

# Taskwarrior

Aufgabenverwaltung auf der Kommandozeile

---



Dirk Deimeke

17. März 2019

Taskwarrior Academy @ Chemnitzer Linux-Tage

**Dirk Deimeke – [d5e.org](https://d5e.org)**

1. Zuerst erzähle ich Euch ein bisschen über Zeitmanagement.
2. Dann erkläre ich, wie Taskwarrior helfen kann, sich zu organisieren.
3. Anschliessend gibt es drei Möglichkeiten, wie es weiter geht: **Ihr habt die Wahl!**
  - Ich erzähle noch etwas über die Philosophie von Taskwarrior.
  - Wir machen ein ausgedehntes Frage- und Antwort-Gespräch.
  - Ihr fragt und ich setze Dinge in Taskwarrior um.

# Zeitplanung und Selbstmanagement

Es gibt kaum etwas individuelleres als Zeitplanung und Selbstmanagement. Über das Thema sind meterweise Bücher geschrieben worden und es gibt viele Methoden und Techniken, die von sich behaupten, die richtige zu sein.

Wichtig ist, dass es sich um **Zeitplanung**, nicht um Zeitmanagement handelt:  
*Zeit kann man nicht managen, nur den sinnvollen Umgang mit ihr.*

Jeder Tag hat für jeden Menschen die gleiche Länge, daran ändert keine Zeit-Management-Methode etwas.

**Alle Tage sind gleich lang,  
aber unterschiedlich breit.**

# Beispiele für Methoden und Techniken

Hier eine *kleine* Auswahl von Methoden und Techniken.

- Pomodoro
- Personal Kanban
- Getting Things Done
- A.L.P.E.N.-Methode
- Eisenhower-Prinzip
- Ziele sind S.M.A.R.T.
- Paretoprinzip
- To-Do-Liste
- Autofocus-System
- Final Version
- Inbox Zero
- The Cycle System
- Emergent Task Planner
- Strikethru (kein Tippfehler)
- (es gibt noch viele mehr)

## Techniken und Werkzeuge

Erst, wenn man verschiedene Techniken kennt, ist man auch in der Lage, die passende Methode oder – besser – den passenden Methodenmix für sich selber zu wählen bzw. zusammenzustellen.

### **Hinweis:**

Bitte nicht die Zeit- und Selbstmanagementmethode nach dem Werkzeug auswählen! Lieber für eine Methode entscheiden und das dazu passende Werkzeug suchen.

Wenn man sich allerdings für ein Werkzeug entscheidet, dann ist es wichtig, dass das Tool unabhängig von der gewählten Methode funktioniert.

**Die eigene Arbeitsweise ist einem stetigen Wandel unterworfen.**

**Welche Anforderungen habt Ihr an ein Werkzeug,  
dass Euch unterstützen soll?**



## Das richtige und passende Werkzeug

Das richtige Werkzeug zum Zeit- und Selbstmanagement hat meiner Ansicht nach die folgenden sechs Eigenschaften:

- Es ist immer dabei.
- Es ist zuverlässig.
- Es ist unabhängig von der Methode, die ich verwenden möchte.
- Es steht nicht im Weg.
- Es lenkt den Fokus auf wenige Aufgaben, die es zu erledigen gilt.
- Es ist ein Werkzeug, das ich gerne benutze.

**Taskwarrior**

## Aufgabenverwaltung auf der Kommandozeile?

Selbst die hartnäckigsten Kommandozeilennutzer unter meinen Kollegen halten mich für einen Sonderling, weil sie sich nicht vorstellen können, ihre Aufgaben auf der Kommandozeile zu verwalten.

## Vorteile?

Welchen Vorteil bringt es nun für mich, ein entsprechendes Werkzeug einzusetzen?

Nun, es erfüllt direkt vier der Eigenschaften, die ich im letzten Abschnitt genannt habe.

## Es ist immer dabei

Ein sehr grosser Teil meiner täglichen Arbeit – deutlich mehr als 80% – findet auf der Kommandozeile statt, damit ist Taskwarrior per Definition schon «immer dabei».

Und ja, ich habe auch eine Kommandozeile – mittels **Termux** – auf meinem Mobiltelefon, und so Taskwarrior immer dabei.

Es gibt einen **Taskserver**, der Aufgaben zwischen verschiedenen Instanzen **mit der gleichen Version** synchronisieren kann.

## Es steht nicht im Weg

Taskwarrior hat sehr schnelle Antwortzeiten.

```
$ /usr/bin/time -v task add "Neue Aufgabe"
```

```
Created task 206.
```

```
    Command being timed: "task add Neue Aufgabe"
```

```
...
```

```
    Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.01
```

```
...
```

In meiner privaten Taskwarrior-Datenbank sind rund 8500 Tasks.

```
$ /usr/bin/time -v task next
```

```
    Command being timed: "task next"
```

```
...
```

```
    Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.05
```

```
...
```

## Es ist ein Werkzeug, das ich gerne benutze

Selbstverständlich ist Taskwarrior ein Programm, das ich sehr gerne nutze, sonst würde ich mich ja auch nicht dafür engagieren oder diesen Vortrag halten.

Das Dateiformat ist robust und hat sich in den letzten Jahren nicht verändert.

Probleme, mit denen ich seit 2010 zu tun hatte, waren der Entwicklerversion oder dem Mischen von verschiedenen Versionen in der Synchronisation geschuldet.



# Einfachheit von Taskwarrior

Es kann so einfach wie in diesem Beispiel sein:

```
$ task add Vortrag für die Chemnitzer Linux-Tage vorbereiten
```

```
$ task list
```

```
$ task 1 done
```

# Komplexe Anwendungsfälle

Aber es können auch deutlich komplexere Anwendungsfälle abgedeckt werden.

```
$ task add \  
  project:privat.korrespondenz \  
  +weihnachten \  
  due:2019-12-17T10:00:00Z \  
  scheduled:2019-12-01 \  
  wait:2019-11-15 \  
  until:due +7days \  
  Weihnachtskarten verschicken
```

```
$ task add \  
  project:job.meeting \  
  due:eom \  
  priority:M \  
  +boring \  
  +important \  
  recurr:monthly \  
  Prepare Meeting\
```

Als User steht man vor der Wahl, die komplexeren Features zu benutzen, Taskwarrior selber zwingt einen aber nicht dazu.

## Funktionen oder Features

Bei den Funktionen und Features möchte ich mich hier insbesondere um die Forderung kümmern, dass ein gutes Werkzeug zur Aufgabenverwaltung den **Fokus auf wenige Aufgaben** lenkt, die es zu bearbeiten gilt.

Wie schon häufiger erwähnt, kann man *alle Features* nutzen, man muss aber nicht.

Der Standardreport in Taskwarrior heisst «next», sein Ziel ist es, Aufgaben zu präsentieren, die als nächstes erledigt werden sollen.

## Urgency – Dringlichkeit

Um die nächsten Aufgaben zu präsentieren ist der «next»-Report nach Dringlichkeit in absteigender Reihenfolge sortiert.

Die Dringlichkeit in Taskwarrior ist das Ergebnis einer Berechnung in die mehr als ein dutzend verschiedene Faktoren einfließen. Die Gewichtung der einzelnen Faktoren ist frei konfigurierbar.

Die Koeffizienten, die die Standardgewichtung ergeben, sind das Ergebnis jahrelanger Erfahrung, ändert diese bitte nur, wenn sich die Dringlichkeit nicht mit Eurer Einschätzung deckt.

## Das Alter einer Aufgabe

Einer der Faktoren mit einem geringen Einfluss ist das Alter einer Aufgabe.

Im folgenden Beispiel steht «yesterday» für den Start – 0:00 Uhr – des gestrigen Tages und «today» entsprechend für 0:00 Uhr heute.

# Einfluss des Alters auf die Dringlichkeit

```
[dirk@taskwarrior ~]$ task add entry:yesterday Aufgabe Eins  
Created task 1.
```

```
[dirk@taskwarrior ~]$ task add entry:today Aufgabe Zwei  
Created task 2.
```

```
[dirk@taskwarrior ~]$ task  
[task next]
```

```
ID Age Description Urg  
1 1d Aufgabe Eins 0.01  
2 9h Aufgabe Zwei 0
```

```
2 tasks
```

Spalte «Urg» zeigt, dass «Aufgabe Eins» dringender gewichtet ist als «Aufgabe Zwei».

# Informationen

```
[dirk@taskwarrior ~]$ task 1 information
```

Name	Value
ID	1
Description	Aufgabe Eins
Status	Pending
Entered	2019-03-08 00:00:00 (1d)
Last modified	2019-03-09 09:26:18 (1min)
Virtual tags	PENDING READY UNBLOCKED
UUID	e154ee33-e6bf-455a-bc23-beff72f4e3bc
Urgency	0.01

```
age 0.003 * 2 = 0.005
          -----
          0.01
```

Der Koeffizient für das Alter hat den Wert 2 und erhöht sich für jeden vollen Tag um  $2 * 1/365$ .

## Bereits definierte Koeffizienten

Config Variable	Value
<code>urgency.active.coefficient</code>	4.0
<code>urgency.age.coefficient</code>	2.0
<code>urgency.age.max</code>	365
<code>urgency.annotations.coefficient</code>	1.0
<code>urgency.blocked.coefficient</code>	-5.0
<code>urgency.blocking.coefficient</code>	8.0
<code>urgency.due.coefficient</code>	12.0
<code>urgency.inherit</code>	off
<code>urgency.project.coefficient</code>	1.0
<code>urgency.scheduled.coefficient</code>	5.0
<code>urgency.tags.coefficient</code>	1.0
<code>urgency.uda.priority.H.coefficient</code>	6.0
<code>urgency.uda.priority.L.coefficient</code>	1.8
<code>urgency.uda.priority.M.coefficient</code>	3.9
<code>urgency.user.tag.next.coefficient</code>	15.0
<code>urgency.waiting.coefficient</code>	-3.0

Die Koeffizienten auf der linken Seite – das Kommando lautet `task show urgency` – sind im Standard definiert und natürlich konfigurierbar.

- Wenn eine Aufgabe Anmerkungen hat, einem Projekt zugewiesen ist, einen Tag zugewiesen hat oder ein Stichtermin eingetragen wurde, wird das in den entsprechenden Koeffizienten berücksichtigt.
- Eine Aufgabe, die eine andere blockiert wird höher bewertet und sollte vor der Aufgabe, die blockiert wird, erledigt werden.
- Die Prioritäten «High», «Medium» und «Low» werden genauso berücksichtigt, wie ein manuell gesetzter Tag «next», der die Dringlichkeit um ganze 15 Punkte verändert.

Mehr Informationen unter [Urgency](#).



## Wait – Aus den Augen, aus dem Sinn

Das Setzen eines Wait-Datums (oder sogar Zeit) ist eine weitere Möglichkeit, die Aufgabenliste übersichtlich zu halten. Alle Datumsformate sind frei definierbar, im Standard unterstützt Taskwarrior neben den selbst definierten Formaten noch das **ISO-8601 Format**.

```
[dirk@taskwarrior ~]$ task 1 mod wait:2019-12-24
Modifying task 1 'Aufgabe Eins'.
Modified 1 task.
```

```
[dirk@taskwarrior ~]$ task
[task next]
```

```
ID Age Description Urg
 2 9h Aufgabe Zwei    0
```

```
1 task
```

# Wait und Information

```
[dirk@taskwarrior ~]$ task 1 information
```

```
Name          Value
ID             1
Description    Aufgabe Eins
Status         Waiting
Entered        2019-03-08 00:00:00 (1d)
Waiting until  2019-12-24 00:00:00
Last modified  2019-03-09 09:30:18 (43s)
Virtual tags   UNBLOCKED WAITING
UUID           e154ee33-e6bf-455a-bc23-beff72f4e3bc
Urgency        -2.995
```

```
waiting      1 *   -3 =   -3
age          0.003 * 2 = 0.005
            -----
            -2.995
```

## Wait und Information (Fortsetzung)

Date	Modification
2019-03-09 09:30:18	Wait set to '2019-12-24 00:00:00'. Status changed from 'pending' to 'waiting'.

## Der «waiting» Report

Da die wartenden Aufgaben nicht für immer – respektive bis zum Zieldatum – verschollen sein sollen, gibt es den «waiting»-Report, der alle wartenden Aufgaben anzeigt.

```
[dirk@taskwarrior ~]$ task waiting
```

```
ID Age Wait      Remaining Description
  1 1d  2019-12-24    9mo Aufgabe Eins
```

```
1 task
```

Anmerkung: Intern speichert Taskwarrior alle Datumfelder als verstrichene Sekunden seit dem 01.01.1970 («The Epoch»), was auch *Unix Zeit* genannt wird. Erst bei der Darstellung wird dieser Sekundenwert auf die benutzte Zeitzone umgerechnet.

## Recurrence – Wiederkehrende Aufgaben

Taskwarrior erlaubt auch, wiederkehrende Aufgaben zu erstellen.

So sorgt beispielsweise das folgende Kommando dafür, dass allmonatlich ein Termin «Miete zahlen» erstellt wird, der Ende des Monats fällig ist.

```
[dirk@taskwarrior ~]$ task add due:eom recur:monthly Miete zahlen  
Created task 3 (recurrence template).
```

```
[dirk@taskwarrior ~]$ task  
[task next]
```

ID	Age	Recur	Due	Description	Urg
4	-	P30D	3w	Miete zahlen	2.4
2	9h			Aufgabe Zwei	0

```
2 tasks
```

```
Creating recurring task instance 'Miete zahlen'
```

## Funktionsweise Wiederkehrende Aufgaben

Intern wird dazu eine Musteraufgabe angelegt, die bei jedem Verstreichen des Stichtermins eine neue Aufgabe mit dem neuen Stichtermin für den Folgemonat anlegt.

Das «eom» im Beispiel steht für «end of month». Taskwarrior kennt eine ganze Reihe solcher Synonyme, die im Abschnitt **Synonyms** auf der Webseite zu Daten in Taskwarrior erläutert werden.

Auch bei den Wiederholungsfrequenzen sind bereits einige vordefiniert wie beispielsweise «daily» oder «weekdays» oder viele andere mehr.

Die Manpage – «man task» – ist dafür ein guter Anlaufpunkt.

## Das kann sich bei zukünftigen Versionen ändern!

Die aktuelle Version von Taskwarrior ist 2.5.1, mit Version 3.0.0 wird das Feature, wiederkehrende Aufgaben anlegen zu können, **komplett überarbeitet** (wir sprechen von «recurrence overhaul») und erlaubt dann auch komplexere und selbstdefinierte Wiederholungsfrequenzen.

# Ausgewählte Synonyme für Datum und Wiederholungsintervall

## Relative wording

task ... due:today

task ... due:yesterday

task ... due:tomorrow

## Day number with ordinal

task ... due:23rd

task ... due:3wks

task ... due:1day

task ... due:9hrs

## Start of next (work) week (Monday), calendar week (Sunday or Monday), month, quarter and year

task ... due:sow

task ... due:soww

task ... due:socw

task ... due:som

task ... due:soq

task ... due:soy

daily, day, 1da, 2da, ...

Every day or a number of days.

weekdays

Mondays, Tuesdays, Wednesdays,  
Thursdays, Fridays and skipping  
weekend days.

weekly, 1wk, 2wks, ...

Every week or a number of weeks.

biweekly, fortnight

Every two weeks.

monthly, month, 1mo, 2mo, ...

Every month.

quarterly, 1qtr, 2qtrs, ...

Every three months, a quarter, or a  
number of quarters.

semiannual

Every six months.



Die vorgestellten Kommandos sollten einen Eindruck davon vermitteln, wie Taskwarrior dabei unterstützt, die Aufgaben zu erledigen, die gerade dringend sind und hilft, den Fokus auf wenige Aufgaben zu setzen.

# Kommandoreferenz – Cheat Sheet

Es gibt noch sehr viel mehr in Taskwarrior, wie die **Kommandoreferenz** zeigt.

**Syntax:** task <override> <filter> <command> <modifications>

### Runtime Override

```
task <override> <value>  
task <override> <value>
```

### Config File Override

```
task <override> <value>  
task <override> <value>
```

### Parser Override

```
task <override> <value>
```

### Configuration

```
task <override> <value>  
task <override> <value>
```

### UDA Configuration

```
task <override> <value>  
task <override> <value>
```

### Sync Configuration

```
task <override> <value>  
task <override> <value>
```

### Date Formats

```
task <override> <value>  
task <override> <value>
```

### Annotations

```
task <override> <value>  
task <override> <value>
```

### Annotation en-passant

```
task <override> <value>  
task <override> <value>
```

### Synchronize

```
task <override> <value>  
task <override> <value>
```

### Import/Export

```
task <override> <value>  
task <override> <value>
```

### Filter

Expression of terms with values, operators.

```
task <override> <value>
```

### Terms

```
task <override> <value>  
task <override> <value>
```

### ID Ranges - use ID or UID

```
task <override> <value>  
task <override> <value>
```

### Searching for text

```
task <override> <value>  
task <override> <value>
```

### Operators

```
task <override> <value>  
task <override> <value>
```

### Attribute Modifiers

```
task <override> <value>  
task <override> <value>
```

### Report Commands

(Run task reports)

```
task <override> <value>  
task <override> <value>
```

### Write Commands

```
task <override> <value>  
task <override> <value>
```

### Read Commands

```
task <override> <value>  
task <override> <value>
```

### Misc Report Commands

```
task <override> <value>  
task <override> <value>
```

### Modifications

```
task <override> <value>  
task <override> <value>
```

### Date Symbols

```
task <override> <value>  
task <override> <value>
```

### Document Object Model (DOM)

```
task <override> <value>  
task <override> <value>
```

### Examples

```
task <override> <value>  
task <override> <value>
```

### Abbreviations

```
task <override> <value>  
task <override> <value>
```

### Task Attributes

```
task <override> <value>  
task <override> <value>
```

### Special Tags

```
task <override> <value>  
task <override> <value>
```

### Virtual Tags

```
task <override> <value>  
task <override> <value>
```

**Copyright (C) 2006 - 2016**  
P Hertzberg  
D Dreyfus  
Revised: 2018-11-25

- Taskwarrior ist immer dabei.
- Taskwarrior ist zuverlässig.
- Taskwarrior ist unabhängig von der Methode, die ich verwenden möchte.
- Taskwarrior steht nicht im Weg.
- Taskwarrior lenkt den Fokus auf wenige Aufgaben, die es zu erledigen gilt.
- Taskwarrior ist ein Werkzeug, das ich gerne benutze.

Vielen Dank!

Dirk Deimeke, 2019, CC-BY

[dirk@deimeke.net](mailto:dirk@deimeke.net)

[d5e.org](https://d5e.org) – [speakerdeck.com/ddeimeke](https://speakerdeck.com/ddeimeke)

PDF bei Speakerdeck herunterladen, dann sind die Links klickbar.

Die Homepage von [Taskwarrior](#) enthält unter anderem [Beispiele](#), inklusive einer 30-Sekunden-Einführung und auch [Muster-Workflows](#).

Basis dieses Vortrags ist ein Artikel, den ich im August 2017 für Informatik Aktuell geschrieben habe: [Taskwarrior – Aufgabenverwaltung auf der Kommandozeile](#)

In einem anderen Vortrag zu [Zeit- und Selbstmanagement](#) finden sich Tipps und Hinweise darauf, wie man sich selber organisieren kann.