

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 funkcionální 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

Úvod

Čím větší a komplexnější váš projekt je, tím je těžší přejít na novou verzi programu. Když mají systémy mnoho závislostí, může se stát, že vydání nové verze je noční můrou. Pokud je specifikovaná závislost příliš striktní, hrozí, že zůstanete na jedné verzi (angl. “version lock”) a nebudete mít možnost přejít na další. V pekle závislostí se nacházíte, jakmile jste vázaný určitou verzí nebo v případě, že není přesně specifikován rozsah verzí, které máte použít. Taková situace brání v bezpečném vývoji kupředu.

Řešením tohoto problému je jednoduchý sled pravidel. Tato pravidla jsou založena (ale nikoliv omezena) na již ověřené praxi a to ve vývoji otevřeného i uzavřeného softwaru. Pro správné fungování systému, si musíte nejdříve nadefinovat způsob, jakým se bude komunikovat s Vaší aplikací (dále API). Toto API může být nadefinováno pomocí rozsáhlé dokumentace nebo zapsáno přímo ve zdrojovém kódu. Je důležité, aby bylo srozumitelné a snadno čitelné. Je třeba mít kompletní dokumentaci. Zápis verzí je ve formátu MAJOR.MINOR.PATCH. Opravy chyb, které nezměnily Vaše API, zvyšují číslo PATCH verze. Zpětně kompatibilní změny v API zvyšují hodnotu MINOR verze a rozdílné verze API, které nejsou zpětně kompatibilní, zvyšují číslo MAJOR verze.

Major: 1.y.z

První číslovka je určena pro *major* verze, které mohou do software přinášet zlomové změny.

Specifikace ale mluví o výjimce pro verze začínající nulou: Major verze s hodnotou 0 (0.y.z) je určena pro počáteční vývoj. Cokoliv se může změnit a API v tomto formátu by nemělo být považováno za stabilní.

Minor: x.2.z

Číslovka za tečkou označuje *minor* verzi. Ta může přinášet tyto změny:

- přidání zpětně kompatibilní funkcionality,
- odebrání funkcionality (jako zastaralé) i pokud neovlivňuje samotný API kód.

Jde prostě o změny, které jsou z pohledu uživatele programu zpětně kompatibilní. V příkladu nahoře je to číslovka 2 na druhé pozici.

Patch: x.y.3

Třetí číslovka je *patch*. Ta se mění při opravách chyb. V ukázce výše je to číslovka 3 na třetí pozici.

h2. Alfa verze

Verze sw/hw, která je zpravidla poskytována pouze v rámci společnosti, vyvíjející tento software. Jedná se o produkt, který většinou obsahuje všechny důležité funkce, avšak také spoustu chyb. Tato verze je proto testována pouze vývojáři, kteří vědí, jak tento sw/hw pracuje. V tomto stádiu vývoje se nalezne a odstraní nejvíce chyb. Jakmile jsou vážné chyby odstraněny, bývá často vydávána Beta verze.

h2. Beta verze

Sw/hw produkt, na kterém je již opravena většina chyb, nicméně je pořád nestabilní a na jeho chování se nedá spolehnout. To, co vznikne z betaverze po odstranění chyb, se nazývá Release Candidate.

Druhy beta verzí:

- * Closed beta – uzavřený test, většinou určený pouze vývojářům.*
- * F&F beta – vývojáři si vyberou sami, kdo bude mít možnost účastnit se testu, většinou vybírají z řad rodiny a přátel (proto F&F = friends and family).*
- * Open beta – je dostupná zdarma komukoli, kdo má zájem se testu účastnit.*

h2. Release candidate (RC)

Testovací verze připravovaného sw/hw. Jedná se o kandidáta na konečnou, finální verzi. Release Candidate je k finální verzi blíže, než betaverze. Ke zkratce RC se přidává i číslice. Vyjít tedy může RC1, RC2, RC3, atd. "RCčka" mohou být značně nestabilní.

Waldemar Kotmel, Dominika Kabátková