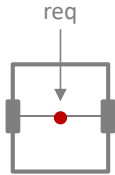


Die Funktion Vernünftige Software ist sparsam und einfach



A.1 Erstellung einer Funktion

Bevorzuge eine klare und verständliche Lösung.

A.3 Ablauf Änderung einer Funktion

Verlasse eine Funktion verständlicher als vorgefunden.

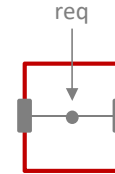
A.2 Umwelt Änderung einer Funktion

Optimiere nur, wenn der Engpass identifiziert wurde und die Verbesserung gemessen werden kann.

A.4 Hintergrund: Technische Schuld

Messen der technischen Schuld mit Hilfe statischer Analysen

Die Komponente Gutes Design ist immer der Aufgabe angemessen



P.1 Prinzip der eindeutigen Verantwortung

Dieses Design-Prinzip besagt, dass jede Funktion eine einzige Verantwortung vollständig umsetzt.

P.3 Prinzip der eindeutigen Aufgabenteilung

Dieses Design-Prinzip besagt, dass jede Funktion einem eindeutigen Aufgabenbereich zugeordnet ist.

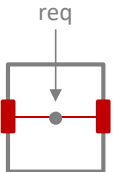
P.3 Prinzip der ordentlichen Abstraktionsstufe

Dieses Design-Prinzip besagt, dass jede Funktion auf einer erkennbaren Abstraktionsebene liegt.

A.5 Hintergrund: Modifizierbarkeit

Uniformität im Code kann nur durch konservative Codekonvention bewahrt werden.

Die Schnittstelle Rein symptomatische Codierung verspricht nur selten dauerhaften Erfolg



P.4 Prinzip der geringsten Überraschung

Dieses Design-Prinzip besagt, dass jede Schnittstelle sich so Verhalten soll wie erwartet.

P.5 Prinzip der Datenkapslung

Dieses Design-Prinzip besagt, dass der direkte Zugriff auf die Komponenten interne Datenstruktur unterbunden wird.

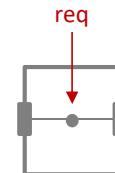
P.6 Prinzip der Redundanz-Vermeidung

Dieses Design-Prinzip besagt, dass eine Information eine einzige, eindeutige, verbindliche Darstellung innerhalb der Software hat.

A.6 Hintergrund Wartung

Periodisches Refactoring erhöht die Modifizierbarkeit durch Tilgung der technischen Schuld.

Die Anforderung Es muss mit Absicht gehandelt werden



P.7 Prinzip der Notwendigkeit

Dieses Design-Prinzip besagt, dass nur klar kommunizierte Anforderungen umgesetzt werden.

P.8 Prinzip der Entwurfsvereinbarung

Dieses Design-Prinzip besagt, dass jede Schnittstelle eindeutig in den Anforderungen definiert ist.

A.7 Löschung einer Funktion

Dieses Design-Prinzip besagt, dass nur Funktionen vollständig unter Berücksichtigung der Anforderungen zu entfernen sind.

A.8 Hintergrund: Review

Reviews dienen der Legitimation jeder geplanten Wartung.