

Verteilte Systeme und Komponenten

C.U.P.I.D.-Eigenschaften

**Keine Konkurrenz, sondern eine sinnvolle
Ergänzung zu den S.O.L.I.D.-Prinzipien**

Roland Gisler



Inhalt

C.U.P.I.D. fasst **fünf** wichtige Designprinzipien zusammen:

- **C**omposable – lässt sich gut mit anderem nutzen.
- **U**nix philosophy – kümmert sich genau um eine Sache.
- **P**redictable – macht das, was man erwartet.
- **I**diomatic – Nutzung fühlt sich «natürlich» an.
- **D**omain-based – sowohl in Sprache als auch Struktur.

Lernziele

- Sie kennen die fünf grundlegenden C.U.P.I.D.-Designprinzipien.
- Sie können die Prinzipien anhand von Beispielen erklären.
- Sie können die Prinzipien in eigenen Entwürfen anwenden.

Idee hinter den C.U.P.I.D.-Eigenschaften

- Entwickelt als Ergänzung zu S.O.L.I.D. von Daniel Terhorst-North und 2021 veröffentlicht.
 - Seine «Provokation» von 2017 wurde oft missverstanden: «Why every single element von SOLID is wrong».
- Motivation für C.U.P.I.D.:

Nicht sture Regeln und Richtlinien (die nicht eingehalten werden) definieren, sondern positive Eigenschaften, die mehr oder weniger erfüllt werden können.
- Wichtig: C.U.P.I.D. ist kein Ersatz oder Gegenplot zu S.O.L.I.D., sondern eine sinnvolle Ergänzung!

Composable

Plays well with others.

Composable - lässt sich gut mit anderem nutzen

- Guter Code hat eine kleine «Oberfläche»: Von Aussen sind nur jene Teile sichtbar, welche für die Entwickler*innen relevant sind.
 - (Wieder-)Verwendung des Codes möglichst einfach halten.
 - Es kann weniger schiefgehen und es entstehen weniger Konflikte, etwa bei Änderungen und Verschiebungen.
 - Hohe Kohäsion, schmale und einfach Schnittstelle.
- Mit eindeutig definierten Schnittstellen ist er einfacher zu verstehen und leichter zu verwenden.
 - Die Intention des Codes soll offensichtlich sein.

Unix Philosophy

Does one thing well.

Unix philosophy - kümmert sich genau um eine Sache

- Guter Code sorgt dafür, dass eine Softwareeinheit (Programm, Modul, Klasse oder Methode) „nur“ genau eine Aufgabe erledigt, diese dafür aber richtig gut.
- Klassisches UNIX-Konzept: Viele, kleine Tools, die sich vielseitig kombinieren lassen.
- Erinnert das zufällig an SRP (Single Responsibility Principle)?
 - Exakt das ist gemeint!

Predictable

Does what you expect.

Predictable - macht das, was man erwartet

- Der Code verhält sich wie erwartet, ohne unliebsame, unerwartete Überraschungen.
 - Fachlicher Formuliert: Keine Nebeneffekte.
- Der Code ist deterministisch in der Ausführung und er ist im technischen Sinne beobachtbar (keine Magie).
 - Vereinfacht sowohl das Debugging als auch die Fehlersuche.
- Der interne Status einer Softwareeinheit kann von den Ausgaben zwar abgeleitet werden, ist aber nicht veränderbar.
 - Datenkapselung

Idiomatic

Feels natural.

Idiomatic - Nutzung fühlt sich «natürlich» an

- Wenn der Code gut ist, dann fühlt sich dessen Nutzung und die Bearbeitung des Codes «natürlich» an.
 - Leicht verständlich.
- Es werden die Idiome des Kontextes (Domäne, Firma, Team) sowie des Ökosystems der Sprache, in der er geschrieben ist, verwendet.
 - Namenskonventionen einhalten!

Domain-based

In language and structure.

Domain-based - sowohl in Sprache als auch Struktur

- Für guten Code wird die Domänensprache und deren Struktur verwendet.
- Die Struktur des Codes soll die Lösung widerspiegeln, und nicht das darunterliegende Framework und/oder die eingesetzten Konzepte.
 - Etwas was in vielen Libraries/Frameworks verletzt wird!
- Anstatt technischer Begriffe/Konzepte (Models, Views, Controllers) sollen Domänenbegriffe (Dokumente, Zahlungen) verwendet werden.
 - Die Domänengrenzen sollen als Modulgrenzen und Deployment-Units beachtet werden.
 - Architektur: Vertikale vs. horizontale Trennung.

C.U.P.I.D.

Schreiben Sie **liebvollen*** Code!

*engl. «cupid» = amor, Liebesgott in der römischen Mythologie

Zusammenfassung – C.U.P.I.D.

- Wird eher durch «positive, bewertbare Eigenschaften» denn als (vermeintlich) «harte Regeln» definiert.
 - Somit für viele Entwickler*innen angenehmer im Umgang?
 - «Kuschel»-Kommunikation?
- Letztlich verfolgt C.U.P.I.D ziemlich exakt die gleichen Ziele wie S.O.L.I.D. – einfach ein bisschen moderner und zeit(geist)gemässer ausformuliert.
- Gute Ergänzung zu S.O.L.I.D. – welche noch immer Gültigkeit hat und absolut unbestritten ist.
 - C.U.P.I.D. ist **kein** Angriff auf S.O.L.I.D.!

Fragen?