Inhaltsverzeichnis

1 Einleitung 3

2 Ordner- und Ablagestruktur 4

2.1. Einleitung 4

2.2. Richtlinien für die Ordner- und Ablagestruktur 4

2.3. Standardordnerstruktur für ein Projekt 5

3 Allgemeine Struktur einer Quellcodedatei 6

3.1. Einleitung 6

3.2. Richtlinien für die Strukturierung einer Quellcodedatei 6

3.3. Struktur einer Quellcode-Datei 7

4 Allgemeine Codierungsrichtlinien 8

4.1. Einleitung 8

4.2. Allgemeine Richtlinien 8

5 Namenskonventionen 9

5.1. Einleitung 9

5.2. Richtlinien für die Benennung (Allgemein) 9

5.3. Richtlinien für Benennung von Namensräumen 10

5.4. Richtlinien für Benennung von Klassen, Schnittstellen und Enumerationen 11

5.5. Richtlinien für Benennung von Methoden 11

5.6. Richtlinien für Benennung von Instanzvariablen 12

5.7. Richtlinien für Benennung von Properties 13

5.8. Richtlinien für die Benennung von Events und Delegates 13

5.9. Präfixe für die Benennung von Steuerelementen 14

6 Quellcode-Dokumentation 16

6.1. Einleitung 16

6.2. Richtlinien für die Quellcode-Dokumentation 16

6.3. Erzeugen der Quellcode API mit YUIDoc 21

7 Fehlerbehandlung 22

7.1. Einleitung 22

7.2. Richtlinien für die Fehlerbehandlung 22

8 Subversion 25

8.1. Einleitung 25

9 Abbildungsverzeichnis 26

# Einleitung

Die Entwicklungsrichtlinien bilden eine Grundlage zum vereinheitlichten Erstellen von Projekten. Sie legen unteranderem fest, wie Projektordner strukturiert, Quellcode zu codieren sind. Diese Konventionen sind umfassend in diesen Entwicklungsrichtlinien festgehalten und müssen angewendet werden.

Jedes Kapitel dieses Dokumentes ist nach dem gleichen Schema aufgebaut. Es beginnt mit einer kurzen Einleitung, welche die Motivation des jeweiligen Themas beschreibt. Anschließend werden die einzuhaltenen Richtlinen des Themas aufgeführt. In den nachfolgenden Abschnitten eines Kapitels werden diese Richtlinien erläutert und deren Umsetzung beschrieben.

Jede Richtlinie hat folgenden Aufbau:

1. **Nr.** – Jede Richtlinie erhält eine Nummer. In jedem Kapitel beginnt die Nummerierung bei 1.
2. **Name** – Kurze Beschreibung der Regel bzw. Richtlinie.
3. **Kategorie** – Kategorie zu der die Richtline gehört.
4. **Beschreibung** – Verständliche Beschreibung der Richtlinie.

|  |  |
| --- | --- |
| **§ 1** | <Kategorie> |
| *<Name der Richtlinie>* | |
| Erläuterung der Richtlinie und Ausführungsbestimmungen! | |

# Ordner- und Ablagestruktur

## Einleitung

Zur Reduzierung der Such- und Einarbeitungszeiten ist die Ordnerstruktur eines Projektes standardisiert. Dies gilt auch für die Titanium Projektordner und Projekte. Die Standardisierung der Ordnerstruktur bildet die Grundlage für die Arbeit mit Subversion (Haupt- und Nebenentwicklungszeigen) und für das Releasemanagement, sowie einen Ansatz für eine entsprechende Toolunterstützung.

## Richtlinien für die Ordner- und Ablagestruktur

|  |  |
| --- | --- |
| **§ 1** | Ordner- und Ablagestruktur |
| *Verwendung der Standardstruktur für Ti-Projektordner und Ti-Projekte* | |
| Für alle Titanium Projektordner und Projekte ist die definierte Ordnerstruktur einzuhalten. | |

|  |  |
| --- | --- |
| **§ 2** | Ordner- und Ablagestruktur |
| *Jedes Ti-Projekt ist mindestens einem Ti-Projektordner zugeordnet* | |
| Jedes Ti-Projekt ist Teil einer Ti-Projektmappe. | |

|  |  |
| --- | --- |
| **§ 3** | Ordner- und Ablagestruktur |
| *Der Name eines SW-Elements muss eindeutig sein* | |
| Unterschiedliche SW-Elemente dürfen nicht denselben Namen haben, dies erhöht nicht nur das allgemeine Verständnis des Systems, sondern erlaubt auch eine eindeutige Zuordnung zwischen Dokumentation (Bereich: Systementwicklung) und Realisierung (Bereich: Implementation). | |

|  |  |
| --- | --- |
| **§ 4** | Ordner- und Ablagestruktur |
| *Es muss für jedes SW-Element eine VS-Projektmappe mit gleichen Namen geben* | |
| Für jedes SW-Element muss eine Ti-Projektordner existieren, welche den gleichen Namen wie das SW-Element besitzt. Dies gilt auch für Prototypen, hier muss ein Ti-Projektordner mit dem Namen des Prototyps existieren. | |

|  |  |
| --- | --- |
| **§ 5** | Ordner- und Ablagestruktur |
| *Jeder Namensraum wird auf einen Ordner abgebildet* | |
| Innerhalb eines Ti-Projektes wird jeder erzeugte Namensraum auf einen Ordner innerhalb des Ordners abgebildet und die entsprechenden Quellcodedateien in diesem abgelegt. | |

## Standardordnerstruktur für ein Projekt

Im Nachfolgenden ist die Standardordnerstruktur für Projekte abgebildet. Die Bezeichnung Projekt ist bzgl. des allg. Projektbegriffes zu verstehen. Nachfolgend wird nur auf die Bereiche eingegangen, welche für einen Entwickler von besonderer Bedeutung sind.

| Ordner | Beschreibung |
| --- | --- |
| Dokumentation | Enthält Dokumentation z.B. Anleitungen, Technische - Dokumentation |
| Implementation | Enthält Prototype, Demoprojekte und Laufende Projekte |
| Management | Enthält Planung – und Steuerung Dokumente, Berichtswesen und Meetings |
| Releases | Enthält fertige Projekte, die zur Auslieferung bereit sind |
| Systementwicklung | Enthält Systemarchitektur und Standardprojekt |
| Tools | Enthält Tools, die zum Entwickeln benötigt werden |