

[Anterior](#) [Home](#) [Próximo](#)

4.2.08 Projektschablonen (Fortsetzung)

Anleitung zum Schablonenbau

Als Beispiel soll ein einfaches Projekt angelegt werden, aus einem Oberprojekt und zwei Unterprojekten mit jeweils zwei Arbeitspaketen. Das Projekt hat zwei Meilensteine. Die Arbeitspakete des ersten Unterprojektes sind nacheinander durchzuführen.

Standardprojekt.projectTemplate

```
// Standard-Projekt
```

```
[Project]:P1
```

```
Name:Hauptprojekt Web $Name
```

```
ParentProject:
```

```
Customer:$Customer
```

```
Contact:$Contact
```

```
Start:$Start
```

```
End:$Start + 50d
```

```
ProjectLeader:
```

```
Currency:EURO
```

```
CostCenter:Technik
```

```
[Project]:P2
```

```
Name:Analyse $Name
```

```
ParentProject:P1
```

```
Start:$P1.Start
```

```
End:$P1.Start + 30%
```

```
[Project]:P3
```

```
Name:Produktion $Name
```

```
ParentProject:P1
```

```
Start:$P2.End
```

```
End:$P1.End - 10%
```

```
[Milestone]:M1
```

```
Project:P1
```

```
Name:Konzeptabnahme
```

```
Date:$P1.Start + 25%
```

```
[Milestone]:M2
```

```
Project:P1
```

```
Name:Testbetrieb
```

```
Date:$P1.End - 15%
```

```
[Job]:J1
```

```
Project:P2
```

Name:Pflichtenheft erstellen
Occupation:Analyse
Duration:12
Start:\$P1.Start
End:\$P1.Start + 5d

[Job]:J2
Project:P2
Name:Konzeptgespräche
Occupation:Beratung
Duration:20
Start:\$P1.Start + 7d
End:\$P1.Start + 14d

[Job]:J3
Project:P3
Name:HTML-Produktion
Occupation:Programmierung
Duration:24
Start:\$P1.Start + 25%
End:\$P1.End - 25%

[Job]:J4
Project:P3
Name:Testen
Occupation:Programmierung
Duration:20
Start:\$P1.End - 50%
End:\$P1.End

[Timing]:T1
ObjectA:J1
ObjectB:J2
Timing:Ende-Start

Die erste Zeile enthält einen Kommentar:

```
// Standard-Projekt
```

Zeilen, die mit // beginnen werden nicht beachtet.

Daran anschließend folgen 3 Projektdefinitionen (PROJECT), 2 Meilensteindefinitionen (MILESTONE), 4 Arbeitspaketdefinitionen (JOB) und eine Definition des Projektablaufes (TIMING).

Grundsätzlich kann eine Schablone beliebige viele Definitionen enthalten. Allerdings gibt es folgende Regeln für die Anordnung:

- Die Reihenfolge der Definitionen ist: Projekte, Meilensteine, Arbeitspakete, Projektablaufe
- Es muss genau ein Hauptprojekt geben (Projekt ohne Oberprojekt)
- Die Projekte müssen nach ihrer Hierarchie geordnet aufgeführt sein: erst das Hauptprojekt, dann die Unterprojekte usw.

Als erste Definition im Beispiel ist das Hauptprojekt aufgeführt:

```
[Project]:P1 Name:Hauptprojekt Web $Name ParentProject: Customer:$Customer Contact:$Contact
Start:$Start End:$Start + 50d ProjectLeader: Currency:EURO CostCenter:Technik
```

Jede Projektdefinition beginnt mit [Project]:<Projektkennung>. Die Projektkennung für das Hauptprojekt im Beispiel lautet P1. Diese Kennung dient nur der internen Bezugnahme auf dieses Projekt und erscheint nicht später im erzeugten Projekt. Die Kennung ist wahlfrei, sollte aber keine Leerzeichen oder Sonderzeichen enthalten. Jede Projektdefinition muss eine eigene Projektkennung haben.

Darunter werden die wichtigsten Felder und Kennzahlen des Projekts aufgeführt.

- **Name:** die Bezeichnung des Projekts - in diesem Fall wird die Bezeichnung aus "Hauptprojekt Web" und

\$Name zusammengesetzt; \$Name ist die Bezeichnung, die der Benutzer beim Erzeugen des Projektes aus der Schablone eingibt

- **ParentProject:** die Projektkennung des Oberprojektes - in diesem Falle kein Oberprojekt
- **Customer:** der Kunde, für den das Projekt erstellt wird - in diesem Fall ist der Kunde \$Customer, was bedeutet, daß der Benutzer den Kunden beim Anlegen des Projektes aus der Schablone eingibt
- **Contact:** der Ansprechpartner beim Kunden - in diesem Falle: \$Contact, d.h. der Nutzer gibt den Ansprechpartner ein
- **Start:** das Startdatum des Projekts - in diesem Falle: \$Start, d.h. der Nutzer gibt das Startdatum ein
- **End:** das Enddatum des Projekts - in diesem Falle: \$Start + 50d, d.h. das Enddatum liegt 50 Tage nach dem Startdatum
- **ProjectLeader:** Projektleiter - in diesem Fall ist kein Projektleiter angegeben, d.h. der Nutzer wird zur Eingabe aufgefordert
- **Currency:** die Projektwährung
- **CostCenter:** Kostenstelle

Verschiedene Variablen stehen zur Verfügung, welche die Benutzereingaben aus dem Dokumenttyp enthalten, die zum Erzeugen des Projektes benutzt werden:

\$Name - Bezeichnung \$Customer - Kunde \$Contact - Ansprechpartner \$Currency - Währung
\$CostCenter - Kostenstelle \$ProjectLeader - Projektleiter \$Start - Beginndatum \$End - Enddatum

Die Definition weiterer Projekte ist jetzt erheblich einfacher, denn viele Werte werden aus dem Hauptprojekt übernommen. Aufgeführt sind nur noch die abweichenden Felder:

```
[Project]:P2 Name:Analyse $Name ParentProject:P1 Start:$P1.Start End:$P1.Start + 30%
```

Dieses Projekt hat die Kennung P2 und den Namen "Analyse \$Name". \$Name ist wieder der Platzhalter für die Eingabe des Nutzers im Feld Bezeichnung. Gibt der Nutzer hier zum Beispiel "Artworks" ein, dann ist die Projektbezeichnung "Analyse Artwoks".

Als Oberprojekt wird das Projekt mit der Kennung P1 ausgewählt, also das Hauptprojekt. Alle fehlenden Angaben werden aus diesem Oberprojekt übernommen.

Der Starttermin dieses Projektes wird auf den Starttermin des Oberprojektes gesetzt. Hierzu wird die

Variable \$P1.Start angegeben, welche sich aus der Projektkennung des verwendeten Projekts gefolgt von "." und dem Feldnamen zusammensetzt. Der Endtermin wird auf Starttermin + 30% gesetzt. Eine Prozentangabe bei der Dauer bezieht sich immer auf den Zeitraum des Hauptprojektes.

Analog wird das zweite Unterprojekt P3 angelegt.

Es folgt die Definition der Meilensteine. Eine Meilensteindefinition beginnt mit [Milestone]:<Meilensteinkennung>. Dann folgt die Definition der Werte eines Meilensteins:

[Milestone]:M1 Project:P1 Name:Konzeptabnahme Date:\$P1.Start + 25%

[Milestone]:M2 Project:P1 Name:Testbetrieb Date:\$P1.End - 15%

Der erste Meilenstein M1 "Konzeptabnahme" wird auf 25% der Gesamtzeit gesetzt, der zweite Meilenstein M2 "Testbetrieb" liegt 15% vor dem Projektende. Beide Meilensteine beziehen sich auf das Hauptprojekt P1.

Es folgt die Definition der Arbeitspakete. Eine Arbeitspaketdefinition beginnt mit [Job]:<Arbeitspaketkennung>. Dann folgt die Definition der Werte eines Arbeitspaketes:

[Job]:J1 Project:P2 Name:Pflichtenheft erstellen Occupation:Analyse Duration:12 Start:\$P1.Start End:\$P1.Start + 5d

Definiert werden die Werte:

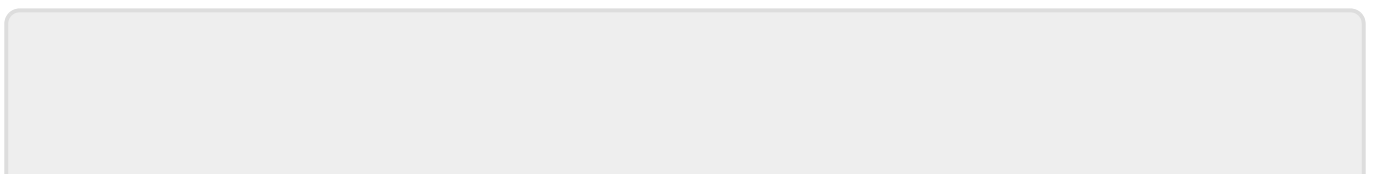
- **Projekt:** Das Projekt, dem das Arbeitspaket zugeordnet ist
- **Name:** Die Bezeichnung des Arbeitspaketes
- **Occupation:** Die Tätigkeit des Arbeitspaketes
- **Duration:** Die Solldauer des Arbeitspaketes in Stunden
- **Start:** Der Soll-Beginn des Arbeitspaketes
- **End:** Das Soll-Ende des Arbeitspaketes

Das Arbeitspaket J1 "Pflichtenheft erstellen" des Projektes P2 hat die Tätigkeit "Analyse" und die Dauer von 12 Stunden. Das Arbeitspaket beginnt zum Starttermin des Projektes P1 und endet 5 Tage danach.

Analog werden drei weitere Arbeitspakete J2, J3 und J4 definiert.

Definitionen für Arbeitspakete enthalten keinen Mitarbeiter. Vielmehr wird der Benutzer aufgefordert die Mitarbeiter auszuwählen, wenn er mit dieser Schablone ein Projekt anlegt. Hier hat der Nutzer auch die Möglichkeit zu jedem Arbeitspaket, das in der Schablone definiert ist mehrere Mitarbeiter zuzuordnen. Da jedes Arbeitspaket in Projectile nur einem Mitarbeiter zugeordnet ist, werden mehrere Arbeitspakete erstellt, die sich nur im Mitarbeiter-Feld und in der Arbeitspaketnummer unterscheiden.

[weiter mit Projektschablonen \(Referenz\)](#)



From:

<https://infodesire.net/dokuwiki/> - **Projectile-Online-Handbuch**

Permanent link:

https://infodesire.net/dokuwiki/doku.php?id=br:handbuch:kapitel_4:4.2.08_projektschablonen_fortsetzung



Last update: **2019/10/25 14:11**