

Hlavní navigace

Zdroják » Glosy » Jak budeme psát webové aplikace za tři roky?

Jak budeme psát webové aplikace za tři roky?

Dnes 0:00 Martin Malý



Pokud se podíváme na tvorbu webových aplikací z nadhledu, zjistíme, že možná mnoho věcí, které pokládáme za základ webařiny, během několika let ztratí svou fundamentální pozici. POZOR: Ke čtení tohoto textu potřebujete schopnost chápat nadsázku a dívat se na neochvějně pravdy vlastní práce z nadhledu.

Všiml jsem si zajímavého jevu. Jestli si vzpomínáte, tak před pár lety dávali tvůrci webů na svá díla takové malé ikonky, které hlásaly: Validní HTML, Validní CSS, Validní RSS... Dnes pomalu mizí. Dilem proto, že validní kód začal být samozřejmostí a už ho nelze prodat jako „konkurenční výhodu“, ale hlavně proto, že žádného čtenáře taková informace nezajímá, protože je pro něj zcela zbytečná. (Pardon, zapomněl jsem na některé čtenáře Zdrojáku a Roota, pro které je validita stránek a to, že nejsou servírovány MS IIS, podstatná informace. S ohledem na ně tedy opravuji: *Nezajímá to žádného soudného čtenáře!*)

U obsahových webů, ale hlavně u webových aplikací, začalo být důležité, CO to dělá, víc než JAK to je uděláno. Pokud je webová aplikace funkční ve většině prohlížečů, dobře ovladatelná a dělá to, co od ní uživatel čeká způsobem, jakým to uživatel čeká, tak je jedno, jestli je ve Flashi nebo v HTML5 [viz: <http://labs.codecomputerlove.com/flashvshhtml5/>]. (Odkaz je na hru PONG, kde je půl hracího pole vytvořeno ve Flashi a půl v HTML5.) Webové aplikace přestávají být „klenotem webařiny“ – tedy: konečně! – a stávají se aplikacemi, se všemi vedlejšími efekty, které z toho vyplývají. Například se mění přístup autorů k tvorbě a přibližuje se způsobům známým z tvorby spotřebního komerčního SW či jiného zboží, kde mezi důležitými zásadami stojí na prvním místě zásada *good enough*.

Good enough

Zásada *good enough* vyjadřuje jednoduchou myšlenku: **Produkt je přijatelný ve chvíli, kdy je dostatečně dobrý. Jakékoli zbytné vylepšování nad hranice „dostatečně dobrého“ je neekonomické.** Co je „dostatečně dobré“ záleží na trhu, pro který je produkt určen. Když budete vyrábět čočky, bude hranice *good enough* jiná pro trh výrobců průmyslových mikroskopů, pro high-end optiku do kamer, a jiná bude i pro trh, na němž nakupují výrobci školních souprav pro výuku optiky na středních školách. Ne že by čočky pro školní soupravy mohly být vysloveně špatné – jen nemusí být tak přesné a kvalitní jako pro profesionální teleobjektivy. Zásada *good enough* tedy neznamená „dělejte špatné věci“ (jak si mohou mnozí myslet), ale **dělejte věci (jen) dostatečně dobré pro zamýšlené použití**. Co je dostatečně dobré, to není třeba v tuto chvíli vylepšovat.

Co znamená zásada *good enough* pro psaní webových aplikací a software vůbec? Software je tak dobrý, jak musí být, aby byl prodejný, ale ne lepší (názor lidí, co se řídí heslem „*jenom to nejlepší je pro mne dost dobrý*“ je tu opominutelný). Lepší může být další verze, až se ukáže, co zákazníci opravdu potřebují (software má tu výhodu, že jeho vývoj lze iterovat po malých krocích; viz agilní programování). Software musí fungovat na „dostatečně dobrém“ vybavení. No a konečně lze tutéž zásadu aplikovat i tak, že **pro vývoj software jsou použity „dostatečně dobré“ nástroje**.

Dostatečně dobrý nástroj

Dostatečně dobrý nástroj je takový, v němž jsou náklady na vývoj (tj. čas a zkušenosti vývojáře) minimální pro dosažení „dostatečně dobrého“ výsledku. Nepřekvapí, že řada je sestupná – od nejnovějších vizuálních nástrojů a jazyků velmi vysoké úrovně abstrakce přes běžné vysokoúrovňové jazyky, přes nízkoúrovňové jazyky a přes „lepší makra“ až k assembleru. Čím nižší úroveň abstrakce, čím hlouběji se jde, tím dražší je vývoj – a tím jemnějších výsledků lze dosáhnout. Nástroje by šlo přirovnat k výtvarnickým: od hrubého dláta a rašple přes pilníky až k nejjemnějšímu smirkovým papírům. Nikdo si nebude brát na dvoumetrovou sochu, která bude stát kdesi v parku, smirkový papír, ani nepůjde tvořit náušnici dlátem. Vezme si „dostatečně dobrý nástroj“.

Vývojáři také používají (alespoň v ideálních podmínkách) většinou „dostatečně dobrý“ nástroj – tedy takový, v němž

dosáhnou „dostatečně dobrého“ výsledku s nejmenšími náklady. Jednoduchý webový CMS napíší v Django nebo RoR a nebudou se patlat s assemblerem; naopak modul do jádra systému napíší v C (nebo právě v assembleru). Použijí takový nástroj, který jim umožní vytvořit to, co vytvořit potřebují, s nejmenšími možnými náklady (do kterých se započítávají i faktory jako „náklady na učení“). Čeština pro to má úsloví „brát kanón na vrabce“, angličtina používá slovo „overkill“.

Vývoj přináší stále nové a nové nástroje, které se (většinou) řadí na začátek řady „dostatečně dobrých“, a postupně ubývá věcí, pro které je nutné sahat k velmi jemným (a složitým, pracným, ...) nástrojům. Dnes už nikdo nesáhá do asm kódu, který vypadne z překladače C, proto, aby ručně optimalizoval některé konstrukce; překladače optimalizují velmi dobře a je jen velmi, velmi málo situací, kdy se vyplatí ruční optimalizace (a jen velmi málo lidí, kteří ji opravdu umí líp než překladač). **Výkon dnešních procesorů a paměť počítačů posunuje hranici „good enough“ pro většinu aplikací daleko za cokoli, co bychom mohli „dooptimalizovat“ ručně v assembleru.**

Není nejmenší důvod pochybovat o tom, že vývoj webových aplikací půjde stejnou cestou. Lze se tedy s celkem slušnou mírou pravděpodobnosti opřít do jednoho z axiomů tvorby webů a říct:

Psaní v HTML, CSS a JavaScriptu bude zítra overkill

Nebylo to na vás moc radikální, doufám?

HTML, CSS i JavaScript jsou technologie srovnatelné s makroassemblerem. Jsou na té nejnižší úrovni nad virtuálním strojem (=prohlížečem). Je velmi pravděpodobné, že se prohlížeče jako aplikační platforma nakonec prosadí, ať si o tom už myslíme cokoli; navzdory tomu, že by bylo mnohem lepší začít na zelené louce s *opravdovou* aplikační platformou. Bohužel, než se jakákoli aplikační platforma prosadí, budou prohlížeče o generaci dál a vývojáři budou tvořit aplikace právě pro ně, protože se jim nevyplatí čekat na novou platformu (a ještě riskovat, že vsadí na tu špatnou). **Prohlížeče jsou dnes typická good enough platforma.**

Stejně jako nikdo přičetný nepíše přímo bytekód pro JVM a místo toho používá Javu nebo Scalu, tak nemá příliš smyslu do budoucna psát aplikace přímo ve *webovém bytekódu*, tedy v JS/HTML/CSS. Na dveře už dnes ťukají první nástroje, které jsou „o generaci dál a o stupeň výš“.

Máme tu GWT [viz: <http://code.google.com/intl/cs/webtoolkit/>] – nástroj od Google, který umožňuje napsat webovou aplikaci v Javě a přeložit ji pro „platformu web“ – tedy do HTML, CSS a JS. Je tu OpenLaszlo [viz: <http://www.openlaszlo.org/>], v němž lze popsat aplikaci pomocí XML a skriptů a přeložit ji do swf (FLASH) nebo do HTML/CSS/JS. Jsou tu i další nástroje, například Cappuccino [viz: <http://zdrojak.root.cz/n/cappuccino/>] nebo haXe [viz: <http://haxe.org/>], které překládají z nějakého „jazyka s vyšší úrovní abstrakce“ do JavaScriptu. Je tu CoffeeScript [viz: <http://jashkenas.github.com/coffee-script/>]. A všechny tyto nástroje začínají být *dostatečně dobré* – JS enginy jsou už velmi rychlé a renderovací jádra prohlížečů také, takže kód, který vypadne z těchto nástrojů, je *good enough* i bez toho, aby se někdo ručně hrabal v HTML/CSS/JS a upravoval ho. Ostatně i VisualStudio produkuje poměrně použitelný kód – resp. webové aplikace napsané v ASP.NET nemá nikdo potřebu „optimalizovat“ na úrovni HTML/JS, protože jsou *dostatečně dobré*.

I pro tvorbu „obyčejných“ obsahových stránek jsou už „dostatečně dobré“ nástroje. Zapomeňte na tragický FrontPage nebo GoLive; nové nástroje, vytvářené a laděné zástupem webdesignérů, vytváří *dostatečně* kvalitní kód, validní, bez podivností, navíc dokážou optimalizovat zápis a ušetřit i práci s psaním hacků či vendor-specific vlastností.

Prasečina, klikači, pro lamy...

Může se vám to zdát jako úpadek řemesla, ale to patří k vývoji. Co dnešní generace považuje za úpadek, to bude pro další přijatelný nástroj a pro další pak *klasika*. První opravdoví programátoři, zvyklí zadávat kód pomocí přepínačů a tlačítek, mohli považovat terminály a FORTRAN za úpadek řemesla. FORTRANovští programátoři považovali za úpadek Pascal, C i Smalltalk. Desktopoví programátoři považují za úpadek webové jazyky. Atakdál. Časem se „úpadková“ technologie stane mainstreamem, hlavně proto, že je *dostatečně dobrá*, tudíž ekonomicky výhodná.

Můj soukromý tip je, že do tří let budou technologie postavené nad HTML, CSS a JS *dostatečně dobré* k tomu, aby získaly významný podíl na trhu. Podobně jako se v posledních letech staly třeba PHP frameworky *dostatečně dobrými*. V roce 2003 každý rubal PHP tak, jak se to naučil: přímo do HTML, knihovny byly „fuj“ a ti největší guruové hlásali, že „*knihovny zdržují a můj čas je příliš drahý na to, abych ošetřoval cizí chyby*“. Dnes jsou frameworky v PHP *good enough*. Totéž platí třeba pro šablonovací nástroje: Před pěti, sedmi lety „zbytečnost, zpomalení, komplikovanost“, dnes už běžně

přijatelná věc, která šetří práci. U jiných jazyků (Ruby, Python, ...) platí totéž.

Stejně tak odejde třeba i *dnešní PHP*. Neříkám že zmizí, na to je příliš rozšířené a příliš používané. Ale frameworky postupně vytlačí psaní „surového PHP“; další logický krok budou *generátory kódu*, které z nějakého jazyka s vyšší úrovní abstrakce vytvoří *dostatečně dobrý* PHP kód nad nějakým frameworkem (a přestane být důležité nad jakým: prostě nad *dostatečně dobrým*). Vývoj se zrychlí a zlevní; psát PHP kód bude stejná *podivnost* jako dnes psát assembler. (*Povinný disclaimer: Autor textu píše dodneška některé aplikace v assembleru a nepovažuje jej za nadávku; přesto si uvědomuje, že dnes je jeho použití omezené na určité minoritní oblasti.*)

V nadsázce řečeno spolu s jednou postavou známého filmu: „*Dávám psaní v HTML rok, nejvýš dva!*“

Kodérina: vymírající řemeslo

Nechci tvrdit, že znalost HTML a CSS bude do tří let zbytečná, to v žádném případě. Jen nebude nezbytně nutné znát je do hloubky, protože *nebudou denním chlebem*. Převod návrhu od designéra do HTML/CSS postupně nebude při vývoji webových aplikací tak důležitá záležitost jako dnes. Nástroje, které budou dejme tomu v roce 2013 (a berte toto datum *cum grano salis*) k dispozici, budou pravděpodobně schopné vytvořit *dostatečně dobrý* kód z nějakého zápisu chování v jazyce velmi vysoké úrovně a z vizuálního návrhu UI (*a to tvrdím s vědomím rizika, že za tři roky tento článek někdo najde a bude se mi posmívat*). Je docela dobře možné, že třeba Flash Builder (dříve *Flex Builder*) nebo Visual Studio dostane přepínač „*Přeložit do HTML/JS/CSS*“ – a nikoho nebude trápit, že výsledný HTML kód nebude tak krásný, jako kdyby ho psal *opravdu dobrý kodér*. Bude „dostatečně dobrý“, ale bude vytvořen a odladěn mnohem rychleji a laciněji než od *opravdu dobrého kodéra*.

Mimochodem: nespolehejte na to, že si *dnes* připadáte jako *vysoce kvalifikovaní odborníci* – trh, a tím i vývoj, jde vždy nakonec cestou minimálních nákladů, a věřit, že *ocení kvalitu*, znamená často zavřít krám. *Na trhu je jen omezený prostor pro drahou ruční práci – proč myslíte, že to bude zrovna ta vaše?*

V oblasti webových aplikací bude pravděpodobně posun k podobným nástrojům, ke „generátorům kódu“, markantnější než u obsahových webů, kde se *kodéřské řemeslo* bude držet delší dobu; obsahové weby bývají unikátnější a podíl ruční práce bývá vyšší. Už dnes ale vím o webdesignérech, kteří jsou perfekcionalisté a dokáží si hrát s detaily – a používají (třeba) Dreamweaver, protože jim umožní soustředit se na dílo, ne na implementační detaily. Pracují rychleji, produktivita jim roste, protože se nezabývají opakovanými prázdnými činnostmi, ale přitom mají stále možnost zasáhnout do detailů. Zbytek kódu, o který se za ně postará nástroj, je i pro ně „dost dobrý“ – jeden z nich mi řekl: „***Napsal bych to asi někde líp, ale trvalo by mi to třikrát tak dlouho.***“

A přesně to vyjadřuje princip *good enough*, a taková bude podle všeho i budoucnost nástrojů pro vývoj webových aplikací.

Jaký je váš názor na budoucnost vývoje webových aplikací? Myslíte si, že se bude dál psát tak, jak píšeme dnes, jen s novými verzemi jazyků, nebo že porostou schopnosti a vliv „generátorů kódu“ natolik, že překročí hranici *good enough* a stanou se primárním nástrojem pro vývoj? Nebo si myslíte, že webové aplikace v prohlížečích budou nahrazeny jinou platformou? Podělte se o svůj **názor na toto téma!**

Martin Malý



Začal programovat v roce 1984 s programovatelnou kalkulačkou. Pokračoval k BASICu, assembleru Z80, Forthu, Pascalu, Céčku, dalším assemblerům, před časem v PHP a teď je rád, že neprogramuje...

zdroják.cz



Je IE 9 opravdu nejlepší prohlížeč?

Vitalia.cz



Učitel je někdy pro studentky alfa-samec

zdroják.cz



Bude z vašeho dítěte programátor?

DigiZone.cz



Jak vznikala reklama s Chuckem Norrisem?

měšec.cz



Akcie přehledně graficky jako na Wall Street!

Zdroják (zdrojak.root.cz), tvorba webových stránek a aplikací. ISSN 1803-5620

Copyright © 2008 – 2010 Internet Info, s.r.o. Všechna práva vyhrazena. Powered by Linux.

tady hlídá
SiUX