



## Buch: Softwareentwicklung für die Naturwissenschaften

Dr. habil. Till Biskup

— Glossar zu Kapitel 07: „Versionsnummern“ —

---

*Hinweis: Die nachfolgend genannten Begriffe und Definitionen erheben keinen Anspruch auf formale Korrektheit, sondern dienen lediglich dem besseren Verständnis der in der Vorlesung behandelten Themen und sind im jeweiligen Kontext zu sehen. Mehrfache, voneinander abweichende Definitionen in unterschiedlichen Kontexten sind daher möglich. Englische Begriffe werden zwar nach Möglichkeit übersetzt, erscheinen aber ggf. unter ihrem englischen Namen in der Liste. Verweise untereinander sind durch ↑ gekennzeichnet.*

**Abwärtskompatibilität** Kompatibilität einer Version einer Software mit einer früheren Version ihrer selbst. Von großer Bedeutung für die Nutzer einer solchen Software. Die Trennung zwischen öffentlichen und internen Schnittstellen ↑API ist hier von entscheidender Bedeutung. Nur erstere werden für die Abwärtskompatibilität berücksichtigt. Ein Versionsnummernschema, das besonders auf die Integration dieser Information Wert legt, ist ↑SemVer.

**API** *application programming interface*, Programmierschnittstelle oder genauer Schnittstelle zur Anwendungsprogrammierung

**CalVer** *Calendar Versioning*, Serie von Schemata für Versionsnummern, die in der einen oder anderen Form ein Datum beinhalten. Für Details vgl. <https://calver.org/>. Ein alternatives, auf ↑Abwärtskompatibilität zielendes und bei ↑freier Software vergleichsweise weit verbreitetes Schema ist ↑SemVer.

**Changelog** Liste von Änderungen, sortiert nach dem Datum ihres Auftretens. Lässt sich häufig vollständig aus den Commit-Kommentaren einer ↑Versionsverwaltung erzeugen. Mitunter wird die Liste auch händisch gepflegt und enthält dann übersichtlich die wichtigsten Änderungen zwischen Versionen inkl. Hinweisen zur ↑Abwärtskompatibilität.

**CODATA** *Committee on Data of the International Council for Science*, in den 1960er

Jahren vom Internationalen Wissenschaftsrat (ICSU) gegründete Organisation, deren Ziel die Verbesserung von Qualität, Zuverlässigkeit und Zugänglichkeit interessanter Daten aus allen Feldern der Wissenschaft und Technologie ist. Vielleicht am bekanntesten sind die CODATA-Empfehlungen für physikalische Konstanten, die einen optimal geschätzten Satz von Werten physikalischer Konstanten und der zugehörigen Standardunsicherheiten darstellen und regelmäßig (aktuell alle vier Jahre) veröffentlicht werden.

**deprecated** überholt, Bezeichnung für Funktionalität, die zwar aktuell noch vorhanden ist, aber in absehbarer Zeit aus einem Programm entfernt werden wird. Die Übergangsfrist sollte dem Nutzungsszenario angemessen gewählt und Nutzer entsprechend informiert werden.

**Eineindeutigkeit** Bijektivität, Eindeutigkeit in beiden Richtungen. Mathematisch die Abbildung eines Elements einer Menge auf genau ein Element einer zweiten Menge und umgekehrt, weshalb Definitions- und Zielmenge die gleiche Mächtigkeit aufweisen. Im Kontext von Versionsnummern die Eindeutigkeit der Zuordnung einer Versionsnummer zu *genau einem* Zustand der Software und umgekehrt der Zuordnung eines Zustandes der Software zu *genau einer* Versionsnummer.

**freie Software** Software, die Endnutzern die Freiheiten der Nutzung, des Überprü-

fens/Studierens, des Teilens und des Modifizierens der Software gewährt. Das erfordert zwingend die Offenlegung des Quellcodes der Software. Der Begriff „frei“ bezieht sich hier auf die Freiheiten des Nutzers, nicht auf die Kosten der Software. Die Begriffe „*open source*“ und „freie Software“ sind nicht gänzlich deckungsgleich.

**Hash** Prüfsumme, mathematisch erzeugter Wert, der es erlaubt, die Integrität von Daten zu überprüfen. Ein sehr einfaches Beispiel wäre die Quersumme. Kryptographische Hashes werden von kryptographischen Hashfunktionen erzeugt, die möglichst kollisionsresistent sind. Das bedeutet, dass es quasi unmöglich ist, zwei verschiedene Eingabewerte zu finden, die zum identischen Hashwert führen.

**inkrementieren** hochzählen

**Konvention** innerhalb einer Gruppe oder einem (lokalen) Kontext getroffene (temporäre) Festlegung. Ziel von Konventionen ist die Vereinheitlichung und damit einhergehend die Befreiung von der Notwendigkeit, jedesmal aufs Neue nachdenken zu müssen, wie z.B. gewisse Prozesse durchgeführt oder Objekte benannt werden sollen. Konventionen sind im Gegensatz zu  $\uparrow$ Standards weniger verbindlich und deutlich flexibler sowie *ad hoc* innerhalb einer Gruppe einführbar.

**Modularisierung** Aufteilung der Gesamtaufgabe in kleinere Abschnitte. Die Aufteilung wird so lange fortgesetzt, bis die Lösung für den aktuellen Abschnitt unmittelbar in Form von Quellcode offensichtlich ist. Setzt die Definition von  $\uparrow$ Schnittstellen voraus.

**monolithisch** aus einem Stück bestehend; zusammenhängend und fugenlos

**Refactoring** Verbesserung der Qualität des Quellcodes einer Software ohne Einfluss auf ihr von außen erkennbares Verhalten. Diszipliniertes Vorgehen zum Aufräumen von Quellcode, das die Wahrscheinlichkeit, Fehler einzuführen, minimiert.

**Release** Veröffentlichung; hier meist bezogen auf eine veröffentlichte Version einer Software.

**Semantik** 1. Bedeutung bzw. Inhalt eines Wortes, Satzes oder Textes; 2. Teilgebiet der Linguistik, dessen Untersuchungsobjekt die Bedeutung sprachlicher Zeichen und Zeichenfolgen ist.

**semantische Information** Bedeutungsebene ( $\uparrow$ Semantik) einer Information.

**SemVer** *Semantic Versioning*, Versionsnummernschema, das insbesondere auf die  $\uparrow$ Abwärtskompatibilität und die Kodierung dieser Information in der Versionsnummer abhebt. Für Details vgl. <https://semver.org/>.

**Standard** von einem oft internationalen und anerkannten Gremium definierte Festlegung. Standards sind im Gegensatz zu  $\uparrow$ Konventionen sehr viel starrer und nicht *ad hoc* von einer Gruppe einführbar.

**Versionsverwaltung** engl. *version control system*, VCS; Software zur Verwaltung unterschiedlicher Versionen von Dateien und Programmen, die den Zugriff auf beliebige ältere als Versionen ( $\uparrow$ Revision) gespeicherte Zustände ermöglicht. Gleichzeitig ein wichtiges Werkzeug für die Softwareentwicklung und wesentlicher Aspekt einer Projektinfrastruktur.