

# Sémantické Verzování

Jan Tančibok 3.IT.

Sémantické verzování je způsob, jakým verzovat software tak, aby bylo možné jej správně používat v balíčkovacích nástrojích.

Jenže jej hodně vývojářů nepoužívá, i když to tak na první pohled vypadá. Existují totiž i jiné metody.

Ukázka:



Shrnutí:

Číslo verzí zapisujeme ve formátu: MAJOR. MINOR. PATCH

Navyšování jednotlivých čísel verzí probíhá následovně:

PATCH – Když se opravila chyba a zůstala kompatibilita

MAJOR – Když nastala změna, která není zpětně kompatibilní s ostatními (API)

MINOR – Když se přidá funkcionality se zachováním zpětné kompatibility

Pomocí předběžných verzí a přidáváním metadat je možné upřesnit informace.

Např.:

1.0.0 - alfa a také 1.0.1 – beta + 2

Dependency hell (peklo závislostí):

Dependency hell znamená, když mají systémy mnoho závislostí tak není možné přejít na další verzi, aniž by se neudělala kompletní instalace.

Pokud není specifikován rozsah verzí tak verze budou na sebe příliš navazovat a bude více kompatibilních verzí, než je nutný a potřebný.

### Proč používat verzování?

Verzování Sémantické není revoluční myšlenka a když vydáváte software, tak už pravděpodobně děláte něco podobného.

Díky Sémantickému verzování se snadno vyhneme problémům se závislostmi (dependency hell).

Sémantické verzování je rozumný způsob, jak vydávat a aktualizovat knihovny tak, abyste nemuseli řešit nové verze závislosti, ušetřili si čas a vyhnuli se zmatkům.

### Příklad verzování:

Začíná se od **0.1.0**

Když je připraven, rozvětví se kód ve zdrojovém repozitáři.

Označí se 0.1.0 a vytvoří se 0.1.0 větev, poté se stane 0.2.0-snapshot nebo něco podobného.

Přidají se nové funkce a opravují do větve a v té době, se změní na 0.1.1, 0.1.2, ...

Deklaruje se verze 1.0.0, když je produkt považován za kompletní a nemá zásadní nedostatky.

Od té doby se každý může rozhodnout, kdy se zvýší hlavní verze.

Zdroje:

<https://www.kutac.cz/pocitace-a-internety/semanticke-romanticke-a-jine-verzovani>

<https://semver.org/lang/cs/>