# Sémantické verzování

Jakub Čapek, Michael Petro

3.IT

# Obsah

# Shrnutí

Číslo verzí zapisujeme ve formátu MAJOR.MINOR.PATCH Navyšování jednotlivých čísel verzí probíhá následovně:

1. MAJOR – když nastala změna, která není zpětně kompatibilní s ostatními (API)
2. MINOR – když se přidá funkcionalita se zachováním zpětné kompatibility
3. PATCH – když se opravila chyba a zůstala kompatibilita

Pomocí předběžných verzí a přidáváním metadat je možné upřesnit informace. Např.:
1.0.0 - alfa, 1.0.1 – beta + 2

Alternativou je označit předběžné verze jako „release candidates“ (kandidáty na uvolnění), takže softwarové balíčky, které budou brzy vydány jako konkrétní verze, mohou nést tuto značku verze následovanou „rc- #“, označující číslo kandidáta na vydání; Po uvolnění konečné verze se značka „rc“ odstraní.

## Proč používat verzování?

Sémantické verzování není revoluční myšlenka a když vydáváte software, tak už pravděpodobně děláte něco podobného. Problém je, že “něco podobného” nestačí.

Tím, že výše uvedeným myšlenkám dáváme přesnou a jasnou definici, je lehčí komunikovat záměry vašeho softwaru jeho uživatelům. 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.

## 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.

# Co bychom měli splňovat

- Software používající Sémantické verzování, musí mít nadefinované API, buďto přímo ve zdrojovém kódu a nebo v externí dokumentaci. V obou případech to musí být hlavně přesné a komplexní.
- Číslo verzí musí být ve formátu X, Y, Z.
- Vždy když program upravíme, měli bychom ho vydat pod novou verzí

## 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.