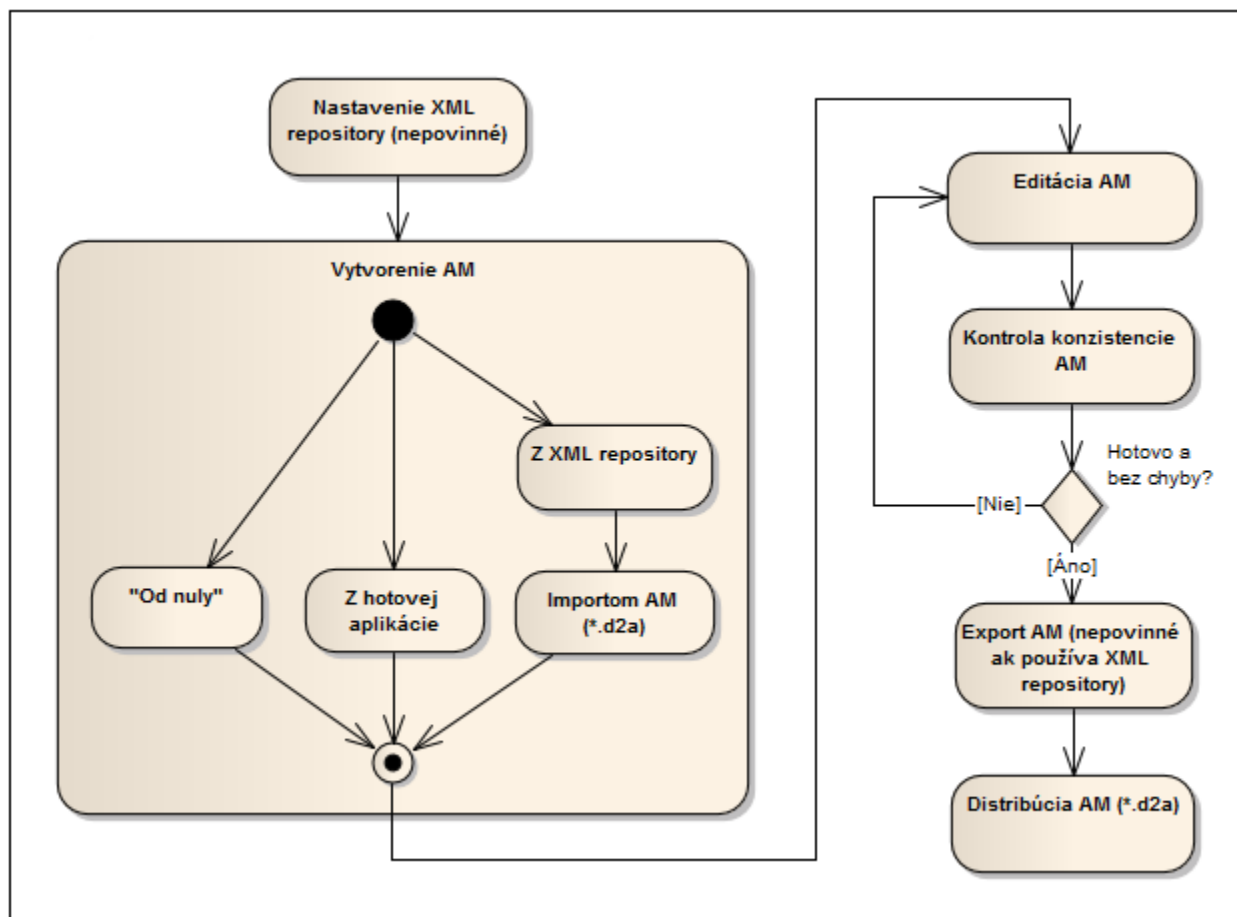


# Vývojový cyklus aplikaného modulu

Vývojový cyklus [aplikaného modulu](#) je možné vyjadriť šiestimi základnými krokmi, ktorých postupnosť je znázornená na nasledovnom diagrame.



## 1. Nastavenie XML Repository pre aplikovaný modul (nepovinné)

Pri vývoji aplikovaného modulu, rovnako ako pri vývoji aplikácie, sú zjavné výhody použitia [XML Repository](#). Toto sa od bežného repository nijako nelíši, je ale špecifické svojou konfiguráciou, ktorá by mala byť nasledovná.

Povolené nastavenia:

- Nastavenie asu súboru podľa asu modifikácie objektu.
- Kontrolovanie konfigurácie objektov pred exportom.
- Nastaviť ID = 0 pre exportované objekty.
- UTF-8 kódovanie exportovaných súborov.
- Export s nepovinným lenstvom v skupinách.
- Export do "exportovacieho podadresára" objektu.
- Zmena rozmeru štruktúrovaných premenných s parametrizovaným rozmerom.
- Vynulovanie parametrizovaných konfiguračných vlastností.

Zakázané nastavenia:

- Rekurzívny export potomkov objektov.
- Rekurzívny export referencovaných objektov.
- Export systémových objektov a premenných.
- Export logov "života objektu".

Takýmto nastavením exportu objektov získame vždy XML súbory vhodné na vytvorenie archívu aplikovaného modulu a následnú distribúciu modulu. Takto vytvorené XML súbory totiž nie sú závislé od aktuálnej konfigurácie parametrizovaných konfiguračných vlastností.

## 2. Vytvorenie aplikovaného modulu

### 1. "Od nuly"

Podľa návodu na [tvorbu aplikovaného modulu](#).

## 2. Z hotovej aplikácie

Vytvorením objektu typu *Aplikovaný modul* a jeho parametrov, [zalenením](#) objektov aplikácie do aplikovaného modulu a [parametrizovaním](#) zvolených konfiguračných vlastností.

## 3. Z XML Repository

XML súbory z [XML Repository](#) pre aplikovaný modul nie je možné priamo nainportovať do aplikácie, o je priamym dôsledkom exportovacieho nastavenia "**Vynulovanie parametrizovaných konfiguračných vlastností**", ktoré môže spôsobiť, že XML súbor obsahuje neplatnú konfiguráciu objektu. Preto je z nich potrebné najprv vytvoriť archív aplikovaného modulu pomocou utility [d2archive](#) a následne sa riadi krokom 4.

## 4. Importom archívu aplikovaného modulu

Importom archívu aplikovaného modulu získame funkčnú konfiguráciu aplikovaného modulu. Hodnoty parametrov je možné kedykoľvek zmeniť cez položku [rozšírených akcií](#) nad objektom aplikovaný modul:

- [Zobrazí/upraví hodnoty parametrov modulu](#) - touto položkou je možné zmeniť konfiguráciu hodnôt parametrov už nainportovaného modulu tak, aby sa zmena prejavila na lenských objektoch.

V prípade, že modul staršej alebo novej verzie už v aplikácii existuje, ešte pred samotným importom môžete získať užitočné informácie z položiek:

- [Porovna parametre modulu](#) - umožňuje odhaliť podozrivé zmeny v parametroch modulu (najmä zmenu závislosti, i odstránenie/pridanie parametra).
- [Porovna verejné leny modulu](#) - umožňuje zistiť zmeny vo verejných lenoch modulu – pridanie, odobratie, i zmenu konfigurácie verejných lenských objektov.

## 3. Editácia aplikovaného modulu

Pod editáciou aplikovaného modulu sa rozumie editácia [parametrov aplikovaného modulu](#) a ich použitia a editácia [lenských objektov](#). Poas editácie modulu sú tiež užitočné možnosti z [rozšírených akcií](#) aktivovaných nad objektom aplikovaný modul ako napríklad:

- Zobrazenie závislostí modulu, ktoré informuje tvorcovi modulu o tom, ktoré objekty iných modulov sú použité v lenských objektoch.
- [Zobrazenie použitia](#) parametrov modulu podáva kompletnú informáciu o tom, kde je aký parameter použitý, pričom upozoruje aj na použitie neplatných (zmazaných) parametrov.
- [Prenomenovanie parametrov modulu](#) umožňuje zmeniť mená parametrov so zachovaním konzistencie modulu tak, že ich premenuje aj pri všetkých použitých KV.
- Náhľad parametrov modulu zobrazuje vzťah [dialógu konfigurácie parametrov](#) modulu pri jeho importe.

## 4. Kontrola aplikovaného modulu

Pravidelným volaním [kontroly aplikovaného modulu](#) počas vývoja zisujeme, o je potrebné v konfigurácii modulu zmeniť tak, aby bolo možné vytvoriť použitý archív AM. Kontrola je automaticky volaná pred exportom AM.

## 5. Export aplikovaného modulu (nepovinné)

Export AM sa vykonáva cez položku [rozšírených akcií](#) otvorených nad objektom aplikovaný modul. Vybratý objekt aplikovaný modul a všetky jeho lenské objekty budú exportované pri použití exportovacích nastavení popísaných v [kroku 1](#). Pred samotným exportom automaticky prebehne aj [kontrola konzistencie](#) modulu, ktorá zabráni exportu v prípade nekonzistentnej konfigurácie modulu. Ak je použité XML Repository, objekty sú exportované automaticky pri každom uložení. Neprebíha však kontrola konzistencie modulu, preto ju je potrebné volať manuálne cez položku [rozšírených akcií](#) nad objektom aplikovaný modul a zamedziť tak nekonzistentnému stavu modulu v XML Repository.

Zmeny vykonané editáciou aplikovaného modulu môžu spôsobiť nekompatibilitu modulu s jeho staršou verziou. Na zistenie takýchto nekompatibilit sú užitočné nasledovné položky [rozšírených akcií](#) pre AM:

- [Porovna parametre modulu](#) - porovnáva konfiguráciu parametrov staršej verzie AM s aktuálnou pomocou [dialógu konfigurácie PAM](#), ktorý zobrazuje parametre, ktoré v novej verzii pribudli, odbudli, alebo zmenili svoju závislosť vzťahom na iný parameter.
- [Porovna verejné leny modulu](#) - porovnaním zoznamu verejných lenov modulu aktuálnej a predchádzajúcej verzie (z archívu AM) získame zoznam verejných lenských objektov, ktoré pribudli, odbudli, alebo boli zmenené. V prípade, že nejaký verejný len v novej verzii odbudol, je potrebné zvýšiť major verziu AM, pretože takto zaniká spätná kompatibilita modulu.

## 6. Distribúcia aplikovaného modulu

Vytvorenie archívu aplikovaného modulu pomocou utility [d2archive](#) z XML súborov exportovaných objektov aplikovaného modulu získané z XML Repository modulu alebo exportom AM.



### Súvisiace stránky:

[Aplikované moduly](#)  
[Utilita D2000Archive](#)