

▼ Nástroje pro týmové studentské projekty

Authorgroup Petr Adámek

Tomáš Pitner

Keywordset Maven Subversion Java

Abstract V rámci řešení týmových studentských projektů v oblasti vývoje software je vhodné používat nástroje podporující týmový vývoj, mezi které patří například systémy pro řízení verzí. Pro úspěšné používání takových nástrojů je však nutné studentům poskytnout patřičnou podporu a pomoci jim překonat vstupní bariéru danou nedostatkem znalostí a zkušeností. Tento článek nabízí řešení postavené na nástrojích Subversion a Maven, které bylo úspěšně ověřeno v praxi na Fakultě informatiky Masarykovy univerzity v Brně.

▼ 1 Úvod

Na Fakultě informatiky Masarykovy univerzity v Brně jsou v rámci některých předmětů věnujících se programování řešeny týmové projekty. Jejich cílem je kromě praktického uplatnění získaných poznatků také nabytí zkušeností s prací v týmu a osvojení příslušných sociálních návyků. Týmy jsou tvořeny většinou čtyřmi studenty, což je dostatečné k řešení i větších projektů a zároveň přiměřené s ohledem na organizaci projektu a komunikaci mezi členy vývojového týmu. Řešení projektu je ukončeno jeho veřejnou obhajobou. Implementačním jazykem je v drtivé většině případu Java.

Původně byla organizace vlastního vývoje a případné používání vývojových a podpůrných nástrojů plně v kompetenci jednotlivých týmů a jediným požadavkem bylo zveřejnění informací o projektu na příslušných www stránkách. Po vyřešení projektu studenti měli na tyto stránky přidat i finální verzi vytvořených produktů a technickou dokumentaci. V prvních letech výuky byly příslušné předměty výběrové a zapisovali si je především studenti vyšších ročníků s hlubším zájmem o danou problematiku. Ačkoliv nebylo vše dokonalé, tento způsob organizace projektů se docela osvědčil.

Později se však změnilo zařazení zmiňovaných předmětů a začalo narůstat množství studentů z nižších ročníků, kteří nedisponují tolikou znalostí a zkušeností. Kvalita projektů poklesla a objevila se řada nových problémů. Někteří studenti začali mít problémy s koordinací práce, nevěděli jaké vhodné nástroje zvolit, případně je neuměli nainstalovat, nastavit a používat. Objevovalo se více technických dotazů, na něž nestačí obecná rada (použijte nástroj *A*), ale na které je nutno odpovědět velmi podrobně a konkrétně (nainstalujte si nástroj *A*, v konfiguraci nastavte v souboru *x* parametr *b* na hodnotu *c*, a pak jej spusťte pomocí příkazu *a xyz*). Dalším novým úskalím byla problematická reprodukovatelnost řešení projektu. Množily se případy, kdy výsledný produkt fungoval pouze na počítači jednoho z členů vývojového týmu a kvůli chybějící dokumentaci jej v podstatě nebylo možné zprovoznit jinde. V takové situaci, kdy jsme nebyli schopni příslušný fragment kódu vůbec přeložit a spustit, bylo velmi obtížné a především časově velmi náročné řešit problémy studentů a například v něm hledat chyby.

Proto jsme se rozhodli přístup ke studentským projektům změnit. Hlavní motivací bylo studentům usnadnit práci, seznámit je s některými užitečnými nástroji a naučit je tyto nástroje používat. Někteří pokročilejší studenti některé z nich používali již předtím, ale naším záměrem bylo je zpřístupnit i ostatním a také je unifikovat, abychom usnadnili práci i sobě.

V první fázi jsme nasadili systém pro řízení verzí Subversion, který usnadňuje týmový vývoj. Postupně jsme získávali další zkušenosti, vytvořili si sadu podpůrných skriptů pro snadné zakládání nových projektů, zpracovali

pravidla pro řešení projektů a používání systému Subversion, a nakonec nasadili i systém Maven, který automatizuje řadu činností spojených s řešením projektů.

Kromě technického řešení bylo nutné nachystat dokumentaci a návody a připravit se na poskytování technické podpory.

▼ 2 Subversion

Subversion je moderní systém pro řízení verzí s open source licencí, který je možné bezplatně používat upravovat i šířit. Je považován za nástupce klasického systému CVS a odstraňuje jeho základní nedostatky. Podobně jako systém CVS používá centrální server, na kterém je uložena repozitory obsahující všechny projekty. Pokud tato koncepce s centrálním serverem nevyhovuje, pro plně distribuovaný vývoj je možné použít např. systém Arch nebo Monotone. Pro naše účely se však jedná a nevhodnější řešení. Jeho používání je jednoduché a centrální repozitory zajistí, že každý vývojář pracuje s aktuální verzí kódu.

Primární úlohou systému pro řízení verzí je ukládat soubory a adresáře, které daný projekt obsahuje, umožnit jejich modifikaci různými vývojáři, všechny změny zaznamenávat a zabránit poškození či ztrátě části kódu způsobené jeho souběžnou modifikací více vývojáři. Systém je transakční a pokud se nepodaří nějakou změnu úspěšně dokončit (např. kvůli náhlému výpadku sítě), kód v repozitory zůstane nedotčen a nehrozí riziko jeho poškození.

Princip práce se systémem Subversion je podobný jako s jinými systémy pro správu verzí. Nejdříve je nutné projekt založit, což obnáší přidání příslušného adresáře do repozitory. Aby bylo možné s projektem pracovat, je nutné získat jeho lokální kopii (označovanou jako *pracovní kopie*). To se provede pomocí příkazu **svn co <url>**, kde <url> je adresa projektu v repozitory. Například níže posaný systém pro správu projektů má adresu <https://kore.fi.muni.cz/repos/fi/tools/2004/projectmanager/trunk>. Pokud již pracovní kopii máme a chceme ji pouze aktualizovat (tj. získat změny, které v projektu mezitím provedli ostatní vývojáři), použijeme příkaz **svn update**. Pokud chceme do projektu přidat další soubor, je třeba to systému oznámit příkazem **svn add <jméno_souboru>** a chceme-li naopak některý ze souborů z projekty odebrat, použijeme příkaz **svn delete <jméno_souboru>**. Pokud chceme změny provedené v pracovní kopii (úprava obsahu souborů, jejich přidání či odebrání apod.) uložit do repozitory, použijeme příkaz **svn commit -m "<popis změn>"**. Možnosti systému Subversion jsou mnohem rozsáhlejší, podrobnosti lze nalézt např. v dokumentaci na adrese <http://svnbook.red-bean.com>.

Díky systému Subversion mohou jednotliví vývojáři na projektu pracovat souběžně a všichni mají k dispozici aktuální verzi projektu. Pokud se objeví nějaká nová chyba nebo se vývoj začne ubírat nesprávným směrem, je snadné se vrátit ke kterékoliv předchozí verzi. Kdokoliv se může podívat do historie změn a zjistit, co jeho kolegové změnili a proč.

Příjemným vedlejším efektem nasazení systému pro řízení verzí je možnost kontroly průběhu práce na projektu, díky které můžeme průběžně ověřovat jeho stav, kvalitu a použitelnost. Při hodnocení projektu lze pak přihlídnout k příspěvku jednotlivých členů týmu a k dodržení harmonogramu jeho řešení (zda byl projekt dokončován na poslední chvíli apod.).

Pokud má být přínos systému Subversion maximální, je nutné dodržet několik základních pravidel, mezi které patří psaní smysluplných a výstižných popisů jednotlivých změn či kontrola správnosti a přeložitelnosti celého projektu před odesláním změn do repozitory. Každá odeslaná změna by měla být ucelená a nesouvisející úpravy by měly být odesílány zvlášť. Do repozitory by se měly vkládat pouze zdrojové soubory, které se nedají vygenerovat z jiných souborů. Rozhodně tam nepatří například přeložené třídy nebo používané knihovny. Tyto soubory totiž zabírají zbytečně místo v repozitory, zpomalují systém a komplikují jeho údržbu.

▼ 3 Maven

Maven je systém, který umožňuje automatizovat většinu činností souvisejících s řešením projektů v jazyce Java. K tomuto účelu se obvykle používají nástroje Make nebo Ant, pro něž se však musí vytvořit příslušný skript. To vede k tomu, že každý den několik vývojářů vymýšlí strukturu svého nového projektu a tráví čas psaním zmíněných skriptů. Při používání systému Maven se žádné skripty psát nemusí; místo toho se projekt popíše pomocí XML dokumentu, který se nazývá *popisovač projektu* (*project descriptor*). Tento popisovač obsahuje základní informaci o projektu a jeho vývojářích, adresy webových stránek, repozitory systému řízení verzí, poštovních konferencí a stránek pro nahlašování chyb, seznam potřebných knihoven a jejich verzí, apod. Na základě těchto informací potom vykonává příslušné operace, který je díky řadě zásuvných modulů obrovské množství.

Kromě základních operací, jako je přeložení zdrojových kódů projektu, spuštění testů, generování dokumentace apod., existují např. operace pro generování různých zpráv a reportů, seznamů změn či kompletních stránek projektu. Například následující příkaz **maven site:deploy** vygeneruje kompletní www stránky projektu a vystaví je na příslušném WWW serveru. Užitečné jsou také moduly kontrolující dodržování zásad pro psaní kódu nebo hledající potenciální chyby v kódu.

Velmi silnou stránkou systému Maven je způsob práce s knihovnami. Většina projektů vyžaduje nějaké knihovny, které je nutné mít nainstalované spolu s aplikací, a které je nutné přidat do proměnné `CLASSPATH`. Tyto knihovny se buď distribují zároveň s aplikací, nebo je nutné je stáhnout zvlášť. V obou případech to přináší komplikace a práci navíc. Vývojáři mají někdy tendenci problém zjednodušit tak, že potřebné knihovny vloží do repozitory s projektem, což je ale špatné řešení (viz předchozí kapitola). Systém Maven řeší tento problém velmi elegantně. Z popisovače projektu zjistí, jaké knihovny daný projekt vyžaduje, tyto knihovny automaticky stáhne prostřednictvím internetu a přidá je do systémové proměnné `CLASSPATH`. Knihovny pochopitelně nejsou stahovány pokaždé, ale pouze v okamžiku, kdy jsou poprvé potřeba.

▼ 4 Technické řešení a jeho vývoj

Jak již bylo zmíněno v úvodu, prvním krokem při zavádění nástrojů pro podporu vývoje týmových projektů bylo nasazení systému Subversion. Nejdříve bylo jeho použití dobrovolné, abychom si ověřili jeho funkčnost a použitelnost na malém vzorku studentů.

Prvním zjištěním byla nutnost vytvoření pomocných nástrojů pro zakládání projektů. Tento proces je totiž netriviální a obnáší několik kroků:

- vytvoření adresáře s projektem v repozitory;
- nastavení přístupových práv k adresáři s projektem v repozitory, aby do něj mohli zapisovat pouze studenti řešící daný projekt;
- založení adresáře pro WWW stránky projektu;
- vytvoření skupiny studentů řešících projekt;
- nastavení přístupových práv k adresáři s WWW stránkou projektu, aby do ní mohli zapisovat členové skupiny vytvořené v předchozím kroku.

Kromě toho, že jde o pracný proces, je zde relativně velké riziko chyby. Proto byla vytvořena sada skriptů, která tento proces zautomatizovala. Vstupem byl XML dokument popisující projekt a jeho vývojáře, který kromě informací potřebných pro založení projektu obsahoval i další informace použitelné pro generování stránek se seznamem projektů.

V dalším semestru bylo již použití systému Subversion povinné, nicméně šlo o výběrový a pokročilý předmět, který navštěvují studenti vyšších ročníků magisterského studia. Kromě objevení několika nedostatků ve skriptech pro zakládání projektů jsme narazili na první vážnější problém. Ačkoliv byli studenti důrazně upozorněni na zákaz vkládání knihoven a přeložených tříd do repozitory, téměř polovina projektů tento zákaz nedodržela. To způsobilo výrazný nárůst velikosti repozitory řádově v desítkách MB, přičemž se dalo očekávat, že by to v budoucnu s

rostoucími počtem projektů mohlo vést ke zbytečným komplikacím. Proto jsme se rozhodli přijmout následující opatření:

- do subversion repozitory doplnit kontrolní skript (*hook*), který nepovolí vložení souborů s příponami zip, class a jar;
- sepsat podrobná a přesná pravidla pro používání Subversion repozitory;
- poskytnout studentům nějaký efektivní mechanismus pro práci s knihovnamy.

Uvědomovali jsme si také fakt, že zejména pro studenty nižších ročníků může být problémem navrhnout optimální adresářovou strukturu projektu a vytvořit skripty pro překlad aplikace. Abychom tuto vstupní bariéru minimalizovali, rozhodli jsme se při vytváření adresáře projektu v repozitory tento adresář naplnit jednoduchým projektem s vhodnou strukturou a hotovým skriptem pro základní operace (překlad projektu, jeho spuštění, generování WWW stránek s projektem apod.). Nejdříve jsme se pokoušeli o vytváření univerzálního skriptu pro nástroj Ant, ale když jsme narazili na výše zmíněný problém s knihovnamy, rozhodli jsme se nasadit nástroj Maven. Jak se později ukázalo, přineslo to řadu dalších výhod. Tou nejvýznamnější je bezesporu snadná reprodukovatelnost řešení, neboť je snadné projekt stáhnout z repozitory a pak provést libovolnou operaci podporovanou nástrojem Maven bez nutnosti provádět jakékoliv další kroky.

Posledním problémem byla integrovaná vývojová prostředí (*IDE*). Chtěli jsme studentům umožnit používat jimi preferované nástroje včetně IDE, jenomže každé IDE má poněkud odlišnou strukturu projektu. Většinou umožňují tuto strukturu přizpůsobit, vyžaduje to však netriviální zásah do projektových souborů daného IDE. Jako nejvhodnější řešení se nám proto jevilo kromě souborů projektu generovat i projektové soubory pro nejpoužívanější IDE, tj. pro NetBeans a Eclipse. To by sice řešilo problém s odlišnou strukturou projektu, při používání IDE by však nebylo možné využít schopností Mavenu v oblasti práce s knihovnamy, neboť každé IDE má svůj vlastní mechanismus správy knihoven.

Naštěstí jsme objevili projekt Maven-ide, který integruje Maven do integrovaných vývojových prostředí NetBeans, Eclipse a Idea. S tímto modulem je možné v těchto prostředích otvírat libovolné projekty využívající Maven. IDE pak místo vlastních projektových souborů používá informace z popisovače projektu a pro všechny operace využívá služeb nástroje Maven (včetně práce s knihovnamy).

5 Současná verze

V současné době jsou při řešení projektů již povinně používány nástroje Subversion i Maven. Studenti, kteří chtějí řešit nějaký projekt, vytvoří jednoduchý XML dokument s popisem projektu a seznamem vývojářů. Jde o zjednodušenou verzi popisovače projektu pro nástroj Maven, která obsahuje pouze základní údaje nutné pro založení projektu.

S využitím tohoto dokumentu jsou provedeny následující kroky:

- Vytvoří se popisovač pro nástroj Maven; kromě údajů ze vstupního dokumentu se do něj vloží údaje, které je možné odvodit (např. adresa stránek projektu, adresa repozitory apod.).
- Vyvoří se vzorový projekt; tento projekt je ve správném balíku, obsahuje všechny náležitosti a je možné jej přeložit, otestovat i spustit.
- Projekt se vloží do svn repozitory.
- Nastaví se přístupová práva k adresáři s projektem v repozitory.
- Založí se adresář pro WWW stránky projektu.
- Nastaví se přístupová práva k adresáři se stránkami projektu.
- Zašle se vývojářům zpráva o založení projektu; tato zpráva obsahuje základní informace pro vývojáře (např. adresy WWW stránek a repozitory projektu).

Po technické stránce je tento postup zajištěn skriptem pro nástroj Ant a sadou XSLT šablon. Jde o plně funkční

avšak prototypové řešení, přičemž momentálně probíhá vývoj sofistikovanějšího nástroje pro údržbu repozitory a projektů.

6 Závěr

Popisované řešení je momentálně nasazeno asi pro 60 projektů a zatím se osvědčilo. Někteří studenti narážejí na různé problémy, většinou jde ale o nějakou drobnou chybu. Nejčastější problém kupodivu spočívá ve spouštění nástroje Maven v nesprávném adresáři, při příští přednášce o používání nástrojů Subversion a Maven bude nutné zdůraznit, že se nástroj Maven musí spouštět v adresáři obsahujícím popisovač projektu. Další problémy se občas objeví při instalaci nástroje Maven na soukromé počítače studentů, což je obvykle způsobeno nějakým nestandardním nastavením

Footnote Problémem bývají například chybějící programy ssh a scp na platformě MS Windows, které jsou zapotřebí pro kopírování WWW stránek na WWW server. Je však možné stránky jenom vygenerovat a na server je zkopírovat ručně.

Nečekaný problém se projevil neschopností studentů správně vyplnit jednoduchý XML dokument. Chyby ve struktuře dokumentu budou eliminovány pomocí XML Schématu, pomocí kterého budou dokumenty před zpracováním validovány. Větším problémem však zůstanou věcné chyby, mezi které patří nesprávné přihlašovací jména studentů nebo adresy elektronické pošty. Tyto chyby jsou bohužel velmi frekventované a jediným řešením je zřejmě zavedení přísných sankcí, které přimějí studenty, aby si dokument před odesláním důkladně zkontrolovali.

Obvyklým problémem u nástrojů Subversion a Maven je jistá vstupní bariéra, kterou je nutné překonat. V našem případě se nám ji podařilo minimalizovat, neboť projekt do Subversion repozitory studentům zavedeme, popisovač projektu pro Maven vygenerujeme a vytvoříme i vzorový projekt.

Při nasazování podobných systémů je velice důležitá osvěta a podpora studentů, přičemž je také nutné poskytnout rady a informace, jak vývoj organizovat. Připravili jsme jednu přednášku, která se věnovala používání nástrojů Subversion a Maven a byla zpracována dokumentace a jednoduché návody na jejich používání. V diskusních fórech promptně odpovídáme na dotazy studentů. Díky tomu drtivá většina studentů s těmito nástroji problém nemá, přesto však některým studentům jejich používání činí potíže. Objevují se bohužel i jedinci, kteří jsou přesvědčeni, že jsou tyto nástroje absolutně zbytečné a že jim tím pouze zbytečně přidáváme práci. Problém spočívá v chybějících zkušenostech studentů, kteří problémy spojené s týmovým vývojem projektů neznají. Proto nejsou schopni ocenit nástroje, které jim tyto problémy ušetří. Další problémovou skupinou jsou studenti, kterým nevyhovují předepsané nástroje nebo případně struktura projektu, protože jsou zvyklí používat něco jiného.

Náročnost pro vyučující vzrostla o trochu více, než jsme očekávali, neboť se objevují dotazy a problémy související s novými nástroji. Přesto jsme přesvědčeni, že přínos projektů pro studenty je dostatečný, aby toto kompenzoval. Navíc očekáváme, že v průběhu užívání systému dojde k vyřešení většiny drobných technických problémů a že se nám podaří na základě dotazů a připomínek studentů zkvalitnit dokumentaci a návody tak, abychom většinu dotazů a nejasností eliminovali.

Plánujeme doplnit podporu pro jiné typy projektů než aplikace v Javě, doplnit podporu pro Java Web Start, rozšířit dokumentaci a přidat nové návody (např. jak zveřejňovat vlastní dokumenty na WWW stránky projektu). Taktéž je nutné vyřešit způsob vystavování hotových projektů. Uvažujeme o nepovinném nasazení dalších nástrojů, které by vývoj usnadnily (např. diskusní fóra, elektronické poštovní konference, systém pro správu chyb apod.). Některé projekty již používají systém Wiki, který se pro tento účel velmi osvědčil.