

Metodika agilního vývoje softwaru na OVSS ÚVT

Vendula Švendová, ÚVT MU

1 Úvod

Tento článek popisuje agilní metodiku vývoje softwaru, kterou úspěšně používáme v našem týmu na Oddělení vývoje systémových služeb. Agilní metodika je způsob rozvržení a ověřování práce. Při důsledném dodržování jejích pravidel se lze vyhnout problémům jako jsou:

1. z pohledu zákazníka:

- software vyvíjený na základě sepsaných požadavků zákazníka po dokončení zákazníkovi nevyhovuje (změnily se či přibýly požadavky) a produkt je třeba přetvořit;
- produkt není dokončen včas a je tudíž v požadovaném čase nefunkční;
- produkt není dokončen v rámci rozpočtu;

2. v rámci týmu:

- jednotliví členové týmu nevědí, na čem pracují jejich kolegové;
- vedoucí týmu se nepotkává dostatečně často se členy týmu a nemá každodenní čerstvé informace, jak projekt postupuje a jaké řeší členové týmu problémy.

Většina terminologie v článku je uvedena v českém překladu. Některé pojmy jsou však ponechány v původní anglické podobě, abychom zachovali srozumitelnost a spojitost s terminologií užívanou ve většinou anglických textech jiných autorů, stejně jako při běžném provozu.

2 Agilní manifest

Co slovo „agilní“ vlastně znamená? Podle slovníku cizích slov čilý, aktivní, horlivý. Základní určující podmínkou agility v kontextu vývoje programů je možnost změny zadání v průběhu celého vývoje. Zákazník tedy má možnost za chodu svoje požadavky modifikovat, aniž by to mělo za následek masivní přetváření již provedené práce, a tím zbytečné plýtvání časem i zdroji všech zúčastněných.

Agilní metodiky vznikly v polovině 90. let minulého století jako reakce na těžké tradiční metodiky, kterým byla vytýkána byrokratičnost, zkosnatělost, neschopnost flexibilně reagovat na změny.

Tradiční programovací metodiky mají na začátku vývoje fixně danou funkcionalitu, které se snaží dosáhnout; naproti tomu čas a zdroje jsou proměnné – podřizují se oné funkcionalitě. Agilní metodiky naopak mají pevně dané zdroje a časové úseky (tzv. iterace), během nichž se postupně implementují jednotlivé vlastnosti produktu dle jejich priority. Po implementaci každé vlastnosti by měl být systém provozuschopný.

Častá, zákaznicky nevídaná, situace „uplynul termín pro dokončení, ale software je stále nefunkční, proto potřebujeme více času“ tak v agilním vývoji nemůže nastat. I v případě, že dojde ke zpoždění, stane se pouze to, že nebudou implementovány vlastnosti s nejnižší prioritou (tudíž nebude ohrožena funkčnost systému).

Agilních metodik existuje několik typů (Extrémní programování, Feature-Driven Development, Dynamic Systems Development Method, Agilní modelování, SCRUM aj.) a jejich společné rysy byly zformulovány v roce 2001 v tzv. *Manifestu agilního vývoje softwaru* [1] organizací Agile Alliance [2]. Sepsání manifestu bylo inspirováno dvěma hlavními myšlenkami:

- umožnit změnu je mnohem efektivnější než se jí snažit zabránit;
- je třeba být připraven na nepředvídatelné události, protože ty stejně nastanou.

Základními body manifestu jsou:

- přednost individualitám a komunikaci před procesy a nástroji;
- přednost funkčnímu SW před obsažnou dokumentací;
- přednost spolupráci se zákazníkem před jednáním o smlouvě;
- přednost reakci na změnu před plněním plánu.

Jinými slovy při dodržení těchto specifik se nemůže stát, že zákazník není spokojen, přestože je smlouva naplněna a obohacena o rozsáhlou dokumentaci.

3 SCRUM

Na našem oddělení používáme jednu z agilních metodik známou jako Scrum. Samotné slovo „scrum“ (zkrácenina slova „scrummage“, do češtiny překládáno jako „mlýn“) je převzato ze sportovní terminologie ragby a je to způsob, jak obnovit hru po ztrátě míče. Míče se zmocní ten tým, jehož členové spolu lépe komunikují a který je tedy lépe koordinovaný. Přesně to je principem

Scrumu – členové týmu spolu neustále úzce komunikují, práci si společně plánují a jsou pravidelně informováni o tom, jak postupuje projekt jakožto celek, na čem pracují jejich kolegové, předávají si své know-how, a tím se práce týmu výrazně zefektivní.

3.1 Tým

Důležitou vlastností, která odlišuje Scrum od tradičních metodik, je zapojení zákazníka do týmu a rozhodování o prioritách implementovaných funkcionalit.

Tým je rozdělen na dva základní typy rolí – kuřata a prasata. Jejich pojmenování vychází z bajky o kuřeti a praseti, kteří se rozhodnou založit restauraci a přemýšlí, jak ji pojmenují. Kuře přijde s nápadem „Ham&Eggs“, což se praseti nelíbí, neboť kuřete by se proces pouze týkal, kdežto prase by bylo přímo zapojeno. Pokud by restaurace byla firma, pak kuřata budou zákazníci, kteří poskytnou požadavky a prostředky na uskutečnění projektu, prasata potom zaměstnanci firmy, kteří zapojením vlastního úsilí vytvoří produkt.

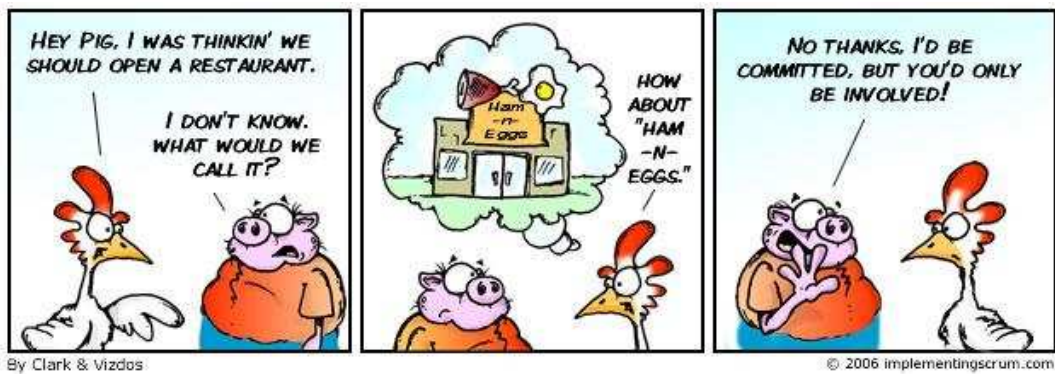
Konkrétně se role v týmu dělí do následujících:

Mezi kuřata patří

- **Zainteresované strany** (stakeholders) – koncoví uživatelé vyvíjeného systému, investoři, manažeři firmy, která software vyvíjí.

Mezi prasata patří

- **Vlastník produktu** – reprezentuje zájmy zákazníka, určuje priority jednotlivých vlastností produktu, pravidelně je upravuje, přijímá/odmítá výsledky práce;
- **Scrum vedoucí** – osoba, jejímž úkolem je zajistit hladký průběh práce – ochraňuje tým před vnějšími vlivy a udržuje ho koncentrovaný na daný úkol, vynucuje dodržování všech pravidel, organizuje pravidelné schůzky, udržuje přehled o úkolech, které je nutno splnit, odhaluje vzniklé překážky a zahrnuje je do plánu, a v neposlední řadě sleduje osobní problémy a konflikty mezi členy týmu a snaží se vytvořit příjemné pracovní prostředí;
- **Vývojáři** – osoby zodpovědné za vytvoření vlastního produktu, vzájemně úzce spolupracující.



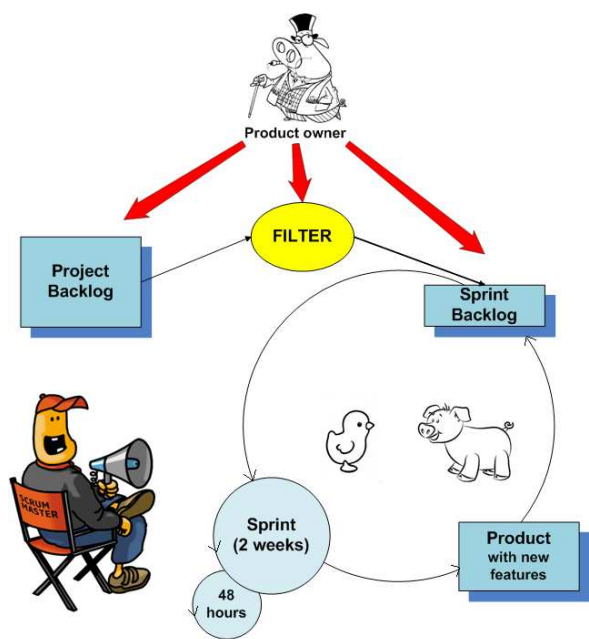
Obrázek 1: Bajka o praseti a kuřeti

3.2 Scrum proces

Celý Scrum proces probíhá ve striktně vymezených časových intervalech tzv. *sprintech*. Proces začíná sepsáním požadavků zákazníka (vlastnosti produktu) a ohodnocením jejich priorit vlastníkem produktu. Na začátku každého sprintu jsou vybrány požadavky dle priorit a v množství úměrném délce sprintu (obvykle 2–4 týdny). Jak už bylo řečeno v úvodu, výsledkem každého sprintu by měl být provozuschopný software. V každé další iteraci pak už jen přibývají vlastnosti produktu podle jejich priorit, přičemž funkčnost softwaru zůstává zachována. Je tedy možné zavést verzování tak, že co sprint, to nová verze.

3.3 Artefakty

Stavebními kameny Scrumu jsou tzv. *položky* (Backlog items). Jsou to ony zákazníkem jasně pojmenované a prioritizované vlastnosti, které by měl výsledný produkt mít (např. aplikace umí na dálku zastřežit všechny sledované místnosti; po zastřežení se zelená barva místnosti změní na modrou apod.). Každý projekt má svůj tzv. *Backlog*, což je seznam všech položek, které se týkají daného produktu. Na začátku každého sprintu se z něj pak vytvoří tzv. *Sprint Backlog*, obsahující pouze položky určené pro daný sprint. Backlog je dynamický v závislosti na nově vzniklých podnětech (mohou se měnit priority jednotlivých požadavků, přibývat nové,...). Zde je vidět zmiňovaný rozdíl oproti tradičním metodikám, ve kterých jsou vlastnosti výsledného produktu fixně dané, a jakékoli vyvstalé komplikace je třeba podřídít původnímu plánu, přes-



Obrázek 2: Scrum proces

tože třeba existuje daleko elegantnější a úspornější řešení.

3.4 Schůzky

Všechny schůzky jsou organizovány a moderovány Scrum vedoucím, který by měl zajistit jejich hladký průběh a veškeré výstupy ze schůzek zaznamenávat.

3.5 Plánování sprintu

Plánování je nejdůležitější a nejdelší schůzkou, kterou je zahájen každý sprint. Účastní se jej všechna prasata a možná, ne však nutná, je i účast kuřat. Na této schůzce vlastník produktu představí Backlog a podle priorit spolu se zbytkem týmu vytvoří program nadcházejícího sprintu – Sprint Backlog. Klíčové je vybrání položky důkladně rozplánovat, což zahrnuje:

- každé položce přidělit vlastníka z řad vývojářů, který je zodpovědný za její dokončení;
- rozepsat každou položku do jednotlivých úkolů, tzv. tasků;
- časově každou položku ohodnotit.

Časové ohodnocení je obzvláště důležitá a obtížná část plánování. Většinu z nás se nejednou stalo, že jsme si na nějakou práci vyhradili nadhodnocený čas s tím, že „se to musí zaručeně stihnout“, a po uplynutí oné přehnané doby jsme se divili jak velké množství „se nestihlo“.

Každý člen týmu by měl mít jasno, kolik hodin bude mít během Sprintu k dispozici (odečíst dovolené, svátky, pracovní schůzky,...) a z tohoto času je třeba odečíst ještě 30 % (pro výskyt nečekaných problémů).

Existuje několik metod jak správně plánovat čas určený pro řešení úkolů. Samozřejmě platí: čím méně komplexní úkol, tím snazší je odhadnout jeho časovou náročnost, proto je třeba plánovat dostatečně podrobně, aby byl odhad co nejpřesnější. Nejčastěji používaná metoda pro plánování je tzv. *plánovací poker*, jehož pravidla jsou následující:

Každý hráč (člen týmu) dostane do rukou karty s hodnotami Fibonaccioho posloupnosti (1, 2, 3, 5, 8, 13, 21, 34,...), které symbolizují časové

úseky (hodiny, dny,...). Stanoví se položka, jejíž časové ohodnocení je cílem kola hry. Vývojář, v dané oblasti nejzkušenější, krátce zbytku týmu představí, o co se jedná, co vše je třeba v dané věci udělat a jaká jsou pravděpodobná úskalí problému. Tým má příležitost klást dotazy a o problému diskutovat. Následně každý z týmu vybere jednu kartu, reprezentující časový odhad položky, a položí ji lícem na desku stolu. Ve chvíli kdy mají všichni odloženou kartu, karty se obrátí a odhady porovnají. Ti, kteří mají nejextrémnější hodnoty, jsou vyzváni k odůvodnění a je vyvolána nová diskuse. Postup hry se opakuje do chvíle, než se všechny odhady shodují.

Výsledkem této schůzky by tedy měl být konkrétní časově ohodnocený plán, který se tím záváže splnit. Tím, že se vývojáři podílí na plánování, je i jejich zainteresovanost na výsledku vyšší než při pouhém plnění úkolů nadřazených a jejich práce je předvídatelná a spolehlivá.

3.6 Stand Up

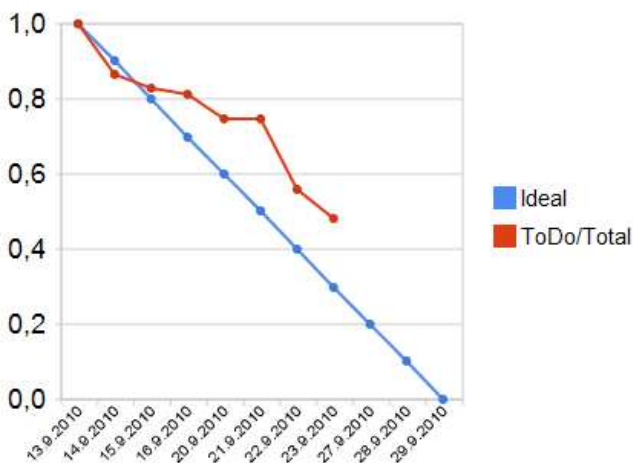
Tým je pravidelně informován o tom, jak postupuje práce všech jeho členů. Tak se děje na (ideálně) každodenní ranní Stand Up schůzce. Název naznačuje charakter takovýchto setkání – všichni přítomní stojí a schůzka netrvá déle než 10 minut. Účast prasat je povinná, účast kuřat možná. Na programu jsou odpovědi všech členů týmu na následující tři otázky:

- Co jsem udělal včera?
- Co udělám dnes?
- Co mě zdržuje v práci? (potřebuji pomoci s laděním programu, rozbil se mi počítač, nemohu sehnat osobu XY, ...)

Stand Up schůzka není určena k řešení problémů, není to ani sběr informací o tom, kdo je pozadu. Cílem je informovanost týmu o postupu celého projektu, o komplikacích, které objevil jeden člen a mohou ovlivnit práci ostatních. Často je práce jednoho člena vázána na dokončení práce jiného a StandUp schůzky jsou chvíle, kdy se tyto informace předávají. Tým práci sám naplánoval a sám sobě se zodpovídá.

3.7 Review

V den následující konec sprintu proběhne tzv. Review schůzka. Během této schůzky prasata



Obrázek 3: Graf Burndown

představí kuřatům výsledky své práce (v nějaké formě prezentace). Jak už bylo několikrát zmíněno, software je funkční po každém sprintu a každá nově přidaná vlastnost jeho funkčnost je nom rozšiřuje.

3.8 Retrospektiva

Po Review následuje poslední schůzka sprintu, tzv. Retrospektiva. Zde se sejdou všechna prasa a společně zhodnotí uplynulý sprint. Všichni členové týmu opět zodpoví několik otázek:

- Co se mi líbilo, co jsme dělali dobře a chci abychom v tom pokračovali?
- Co se mi nelíbilo a chtěl bych změnit?
- Co by se dalo vylepšit nebo zefektivnit?

Smyslem Retrospektivy je subjektivní zhodnocení sprintu členy týmu (nikoli pouze jejich vedoucím), čímž jsou zapojeni do rozhodování o procesu, každý vyjádří svůj názor, poslechne názory ostatních a společně se snaží vyvarovat chyb v příštím sprintu.

3.9 Sledování průběhu Sprintu

Průběh sprintu je zaznamenáván pomocí různých typů grafů. Nejdůležitějším z nich je tzv. *Burndown*, viz obrázek 3.

Na svislé ose je množství práce, kterou je třeba během sprintu udělat (formou procent či hodin), na vodorovné ose čas. Lomenou čarou je vyznačen skutečný průběh sprintu (ToDo/Total),

druhá čára znázorňuje ideální průběh sprintu (Ideal), tedy rovnoměrné snižování množství práce. Na tomto grafu je vidět, že během prvních dvou dní tým pracoval rychleji, od třetího dne však nestihá. To může být způsobeno výskyt nečekaných komplikací, případně špatným plánováním (což by při dodržení výše uvedeného postupu nemělo nastat). Pomocí takového grafu lze jedním pohledem zjistit, v jakém stavu projekt je. Samozřejmě to vyžaduje důsledné každodenní zaznamenávání provedené práce do vhodného nástroje.

Pro záznam veškerých artefaktů, statistik, průběhů sprintů, grafů existuje celá škála nástrojů. My máme zkušenosti s produktem VersionOne Enterprise [3] společnosti VersionOne, Inc., který jsme používali rok. Vzhledem k jeho obsáhlosti a finanční náročnosti jsme se rozhodli licenci na další rok nekoupit a používáme nyní námi upravenou formu Google Spreadsheet, viz obr. 4.

4 SCRUM na oddělení OVSS

S implementací Scrumu jsme na našem oddělení začali zhruba před rokem. Počáteční nesmělé kroky zahrnovaly špatně naplánované a především špatně pojmenované položky a nedůsledné dodržování pravidel. Během několika měsíců jsme se však přesunuli k poměrně zaběhnutému režimu, jehož přínos je nezpochybnitelný.

Naše modifikace:

- položky označujeme písmeny M, S, W (Must, Should, Would), které ještě před konkrétním rozplánováním stanoví jejich prioritu.
- Burndown graf aktualizujeme na tabuli na chodbě, takže každý zjistí při příchodu do práce jediným pohledem, v jakém stavu je aktuální sprint.
- Pro záznam podrobností celého sprintu jsme si upravili Google spreadsheet, který je dostupný odkudkoli přes Gmail.
- Vedle Burndown grafu vytváříme i grafy pro sledování časů skutečně strávených při vývoji (ideálně by se měly shodovat s naplánovanými) a stavů položek (nová, rozpracovaná, hotová, nestihám, nenaplánovaná).

Item	ID	Owners	Časový odhad	Task	Owner	Časový odhad	Effort	ToDo	Status
Křížkamu půjde připojit automat	B-01249	Dáko	26	Příprava pro demonstraci malopros-pro-ISISAM Úprava AutomateWebService-pro-identifikaci-pomoci-karset Úprava AutomateWebService-pro-identifikaci-pomoci-karset Dodání návazací AutomateWebService-na-FBS-pro-prostý-do-ISISAMu Odstranění závislosti automatu na GUIFO-a navázání na FBS Přesunuti metod pro-Stericky-z-AWS-do-vlastní služby	Dáko	6	6	0	Done
Lze vytisknout soupis reklamaci (podobně jako Dodák)	B-01250	Dáko, Filip	8	Úprava AutomateWebService-podle-Filipových-pozadavku Tržní dodávkového pdf	Dáko Filip	3 5	2 5	0 0	Done Done
V Iris lze zastřežit např. servovna na Gotexu	B-01252	Matej, Orla	15	Úprava GUI zastřežení modul zastřežuje-ozajstný-smart-modul	Orla Matej	13 2	15 2	3 0	In Progress Done
Každý v Iris vidí to na co má oprávnění	B-01253	Matej	13	logika zbarování oprávnění-zo-SES filrování-a-vyvození-oprávnění	Matej	8	4	0	Done

Obrázek 4: Používaný Google Spreadsheet

- Vzhledem k vysokému procentu dohodářů v našem týmu pořádáme Stand Up schůzky pouze každý druhý den.
- Detailní rozplánování dílčích částí položek (tasků) je v režii vlastníka položky, který za její dokončení zodpovídá.

Přínos:

- vlastnost, kterou na začátku sprintu naplánujeme, je na konci sprintu hotová;
- pracujeme primárně na věcech, které jsou nejdůležitější;
- vedoucí týmu přesně ví, v jaké fázi produkt aktuálně je a které funkcionality bude na konci sprintu obsahovat;
- každý člen oddělení ví, na čem pracují ostatní a vzniklé problémy společně řeší.

5 Závěr

Scrum není jedinou agilní metodikou, ale vybrali jsme si ji, protože je relativně jednoduchá a její zavedení bylo v našem prostředí rychlé a efektivní. Mezi její hlavní výhody patří zejména úzké propojení uživatele aplikace s vývojovým procesem, krátké a rychlé iterace, které přinášejí zákazníkovi hodnoty průběžně a poměrně často, čímž zvyšují jeho spokojenost.

Nesporným přínosem je Scrum v univerzitním prostředí při zapojení studentů do vývoje. Díky flexibilitě plánování je zaručeno, že je možno proces přizpůsobit jejich individuálním časovým možnostem – StandUp schůzky zvládnou hravě mezi výukou a v každém sprintu je jim naplánováno pouze tolik hodin, kolik mají skutečně

k dispozici. Zejména pro studenty začínající s vývojem, jsou 14denní sprinty ideálním dávkovacím prací, a pro jejich nadřazené výbornou možností kontroly.

Literatura

- [1] Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>
- [2] Agile Alliance. <http://www.agilealliance.org/>
- [3] Versionone. <http://www.versionone.com/>
- [4] Wikipedia. Scrum (development). [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development)) □

Trendy v bezpečnosti počítačových sítí

Václav Lorenc, Honeywell, spol. s r.o.

*Netflou sondy, fájrvóly,
bezpečnostní Velká zed'.
Dat nám z toho lezou hory,
jenže ... co s tím dělat teď?*

Ve světě počítačových sítí je možné pozorovat poměrně rychlý vývoj techniky, která lidem umožňuje vyměňovat si vzájemně informace. V souvislosti s tím se však objevuje celá řada problémů, zejména stran bezpečnosti uživatelů a dat. Jaké pokroky a vyhlídky je možné očekávat v této oblasti?

1 Čeho je moc, toho je příliš

Počítačové sítě byly původně prostředím přátelským, ve kterém se jednotliví účastníci snažili k sobě chovat slušně a spořádaně. Později přišly první útoky a s nimi i první obranné mechanismy. S většími sítěmi a méně technicky znalými uživateli se ukázalo, že bude možná vhodnější umístit různé linie obrany ještě před nicnetujícími uživateli. A tak se začala objevovat jednoduchá pravidla bránící důležité součásti firemních i akademických sítí. Stejně jako slavnostní porizení prvního modemu, prvního serveru a prvního displeje z tekutých krystalů, i provoz firewallů byl pln očekávání a obav, naplněných i lichých.

Jedna ze zajímavých průvodních vlastností firewallů je ta, že rády povídají svým administrátorům, co všechno se jim děje. To samé umí i chytřejší aktivní síťové prvky. Jsou-li takových zařízení nejvýše jednotky, ještě se to dá vše zvládnout. Pokud jich je několik desítek, přestává už být v silách i těch nejsikovnějších správců pročitat denní hlášení a včas reagovat na potenciální problémy.

Na ÚVT vznikla před několika lety pro potřeby Oddělení datových sítí závěrečná práce, která zpracovávala část informací generovaných aktivními prvky – systémové logy, tzv. syslog. Díky tomu je možné každé ráno obdržet nejen souhrnný přehled toho, co si jednotlivé stroje myslí o chování vlastním i připojených zařízení, ale i výčet podivností, rozličných nekonzistencí a případných bezpečnostních incidentů, které tato zařízení uměla rozpoznat a povšimla si jich za minulý den.

Jednalo se o výrazné, byť pouze dočasné zlepšení. Záhy se totiž objevila nová vlna internetových červů, které mají v popisu práce vyhledávat zranitelnosti připojených strojů a co nejrychleji se šířit dál. Hlášení podávané ráno tak bylo stále zajímavé, ale aktuální problémy bylo nutné odhalovat a řešit rychleji.

Navíc se kvůli rozvoji služeb instalovala celá řada dalších a dalších zařízení dožadujících se moudrého a vlídného dohledu – bezdrátové sítě a jejich centrální řídicí prvky, autentizační a autorizační servery, systémy na detekci a prevenci síťových útoků (IDS, IPS), sondy sbírající informace

o provozu v síti, síť Eduroam a s ní i autentizační protokol 802.1x...

Ačkoliv každé zařízení hlásilo něco jiného, jednu složku všechny zprávy měly společnou – jednalo se o hlášení související s počítačovou sítí, v tomto případě sítí Masarykovy univerzity. Z různých pohledů a aspektů, které byly natolik roztržštěné, že zvažovat jeden jediný nemuselo dávat kdovíjak smysluplné či důležité informace.

2 SIEM – Security Information and Events Management

Zkrátka – ukazuje se, že mnoho šikovných zařízení má o provozu na síti co říci. Současně se zdá zřejmé, že různé informace mají různou prioritu. A že události, které na první pohled a s odstupem pár minut vypadají neškodně a bezzubě, se mohou v denním či týdenním přehledu objevit ve vší své zubaté žraločí kráse. Jeden z větších takových projektů, který se snažil obsáhnout a zjednodušit analýzu souvisejících událostí v počítačové síti byl ambiciózní produkt fy Cisco, *CS Mars (Cisco Security Monitoring, Analysis and Response System)*. I přes odvážná tvrzení masivní obchodní kampaně se začaly ukazovat různé drobnosti, které nakonec vedly k tomu, že společnost Cisco celý projekt nedávno oficiálně ukončila.

Problém nebyl ani tak ve špatné základní myšlence, tedy centrální analýze a korelaci událostí, ale zejména v tom, že úvodní nastavení celého produktu bylo natolik komplikované, že v rozsáhlých sítích v podstatě nemohlo dojít k jeho úspěšnému nasazení. Správci byli buď zahlceni zbytečnými událostmi nebo naopak ani nedostávali ty důležité – správné vyhodnocení toho, co pro danou síť důležité je a co není, nebylo a není triviální úlohou.

Jiný problém byl současně i v tom, že ne všechny upovídáné technologie jsou od jediného dodavatele. A pokud se každý výrobce vcelku oprávněně rozhodne, že nad svými zařízeními umí dělat inteligentní dohled lépe, než jiní, nastává trochu jiná verze původního problému – sice už někdo data předzvíkal a zjemnil do grafické a koláčové podoby, ale stále každému z těchto systémů může chybět důležitý kousek skládačky.

A tak se objevil nový marketingový pojem, tzv. „Security Information a Events Management“, tedy správa bezpečnostních informací a událostí. Cílem je, kromě prodeje nových produktů a nových krabiček, vylepšit výsledky síťových dohledových systémů. A to nejlépe tak, že se budou integrovat výsledky jedné dohledy do dohledů jiných, nad tím vším se provede nějaká rozumná úvaha a půjde-li vše dobře, dojde k propojení různých událostí správným způsobem a zlotřilý program bude zakázán programem moudrým.

3 Síť bez hranic

Ačkoliv se jedná o další z termínů používaných firmou Cisco, lze i na myšlence Síť bez hranic (Borderless Networks) demonstrovat další zajímavý posun toho, jak se chování uživatelů v síti mění.

Dříve, bez existence mobilních zařízení, byla situace poměrně jednoduchá a přímočará. Pracovní počítač byl v práci, domácí spokojeně dlel doma. Navíc domácí počítače často nemusely mít ani připojení k internetu. Pracovní síť tak mohla být uzavřeným jezírkem – zlé věci se děly venku a bylo třeba správně kontrolovat přítoky, aby štiky nevpadly dovnitř.

Notebooky, bezdrátové sítě a chytré mobilní telefony přinesly do poklidných firemních rybníčků hejna dravých ryb. Podnikové sítě se tak stávají mnohem zranitelnějšími, neboť notebook, pečlivě chráněný před internetovým Zlem, se přenesením do kaváren, letištních sítí či sítí univerzit stává terčem nemalých zkoušek. A chytré mobilní telefony? Ty klidně snad ani nezažily. Neodolají-li svodům vnějšího světa, jsou po připojení zpět do podnikové sítě zdrojem nenápadné interní náklady.

Aby však bylo možné chránit i součásti sítí, které jsou velmi mobilní, bylo potřeba část centralizované inteligence schopné rozpoznat útoky přenést i na počítače a mobilní telefony. A s každým takovým zařízením přibývá pochopitelně i možných hlášení toho, co se jim děje za příkoří.

Jeden ze způsobů, kterak kontrolovat bezpečnost koncových stanic a alespoň trochu zmenšit množství generovaných informací, se skrývá

pod jménem *Unified Access Control*. Celé řešení umožňuje kontroly připojovaných koncových bodů, např. aktualizace operačních systémů a antivirových programů, a to tak, aby v případě problémů mohly co nejrychleji, tedy automaticky, zareagovat například firewally a umístit takový stroj do karantény. Zajímavá na UAC není ani tak myšlenka spolupráce všech důležitých prvků a zařízení různých výrobců, ale skutečnost, že se tak autoři pokusili učinit prostřednictvím protokolu IF-MAP (*Interface for the Metadata Access Point*). Neboť jde o protokol otevřený, existují v současné době i volně dostupné implementace jeho serverové části.

4 Firewally nové generace

S rozvojem sítí, jejich rychlostí, schopností a provázaností, se objevily i nové typy poskytování služeb.

Problém cestujících uživatelů nutil k zamyšlení i v jiném ohledu – když já jsem v Austrálii, kolega v Kolumbii a servery máme pouze na Jižní Moravě... Je to optimální? Například služby firmy Google jsou roz distribuované po celém světě tak, aby si každý mohl najít „svůj nejbližší Google“. Tím se jednak zmenšuje zátěž tranzitních sítí, a jednak zlepšuje uživatelský zážitek z takové služby.

Umíte si však představit administrátora, jenž je postaven před úkol zabránit škodlivým serverům, aby traumatizovaly uživatele firemních notebooků? Pokud by chtěl povolit přístup na stránky Masarykovy univerzity či IS MU, má úkol v podstatě snadný, tyto servery opravdu na Jižní Moravě sídlí. Jenže co ostatní služby, rozstřelené promyšleně po celém globu?

Dá se říci, že IP adresa jako identifikátor serveru a související služby již nemusí dostačovat. Uživatelé totiž zpravidla chtějí „povolit Facebook“, „video z Youtube“, případně si „pokecat na ICQ“. IP adresy pro ně nejsou důležité, důležitější jsou služby, které vyžadují.

A tak nezbylo, než vymyslet nový koncept zařízení, tentokrát z analytické společnosti Gartner, označovaných termínem „Next Generation Firewalls“. Jejich úkolem je usnadnit administrátorům orientaci v existujících službách, zkoumat

důležité kousky provozu a následně zkusit odhadnout, o jaké služby se vlastně jedná. A buď je včas povolit, nebo zakázat.

„Chci používat P2P sítě typu XYZ“, „Potřebuji, aby mi fungovala videokonference a hlasové hovory“, „Což takhle ochránit náš SQL server před škodlivými dotazy?“. To je zhruba cíl toho, co by zkratka NGFW měla v budoucnu dokázat. S maximálními možnými rychlostmi a co nejvyšším uživatelským komfortem. V některých navrhovaných variantách se prý uvažuje i o tom, že by před povolením služby vyplňovali uživatelé dotazník, v němž by vysvětlili důvody pro povolení požadované služby a případně i spokojenost s nimi. Což opět může zrychlit procesy i evidenci existujících výjimek a umožňuje efektivně realizovat změny v pravidlech firewallů na základě zpětné vazby od uživatelů.

5 Kudy dál?

Zdá se tedy, že snahou výrobců bezpečnostních zařízení je umožnit uživatelům, aby používali své oblíbené služby bezpečně a současně správcům, aby ono bezpečí mohli zajistit v rámci nastavených firemních pravidel. S rozvojem nových konceptů a distribuovaných služeb se to i za cenu mírných změn pracovních postupů nejspíše daří. Nikoliv nějakými revolučními myšlenkami, ale spíš postupným vývojem těch existujících.

Jako u mnoha dalších oborů, i zde se ukazuje, že není problém vyrobit a vygenerovat vagóny dat, je problém v nich dostatečně rychle najít nejzajímavější informace pro konkrétní skupinu lidí. A v tom vidí budoucnost pravděpodobně i velké společnosti. Bezpečnost, vysokorychlostní sítě, distribuované aplikace a maximální mobilita uživatelů, to vše by mělo zajistit hladký chod aplikací a vznik nových myšlenek a konceptů.

Právě proto se na ÚVT MU testovalo nasazení SIEM nástrojů a uvažuje se o rozšíření projektu, v rámci kterého byla vyzkoušena a zdokonalena infrastruktura pro použití nástrojů unifikovaného řízení přístupu k síti. Kombinací těchto přístupů by bez omezování uživatelů mohlo dojít k výraznému posunu v obraně před nezvanými návštěvníky a snižování škod vzniklých nevhodnými programy.

Celé povídání by se dalo snad shrnout pod jediné slovo – spolupráce. Ať již mezi výrobci, jednotlivými zařízeními nebo správci a uživateli. Bez spolupráce na některé z těchto úrovní, nebo ideálně na všech, to zkrátka nejspíš nepůjde.

Literatura

- [1] IF-MAP. Dostupné na <http://en.wikipedia.org/wiki/IF-MAP>.
- [2] NGFW. Dostupné na <http://img1.custompublish.com/getfile.php/1434855.1861.sqycbrdwq/Defining+the+Next-Generation+Firewall.pdf> □

Požadavkové systémy v Inetu

Zdeněk Machač, ÚVT MU

Všichni to známe: úkoly, termíny, plnění... S rostoucím počtem úkolů, termínů řešení a dalších doprovodných dat a snižující se schopností naší paměti udržet toto množství informací v hlavě nám nezbývá, než si vypomoci některým ze známých způsobů.

Někteří si úkoly zapisují do sešitů či na různé papíry, papírky nebo různobarevné lepící čtverečky, které jsou pak rozesety po pracovním stole, monitoru nebo klávesnici (tak často činí i autor). Jiní na to jdou v dnešní elektronizované době pokrokově a zaznamenávají si požadavky do elektronických dokumentů v počítači nebo si je zapisují a organizují v e-mailové schránce. Ti zdatnější používají různé k tomu určené nástroje dostupné on-line na Internetu. V případě, že je řešitelem úkolů/požadavků skupina více osob, je poslední jmenovaná možnost téměř nutností. O sběru požadavků v takovém nástroji vyvíjeném v rámci informačního systému Inet MU je tento článek.

1 Modelový příklad

Abychom si odpověděli na otázku, v čem nám (obecně) nástroje pro sběr požadavků mohou pomoci, uveďme si modelový příklad z praxe ÚVT – z vývoje a provozu Inetu a také ekonomického informačního systému firmy Magion (EIS

Magion). Uživatelé se na provozní tým denně obraceli s žádostmi o pomoc při řešení problémů s aplikacemi, o přidělení práv, o zjištění či upřesnění informací z jednotlivých agend nebo jen o radu, jak postupovat při práci v jednom či druhém informačním systému. Běžnou, a také dříve hojně využívanou cestou, byla e-mailová komunikace, která má však v našem asi 20členném týmu svá úskalí.

I přes zavedení skupinových aliasů se uživatelé často e-mailem obraceli přímo na konkrétní osobu, jejíž případná nepřítomnost vyřešení problému neúměrně prodlužuje. I skupinové aliasy mají svá úskalí: Stane se, že je řešení požadavku započato více vývojáři zároveň, nebo naopak vývojáři spoléhají jeden na druhého a řešení požadavku stojí. Špatně adresované zprávy (na špatný alias nebo osobu) nebo předání řešení jinému vývojáři se řeší přeposíláním e-mailů ze schránky do schránky, a tím narůstá pracnost a čas.

Shrneme-li výše popsané, pak komunikace a informace o řešení jednoho požadavku může být distribuována ve více e-mailech a mezi více e-mailových schránek. Zvládnout takový proces řízení – kdo má problém řešit, jestli je již řešen nebo vyřešen, a zda je zadavateli korektně odpovězeno – je značně obtížné. Systémy pro sběr požadavků, pokud jsou správně používány, nám mohou účinně pomoci tyto obtíže zvládat.

2 Historický exkurz

V Inetu je již od roku 2004 v provozu aplikace pro sběr požadavků na změnu dat v telefonní ústředně, kterou využívají správci na součástech univerzity na straně zadavatelů, a dvě oddělení ÚVT jako řešitelé. O něco později vznikla aplikace pro sběr požadavků na změnu práv ve zmiňovaném EIS Magion. Obě aplikace byly šity na míru daným potřebám, a nedaly se tak použít pro jiné účely.

Prvotními impulzy pro zavedení obecného požadavkového systému v rámci Inetu byla žádost správců IT z lékařské fakulty (kteří si podobnou aplikaci vytvořili sami, ale nechtěli ji už více rozvíjet a udržovat) a také obtížně udržitelný stav řešení úkolů a požadavků v našem týmu vývojářů. Nejdříve jsme hledali volně dostupné již

hotové řešení, které by vyhovovalo našim potřebám a bylo do Inetu snadno včlenitelné, abychom mohli využít potenciálních uživatelů a nám dostupných dat (personalistika, ekonomika, provoz a správa). V hledání jsme však neuspěli, a tak nezbylo, než se pustit do vývoje vlastními silami.

Využili jsme idejí ze školní seminární práce tehdejších našich dvou dohodářů Pavla Budíka a Martina Jakubičky, a vývoj pilotní verze systému později zadali jako diplomovou práci pro Juraje Skubáka. Výsledek diplomové práce byl nasazen, a také studentem úspěšně obhájen, ale k dnešnímu stavu vedla ještě dlouhá cesta, kdy se systémem upravoval, přepracovával a zobecňoval, aby dostal žádané vlastnosti, které si popíšeme dále. Hned na počátku však zdůrazníme, že naším cílem nebylo vytvořit ani systém pro vývoj softwaru, ani systém pro řízení týmových projektů, které mají mnoho společných rysů, ale metodicky jsou lépe zvládnuty v existujících komerčních i bezplatných aplikacích.

3 Technické podrobnosti

Aby se čtenář neztratil v pouhém výčtu technických vlastností, rozdělíme si je do několika logických celků, a pokud to bude možné, popíšeme v ukázkách na našem modelovém příkladě.

3.1 Skupiny osob

Osoby jsou přebírány z centrální evidence MU, tj. z personálního a studijního systému, a pro definování jejich skupin byla využita existující komponenta používaná v jiných subsystémech Inetu. V jedné instanci požadavkového systému je možné použít neomezené množství skupin, které mohou být definovány jako:

- vyjmenovaný seznam osob,
- osoby mající aktivní vztah k vybranému pracovišti s výběrem typu vztahu (zaměstnanec, dohodář, student),
- osoby s funkcí na univerzitě nebo součásti univerzity (rektor, kvestor, děkan, tajemník, ...),
- kombinace předešlých,
- všichni uživatelé Inetu.

Příkladem skupin v požadavkovém systému mohou být např. zadavatelé, možní řešitelé nebo

„dohlížitelé“ – odpovědné osoby. Mimo pevně zadané skupiny jsou v systému také dynamické skupiny odvozené od konkrétního požadavku: např. jeho zadavatel, přiřazený řešitel nebo výběr skupiny podle hodnoty některého z atributů.

3.2 Data, atributy a datové typy

Základem každého požadavku či úkolu jsou informace o něm. Některé jsou společné všem typům použití (instancím systému), např. kdo požadavek zadal a kdy, identifikace požadavku atd. Jiné údaje se velmi různí pro různé instance, například: popis a priorita požadavku, příloha, řešitelé a jejich komentáře atd. Systém tak pro každou instanci definuje seznam tzv. atributů, tj. seznam údajů, jejich datových typů a také omezení. Např. popis problému je text o maximální délce 4 000 znaků, priorita je pevný seznam položek, ze kterých si uživatel povinně vybírá, řešitelem je osoba z definované skupiny (viz výše). Mezi základní typy patří:

- krátký a velmi dlouhý text,
- celé a desetinné číslo,
- datum a čas,
- osoba,
- soubor,
- seznam pevně daných hodnot.

Nový datový typ však může být do systému s určitou námahou přidán. K dispozici je také datový typ sdružující atributy do větších celků (např. adresa bydliště se může skládat z obce, ulice, č. popisného a PSČ) a také typ pro násobné uložení různých hodnot atributů (např. více příloh, více osob, ale také v kombinaci s předchozím – více adres).

Možná omezení rozsahu hodnot jsou různá podle datového typu:

- délka a vzor textu (PSČ, telefon, e-mail atd.),
- minimální a maximální hodnota čísla nebo data,
- velikost souboru a jeho typ.

Systém tedy zajišťuje kontrolu a správné ukládání dat podle definice atributu. Žádný ze zadaných údajů se při změně hodnoty nepřepisuje; vše je uchováno historicky, a je tak možné se zpětně podívat, jak a kdo požadavek v čase měnil. U některých položek je však žádoucí vidět

ihned všechny uložené hodnoty – typickým příkladem jsou komentáře řešitelů v interní komunikaci.

3.3 Stav a jejich workflow

Požadavky nejsou neměnné a během svého životního cyklu procházejí různými stavy. V našem modelovém případě je požadavek na počátku ve stavu založený, poté je převzat, řešen, vyřešen, a nakonec je potvrzena správnost řešení. Nad seznamem stavů je definován workflow, tj. zda je z jednoho stavu pro vybranou skupinu osob povoleno přejít do stavu druhého. Ve výše popisované aplikaci pro telefonii na MU se ukázala potřeba mít v dané instanci více nezávislých workflow, a tedy i více seznamů stavů, protože stavy řešení pro obě skupiny řešitelů jsou vzájemně nezávislé. V systému nejsou omezeny ani počet stavů ani možné přechody.

3.4 Viditelnost a práva

Dáme-li předchozí tři části systému dohromady, určíme tím pro každou trojici [skupina osob \times stav každého workflow \times atribut], zda je/není hodnota atributu vidět nebo zda je možné/povinné hodnotu vyplnit či změnit. Ukázka nastavení práv modelového příkladu v našem editoru je vidět na obrázku 1.

V editoru je například vidět, že při uzavření požadavku již není možné data kýmkoliv měnit (pouze číst – R), interní komunikaci mohou (CW) vyplňovat pouze administrátoři či řešitelé, zadavatel musí vyplnit popis (MW) nebo administrátor může změnit kategorii, ale musí být vyplněna (CMW).

3.5 Upozornování

Nesmíme také zapomenout na informování uživatelů o vzniku a změnách požadavku. Pro větší flexibilitu lze v instanci systému povolit posílání e-mailových upozornění buď při změně stavu, nebo při změně hodnoty kteréhokoliv atributu – samozřejmě pro každou skupinu osob zvlášť. Obsah e-mailu je pak dán vybranou šablonou (pevné i proměnlivé texty a jejich rozmístění) a také výběrem atributů, které se v něm mají zobrazit.

		zadavatel	popis	interni_komunikace	priloha	priorita	kategorie	odpovedna_osoba	resitel	nazev
zadáva se	ADMINI_IHELP									
	RESITEL_IHELP(adm,tech)									
	ZADAVATELE_IHELP	MW	MW		CW	MW	MW			MW
	ZADAVATEL(sub)									
	RESITEL(own)									
neřešeno	ADMINI_IHELP	R	R	CW	R	R	CMW	CMW	CMW	R
	RESITEL_IHELP(adm,tech)									
	ZADAVATELE_IHELP									
	ZADAVATEL(sub)	R	CW		CW	CMW	R	R	R	R
	RESITEL(own)	R	CW	CW	R	R	CMW	R	CMW	R
řeší se	ADMINI_IHELP	R	CW	CW	R	R	CMW	CMW	CMW	R
	RESITEL_IHELP(adm,tech)									
	ZADAVATELE_IHELP									
	ZADAVATEL(sub)	R	CW		CW	CMW	R	R	R	R
	RESITEL(own)	R	CW	CW	R	R	CMW	R	CMW	R
vyřešeno	ADMINI_IHELP	R	CW	R	R	R	R	R	R	R
	RESITEL_IHELP(adm,tech)									
	ZADAVATELE_IHELP									
	ZADAVATEL(sub)	R	R		R	R	R	R	R	R
	RESITEL(own)	R	CW	CW	R	R	R	R	R	R
uzavřeno	ADMINI_IHELP	R	R	R	R	R	R	R	R	R
	RESITEL_IHELP(adm,tech)									
	ZADAVATELE_IHELP									
	ZADAVATEL(sub)	R	R		R	R		R	R	R
	RESITEL(own)	R	R	R	R	R	R	R	R	R

Obrázek 1: Ukázka nastavení práv

4 Vzhled a chování

V předchozích částech jsme si popsali vlastnosti našeho systému, ale chybí nám celkový pohled na něj. Každá instance systému je rozdělena do 3 záložek; na první jsou seznamy požadavků, na druhé detail a na třetí vyhledávání požadavků. Na přání je však možné přidat i další záložky se specifickými údaji. Při vstupu do aplikace systém zkontroluje, zda je přihlášený uživatel alespoň v jedné ze skupin oprávněných osob, a v každé ze záložek pak zobrazuje jen ty informace, na něž má uživatel právo. Ukázky uživatelského rozhraní jsou na obrázcích 2 a 3.

První (vstupní) záložka může obsahovat několik seznamů požadavků nalezených podle zadaných kritérií a práv. Mohou to být např. seznamy mnou zadaných, mnou řešených nebo posledních 10 vyřešených požadavků. Počet a kritéria jsou pro každou instanci volitelná a volitelné jsou také zobrazené atributy pro každý požadavek. U každé položky v seznamu je odkaz na detail. Druhá záložka slouží pro zadávání nových a také editaci a zobrazení detailu a historie již existujících požadavků. A konečně na třetí zá-

ložce lze hledat požadavky podle zadaných hodnot atributů.

Námi popisovaný modelový případ jsme implementovali a zpřístupnili v menu Inetu jako *Požadavkový systém uživatelské podpory Inetu* (zkráceně *iHelp*), na něž vedou odkazy z každé aplikační stránky (zápatí vpravo). Přimo z aplikace tedy může každý uživatel zadat svůj požadavek na poskytnutí pomoci. Jediné, co je nutné vyplnit (podobně jako v e-mailu), je název, popis, kategorie a priorita problému či úkolu, a případně přiložit soubor ve formě přílohy.

5 Závěr

V době psaní tohoto článku jsou v Inetu k dispozici tyto instance požadavkového systému:

- zadávání požadavků pro Centrum informačních technologií SUKB (tato instance vznikla z původního požadavkového systému správců IT z lékařské fakulty),
- požadavky pro správu stavebního pasportu,
- zmiňovaný systém uživatelské podpory Inetu,
- dva systémy pro interní podporu vývoje Inetu a veřejných stránek univerzity,

Hlavní stránka Zadání nového úkolu Vyhledávání úkolů

Nepřřazené
Nebyly nalezeny žádné úkoly.

Přřazené ostatním

Nastavení

Strana: 1 z 3 (počet položek: 26)

	Zadáno	Název	Priorita	Stav	Kategorie	Zadavatel	Odpovědná osoba	Řešitel
1	18.03.2011	Žádost o aktualizaci katalogu DNS ICT č.9	střední	řeší se	DNS	Michal Baudys	Šárka Ocelková	Šárka Ocelková
1	18.03.2011	Nastavení práv pro Ing. Voráčovou	střední	vyřešeno	W3MU	Vladimír Šmíd	Šárka Ocelková	Šárka Ocelková
1	18.03.2011	sestava pracovních cest v INETu	střední	řeší se	Personalistika	Jaroslava Stanková	Šárka Ocelková	Jana Haluzová
1	18.03.2011	práva pro J. Tomanovou	vysoká	vyřešeno	W3MU	Vladimír Šmíd	Ivan Burian	Ivan Burian
1	17.03.2011	Řídící a akademické funkce na MU - Žádost o aktualizaci	vysoká	řeší se	Personalistika	Martin Celhoffer	Petr Vokřínek	Petr Vokřínek
1	17.03.2011	53177 Celhoffer: změna tel., změna čísla pracovní	vysoká	vyřešeno	Kontakty	Martin Celhoffer	Petr Písek	Petr Písek
1	17.03.2011	Aktualizace katalogu - TNT	střední	vyřešeno	DNS	Nadja Voráčová	Šárka Ocelková	Šárka Ocelková
1	17.03.2011	Aktualizace katalogu - KP	střední	vyřešeno	DNS	Nadja Voráčová	Hana Vališová	Hana Vališová

Obrázek 2: Vstupní stránka se seznamy požadavků/úkolů v IHelpu

Hlavní stránka Zadání nového úkolu Vyhledávání úkolů

Zadání nového úkolu

Název:

Kategorie:

Popis:

Příloha: Procházet...

Priorita:

Obrázek 3: Ukázka zadání požadavku do IHelpu

- objednávky tisku velkoformátových posterů v ÚVT - ukázka této nejsložitější implementace je na obrázku 4 (více posterů s rozdílným nastavením tisku, při výběru platby přes systém SUPO kontrola na existenci a platnost účtu, a odhad výsledné ceny podle plochy pokrytí).

V přípravě je dále systém pro požadavky na elektronické podepisování dokumentů (s externí aplikací usnadňující celý proces podepisování), a systém pro sběr a evidenci požadavků resp. problémů týkajících se EIS Magion, z nichž druhý jmenovaný je určen nejen pro potřeby MU, ale také ostatních vysokých škol provozujících

stejný ekonomický informační systém.

V předchozím textu jsme se pokusili ukázat možnosti a komplexnost celého systému a jsme otevření implementacím dalších instancí pro potřeby zaměstnanců univerzity. Případní zájemci se mohou obracet na e-mailovou adresu inet@ics.muni.cz. □

Přechodové mechanismy k IPv6 (2)

David Rohleder, ÚVT MU

V minulém díle jsme si popsali základní principy přechodových mechanismů mezi IPv4 a IPv6 a

Pokud si nevíte rady, najedte myši na ikonu nápověda.

Platba:

Kontaktní email:

Telefon:

Požadované datum vyzvednutí: čas:

Zapůjčit tubus:

Poster:

Typ papíru:

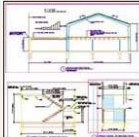



Rozměry:

Počet kopií:

Ořez:

Soubor:

Verze powerpointu:

A	B	C	D
			
169 Kč	191 Kč	253 Kč	405 Kč

Cena závisí na potisknuté ploše. Podle statistik můžete porovnat ceny. Pro běžné postery se cena pohybuje mezi skupinou B a C.

Obrázek 4: Objednávka tisku posterů

popsali jsme protokol ISATAP (používaný hlavně v organizacích). V dnešním díle se zaměříme na zbývající dva nejrozšířenější mechanismy, a to 6to4 a Teredo.

1 6to4

Technologie 6to4 byla definována v RFC 3056 jako jeden z automatických mechanismů spojování IPv6 ostrůvků přes IPv4 síť. 6to4 používá celý IPv4 internet jako jednu spojovací síť.

6to4 používá pro své adresování IPv6 adres z rozsahu 2002::/16 a zbytek IPv6 adresy se pak vytváří z přidělené IPv4 adresy.

16b	32b	16b	64b
2002	WWXX:YYZZ	Subnet ID	Interface ID

6to4 rozlišuje tři základní druhy uzlů zapojených do tohoto mechanismu:

- *6to4 host* - zařízení ve vnitřní síti, které získá 6to4 IPv6 adresu ze směrovače prostřednictvím standardních mechanismů autokonfigurace (nebo jiným způsobem - DHCPv6, staticky).
- *6to4 router* - uzel s veřejnou IPv4 adresou, kterému může být přímo vytvořena 6to4 IPv6 adresa, a který je připojen k IPv4 internetu.

- *6to4 relay* - uzel, který je jedním rozhraním připojen do IPv4 internetu, druhým do IPv6 internetu a obstarává komunikaci mezi IPv4 internetem a IPv6 internetem.

1.1 Přidělení IPv6 adres

Každý IPv6 ostrůvek dostane svůj IPv6 prefix sestavený následovně: 2002:WWXX:YYZZ::/48, kde WWXX:YYZZ jsou čtyři bajty přidělené IPv4 adresy. Prefix /48 je standardizovaný prefix přidělovaný koncovým zákazníkům v IPv6. To umožňuje mít ve vnitřní síti až 65535 IPv6 podsítí, a tedy dostatečnou kapacitu pro adresaci všech vnitřních počítačů. Tam je možné použít standardní mechanismy pro přidělování IPv6 adres, včetně adresy směrovače. Tj. použijeme buď statickou konfiguraci, bezstavovou automatickou konfiguraci SLAAC nebo DHCPv6.

Pokud přidělujeme 6to4 adresu jedinému počítači (6to4 host), pak je možné adresu konstruovat libovolným způsobem odpovídajícím danému prefixu. Např. 2002:WWXX:YYZZ::1. Operační systémy z rodiny Microsoft Windows konstruuji adresy jako 2002:WWXX:YYZZ::WWXX:YYZZ.

1.2 Směrování

IPv6 adresy máme tedy přiděleny, co nám zbývá, je vyřešit směrování mezi jednotlivými počítači. To je překvapivě jednoduché. Máme tři možnosti podle toho, kde se koncový počítač nachází:

1. Pokud se koncový počítač nachází podle IPv6 adresy ve stejné síti, pak se použije klasické směrování v IPv6.
2. V případě, že nenastal případ 1 a adresa cílového počítače začíná prefixem 2002::/16, znamená to, že koncový počítač leží v jiném IPv6 ostrůvku. Ostrůvky jsou spojené prostřednictvím IPv4 sítě, přičemž z konstrukce 6to4 adresy známe koncovou IPv4 adresu 6to4 routeru/uzlu (ony WWXX:YYZZ). Vezmeme tedy IPv6 paket a zabalíme ho do IPv4 paketu s cílovou adresou WW.XX.YY.ZZ a pak pošleme standardní IPv4 síti, která ostrůvky propojuje. Cílový 6to4 router/uzel pak IPv6 paket vybalí z IPv4 obálky a pošle do vnitřní IPv6 sítě standardním směrováním.
3. Poslední varianta je ta, že cílový počítač neleží ani v místní síti ani v 6to4 ostrůvku, tj. leží ve standardní IPv6 síti. V této chvíli využijeme tzv. 6to4 relay, který je jedním rozhraním připojen do IPv4 internetu a druhým rozhraním do IPv6 internetu. Zdrojový počítač vyšle IPv6 paket vnitřní síti k defaultnímu směrovači, který IPv6 paket zabalí do IPv4 paketu a pošle jej na IPv4 adresu 6to4 relaye. 6to4 relay vybalí IPv6 paket a pošle jej standardním mechanismem do IPv6 internetu.

Pro poslední variantu zbývá vyřešit pouze dvě drobnosti. Tou první je, jak zjistit IPv4 adresu 6to4 relaye. K tomu máme tři možnosti:

- IPv4 adresu 6to4 relaye známe a staticky ji nakonfigurujeme do 6to4 routeru/uzlu;
- použijeme DNS. Microsoft windows používá k nalezení 6to4 relaye jméno 6to4.ipv6.microsoft.com;
- použijeme tzv. anycastovou IPv4 adresu. Anycast sice nebyl původně v IPv4 definován, nicméně funguje stejně jako v IPv6. Tj. uzlů s jednou IPv4 adresou může být v Internetu několik a směrovače pošlou paket vždy

k tomu nejbližšímu. Pro účely 6to4 tunelu se používá adresa 192.88.99.1.

Druhou překážku představuje cesta z IPv6 internetu zpět k 6to4 relayi. Zpáteční cesta je zajištěna tak, že každá 6to4 relay oznamuje do IPv6 internetu, že se za ním nachází síť 2002::/16. Zpáteční paket do 6to4 ostrůvku si pak vybere nejbližší 6to4 relay automaticky. Tento způsob řešení může mimo jiné způsobit asymetrickou cestu, kdy odesílající 6to4 relay bude jiná než 6to4 relay přijímající zpáteční paket.

Mechanismus 6to4 je tedy relativně jednoduché řešení, které by vždy fungovalo nebýt NAT na straně poskytovatele připojení. Pokud totiž nemá 6to4 router/uzel veřejnou IPv4 adresu, není možné zkonstruovat 6to4 IPv6 adresu tak, aby se zpáteční paket vrátil k původnímu odesílateli. Tento problém řeší až následující protokol.

2 Teredo

V dnešní době je Internet kvůli nedostatku IPv4 adres zaplevelen různými druhy NATu, které za jednu IPv4 adresou schovávají celé velké sítě. Internet byl přitom původně založen na principu přímé konektivity mezi koncovými uzly, čemuž NAT úspěšně brání. Většina dříve popsanych přechodových mechanismů nebude na počítačích za NATem fungovat (nebo pouze za zvláštních podmínek).

Aby bylo možné používat IPv6 i v takové situaci (ve které se nachází velká část domácích uživatelů, minimálně v ČR), byl vyvinut protokol Teredo, který si s NATem dokáže většinou poradit. Před tím, než se pustíme do popisu fungování samotného Teredo mechanismu, musíme se trochu seznámit s fungováním technologie NAT.

2.1 NAT

NAT (Network Address Translation) byla původně technologie pro přepisování IPv4 adres v IPv4 paketech. Sloužila hlavně jako dočasné řešení při přečíslování sítě, aby se adresy nemusely měnit na všech počítačích vnitřní sítě hned, ale mohla se využít nějaká dočasná doba, kdy počítače fungovaly v nové síti i se starými adresami. NAT byl provizorní řešení, protože některé protokoly kvůli tomu nefungovaly (FTP). Překlad byl

nestavový, a kvůli tomu jej bylo možné provádět vždy pouze v poměru 1:1.

Později vyvstala potřeba překládat adresy ve větším poměru než 1:1, aby se dosáhlo ušetření adresového prostoru. Pro tyto účely byl NAT rozšířen i o použití hlaviček čtvrté vrstvy (tj. např. UDP a TCP). Tímto vznikla čtveřice [src_IP, src_port, dst_IP, dst_port], a díky přemapování zdrojového portu bylo možné detekovat, ke kterému odchozímu paketu patří vracující se pakety. Tento princip se správně nazývá NAPT (Network Address and Port Translation), nicméně dnes se poměrně běžně zaměňuje za NAT, a NAT je takto v novém významu i vnímán. Pro tento způsob NATu je nutné udržovat stavovou tabulku překladů, což může v některých případech (málo výkonný hardware, příliš provozu) způsobovat problémy.

Použití informací z vyšších vrstev hlaviček paketů v NATu představuje pro standardní IPv6 tunelovací mechanismy problém, protože NAT obvykle pracuje pouze s protokoly TCP a UDP, jiným protokolům většina NAT směrovačů nerozumí. Ostatní IPv6 přechodové mechanismy totiž nepoužívají TCP ani UDP a pro zapouzdření IPv6 paketu do IPv4 paketu používají protokol číslo 41, ve kterém není nic, co by se dalo označit za číslo portu.

Teredo tedy zapouzdřuje IPv6 paket do UDP paketu, který snadněji projde NATem.

Pro přesnější členění různých druhů NATu rozlišujeme následující typy (podle RFC 3489):

- *cone NAT* (trychtýřovitý NAT) - vytvoří vazbu mezi vnitřní IP adresou a portem a vnější adresou a portem. Všechny pakety, které pak přicházejí na vnější adresu a port zvenku, jsou přeposílány na vnitřní adresu a vnitřní port (tj. je možné komunikovat z libovolné vnější adresy na správný vnitřní port vnitřního počítače, pokud před tím byla zevnitř vytvořena vazba vnitřní adresy a portu na vnější adresu a port. Tato vazba se vytvoří vysláním paketu směrem zevnitř ven).
- *restricted NAT* - podobná situace jako cone NAT, pouze s tím omezením, že s vnitřní adresou a portem může komunikovat pouze vnější

počítač, na který bylo navázáno toto spojení zevnitř, nezávisle na čísle portu.

- *symetrický NAT* - mapuje vnitřní IP a port na různé vnější IP a porty, s Teredo mechanismem funguje pouze v případě, že za takovým NATem je maximálně jeden z komunikujících partnerů.

2.2 Součásti Teredo mechanismu

Teredo rozlišuje tři součásti nutné ke správnému fungování:

- Teredo klient - koncový uzel podporující Teredo mechanismus;
- Teredo server - uzel připojený jedním rozhraním do IPv4 internetu a druhým rozhraním do IPv6 internetu. Teredo server slouží k inicializaci spojení mezi Teredo klienty nebo Teredo klientem a IPv6 uzlem;
- Teredo relay - uzel, který dokáže ukončovat Teredo tunely a směřovat pakety mezi Teredo klienty na IPv4 internetu a IPv6 uzly na IPv6 internetu.

2.3 Teredo adresace

IPv6 adresa Teredo klienta vypadá následovně:

32b	32b	16b	16b	32b
Teredo prefix	IPv4 adresa Teredo serveru	Příznaky	Modifikovaný externí port	Modifikovaná externí adresa

Teredo prefix: aby bylo možné rozeznat, že se jedná o protokol Teredo, je pro tento protokol vyhrazen prefix 2001:0000::/32.

Teredo server IPv4 adresa: adresa Teredo serveru, který spolupracoval na vytvoření Teredo adresy na klientovi.

Příznaky: náhodná data sloužící jako ochrana proti některým druhům útoků.

Modifikovaný externí port: je upravené číslo externího portu, na který se mapuje vnitřní IPv4 adresa. Tento port se systém dozví až po první komunikaci s Teredo serverem, protože mezitím prošel IPv4 paket NATem, který původní vnitřní port změnil.

Modifikovaná externí adresa: upravená externí IPv4 adresa, na kterou se mapuje vnitřní IPv4 adresa. Tuto adresu se klient dozví až po první komunikaci s Teredo serverem.

Jak si Teredo klient zjistí svoji IPv6 Teredo adresu? Celkem překvapivě jednoduše pomocí téměř standardního IPv6 ohlášení směrovače. Klient vytvoří IPv6 paket obsahující zprávu Router Solicitation, zabalí ji do UDP IPv4 paketu s cílovou adresou Teredo serveru. Tento paket projde přes NAT, dorazí na Teredo server, a ten odpoví upravenou RA zprávou, která v sobě obsahuje informaci o externí adrese a portu klienta za NATem. Klient pak tyto informace vezme a vytvoří si z nich podle výše uvedeného návodu svoji vlastní IPv6 Teredo adresu.

Tímto postupem si Teredo klient vytvořil také patřičný záznam v NAT tabulkách, takže je možné, aby Teredo server komunikoval s Teredo klientem (už může posílat pakety dovnitř, což se nám bude za chvíli hodit).

Mechanismus Teredo používá techniku, které se říká „NAT punching“ – proděravění NATu. Komunikace mezi Teredo klientem A a Teredo klientem B vypadá potom následovně:

1. Klient A, který chce komunikovat s klientem B, vyšle Teredo paket na externí adresu NATu klienta B (tuto IPv4 adresu odvodí z IPv6 adresy klienta B). Tím také vytvoří v NAT tabulkách díru, díky které bude možné v budoucnosti přijímat pakety od klienta B. NAT zařízení před klientem B tento paket zahodí, protože v jeho NAT tabulkách neexistuje žádný záznam, který by odpovídal danému paketu. Ale to nevadí, protože následuje:
2. Klient A vyšle paket (říká se mu bublina) Teredo serveru klienta B (IPv4 adresu tohoto serveru opět získá z IPv6 adresy klienta B). Teredo server klienta B už může komunikovat s klientem B díky již existující vazbě v NAT tabulkách zařízení klienta B.
3. Teredo server B předá tuto bublinu klientovi B. Klient B se tedy právě dozvěděl, že s ním chce komunikovat klient A.
4. Klient B vyšle bublinu přímo směrem ke klientovi A (jeho externí IPv4 adresu a port odvodí z jeho IPv6 adresy). Tím se vytvoří na NAT zařízení B záznam pro komunikaci od klienta B ke klientovi A.
5. Protože v prvním kroku jsme vytvořili na NAT zařízení A záznam pro komunikaci s klientem B, tak NAT zařízení A propustí paket až ke

klientovi A. Tímto je komunikace sestavena a klienti A a B si nyní mohou vyměňovat pakety navzájem.

Komunikace mezi Teredo klientem A a IPv6 only klientem C probíhá odlišně:

1. Teredo klient A vyšle ping přes Teredo server ke klientovi C.
2. Teredo server přepośle bublinu klientovi C.
3. Klient C odpoví, paket dojde na Teredo relay nejbližší ke klientovi C.
4. Teredo relay vyšle bublinu pro klienta A přes Teredo server klienta A (Teredo relay zatím nemůže komunikovat s klientem A napřímou, Teredo server už ale ano). Odpověď na ping paket zatím relay zařadí do fronty na vyřízení.
5. Teredo server přepośle bublinu klientovi A.
6. Klient A z bubliny zjistí IPv4 adresu Teredo relaye pro klienta C (relaye jsou pro různé IPv6 počítače různé). Klient A vyšle bublinu směrem k Teredo relayi a vyrobí tím tedy díru do NATu.
7. Teredo relay dostane bublinu, vezme odpověď na ping z fronty a pošle ji na IPv4 adresu NAT zařízení klienta A. NAT už má vytvořenou vazbu a paket přepośle. Tímto byla komunikace mezi A a C ustanovena a uzly tak mohou komunikovat mezi sebou.

2.4 Bezpečnostní rizika

O NATu se často říká, že je jistá forma bezpečnosti, protože na počítače není možné navazovat spojení z vnější sítě. Jak jsme si předvedli v případě mechanismu Teredo, není to pravda. Komunikace dokonce může probíhat mezi dvěma uzly, které oba jsou za NATem.

Obecně všechny přechodové mechanismy mohou představovat jisté bezpečnostní riziko, protože se často tvoří samy automaticky, bez nutnosti zásahu administrátora. Firewally nejsou většinou na takovou tunelovanou komunikaci připraveny a může se stát, že propustí pakety, které by v IPv4 zahodily.

O nebezpečnosti tohoto mechanismu hovoří už samotný název. Původně se totiž tento mechanismus jmenoval Shipworm, nicméně byl přejmenován na Teredo, protože worm (červ) nemá

v počítačových sítích zrovna dobré konotace. Autoři vtipně přejmenovali mechanismus na Teredo, což je latinský název pro anglický shipworm, v češtině *sášeň lodní* (drobný mlž dělající díry do dřevěných lodí). Je vidět, že i autoři suchých technických specifikací mají smysl pro humor.

Literatura

- [1] Toreda overview. <http://www.microsoft.com/technet/network/ipv6/teredo.mspx>
□

(N)evernote: poznámky vždy po ruce Tomáš Pitner, FI MU

1 Kde končí řízení ostatních a začíná řízení sebe

V univerzitním prostředí MU máme pro řízení projektů několik specializovaných systémů či agend – jak v Inetu, tak v IS MU. Přesto je řada situací, kdy komplexní řešení pro management projektů nevyhovuje. Problémy nastávají už tam, kde je v projektu angažováno více lidí volnějším způsobem (např. jako pozorovatelé, studenti, členové komunity kolem vývoje určitého SW, spolupracovníci z jiných institucí apod.). Aby mohli plnohodnotně pracovat s univerzitními nástroji, znamenalo by to mít je podchycené v určitém formálním vztahu s univerzitou, zavedené do centrálních databází apod., což je mnohdy dosti neoperativní nebo to ani nejde. Další případ, kdy se vůbec nehodí řešit to složitě přes podporu řízení projektů, jsou aktivity sice pracovní, ale osobního charakteru: mám napsat recenzi pro časopis, pročíst diplomku, splnit deadline na projekt, článek, výuku, potřebuji si udělat rychlou poznámku z jednání,.... Takové jednorázové úkoly nemá smysl podchycovat pod něco jako „projekt“, bylo by to umělé, protože neexistuje něco jako jeden cíl takového projektu, jeho kritéria úspěšnosti, etapizace apod. Pro tyto

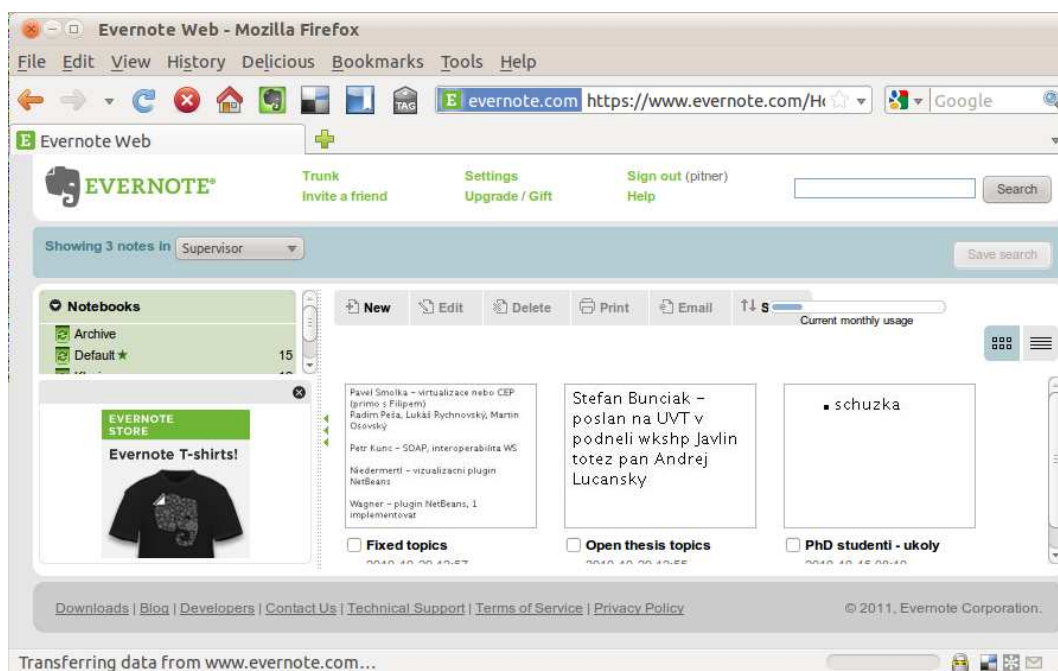
účely saháme po nástrojích jednodušších, označovaných jako *PIM* (Personal Information Management). Obvykle v nich můžeme rychle napsat krátkou poznámku, poznačit si úkol, případně ofotografovat třeba diagram z tabule.

2 Poznámky a úkoly na desktopech

V desktopovém prostředí všech běžných obecně používaných operačních systémů máme nástroje na psaní poznámek. Čím modernější systém, tím je komfort vyšší a styl práce se přibližuje tomu, na co jsme zvyklí z papírové agendy – lístečky nalepené na monitoru či nástěnce, ale navíc s možností prohledávání, upozorňování na termíny apod. Příkladem jsou poznámky na desktopu ve Windows 7 nebo na Linuxech zdařilé aplikace Tomboy, KNotes, Gnote, KJots a další. Jejich možnosti ale obvykle končí ve chvíli, kdy počítač vypínáme. Pak musíme vystačit s poznámkami na papírcích po kapsách, ti systematictější s klasickým papírovým nebo elektronickým diářem či PDA, které se alespoň tak často neztrácejí :-).

3 Online systémy

Dalším stupněm v použitelnosti jsou online služby s podobným účelem, jaký byl naznačen výše. Databáze poznámek je hostovaná u poskytovatele služby a lze k ní přistupovat odkudkoli, kde je webový prohlížeč. Poněkud sice klesá interaktivita a komfort není plný – služba je pomalejší, není tak dobrá integrace s desktopovým prostředím (např. nelze snadno drag-and-drop přetahovat položky z desktopu počítače) – ale zato máme své poznámky kdekoli, kde je počítač a připojení k internetu. V dnešním online propojeném světě ale chceme ještě více. Moderní chytré telefony nás udržují online 24 hodin denně a není to jen proto, abychom sledovali Facebook a brouzdali po internetu. Standardem systémů na bázi iOS (Apple iPhone), stejně jako Microsoft Phone nebo Android, je alespoň *synchronizace kontaktů* (telefonních čísel, ale i e-mailových adres a dalších údajů o osobách a skupinách) s online databází. To je základ, který sám o sobě představuje značný pokrok. Sám jsem



Obrázek 1: Evernote z webu

dlouho nemohl najít optimální propojení a sdílení poznámek. Protože jsem „závislý“ na službách Google – denně používám Gmail, Google Calendar, Google Sites, Google Documents a často i Picasa – byla hledaným požadavkem (byť ne úplně nezbytným) i integrace s těmito službami.

Na vedení rozsáhlejších projektů vyhovovala služba Google Wave, vloni zmiňovaná na stránkách Zpravodaje. Bohužel Google krátce poté její intezivní vývoj zastavil a službu nadále již pouze udržuje. Ukázalo se, že implementačně velmi složitý, technologicky jeden z nejnáročnějších kolaborativních projektů Googlu, by si býval musel najít daleko hlasitější uživatelskou odezvu, aby byl udržitelný. Není to jednoduchá jednouúčelová služba, navíc by byla prakticky nepoužitelná z mobilního zařízení. Podobný smutný konec ale potkal i službu řádově prostší, která na poznámky – včetně těch mobilních – dostávala: Google Notes, čili Poznámky Google. Tu lze sice také nadále používat, ale rovněž není vyvíjena a pravděpodobně bude brzy stažena.

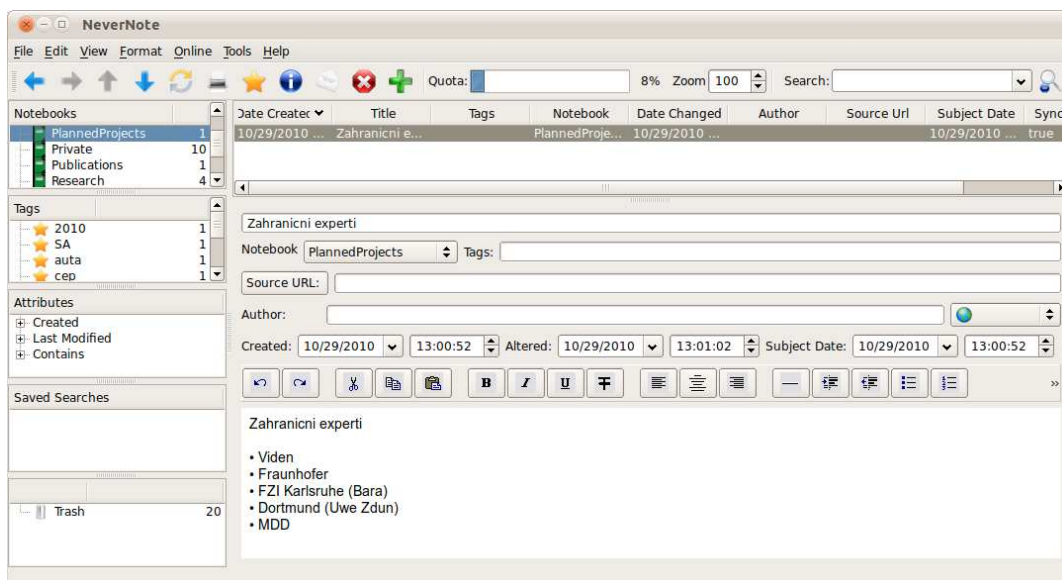
4 Co tedy opravdu chceme?

Bylo nutné z požadavku „všechno z jedné stáje“ ustoupit a najít systém, který bude: (A) použitelný na pracovišti při běžné práci na PC při-

pojeném v síti, (B) informace (poznámky) budou vidět v případě potřeby i z PC jiného, tzn. budou online. To ale není všechno: často máme pracovní notebook s sebou, ale bez připojení 24/7, např. při cestování v zahraničí. Systém by měl umět (C) spravovat poznámky lokálně a po připojení k síti je synchronizovat. Konečně za (D) poznámky by měly být nejen čitelné, ale též vkladatelné z chytrého telefonu na platformě Android, opět s funkcí synchronizace.

4.1 Na webu: Evernote

Těmto náročným kritériím vyhovuje systém Evernote (www.evernote.com), viz obr. 1. Jedná se původně o online projekt, webovou aplikaci, která umí komfortně online spravovat poznámky, je volně dostupná po registraci a do jistého objemu dat funguje zdarma. Čerpání kvóty se navíc po měsíci nuluje, takže nejsme-li přespříliš aktivní (kvóta je 60 MB vložených dat měsíčně), máme službu zdarma stále. Požadavek (B) na online viditelnost je splněn. Přístupový kanál je standardně zabezpečen (SSL); data sice nejsou ukládána šifrovaně, ale na utajované informace to nepoužívám.



Obrázek 2: Desktopový klient pro Linux: Nevernote

4.2 Z počítače: Nevernote

Nyní požadavek (A), plnohodnotná dostupnost z desktopu. Systém nabízí klientský software Evernote pro Windows, Mac; ale i pro Linuxy je zde volně dostupný software – Nevernote, viz obr. 2. Jedná se o javovou aplikaci, která ale disponuje slušným stupněm integrace s prostředím operačního systému. Do vkládaných poznámek je možné přetažením ze správce souborů např. vložit obrázek, hudební/hlasový soubor, prostě cokoli. Není ale možné na jedno kliknutí kamerou počítače nebo scannerem nasnímat obrázek – je nutné ho nejdříve vně pořídit a uložit a pak přetáhnout do Nevernote. Aplikace pro Windows i Mac jsou v tomto ohledu integrované ještě lépe. V každém případě ale textové poznámky nemusejí být jen holý text, editor disponuje slušnými možnostmi formátování (řezy, barvy a velikosti písma, seznamy, zarovnávání).

4.3 Z mobilu: Mobilní Evernote

Skutečnou lahůdkou a příjemným překvapením je jednoduchost a komfort mobilních klientů Evernote jak pro starší i nová Windows Mobile/Windows Phone, tak pro Google Android, tj. novější chytré telefony značek HTC, Samsung, LG a do budoucna věrme i dalších. Teprve na chytrém telefonu se projeví plné výhody systému na poznámky. Na běžném PC máme totiž řadu

alternativních možností, kam poznámky psát – třeba do souboru Google Documents nebo je třeba synchronizovat starým dobrým FTP na běžném síťovém serveru. Na mobilních zařízeních ale chceme, abychom pohotově na místě vyfotografovali, co potřebujeme, rychle nahráli hlasový záznam s úkoly, udělali si na cestách poznámku, prostě chceme daleko pružněji řešit náročnější úlohy. To vše mobilní Evernote s vysokým komfortem umí.

Ani pomalejší (ale stále) připojení k internetu nevádí. Pouze synchronizace déle trvá – třeba i minuty. Takto si běžně hlasově zaznamenávám denní úkoly, fotografuji popsanou tabuli (než se smaže), ofotím pár stránek z knížky, která se musí vrátit, atd. Následný management poznámek zabere nulový čas.

4.4 Manipulace přímo přes e-mail a sdílení

Jak vkládání tak následné posílání ostatním se může dít i prostřednictvím běžného e-mailu. Jako registrovaní uživatelé obdržíme unikátní e-mailovou adresu (privátní, určenou jen nám, něco jako `pitner.x1234@m.evernote.com`), kam když odkudkoli pošleme e-mail, zařadí se mezi naše poznámky. Naopak stávající poznámky můžeme e-mailem komukoli poslat. *Sdílení* funguje tak, že celé poznámkové bloky lze zveřejnit pod „hezkým“ URL celému světu nebo

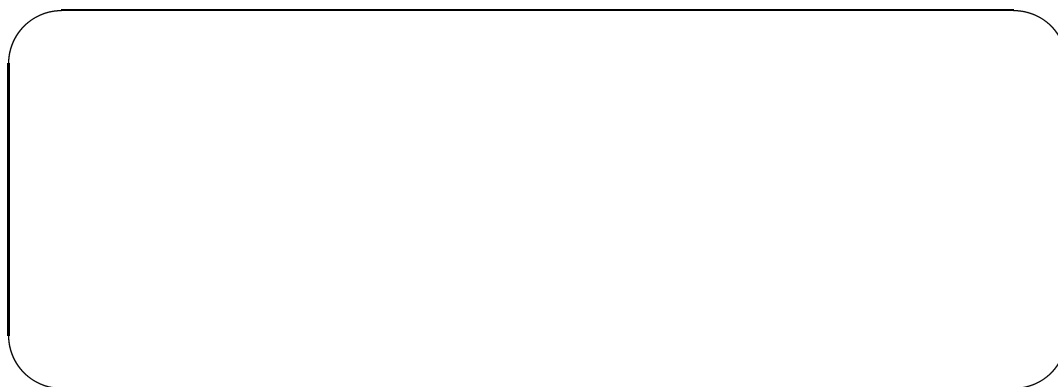
jen jednotlivým lidem, kteří buď musejí nebo nemusejí být přihlášení na Evernote. V placené verzi lze sdílet i pro zápis, tzn. ostatní mohou poznámky v našich blocích vytvářet a upravovat.

5 Organizace poznámek

Každá poznámka, ať už *textová*, *zvuková* nebo *grafická* (fotka, obrázek), je vždy opatřena základními metadaty: název (titulek), popis (krátký text) a pak obsahuje vlastní data. Pamatuje se samozřejmě i na datum vytvoření, poslední modifikace a datum vztahující se k předmětu poznámky – např. k události, která je poznámkou (třeba i s významným zpožděním) zachycena. Základní organizace poznámek spočívá v jejich (opět i dodatečnému) vřazení do právě jednoho z *Poznámkových bloků* (Notebooks). Kromě toho, jak bývá na Webu 2.0 zvykem, můžeme poznámky značkovat pomocí několika *štítků* („tagů“). Toto vše má za úkol především usnadnit rychlé vyhledávání. Vyhledávací dotazy lze uložit ke znovupoužití, takže běžné dotazy za čas bez námahy zopakujeme.

Obsah

Metodika agilního vývoje softwaru na OVSS ÚVT, Vendula Švendová, ÚVT MU	1
Trendy v bezpečnosti počítačových sítí, Václav Lorenc, Honeywell, spol. s r.o.	6
Požadavkové systémy v Inetu, Zdeněk Machač, ÚVT MU	9
Přechodové mechanismy k IPv6 (2), David Rohleder, ÚVT MU	13
(N)evernote: poznámky vždy po ruce, Tomáš Pitner, FI MU	18



K organizaci alespoň trochu cenných dat patří zálohování. První „jistotou“ je už fakt, že poznámky máme na třech místech: online na serverech Evernote, na vlastním počítači v desktopovém klientu (např. Nevertime pod Linuxem) a konečně v mobilní verzi, používáme-li ji. Kromě toho lze data jednoduše exportovat (a tak jako soubor zazálohovat) z desktopového klienta do souborů .nncx. Nechybí ani možnost tisku jednotlivých poznámek.

6 Závěrem

Zkrátka řečeno, na online systém poznámek Evernote a vše okolo (mobilního a desktopového klienta) si zvykneme velmi rychle, není v tom žádná věda a výsledek jednoznačně stojí za pár minut registrace. Velmi podstatná pro budoucnost je také existence celé sféry navazujících služeb přinášejících další přidanou hodnotu, např. rozpoznání a převod namluvené poznámky do textu kvůli vyhledávání, speciální aplikace existující pro „čisté“ snímání tabule, skenování a rozpoznání písma nafocením pomocí iPhone a mnoho dalších. □