI	minor	new	2009-01-28	Test fichier attache
	0005090				GUI	minor	assigned (flandersen)	2009-01-28	Needs attention
^	0005089		3		Website	major	resolved (Russell)	2009-01-28	Problems loading JavaScript on Main.html
^^	0005076		1		GUI	feature	assigned (darksaboteur)	2009-01-28	I can't poo...
^^	0005088				GUI	tweak	assigned (abarbosa)	2009-01-27	sadasdasda
	0005087				GUI	trivial	assigned (ramyap)	2009-01-27	asdasddsad
v	0005086		9		GUI	trivial	assigned (Element)	2009-01-27	Detailbereich wird nicht angezeigt
	0005083		1		Other	minor	acknowledged (aaloc)	2009-01-27	test
	0005085				GUI	major	assigned (celso)	2009-01-27	teste com erro e associacão com arquivo de repositório
	0005081				GUI	minor	resolved (deepak84)	2009-01-27	Sql Error
^	0005080				GUI	feature	assigned (mmiat)	2009-01-27	????
	0005079				GUI	minor	assigned (kedar)	2009-01-26	test
	0005078		4		GUI	text	assigned (jodji)	2009-01-26	test
	0005077				GUI	minor	new	2009-01-26	cant abc
	0005065		3		GUI	tweak	new	2009-01-26	hhhh
	0005074		2		Other	minor	assigned (maxadmin)	2009-01-26	VIZ funktioniert nicht
^	0005075				Website	major	assigned (patriciagomes)	2009-01-26	Erro no site
^	0005071		2		Website	major	assigned (deepak84)	2009-01-26	Website Issue
	0005072				GUI	minor	assigned (tym)	2009-01-26	Alles Scheiße



Überblick über Zustände der Aufgaben

24

Prof. U. Alßmann, Softwareentwicklungswerkzeuge (SEW)

The screenshot shows the MantisBT DEMO site interface. At the top, there is a navigation bar with links for [Main](#), [My View](#), [View Issues](#), [Change Log](#), [Roadmap](#), [Docs](#), [Wiki](#), and [Billing](#). The current date and time are 2009-01-28 14:21 EST. The project is set to 'Demo'. The main content area is divided into three sections:

- Unassigned [^] (1 - 10 / 271)**: A list of 10 unassigned tasks, each with a unique ID, description, category, and timestamp. The tasks are color-coded in shades of red and pink.
- Resolved [^] (1 - 10 / 154)**: A list of 10 resolved tasks, each with a unique ID, description, category, and timestamp. The tasks are color-coded in shades of green.
- Recently Modified [^] (1 - 10 / 2234)**: A list of 10 recently modified tasks, each with a unique ID, description, category, and timestamp. The tasks are color-coded in various colors including grey, blue, yellow, and green.

The interface also includes a search bar, a sidebar with navigation links, and a footer with the ST logo.



The End