

KONFIGURÁCIA SOFTVÉRU – KEDY MÔŽE BYŤ NEVÝHODOU?

Najprv plánovať a až potom vyvíjať

Bálint Szilva

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
sylvla1717@gmail.com

Abstrakt. Oblasť softvérových a informačných systémov sa neustále zväčšuje a každým dňom pracujú tisícky ľudí na tom, aby tieto systémy boli čo najmodernejšie a čo najspoľahlivejšie. Pre správne fungovanie týchto systémov je nevyhnutné, aby boli správne konfigurované, alebo boli konfigurovateľné. Niekedy je najlepšie, keď používateľ môže softvér nakonfigurovať a nastaviť sám pre seba. V tejto eseji sú zahrnuté najdôležitejšie myšlienky o tom, kedy sú konfigurácie vhodné a kedy nie, kedy nechať používateľovi možnosť softvér nakonfigurovať sám. V eseji sa ďalej nachádzajú úvahy o možnostiach a typov konfigurácií rôznych softvérových systémov, najlepšie cesty konfigurácie ako aj možnosti konfigurácií na určitých úrovniach softvéru.

Kľúčové slová: konfigurácia softvéru, možnosti konfigurácie softvéru, konfigurovateľnosť

Úvod

V dnešnom svete sa softvérové systémy neustále menia a oblasť softvérových a informačných systémov zväčšuje, preto je nevyhnutné správne plánovanie, správny manažment pri realizácii a údržbe softvérových systémov a správna konfigurácia. Bez manažmentu vývoj a údržba systému môžu trvať dlhšie, nehovoriac o tom, že investície sa môžu drasticky zvýšiť. Bez správnej konfigurácie softvérový systém môže byť nepoužiteľný. Z týchto dôvodov je jasné, že konfigurácia softvéru je jedna z najdôležitejších vecí, ale môžeme si to dovoliť v každom prípade? Kedy je vhodné vopred konfigurovať a kedy nechať používateľovi možnosť nastaviť systém pre seba? Na

akej úrovni je najlepšie konfigurovať? V nasledujúcich kapitolách sú zahrnuté úvahy a odpovede na tieto otázky.

Konfigurácie – výhoda v každom prípade?

Z predchádzajúcich riadkov sa vyplýva, že konfigurácia je dôležitá časť softvéru. Väčšinou softvéry bez správnej konfigurácie sú nefunkčné, dokonca niekedy používateľ musí systém nakonfigurovať sám pre seba. Treba si porozmýšľať o tom, či je vždy najsprávnejšia cesta konfigurovať, alebo niekedy má svoje nevýhody. Určite otázka, ktorá napadne každému ako prvá: „Prečo používame konfigurácie a prečo môžu byť nevhodné?“.

Nevýhoda konfigurácie je riziko. Riziko, že používateľ môže vykonať alebo nastaviť softvérový systém tak, aby nefungovalo. Riziko, že nastaví niečo, čo by nemal byť nastavený. Riziko, že používateľ, ktorý je laik v oblasti v ktorej sa softvérový systém používa, vykoná v systéme zmenu, ktorá je neodstrániteľná. Čo by sa stalo, keby si každý mohol nastavovať a konfigurovať systémy, s ktorými pracujú banky? Na otázku odpoveď nie je až taká jednoznačná a závisí iba od používateľa. Keby sme mohli predpokladať, že každý používateľ je informatik, programátor alebo má určité znalosti v oblasti softvérových systémov a informatiky, nebolo by potrebné niekedy konfigurácie skryť, alebo nedovoliť. Keby sme mohli predpokladať, že každý používateľ bude používať softvér podľa ciele softvéru a nie na vlastné účely, tiež by mohli byť tieto systémy konfigurovateľné. Bohužiaľ, v reálnom svete to nefunguje tak. Preto banky, organizácie v oblasti vojenstva a organizácie vlády používajú systémy, ktorých nie je možné konfigurovať, ale na druhej strane existujú systémy, ktoré sa používajú práve preto, lebo sú konfigurovateľné na vysokej úrovni.

Tieto systémy sú napríklad systémy CMS (z anglickej frázy *Content Management System*), ktoré sa používajú pri tvorbe webových stránok a informačný systémov. Systémy CMS sú používané práve preto, lebo dovoľia používateľom vykonať rôzne zmeny v systéme, rôzne nastavenie a konfigurácie. Tieto systémy sú otvorené, čo znamená že používateľ má prístup aj ku zdrojovému kódu a môže slobodne zmeniť a nastavovať celý systém podľa svojich nápadov. Systémy sú vytvorené tak, aby s rôznymi konfiguráciami mohli používatelia vytvoriť systém, ako napríklad blog, v používateľskom rozhraní a nemuseli ani riadok programovať. Sú zabezpečené nastavenia, aby každý mohol mať svoj vlastný princíp, svoju vlastnú tému a dizajn. Keď sa niečo nepodarí, alebo nefunguje správne kvôli nastaveniam, systém sa ľahko preinštaluje alebo vráti do pôvodného stavu a nič sa nestane.

Z hore uvedených príkladov vidieť, že konfigurácie nie sú vždy výhodou, niekedy si nemôžeme ich ani dovoliť. Keby sme si dovolili, v oblastiach, ktoré musia fungovať bez chýb a nemôžu zastaviť kvôli neočakávanej chybe, by mohol vzniknúť chaos. Môže existovať zlatá stredná cesta? V nasledujúcej kapitole skúsime odpovedať na otázku.

Existuje zlatá stredná cesta?

Zlatou strednou cestou môžu byť napríklad informačné systémy, ktoré dovoľia niektoré časti konfigurovať, ale niektoré časti sú skryté alebo uzavreté. Tieto systémy nechajú možnosť používateľovi vykonať zmeny v systéme, aby mohli nastaviť systém podľa

vlastnej potreby a aby mohli v ňom čo najrýchlejšie pracovať. Uzavreté časti zabezpečujú úlohy, ktoré sú dôležité. Treba si predstaviť informačný systém v škole, kde je možné nastaviť tému, poradie jednotlivých častí, uložiť najviac používané kontakty pri správach, ale nie je možné zmeniť napríklad adresu, na ktorú sú odovzdávané úlohy a projekty študentov.

Na internete môžeme nájsť kopu informačných systémov, ktoré nejakým spôsobom môžeme nazvať ako zlatá stredná cesta. Dovolia používateľom nastavenia, ktoré uľahčia a urýchlia ich prácu, ale na druhej strane sú zabezpečené správne práva pre každého, aby nemohli vykonať niečo, s čím by urobili chybu v systéme.

Z predchádzajúcich úvahách sa vyplýva, že konfigurácia sa najviac závisí od toho, akí používatelia budú náš softvér používať. Keď môžeme predpokladať, že používateľ, ktorý je laik v oblasti softvérových systémov, a konfiguráciu vykonáva prvý krát v živote, je lepšie, keď nemôže vykonať zmenu v systéme alebo nastaviť systém nechtiac nesprávne. Na druhej strane existujú aj používatelia, ktorí majú za sebou skúsenosť a majú pokročilé znalosti v oblasti softvérových systémov. V takomto prípade môžeme klúdne dovoliť konfigurácie používateľovi, ale tiež sa treba rátať s rizikami, ako napríklad nechtiac vykonaná zmena softvéru. Na tieto účely je najlepšie definovať práva, a rozdeliť tieto práva medzi používateľmi podľa ich schopností. V tomto prípade nenastane situácia neplánovanej odstavky. V opačnom prípade je lepšia cesta, keď nedovolíme používateľom žiadne zmeny systému. Stredná cesta môže byť teda správne nastavené právo pre každého používateľa systému, aby mohli vykonať zmenu iba podľa ich schopností. V praxi to funguje tak, že sú vopred definované práva, ešte lepšie by bolo, keby každý dostal právo podľa vlastných schopností.

Rôzne úrovne, rôzne možnosti konfigurácie

Každý softvérový systém má rôzne úrovne. Konfiguráciu si môžeme vykonať na rôznych úrovniach. Tieto úrovne najčastejšie sú:

- Úroveň používateľského rozhrania
- Úroveň tried
- Úroveň zdrojového kódu.

Na úrovni používateľského rozhrania je možné implementovať nastavenie priamo do používateľského rozhrania, ktorá je najvyššia úroveň softvéru, kde každý môže používať vlastné nastavenia, ale nemôže vykonať niečo, čo by mohol fungovanie systému zastaviť. Na tomto princípe fungujú softvérové systémy, ktoré sú vytvorené pre používateľov, ktoré potrebujú softvér na vykonávanie ich roboty a nepotrebujú ďalšie konfigurácie, iba menšie zmeny v používateľskom rozhraní.

Ďalšou možnosťou je konfigurácia na úrovni tried. Tieto konfigurácie nie je možné zmeniť z používateľského rozhrania, čo znižuje riziko nesprávnej funkcionality softvéru. Na druhej strane, keď v systéme je potrebné vykonať zmenu nastavení, je potrebné, aby bol v blízkosti človek, ktorý tieto zmeny vie vykonať bez toho, aby nenastala neprijemná situácia.

Konfigurácia na najnižšej úrovni je už priama konfigurácia v zdrojovom kóde. Softvéry väčšinou používajú uzavretý kód, preto tieto konfigurácie môžu zmeniť iba ľudia,

ktorí softvérový systém vyvíjali. Softvér, ktorý má otvorený kód, je konfigurovateľný aj na úrovni zdrojového kódu. K týmto softvérom väčšinou pridajú aj dokumentáciu a tutoriál pre vykonávanie zmien v systéme.

Konfigurácia na najvyššej úrovni je teda vhodné riešenie, keď v systéme nie je potrebné vykonať väčšie zmeny. Na úrovni tried a na úrovni zdrojového kódu je potrebné mať k dispozícii odborníka, ktorý určité konfigurácie vie zmeniť a nastaviť systém tak, aby fungoval správne. Ďalšia možnosť je tréning a školenie zamestnancov, s ktorou sa zaoberáme v ďalšej kapitole.

Tréningy a školenia zamestnancov

Firmy hore uvedené problémy často riešia školením zamestnancov a odborným tréningom. Tieto školenia sú buď vykonané priamo vo firme, v ktorej zamestnanec pracuje, alebo určitý počet zamestnancov je poslaný na odborné školenie k inej firme, ktorá sa zaoberá tréningom a školením. Školenia väčšinou trvajú niekoľko dní alebo niekoľko týždňov v závislosti od toho, aké základné znalosti a skúsenosti má zamestnanec, alebo aký komplikovaný je vykonanie zmeny konfigurácie v softvérovom systéme.

Tieto školenia nie sú zadarmo. V lepšom prípade školenie zaplatí zamestnávateľ, ale niekedy sa môže stať, že firma zaplatí iba určitú čiastku a zvyšok musí uhradiť zamestnanec.

Výhoda prvého prípadu z pohľadu zamestnanca je to, že sa môže zadarmo naučiť niečo nové, čo po prípade môže používať potom aj u iného zamestnávateľa. Výhoda z pohľadu firmy je, že za určité náklady bude mať k dispozícii odborníka pre danú oblasť a tým v budúcnosti šetriť čas a peniaze. Výhodou druhého prípadu je, že firma zaplatí za školenia menej a za menšie náklady bude mať k dispozícii odborníka. Tento prípad sa skôr používa vtedy, keď firma spolupracuje s odborníkom, ktorý už má určité skúsenosti za sebou, ale potrebuje si rozšíriť svoje zručnosti a znalosti aby vedel vykonať určitú úlohu. Títo odborníci získané znalosti môžu uplatniť a za školenie a certifikát museli zaplatiť menej.

Tréningy a školenia jednoznačne sú výhodou, aj keď niekedy sú drahé. Firmám sa oplatí poslať zamestnancov, aby získali nové znalosti. Pri konfigurácii to môže byť výhoda vtedy, keď je potrebné pracovať so systémom, v ktorom vie vykonať zmenu a nakonfigurovať systém iba skúsený odborník. V prípade potreby potom je k dispozícii odborník, ktorý vie systém správne nastaviť a nie je potrebné zavolať a zaplatiť inému odborníkovi. Školenia sú výhodou aj pre človeka, ktorý sa zúčastňuje. V budúcnosti získané skúsenosti môže uplatniť v živote ako aj u nového zamestnávateľa.

Záver

O tom, že konfigurácia softvéru má určite svoje výhody, neni pochyb. Na druhej strane, nemôžeme si to vždy dovoliť. Existujú systémy, pri ktorých konfigurácie by boli nevhodné a nesprávna konfigurácia by otvorila priestor pre chyby a pre nesprávne fungovanie softvéru. V najhoršom prípade by to mohlo znamenať zastavenie funkčnosti celého softvérového systému. Všetko závisí aj od znalostí človeka, ktorý konfiguráciu robí a nastavuje softvérový systém počas svojho behu. Najlepšie riešenie je používanie

používateľských práv a nastavenie administrátorského práva iba správne použiteľovi, alebo odborné školenie a tréning zamestnancov. Pri konfigurácii je potrebné premyslieť, aké konfigurácie sú dovolené, kto bude softvér používať, čo všetko je možné nastaviť v systéme a nastaviť správne práva používateľom. Táto by mohla byť zlatá stredná cesta, ale v praxi niekedy je ľahšie konfigurácie nedovoliť. Dúfam, že esej otvorila niekoľko otázok a poskytla možnosti k ďalším úvahám a myšlienkam.

Použitá literatúra

1. L. Bendix, L. Borracci: *Towards a Suite of Software Configuration Management Metrics*. Lund Institute of Technology, Lund, Sweden, 2005.
2. Merant Inc. : *The Benefits of Software Configuration Management*.
http://pvcs.synergex.com/common/downloads/Benefits_SCM.pdf
3. A. M. J. Hass: *Configuration Management Principles and Practice*, Addison-Wesley Publishing Company, 2003

Annotation

The disadvantages of Software Configuration

Software systems are day-by-day bigger and more complex and every day thousands of people are working on these systems to keep them up to date. To reach the best performance of software systems they had to be configured in the right way. Sometimes the best way is to allow for users configure their software system alone for themselves. Essay opens a few questions about configurations and contains ideas about possibilities in software configuration. When and why should we allow users to configure software systems and why we shouldn't? Which level of the software is the best to configure our software?