

Steven Black
steveb@stevenblack.com
<http://stevenblack.com>
<http://fox.wikis.com>

Steven Black se specializuje na vývoj vícejazyčných aplikací, programů pro souběžný provoz na více místech a na další náročné softwarové úkoly, včetně přepracování a vyčištění projektů. Vytvořil INTL Toolkit, vícejazyčný framework pro FoxPro a Visual FoxPro. Pravidelně přednáší na konferencích o Visual FoxPro a jeho příspěvky občas zaplňují stránky knih a časopisů. Je zakladatelem a správcem webu FoxPro Wiki na adrese <http://fox.wikis.com>.

Květen 2007

Úvod

Tato přednáška pojednává o budování *prodejních a technických odůvodnění* Visual FoxPro (VFP) jakožto vývojové platformy pro nový projekt nebo dalšího použití VFP v existující infrastruktuře a provozních činnostech IT.

Vytváření správných prodejních odůvodnění je umění i věda. Je předmětem řádného i postgraduálního studia na významných vysokých školách. Proto je tato přednáška (i tento dokument) velmi důležitá. Jejím cílem je nastínit témata, o nichž byste měli přemýšlet, když máte předložit přesvědčivé argumenty, proč pro budoucí vývoj použít VFP.

Část I – Prodejní odůvodnění obecně

V této sekci stručně probereme základní prvky prodejních odůvodnění.

Berte prosím na vědomí, že prodejní odůvodnění by *vždy* měla být budována s náležitým uvážením všech faktorů specifických pro danou situaci, které mohou významně ovlivnit typ, vyznění a obsah (smysl) prodejního odůvodnění. Každé podnikání je do jisté míry odlišné a totéž platí i pro každý projekt. Proto jsou následující řádky jen obecným vodítkem.

Co to je "prodejní odůvodnění"?

Prodejní odůvodnění je strukturovaná prezentace vyrobená pro management na podporu určitého postupu nebo způsobu jednání. Obvykle je nutné, když je navrhována nějaká významnější změna směru. Může to být dokument nebo sbírka dokumentů, a většinou zahrnuje nějaký druh prezentace, i když je představováno neformálně na jedné nebo několika schůzkách.

Prodejní odůvodnění sestává z identifikace a analýzy (kvalitativní i kvantitativní) následujících záležitostí:

potřeby podniku s souladu s cíli podnikání,

identifikace příležitostí a hrozeb,

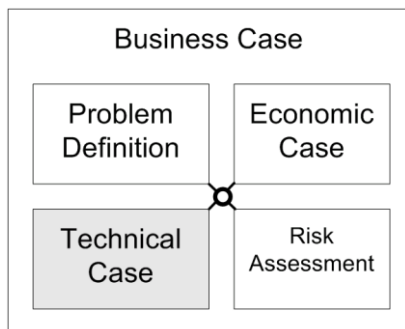
možnosti a dosažitelné výsledky,

náklady, přínosy a rizika,

odhady investic, času potřebného na vývoj a termíny dodávek,

výpočty návratnosti investic (ROI).

Z pohledu informačních technologií (IT) by mělo dobré prodejní odůvodnění přesně zapadat do celkové koncepce informační technologií organizace. Technické odůvodnění je ve skutečnosti podмноžina širšího pojetí prodejního odůvodnění.



Obrázek 1 Tento dokument je zasazen do širšího kontextu prodejního odůvodnění a jeho dalších vzájemně provázaných prvků.

Dobré prodejní odůvodnění převádí technickou problematiku do podnikatelských pojmů, aby ji vedoucí pracovníci lépe pochopili, a identifikuje důsledky chybného podnikatelského rozhodnutí. Představuje důležité skutečnosti, které je nutno brát v úvahu, když má vedení učinit kvalifikované rozhodnutí. Jednoduše řečeno, prodejní odůvodnění je na faktech založené odůvodnění změny a je podepřeno technickým odůvodněním.

Prvky správného prodejního odůvodnění

Správné prodejní odůvodnění usiluje o dosažení všech těchto cílů:

Má jasně a přesvědčivě prezentovat fakta, analýzu a doporučení.

Má patřičně formulovat podnikatelský problém, jehož řešení navrhuje.

Má rozpoznat a ukázat, co od změny očekávají vlastníci.

Má projevít „náležitou příčinnost“, prozkoumat různé alternativy a představit jejich "pro" a "proti".

Má jasně ukázat, proč je doporučovaná volba nejlepší.

Obecné tipy pro vytvoření přesvědčivého prodejního odůvodnění

TIP: Ujistěte se, že **znáte strategii a priority dané organizace**. Řada organizací tyto informace zveřejňuje, například ve výroční zprávě. Někdy jsou uvedeny v propozicích projektu, a občas musíte být vynalézaví a odněkud je vydolovat. Buďte však obezřetní! To, že jste třeba našli větu o strategii a prioritách, ještě neznamená, že máte přestat hledat „skryté agendy“ a „nevyřčené cíle“ v oblastech, kam směřuje vaše prodejní odůvodnění.

TIP: Pokud možno byste měli **brzy dosáhnout závazku**. Závazek neznamená schválení; znamená to, že vedoucí IT a obchodníci souhlasí s tím, že váš námět dává smysl a stojí za to ho rozvinout do návrhu k posouzení. Není nic horšího než spřádat grandiózní plány, které po odhalení každého ohromí. Naopak, nejlepší taktický závěr je ten, že prezentace plánu je spíše povinná formalita, jakési „orazítkování“. To je vždycky lepší než „odhalení“.

TIP: Držte se písemných pokynů, pokud existují. Ty vám řeknou, co od vás vedoucí pracovníci chtějí slyšet a v jaké podobě. Využijte pokynů při sestavování a utřídění svého návrhu; pomohou vám zajistit, že v něm bude všechno, a ušetří vám spoustu rozhodování, o čem psát.

TIP: Poctivě vyčíslete všechny náklady. Důvěryhodnost prodejního odůvodnění spočívá do značné míry v jeho odhadu výhod i nevýhod. Mezi zdroje nákladů patří:

- * shromáždění požadavků
- * řízení projektu
- * testování
- * správa systému, zahrnující vývoj, testování a nasazení systémů
- * pořízení vstupního hardwaru a softwaru pro vývoj, testování a nasazení systémů
- * podpora výroby
- * objemy dat nebo transakcí, které si vynutí vyšší investice do hardwarového vybavení
- * počáteční náklady na implementaci (hardware a software, náklady na instalaci, náklady na vývoj a testování, náklady na zkušební provoz, náklady na nasazení)
- * trvalé provozní náklady (personální náklady na obsluhu i technickou podporu, náklady na údržbu systému, na údržbu hardwaru a softwaru, komunikace, softwarové licence, ekologické náklady)
- * náklady na rozvoj a růst (hardware, přepracování a vylepšení softwaru).

Část II – hlediska VFP v prodejních odůvodněních

V této sekci probereme některé přístupy k prodejním odůvodněním spjaté s VFP. Rozhodl jsem se je rozdělit do čtyř „kategorií“ podle svévolně přiřazeného stupně důležitosti a obtížnosti. Berte to tak, že každá situace je odlišná, a proto jsou tyto kategorie nastaveny jen pro účely tohoto pojednání a můžete si je upravit podle své potřeby.

Zde jsou čtyři kategorie, definující strukturu zbytku tohoto dokumentu.

Základní body, v nichž musíte vyhrát. Tyto prvky jsou nezbytné už jen kvůli naději na úspěch. Vynecháte-li kterýkoliv z nich, nejspíš nebudete mít žádnou šanci.

Body, které pravděpodobně vyhraje, vzhledem k relativním přednostem VFP a, předpokládejme, vašim nabytým programátorským zkušenostem a schopnostem.

Najdou se i **těžké bitvy, které se však dají vyhrát**, známe-li předem obavy těch, kdo ve firmách rozhodují.

A pak jsou samozřejmě i **bitvy, které nejspíš prohrajete**, tak s tím počítejte a hlavně mějte připraveny solidní alternativy.

Základní body, v nichž musíte vyhrát

Toto jsou body, které *musíte* získat, má-li být vaše prodejní odůvodnění úspěšné.

Vaše „historie“ a vaše kvality partnera – vývojáře

Když vytváříte prodejní odůvodnění, které zahrnuje využití určité konkrétní technologie pro vývoj, pak to, co *opravdu* nabízáte, bez ohledu na projekt, je blízký obchodní i vývojářský vztah mezi vámi a vašim zákazníkem. V tomto kontextu to, jak jste v minulosti plnili své sliby, znamená hodně. „Historie“ je přehled prokazatelných výsledků; je to seznam vynikajících minulých výkonů.

Vedle toho je důležité si uvědomit, že výběr určité technologie je jen zřídka čistě technickým rozhodnutím. Když něco zahrnuje vysoký stupeň osobní služby, a to pro vývoj softwaru platí vždy, soulad mezi jednotlivci hraje důležitou a někdy i určující roli v procesu rozhodování.

„Technické odůvodnění“ VFP

Váš návrh, proč použít pro vývoj právě VFP, musí samozřejmě splnit všechny nezbytné technické předpoklady. Když se budete zabývat technickými stránkami prodejního odůvodnění, měli byste vzít v úvahu následující okolnosti.

Vhodnost k danému úkolu

Musíte ukázat, že VFP dokáže splnit předvídatelné požadavky navrhovaného systému s ohledem na funkčnost i okolnosti s funkčností nesouvisející. Patří sem například:

Doba odezvy. Jak dobře splní aplikace ve VFP očekávání koncových uživatelů týkající se odezvy?

Průchodnost. Musíte ukázat, že aplikace ve VFP zvládne očekávané zatížení transakcemi a výběry dat.

Prostředí. Mám na mysli operační systémy (OS) a hardware. Pochopitelně budete muset do značné míry vyhovět všem omezením, která v dané organizaci platí pro verze OS a pro hardware. Tato oblast je pro zastánce VFP obvykle velmi příznivá, protože VFP běží na mnoha starších platformách Windows. Viz <http://fox.wikis.com/wc.dll?Wiki~VFPSupportedOS>.

Přizpůsobivost a udržitelnost. Významný podíl z celkových nákladů na životní cyklus všech úspěšných systémů obvykle ukousnou činnosti, které následují po vývoji, jako údržba, adaptace systému a jeho rozšiřování. Musíte ukázat, že navrhovaný systém bude mít takové provozní vlastnosti, že bude fungovat ještě dlouho po počátečním nasazení.

Spolehlivost a odolnost vůči chybám. Musíte ukázat, že navrhovaný systém s vysokou pravděpodobností splní očekávání týkající se provozuschopnosti, robustnosti a předvídatelných událostí, jako jsou výpadky proudu, havárie systému, běžné chyby uživatelů a podobně.

Bezpečnost. Možná budete muset zaručit, že systém vyhoví všem požadavkům na bezpečnost.

Škálovatelnost. Existují různé aspekty škálovatelnosti a důraz na některé z nich lze do jisté míry očekávat u všech úspěšných systémů.

Když se mluví o „škálovatelnosti“ obecně, obvykle se tím rozumí tyto *klasické aspekty škálovatelnosti*.

Objem dat. Někdy se používá termín „size up“ – schopnost přizpůsobit se narůstajícímu objemu dat.

Rychlost. Někdy se používá termín „speed up“ – schopnost navýšit rychlost odezvy.

Pracovní zatížení. Někdy se používá termín „scale up“ – schopnost zvládnout nárůst zátěže.

Cena transakce, která je v softwarových systémech skoro vždy těsně spjata s *dobou trvání transakce*. Proč? Protože délka transakce určuje horní hranici počtu transakcí, které určitý server nebo skupina serverů dokáže zpracovat, aniž by bylo nutné přidávat nové servery, jejichž celkový počet může být omezen v závislosti na infrastruktuře sítě.

Méně běžné aspekty škálovatelnosti jsou:

Hardwarové platformy. Zákazník může očekávat, že aplikace bude fungovat na obřích serverových farmách nebo naopak na příručních zařízeních.

Sdílení důvěry. Systémy mají tu a tam těsně spolupracovat s jinými systémy a sdílení důvěry se vztahuje k obsluhování uživatelů, kteří už byli ověřeni jinými systémy.

Šířka pásma. Jak se projeví zvýšený provoz po síti v praxi.

Dostupnost služby. Jak se bude měnit dostupnost služby s rostoucími hodnotami ostatních aspektů škálovatelnosti.

Zobrazování informací koncovým uživatelům. Jak se systém vypořádá s novými službami, produkty a kategoriemi, které budou časem dostupné online?

TIP: Kdykoliv přijde téma škálovatelnosti na přetřes, ať už o ní začnete mluvit vy nebo někdo jiný, vždycky zaměřte diskusi na konkrétní typ škálovatelnosti a jednejte výhradně o něm, a to v kontextu navrhovaného systému. *Nesmíte* zabřednout do debaty o škálovatelnosti obecně, protože ta vždycky nastolí více otázek, než kolik jich přesvědčivě zodpoví, a to *nikdy* není dobrá pozice pro prodejní odůvodnění. Proto si uvědomte, že škálovatelnost má specifické aspekty, jejichž hranice a možnosti je nutno probírat jednotlivě.

Další odkazy na témata, která by vám mohla pomoci při sestavování technických stránek prodejního odůvodnění:

<http://fox.wikis.com/wc.dll?Wiki~WhyVFPsTheToolOfYourChoice>

<http://fox.wikis.com/wc.dll?Wiki~VisualFoxProBulletPoints>

„Slučování“ VFP s problematickými aplikacemi

Rick Schummer napsal velmi zajímavý článek o jednom ze svých zákazníků, kde uživatelé o jedné naprogramované aplikaci mluvili jako o „FoxPro“.

Viz http://rickschummer.com/blog/2005_06_01_archive.html#111818196732633978

To je něco, s čím jsem se setkal už mnohokrát: VFP získala špatnou pověst, protože je nějak spojována se stárnoucí, tedy dávno naprogramovanou aplikací.

Když k tomu dojde, čeká vás tvrdý boj, protože dojmy těch lidí se nejspíš utvářely a dozrávaly docela dlouho. Obvykle záleží na tom, aby si klíčoví vedoucí pracovníci uvědomili, že VFP a aplikace napsaná ve VFP jsou dvě různé věci a že nedostatky té aplikace jsou, s nejvyšší pravděpodobností, čistě záležitostí implementace a údržby.

Obviňujete snad z mizerně napsaných nebo zastaralých dokumentů Microsoft Word? Ne. Je za promeškané obchody a nedosažené zisky zodpovědný Microsoft Excel? Ne. Když to formulujete tímto způsobem, lidé obvykle pochopí rozdíl mezi „FoxPro“ a aplikací, kterou provozují, a obvykle přistoupí na tento úhel pohledu. Možnou výjimkou jsou případy, kdy má „FoxPro“ spojitost také s dřívějšími programátory a konzultanty, kteří teď mají v dané organizaci mizernou pověst. To se dá zmírnit podobnou taktikou: oddělte od sebe „FoxPro“ a onen problém. Problémem není skoro nikdy FoxPro, nýbrž to, co bylo ve FoxPro vytvořeno, a také kým, a jak to potom bylo udržováno a dále rozvíjeno.

Dostupnost odborníků

Dostupnost kvalifikovaných foxařů se v různých oblastech výrazně liší. Měli byste být schopni předvést, že existuje dostatek dobrých programátorů, na něž se může tento potenciální zákazník v případě potřeby obrátit. Informace vám poskytnou např. regionální sdružení foxařů. Nic vám nebrání založit si takové sdružení sami nebo oživit existující.

Body, které pravděpodobně vyhrajete

Vzhledem ke stěžejním silným stránkám VFP půjdete po správné cestě, budete-li se držet následujících argumentů:

Vysvětlení pozice VFP vedoucím pracovníkům

Když zdůvodňujete použití VFP, skončí to pokaždé stejně: vedoucí pracovníci nerozumí tomu, jaká je pozice VFP v marketingové směsici produktů Microsoft.

Nabízím pár odkazů na FoxPro Wiki, které se těmito otázkami podrobně zabývají.

<http://fox.wikis.com/wc.dll?Wiki~VisualFoxProBulletPoints>

<http://fox.wikis.com/wc.dll?Wiki~VFPRevenueModelProblems>

Zjistil jsem, že je obvykle možné, s použitím přímočaré, avšak předem naplánované řady argumentů, přivést většinu vedoucích pracovníků k mnohem pozitivnějšímu pohledu na VFP, než jaký měli předtím. Existují důvody, proč Microsoft VFP nevychvaluje, a mají jen málo společného se schopnostmi VFP nebo s její vhodností k vývoji aplikací.

Rizika vývoje softwaru

Jen málo věcí zní zákazníkům lépe než plánování s vědomím rizik a vyhlídky na vývoj a implementaci softwaru s averzí vůči rizikům.

Existují dva obecné typy rizik:

Jak technicky a obchodně argumentovat ve prospěch Visual FoxPro
© 2007 Steven Black

Visual FoxPro DevCon, Praha 2007
FOX20 • 5

Neregulovatelné, známé jako *exogenní (vnější) riziko*.

Regulovatelné, známé jako *endogenní (vnitřní) riziko*.

Analýza rizik slouží k rozpoznání a posouzení faktorů, jež mohou ohrozit úspěch projektu nebo dosažení cíle. Dobrá analýza také pomáhá definovat preventivní opatření snižující pravděpodobnost, že se tyto faktory vyskytnou, a stanovuje, jak se s nimi úspěšně vypořádat, když už nastanou.

Díky své dlouhé životnosti, kvalitě a vyzrálosti komerčních frameworků a všeobecnému povědomí o jejich silných a slabých stránkách má VFP dobrou relativní pozici mezi mnoha dalšími vývojovými nástroji, když dojde na regulaci endogenních rizik vývoje softwaru. V přednášce si ukážeme několik příkladů.

Náklady na licence a další ekonomické aspekty

VFP má neomezený runtime zdarma, což je současně požehnáním i prokletím. Programátorům, kteří potřebují nasadit své aplikace na značný počet pracovních stanic, toto licenční schéma maximálně vyhovuje. S podstatně inovovanými možnostmi tvorby sestav uvedenými ve VFP 9 získává VFP další obrovskou výhodu: téměř neomezený reportovací potenciál s nulovými náklady na distribuci bez ohledu na počet uživatelů.

Nevýhodou je, že Microsoft nemá ze vzrůstající distribuce runtime VFP žádný další zisk, a to je hlavní důvod, proč nemá VFP v marketingu svou váhu.

Zde je odkaz na FoxPro wiki, kde jsou do hloubky probrány otázky týkající se neomezeného runtime zdarma: <http://fox.wikis.com/wc.dll?Wiki~UnlimitedFreeRuntime>

A vedle tohoto způsobu licencování runtime se dá plné vývojové prostředí VFP zakoupit a upgradovat relativně levně.

Aby toho nebylo málo, i ty nejstarší verze VFP poběží docela dobře na počítačích se staršími verzemi Windows. Runtime je v tomto ohledu mnohem benevolentnější než vývojové prostředí VFP, ale zaslechl jsem, že IDE VFP funguje uspokojivě i ve starších operačních systémech, které oficiálně nejsou „podporovány“. V některých důležitých případech poskytuje komunita VFP záplaty a úpravy, které rozchodí VFP i pod staršími verzemi Windows.

Nezapomeňte se tedy podívat, jak vypadá „vozový park“ pracovních stanic a serverů, než začnete ve svém odůvodnění nového vývoje nebo dalšího používání existujících systémů naprogramovaných ve VFP vyčíslovat náklady. VFP nejspíš spokojeně poběží na hardwaru, který už máte, takže vás nečeká série upgradů vynucená Microsoftem, a to vždycky znamená hodně.

Více informací najdete na <http://fox.wikis.com/wc.dll?Wiki~VFPSupportedOS>.

Jisté je jedno: vývoj ve VFP skoro zaručuje, že se prakticky osvobodíte z pout rostoucí nenasytosti Microsoftu. Pro provoz aplikací VFP si pravděpodobně nebudete muset pořizovat nový hardware, ani instalovat spoustu novějších verzí Windows na starší počítače. Na pocitu, že nemusíte pořád něco upgradovat, je cosi neodolatelného a většině vedoucích pracovníků IT to zní jako rajska hudba.

Co vyplývá ze zralosti produktu

VFP se objevila ve verzi 3.0 v roce 1995, ale mnohé prvky jazyka i vývojového prostředí známe už podstatně delší dobu.

Stabilita

S věkem přichází zocelení a VFP rozhodně je ostřílený voják. S každou novou verzí a service packem získává vyšší stabilitu.

Zaměření na data

Některé oblíbené vývojové nástroje nezacházejí s daty zrovna skvěle nebo aspoň slušně. Například marketingové monstrem pro .NET začalo slibovat těsnější datovou integraci v *přes příští verzi*.

Těžko se dají zveličtí výhody plynoucí ze schopnosti pracovat s daty nativně. Možnost pohodlně vybírat data a zpracovávat je ve stejném logickém prostoru patří pro zkušené foxaře k nejpřesvědčivějším přednostem VFP.

Produktivita vývoje

K VFP se dodává široký výběr vynikajících a osvědčených vývojových frameworků a doplňků od jiných výrobců pro rozmanité oblasti použití, včetně zcela specifických balíčků pro účetnictví, online obchody na webu, komunikace, tvorbu sestav atd. U většiny z nich je k dispozici kompletní zdrojový kód, mimořádná podpora v rámci komunity, a často i pravidelná vylepšení a upgrady. Proto je nesmírně důležité zajistit, aby vedoucí pracovníci pochopili, že „prostor VFP“ je plný hospodárných produktů a nástrojů, díky nimž je vývoj rychlejší, robustnější a méně riskantní.

Jiný významný aspekt produktivity vývoje se týká samotných programátorů. Vývojářské zkušenosti se obvykle liší co do šířky i hloubky. Podle různých zdrojů – výzkumu i názorů – se zdá, že rozdíl v produktivitě mezi vynikajícím a průměrným tvůrcem softwaru může být až 20:1. Vzhledem k tomu, že VFP je tady už dlouhou dobu, narazíte dost pravděpodobně na programátory ve VFP, kteří mají letité zkušenosti.

Argumenty týkající se produktivity vývoje očividně mají značnou váhu. Můžete-li je uplatnit, rozhodně je zdůrazněte.

Horizont životního cyklu produktu

Podle toho, co oznámil Microsoft, má VFP verze 9 naplánovanou podporu do roku 2014. Viz <http://support.microsoft.com/gp/lifeselectindex>. V době vzniku tohoto článku je to podstatně pozdější termín než u většiny ostatních vývojových nástrojů Microsoft.

VFP není kriticky závislá na Windows

To je obrovské „plus“. Můžete najisto říci svému zákazníkovi, že řešení založené na VFP jen zřídka vyžaduje zásadní upgrade hardwaru případně Windows v celém systému. Fakt, že VFP nejspíš bude hladce fungovat na dosavadních počítačích a operačních systémech, je velikou výhodou.

Další informace najdete na <http://fox.wikis.com/wc.dll?Wiki~VFPsupportedOS>.

Webové schopnosti

Díky řadě frameworků a toolkitů se FoxPro výborně hodí k vytváření webových aplikací. Vedle toho je Visual FoxPro opravdu dobrý nástroj k tvorbě komponent, které se pak používají ve specializovaných webových vývojových prostředích.

Interoperabilita

Dodnes je VFP vzorem interoperability: dokáže snadno číst, importovat a exportovat širokou paletu datových formátů. A pokud se něco nedá provést přímo (nativně), obvykle se to dá snadno naprogramovat, protože jestli VFP něco dělá jasně a jednoduše, jsou to data a jejich zpracování.

Bonusové body za nadšení a kompetenci

Vzpomeňte si na některá velká rozhodnutí, která jste v životě učinili: kde budete bydlet, na kterou školu půjdete, na kterou školu půjdou vaše děti, jaké auto si koupíte (nebo nekoupíte), kam pojedete na dovolenou, jaký notebook si koupíte a tak podobně. Většinu těchto důležitých rozhodnutí asi spoluutvářela doporučení lidí, jimž důvěřujete, nebo lidí, kteří na dané téma mluvili s nadšením a přesvědčivě.

Dobré doporučení se dá charakterizovat pěti základními body:

1. Osobní zkušenost.
2. Nadšení.
3. Konkrétnost.
4. Upřímnost.
5. Srozumitelnost.

Jak by vás v těchto pěti bodech ohodnotili vaši zákazníci?

Těžké boje, které se však dají vyhrát

Vysvětlení problémů s marketingem VFP

Microsoft nepochybně VFP zanedbává. Asi se vám to nelíbí, ale smířte se s tím, protože se to hned tak nezmění.

K jádru věci: „marketingovou značku“ v myslích cílových zákazníků tvoří tři P – **Produkt** sám, jeho **Povaha** a jeho **Příslib**.

Protože to lidé rozhodující o daném projektu stěží slyšeli od Microsoftu, potřebují to slyšet od vás. Toto jsou prvky značky VFP v pojetí tří P.

Produkt VFP

Vývojové prostředí ve Windows.

Runtime bez licenčních poplatků.

Výjimečně ucelený a bohatý programovací jazyk.

Zcela otevřená metadata přímo ve formátu DBF.

Mimořádná konektivita k různým datovým formátům pro vstup i výstup.

Málo rozdílů mezi vývojovým a produkčním prostředím.

Zpětná kompatibilita až k nejstarším verzím (cca 1985).

Nativní práce s daty na úrovni řádků (vět).

Povaha VFP

Rychlá.

Velmi spolehlivá (v porovnání se všemi ostatními současnými standardy).

Objektově orientovaná.

Vývojové prostředí zaměřené na data. Když pracujete s VFP, jste rovnou v „datové vrstvě“.

Není to žádná nová technologie. Když to pečlivě uvážíte, tohle je dobře! V několika posledních verzích byla VFP zdokonalována a aktualizována, ale nikoliv přepracována.

Prvotřídní produkty od jiných výrobců.

Mimořádně rozšiřitelná.

Příslib VFP

Můžete být vysoce produktivní.

Prostřednictvím webu se vám dostane výborné podpory od dalších foxařů.

Vytvoříte výkonné a stabilní datově orientované aplikace rychleji než pomocí jiných nástrojů.

Microsoft si vás nebude vážit.

Budou vás zesměšňovat a posmívat se vám za to, že používáte VFP.

Novináři si vás nebudou vážit.

Prostě stačí znát meze. Pokud chce zákazník to, co je zrovna v módě, VFP mu to nezajistí. Pokud chce zákazník poměrně rozumné vývojové prostředí se spoustou možností, pak VFP pravděpodobně splní jeho požadavky.

Další informace o značce VFP obecně a o některých jejích problémech najdete zde:

<http://fox.wikis.com/wc.dll?Wiki~VFPBrandMarketingProblem>

<http://fox.wikis.com/wc.dll?Wiki~VFPRevenueModel>

<http://fox.wikis.com/wc.dll?Wiki~VFPRevenueModelProblems>

Bitvy, které nejspíš prohrajete

V určitém okamžiku, zvláště při jednání s rozsáhlou organizací, jejíž centralizované databáze udržuje tým databázových a síťových administrátorů, narazíte na značný tlak, který se vám prostě nepodaří překonat. Toto jsou některé z bodů, které pravděpodobně nikdy nevyhrajete.

Aplikační data uložená v souborech .DBF

Nedivte se, že často, zejména u středních až velkých společností a *zvláště* u společností s centralizovanými datovými středisky, kde databázoví administrátoři udržují aplikace pracující s databázovými servery, je každý návrh ukládat data do souborů .DBF předem ztracený případ.

Je to zcela pochopitelné. V mnoha společnostech se představa otevření části disků na síti pro přímý přístup k souborům s daty *nikdy* neujme. Navrhují: zvykněte si na to.

TIP: Zjistil jsem, že často je dobré ze začátku založit prodejní odůvodnění na tom, že VFP bude pracovat s databázemi jako MySQL, MS-SQL nebo Oracle. Tak se vyhnete problémům plynoucím z dat v souborech DBF ve sdílených síťových adresářích. Později, když se vedoucí pracovníci zarazí nad licenčními nebo logistickými náklady souvisejícími s těmito databázemi SQL, můžete vždycky vyrukovat s námětem ukládat data do souborů DBF.

Malá zařízení

VFP nebude dobrou vývojovou platformou pro programování příručních zařízení, a nepřipadá ani v úvahu, aby se na těchto malých zařízeních provozovala. To ale neznamená, že VFP nemůže být hlavní součástí globálního

systemu (v němž příruční zařízení obvykle tvoří jen malou část). Dokonce i .NET Compact Framework má své „mouchy“ a zatím to vypadá, že v této oblasti nejsou žádné snadné cesty.

Vím o několika impozantních řešeních, kde na malých zařízeních a webových stránkách běží ASP.NET nebo Java, zatímco VFP obstarává podstatnou část zpracování na pozadí jakožto middleware. Hybridní řešení jsou v tomto oboru normou a to, že VFP nelze provozovat na zařízeních, která nemají Windows (nebo aspoň ne „plnohodnotnou verzi“ Windows), ještě neznamená, že se nehodí pro nesčetné další prvky daného systému.

Vyznavači doctrine-du-jour Microsoftu

Občas narazíte na jednotlivce, které Microsoft totálně „obrátil na svou víru“. Typickým poznávacím znamením takových lidí je zjevná a někdy až do krajnosti jdoucí loajalita k produktům Microsoftu, zvláště pak k těm, které jsou právě v módě, a dosti znepokojivý sklon k interpretacím a prognózám založeným na marketingových signálech Microsoftu.

Ačkoliv počet těchto lidí a jejich horlivost (nemluvě o jejich důvěryhodnosti obecně) nápadně klesá, pořád se najdou, připraveni prohodit do klíčové diskuse, že toho poslední dobou o VFP moc neslyšeli od dalších podlézavců Microsoftu, kteří zřejmě představují jejich hlavní zdroj technických informací. Z těchto míst také obvykle pochází šeptanda o tom, že zánik VFP je na spadnutí, založená buď na „Jedna paní povídala...“ nebo na „Mám takové tušení...“.

V takové situaci je vždycky nejlepší oddělit tvrzení a toho, kdo je pronáší (tedy „odosobnit“ problém), a zaměřit se na fakta:

existuje devátá verze VFP,

k VFP 9 vyjde co nevidět další service pack (pozn. redakce: právě vyšel SP2),

Microsoft slibuje podporu VFP 9 až do roku 2014, což je dále než u některých jejich oblíbených technologií, čekáme na vydání nadstavbových částí k VFP, jejichž smyslem je interoperabilita s existujícími a budoucími technologiemi,

VFP *pořád ještě* pracuje s daty (ve všech podobách) lépe než **jakýkoliv** jiný produkt Microsoftu, ať už desktopový nebo middleware,

VFP je stabilní a vyzrálý programovací jazyk, který na stupnici „sexuální přitažlivosti“, oblíbené u pracovníků marketingu a autorů, sotva uspěje, ale na tom, pokud jde o projekty IT, rozhodně není nic špatného,

VFP umí výborně COM, umí výborně XML, umí výborně pracovat s daty, umí výborně využívat metadata, má vyzrálý a zdatný programovací jazyk, má vynikající výkonnost, má vynikající schopnosti nasazení,

...a tak dále.

Zkrátka, měli byste být připraveni reagovat na připomínky tohoto druhu věcně a s chladnou hlavou. Podle mých zkušeností může otevřené jednání s těmito oponenty dopadnout dvěma způsoby a musíte se připravit na oba závěry. Buď se nedostanete nikam, nebo se vám podaří to, co původně vypadalo jako předem ztracený případ, zvrátit v přesvědčivé vítězství. Jen zřídka je výsledkem něco mezi tím. Jednoduše řečeno, tak to má být: buď získáte pro svůj návrh plnou podporu, a v tom případě se do toho pusťte, ale pokud ne, je často nejlepší od daného projektu odstoupit. Není totiž nic horšího, než když vám váhající zákazník dá potvrzení „tak napůl“.

Závěr

Navrhovat pro vývoj VFP vám někdy může připadat jako nerovný boj a často tomu tak opravdu je. Existují však systematické způsoby, jak k těmto otázkám přistupovat a jak budovat pádné argumenty pro technické a prodejní odůvodnění VFP. Doufám, že tato přednáška a tento dokument vám některé z vašich budoucích těžkých bitev trochu usnadní.

Část III – Příprava technického odůvodnění VFP

Váš návrh, proč použít pro vývoj právě VFP, musí samozřejmě splňovat všechny nezbytné technické podmínky. Musíte ukázat, že VFP dokáže splnit předvídatelné požadavky navrhovaného systému s ohledem na funkčnost i okolnosti s funkčností nesouvisející.

Souhrnně by technická část prodejního odůvodnění měla:

Prezentovat technické možnosti a přehled důsledků, které z nich plynou.

Poctivě ohodnotit technické aspekty různých alternativ.

Správně formulovat, v technických pojmech, technické otázky, příležitosti a potenciální hrozby.

Posoudit rizika, zejména ta technická.

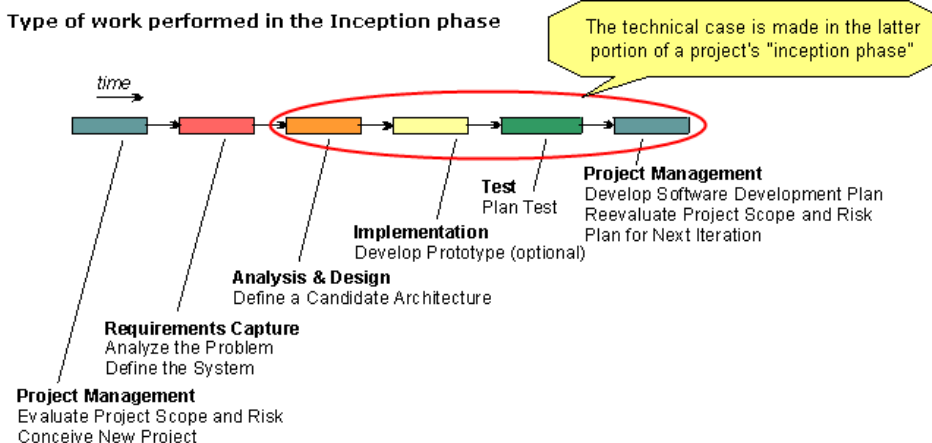
V maximální míře podpořit navrhovaný postup.

Ukázat jasné vazby mezi náklady a přínosy.

Úkol číslo 1: Porozumět systémovým požadavkům

Nemůžete se pustit do technického odůvodnění, dokud neznáte požadavky. Bez znalosti požadavků nemůžete vytvořit důvěryhodné technické odůvodnění vývoje, protože ve skutečnosti nevíte, co všechno je potřeba udělat. Obrázek 2 ukazuje počáteční fázi obecného projektu a také objasňuje, že technické záležitosti logicky přijdou na řadu až po obecném pochopení požadavků.

Pokud tedy zjistíte, že doporučujete VFP a přitom ještě nevíte, co je ve skutečnosti zapotřebí, je na vašem přístupu cosi od základů špatně; máte řešení problému, ale ten se teprve chystáte hledat.



Obrázek 2 Technické odůvodnění určité vývojové technologie tvoří důležitou část většiny činností, které nastanou po seznámení se s požadavky. Zde jsou zobrazeny různé kroky počáteční fáze v souladu se standardy používanými firmou Rational.

Slovníček požadavků

Bez důkladného pochopení požadavků je vaše technické odůvodnění k ničemu, takže představu, že byste mohli použít Visual FoxPro k řešení neznámého problému, prostě zavrhněte.

Na začátku je dobré vědět, že požadavky se dělí do různých kategorií a že je důležité těmto kategoriím rozumět, protože významně ovlivňují plánování a implementaci. Pro splnění některých typů požadavků může, ale také nemusí, být Visual FoxPro přínosem, zvlášť když jsou to požadavky *nejasné* nebo prakticky *chybějící*.

Obecně řečeno se dají požadavky rozdělit do následujících čtyř kategorií:

Požadavky na funkčnost jsou funkce, které musí systém poskytovat. Tyto požadavky jsou často formulovány jako úkoly nebo činnosti, které musí systém zajistit.

Požadavky s funkčností nesouvisející zahrnují takové vlastnosti jako výkonnost, použitelnost, udržitelnost, a také **Systémové požadavky**, jež můžete a nemusíte mít kontrolou a které odrážejí aktuální stav nainstalovaného hardwaru a operačních systémů. Požadavky s funkčností nesouvisející znamenají omezení různých atributů těchto funkcí nebo úkolů.

Omezení jsou obvykle předem definována a pokud ne, budou definována dodatečně, a mohou být zákeřná. Typická omezení se týkají času, nákladů, spolupráce s jinými systémy, vývojových standardů, personálu, hardwaru, již nainstalovaných součástí, školení aj.

Požadavky lze blíže vymezit atributy, mezi něž patří:

Nesmílované požadavky jsou ty, které je nutno splnit. Pokud některý z nich vynecháte, dojde k chybě nebo ke katastrofě nebo, v krajním případě, někdo možná přijde o život.

Mírnější požadavky jsou takové, že jim lze vyhovět v různé míře. Mají nepříjemný zvyk změnit se v nejnepříhodnější dobu na nesmlouvavé požadavky.

Nejasné požadavky jsou ty, které nejsou úplně definovány nebo jasně vysvětleny. Je o nich také známo, že bývají upřesněny *poté* co jste dokončili plánování a předběžné kalkulace. A jak jste jistě uhádli, i z nich se pravděpodobně stanou nesmlouvavé požadavky.

Chybějící požadavky jsou ty, které teprve budou odhaleny nebo správně formulovány. To nejlepší, co o nich kdy bylo řečeno, je: „*Dříve nebo později se určitě náležitě dozvíte, jaké jsou požadavky*“.

Odvozené požadavky jsou takové, které vyvstanou, až se dostanete od fáze požadavků do fáze vývoje a budování. To, co mohlo být z hlediska definice problému jasné a přímočaré, se může z hlediska řešení značně zkomplikovat. Seznam odvozených požadavků je obvykle mnohem delší než seznam požadavků vzniklých při definici a rozpracování problému.

Přibližné rozčlenění požadavků podle oblastí

Každá z níže uvedených oblastí jednak zahrnuje konkrétní funkce požadované projektem, ale vedle toho je i příležitostí ke zdůraznění předností VFP pro implementaci, nebo naopak situací, v níž je nutno uplatnit uklidňující strategie a argumenty.

Prostředí. Mám na mysli operační systémy (OS) a hardware. Pochopitelně budete muset do značné míry vyhovět všem omezením, která v dané organizaci platí pro verze OS a pro hardware. Tato oblast je pro zastávce VFP obvykle velmi příznivá, protože VFP běží na mnoha starších platformách Windows. Viz <http://fox.wikis.com/wc.dll?Wiki~VFPsupportedOS>.

Bezpečnost. Možná budete muset zaručit, že systém vyhoví všem požadavkům na bezpečnost. Je nutno vzít v úvahu, že VFP v této oblasti jen stěží zaboduje, jsou-li aplikační data uložena ve formátu .DBF.

Škálovatelnost. Existují různé aspekty škálovatelnosti a důraz na některé z nich lze do jisté míry očekávat u všech úspěšných systémů.

Když se mluví o „škálovatelnosti“ obecně, obvykle se tím rozumí tyto klasické aspekty škálovatelnosti.

- *Objem dat.* Někdy se používá termín „size up“ – schopnost přizpůsobit se narůstajícímu objemu dat.
- *Rychlost.* Někdy se používá termín „speed up“ – schopnost navýšit rychlost odezvy.
- *Pracovní zatížení.* Někdy se používá termín „scale up“ – schopnost zvládnout nárůst zátěže.
- *Cena transakce,* která je v softwarových systémech skoro vždy těsně spjata s *dobou trvání transakce.* Proč? Protože délka transakce určuje horní hranici počtu transakcí, které určitý server nebo skupina serverů dokáže zpracovat, aniž by bylo nutné přidávat nové servery, jejichž celkový počet může být omezen v závislosti na infrastruktuře sítě.

Méně běžné aspekty škálovatelnosti jsou:

- *Hardwarové platformy.* Zákazník může očekávat, že aplikace bude fungovat na oběh serverových farmách nebo naopak na příručích zařízeních.
- *Sdílení důvěry.* Systémy mají tu a tam těsně spolupracovat s jinými systémy a sdílení důvěry se vztahuje k obslužení uživatelů, kteří už byli ověřeni jinými systémy.
- *Šířka pásma.* Jak se projeví zvýšený provoz po síti v praxi.
- *Dostupnost služby.* Jak se bude měnit dostupnost služby s rostoucími hodnotami ostatních aspektů škálovatelnosti?
- *Zobrazování informací koncovým uživatelům.* Jak se systém vypořádá s novými službami, produkty a kategoriemi, které budou časem dostupné online?

TIP: Kdykoliv přijde téma škálovatelnosti na přetřes, ať už o ní začnete mluvit vy nebo někdo jiný, vždycky zaměřte diskusi na konkrétní typ škálovatelnosti a jednejte výhradně o něm, a to v kontextu navrhovaného systému. *Nesmíte* zabřednout do debaty o škálovatelnosti obecně, protože ta vždycky nastolí více otázek, než kolik jich přesvědčivě zodpoví, a to *nikdy* není dobrá pozice pro prodejní odůvodnění. Proto si uvědomte, že škálovatelnost má specifické aspekty, jejichž hranice a možnosti je nutno probírat jednotlivě.

Doba odezvy. Jak dobře splní aplikace ve VFP očekávání koncových uživatelů týkající se odezvy? Přestože v to věří mnoho programátorů, není předem dáno, že foxová aplikace bude mít úžasný výkon. Systémy, jež fungují velmi přesvědčivě s lokálními daty, se mohou sotva ploužit, když se ke stejným datům přistupuje po síti nebo když v systému přibudou desítky souběžně pracujících uživatelů. Projektování a okolnosti optimalizace by se nikdy neměly považovat za předem dané.

Průchodnost. Musíte ukázat, že aplikace ve VFP zvládne očekávané zatížení transakcemi a výběry dat. Překážky se mohou objevit na překvapivých místech v každém systému a VFP v tom není výjimkou.

Přípůsobivost a udržovatelnost. Významný podíl z celkových nákladů na životní cyklus všech úspěšných systémů obvykle ukousnou činnosti, které následují po vývoji, jako údržba, adaptace systému a jeho rozšiřování. Musíte ukázat, že navrhovaný systém bude mít takové provozní vlastnosti, že bude fungovat ještě dlouho po počátečním nasazení.

Spolehlivost a odolnost vůči chybám. Musíte ukázat, že navrhovaný systém s vysokou pravděpodobností splní očekávání týkající se provozuschopnosti, robustnosti a předvídatelných událostí, jako jsou výpadky proudu, havárie systému, běžné chyby uživatelů a podobně.

Rychlost implementace. Zákazníci budou potřebovat odhad času potřebného k implementaci kompletního fungujícího řešení. Dají se zde očekávat dvě měřítka: čas do nasazení první aplikace koncovému uživateli a celkový čas implementace celého projektu.

Flexibilita implementace. Jak pružné je to, co budujete? O softwaru se říká: „Je jako beton – můžete ho nalít do jakékoliv formy, ale když vytvrdne, už s ním nic nenaděláte“. Proto by podrobné technické odůvodnění mělo pojednávat i o tom, jak budou ošetřeny budoucí úpravy a rozšíření systému.

Snadnost používání: Jak dlouho se bude typický uživatel učit obsluhovat tuto aplikaci. Hodiny, dny, týdny? Kolik dnů školení je plánováno pro koncové uživatele a kolik pro odborníky IT?

Dostupnost odborníků. Dostupnost kvalifikovaných foxařů se v různých oblastech výrazně liší. Měli byste být schopni předvést, že existuje dostatek dobrých programátorů, na něž se může tento potenciální zákazník v případě potřeby obrátit. Informace vám poskytnou např. regionální sdružení foxařů. Nic vám nebrání založit si takové sdružení sami nebo oživit existující.

Další odkazy na témata, která by vám mohla pomoci při sestavování technických stránek prodejního odůvodnění pro VFP.

<http://fox.wikis.com/wc.dll?Wiki~WhyVFPIsTheToolOfYourChoice>

<http://fox.wikis.com/wc.dll?Wiki~VisualFoxProBulletPoints>

Někdy je na místě stručné technické odůvodnění

Nejpřesvědčivějšími technickými argumenty jsou někdy ty, které jsou jednoduché a přímočaré. Tato část slouží jako dobrý start při tvorbě stručného a jasného technického odůvodnění.

Stručný argument 1: Pracujte s tím, co máte

Technické odůvodnění někdy může být docela jednoduché. Vývoj softwaru má *vždycky* tato tři omezení:

- Čas
- Talent (Řízení, návrh a vývoj)
- Peníze

Některé situace jsou dány tím, co můžete udělat s prostředky, které máte k dispozici. Vaši volbu například výrazně ovlivňují takové faktory jako tým zkušených foxařů nebo významné investice vaší firmy do aplikací založených na VFP nebo omezené časové či finanční zdroje. To se může zdát na pohled samozřejmé, ale když máte jen málo možností, pokuste se výhodně prodat to, na co se můžete spolehnout.

Stručný argument 2: Spokojenost koncového uživatele

Spokojenost uživatele je mlhavý pojem, který však lze popsat obecnými termíny. Nespočívá jen ve splnění požadavků a nedá se jí docílit pouhou kvalitou systému. Hodně znamená také včasné dodání a cena systému.

Obecně se dá spokojenost uživatele definovat jako součet nesourodých prvků:

**Spokojenost
uživatele =**

**splnění požadavků + dodávka v pravý čas
+ odpovídající cena + kvalitní produkt**

Proto můžete rozdělit své odůvodnění na čtyři segmenty odpovídající pravé straně rovnice a v nich porovnat VFP s různými jinými variantami, které připadají v úvahu. Máte-li vy osobně nebo váš tým bohaté zkušenosti s VFP, a zvláště používáte-li nějaký framework, což při vývoji snižuje řadu rizik, nejspíš s tímto přístupem docela slušně zabodujete.

Rizika vývoje softwaru

Jen málo věcí zní zákazníkům lépe než plánování s vědomím rizik a vyhlídky na vývoj a implementaci softwaru s averzí vůči rizikům.

Existují dva obecné typy rizik: neregulovatelné, známé jako *exogenní (vnější) riziko*, a regulovatelné, známé jako *endogenní (vnitřní) riziko*. Je potřeba rozpoznat a minimalizovat všechny potenciální zdroje exogenních rizik a co nejlépe zvládnout endogenní rizika.

Analýza rizik slouží k rozpoznání a posouzení faktorů, jež mohou ohrozit úspěch projektu nebo dosažení cíle. Dobrá analýza také pomáhá definovat preventivní opatření snižující pravděpodobnost, že se tyto faktory vyskytnou, a stanovuje, jak se s nimi úspěšně vypořádat, když už nastanou.

Autor Steve McConnell vypočítává ve své zásadní knize *Rapid Development* 36 "[klasických omylů](#)", jejichž seznam následuje. Když vezmeme v úvahu konkrétní alternativy za VFP, o nichž uvažují vedoucí pracovníci ve vašem případě, které omyly z tohoto seznamu jsou v prostředí těch alternativ pravděpodobnější? Možná se budete divit.

Omyly spjaté s lidmi	Omyly spjaté s procesy	Omyly spjaté s produkty	Omyly spjaté s technologiemi
1. Podkopaná motivace	14. Přehnaně optimistický časový plán	28. Přeceňování některých požadavků	33. Syndrom kouzelné hůlky (všeléku)
2. Slabá posádka	15. Nedostatečné zvládnání rizik	29. Změny funkcionality	34. Přeceňování úspor dosažených díky novým nástrojům nebo metodám
3. Nezvladatelní problémoví zaměstnanci	16. Selhání subdodavatele	30. Vývojářské výstřelky	35. Přechod na jiné nástroje během projektu
4. Hrdinství	17. Nedostatečné plánování	31. Rozšíření zadání výměnou za toleranci zpoždění	36. Absence automatizovaného řízení změn (source-code control)
5. Přibírání lidí do zpožděného projektu	18. V tísní se upustí od plánování	32. Vývoj zaměřený na výzkum	
6. Hlučné a přeplněné kanceláře	19. Vyplýtvání času ve fázi rozplzlého sjednávání spolupráce		
7. Třeníce mezi programátory a zákazníky	20. Omezení „vedlejších“ aktivit		
8. Nerealistická očekávání	21. Nedostatečný design		
9. Nedostatek účinné podpory projektu	22. Ošizené řízení kvality		
10. Nedostatečné stvrzení od akcionářů (vlastníků)	23. Nedostatečná kontrola plnění		
11. Nedostatek podkladů od uživatelů	24. Předčasné nebo příliš časté spouštění finalizačních úkonů		
12. Politika stavěná nad obsah	25. V odhadech jsou opomenuty nezbytné úkoly		
13. Zbožná přání	26. Plánování „doženeme to později“		
	27. Snaha rychle generovat co nejvíce kódu		

Rizika vývoje se obvykle s výzrálostí produktu snižují

VFP se objevila ve verzi 3.0 v roce 1995, ale mnohé prvky jazyka i vývojového prostředí známe už podstatně delší dobu.

Díky své dlouhé životnosti, kvalitě a vyzrálosti komerčních frameworků a všeobecnému povědomí o jejich silných a slabých stránkách má VFP dobrou relativní pozici mezi mnoha dalšími vývojovými nástroji, když dojde na regulaci endogenních rizik vývoje softwaru.

Vyzrálost VFP a kvalitu frameworků od jiných výrobců byste určitě měli ve svém technickém odůvodnění zdůraznit. Některé aspekty jsou:

Stabilita

S věkem přichází zocelení a VFP rozhodně je ostřílený voják. S každou novou verzí a service packem získává vyšší stabilitu. Všechna vývojová prostředí mají svá pro a proti a ani VFP na tom není jinak. U VFP ovšem máte dost slušnou jistotu, že nenarazíte na nepřekonatelné překážky, případně na chybně fungující nebo nedodělané součásti, které dostatečně neodzkoušeli programátoři v reálných podmínkách.

Produktivita vývoje

K VFP se dodává široký výběr vynikajících a osvědčených vývojových frameworků a doplňků od jiných výrobců pro rozmanité oblasti použití, včetně zcela specifických balíčků pro účetnictví, online obchody na webu, komunikace, tvorbu sestav atd. U většiny z nich je k dispozici kompletní zdrojový kód, mimořádná podpora v rámci komunity, a často i pravidelná vylepšení a upgrady. Proto je nesmírně důležité zajistit, aby vedoucí pracovníci pochopili, že „prostor VFP“ je plný hospodárných produktů a nástrojů, díky nimž je vývoj rychlejší, robustnější a méně riskantní.

Jiný významný aspekt produktivity vývoje se týká samotných programátorů. Vývojářské zkušenosti se obvykle liší co do šířky i hloubky. Podle různých zdrojů – výzkumu i názorů – se zdá, že rozdíl v produktivitě mezi vynikajícím a průměrným tvůrcem softwaru může být až 20:1. Vzhledem k tomu, že VFP je tady už dlouhou dobu, narazíte dost pravděpodobně na programátory ve VFP, kteří mají letité zkušenosti.

Argumenty týkající se produktivity vývoje očividně mají značnou váhu. Můžete-li je uplatnit, rozhodně je zdůrazněte.

Obavy z pravděpodobné zastaralosti

Snad žádný faktor není pro vedoucí pracovníky přesvědčivější než očividný nedostatek podpory VFP ze strany Microsoftu. To přirozeně vede k jejich obavám z pravděpodobné zastaralosti, kterým můžete čelit různými způsoby:

Horizont životního cyklu produktu

Podle toho, co oznámil Microsoft, má VFP verze 9 naplánovanou podporu do roku 2014. Viz <http://support.microsoft.com/gp/lifeselectindex>. V době vzniku tohoto článku je to podstatně pozdější termín než u většiny ostatních vývojových nástrojů Microsoft.

VFP není kriticky závislá na Windows

To je obrovské „plus“. Můžete najisto říci svému zákazníkovi, že řešení založené na VFP jen zřídka vyžaduje zásadní upgrade hardwaru případně Windows v celém systému. Fakt, že VFP nejspíš bude hladce fungovat na dosavadních počítačích a operačních systémech, je velikou výhodou.

Další informace najdete na <http://fox.wikis.com/wc.dll?Wiki~VFPSupportedOS>.

Srovnajte to s řadou dalších produktů Microsoftu těsně spjatých s Windows. Protože se Windows i tyto produkty vyvíjejí, dosáhnou pravděpodobně „zastaralosti“ nebo „ukončení“ mnohem dřív než VFP. Například už bylo oznámeno, že Office verze 12 bude vyžadovat Windows XP nebo novější. Když na chvíli zapomeneme, že jediný, kdo opravdu potřebuje novou verzi Office, je Microsoft, máte pocit, že teď používáte zastaralý textový procesor? Zastaralý tabulkový procesor? Nestojí vaše nynější dokumenty nebo tabulky za nic? Zůstanete na holičkách, když se Office bude ubírat jinam?

Dostupnost odborníků

To je v současné době hlavním zdrojem starostí všech foxařů. Když si vybereme za vývojové prostředí VFP, jaké jsou záruky dlouhodobé dostupnosti odborníků, které by toto rozhodnutí podpořily? Odpověď záleží na vás a na vašem vztahu k nejbližší i vzdálenější komunitě VFP. Jste členem regionálního sdružení foxařů? Víte o dalších programátorech VFP ve vašem městě nebo kraji? Máte možnost spojení online s dalšími foxaři, ať už blízkými či vzdálenými? Pokud ano, máte dobrou příležitost rozvíjet a povzbuzovat talentované našince a tak přispět k budoucímu vývoji. Pokud se naopak nezapojujete do žádné bližší ani vzdálenější foxařské komunity, nemůžete nijak ovlivnit, co se stane, až bude tato otázka nastolena.

Obavy z technických změn

Vždycky mě podráždí věty jako „Jediné, co je v našem oboru stálé, jsou změny“ nebo „metodologie a postupy vývoje aplikací se neustále mění“. Nepodléhejte tomu; jsou to žvásty. Podobné věty obvykle ve skutečnosti znamenají: „Ukazuje se, že loňský rok jsme přechválili a předloňský rok už vůbec nedopadl dobře, takže teď chceme to, co je právě v módě“.

Jistě, ke změnám dochází. Je ale rozdíl mezi tím, když se přizpůsobíme postupným vylepšením a zdokonalením, a když vychvalujeme změnu samu o sobě. Dříve nebo později musí dodávka zákazníkům dostat přednost před zamilovaností do technologie.

V našem oboru je opravdu stálé toto:

Mnozí (ne-li všichni) vývojáři softwaru využívají možnosti těch vývojových nástrojů, které už mají, v žalostně malé míře.

Mnohé (ne-li všechny) podniky využívají v žalostně malé míře výhody hardwaru a softwaru, který mají (nebo dlouho měli), a většina podniků vytěží žalostně málo ze svých investic do IT.

Většina podniků má víc počítačů a softwarových licencí, než kolik má lidí, kteří by s nimi pracovali.

Rozdíl v produktivitě mezi vynikajícím a sotva průměrným programátorem může být až 20:1.

Vývojáři, kteří stěží dokážou úspěšně dokončit projekt a udržet si zákazníky, když používají Vývojové Prostředí A, zjistí po přesedlání na Vývojové Prostředí B, že stěží dokážou úspěšně dokončit projekt a udržet si zákazníky.

Vývojářské postupy s vědomím rizik vždycky zní sponzorům systémů IT velmi dobře. Je ovšem příliš snadné se ošklivě spálit i pro programátory, kteří vědí, co dělají, natožpak pro ty, kteří to nevědí.

Dejte si pozor na směšně drobná vylepšení v moderním přestrojení ohlášená jako největší trhák příští sezóny.

Výhoda na webu

VFP má svou charakteristickou syntaxi, takže Google obvykle dokáže najít odpověď na váš požadavek velmi rychle.

Zkuste například vyhledat řetězec: `[CREATE CURSOR AutoInc]`.

Kvalita zdrojů informací dostupných online je určitě výhodou VFP.

Hlavní silné stránky VFP mají svou váhu

V některých oblastech je VFP opravdu silná; patří sem například:

Zaměření na data

Těžko se dají zveličít výhody plynoucí ze schopnosti pracovat s daty nativně. Možnost pohodlně vybírat data a zpracovávat je ve stejném logickém prostoru patří pro zkušené foxaře k nejpřesvědčivějším přednostem VFP.

Interoperabilita

Dodnes je VFP vzorem interoperability: dokáže snadno číst, importovat a exportovat širokou paletu datových formátů. VFP má vynikající schopnosti v oblasti COM a XML. A pokud se něco nedá provést přímo (nativně), obvykle se to dá snadno naprogramovat, protože jestli VFP něco dělá jasně a jednoduše, jsou to data a jejich zpracování.

Všestrannost

Ve VFP se dají vytvářet desktopové aplikace, internetové aplikace, uživatelská rozhraní pro přístup k úložištím dat, middlewarové komponenty, komponenty COM, a pro každou z těchto implementací je bohatý jazyk a schopnosti práce s daty VFP k dispozici přímo *uvnitř* vrstvy, v níž právě pracujete.

Kde s VFP nejspíš neprorazíte

V určitém okamžiku, zvláště při jednání s rozsáhlou organizací, jejíž centralizované databáze udržuje tým databázových a síťových administrátorů, narazíte na značný tlak, který se vám prostě nepodaří překonat. Toto jsou některé z bodů, které pravděpodobně nikdy nevyhrajete.

Aplikační data uložená v souborech .DBF

Nedivte se, že často, zejména u středních až velkých společností a *zvláště* u společností s centralizovanými datovými středisky, kde databázoví administrátoři udržují aplikace pracující s databázovými servery, je každý návrh ukládat data do souborů .DBF předem ztracený případ.

Je to zcela pochopitelné. V mnoha společnostech se představa otevření části disků na síti pro přímý přístup k souborům s daty *nikdy* neujmeme. Navrhují: zvykněte si na to.

TIP: Zjistil jsem, že často je dobré ze začátku založit prodejní odůvodnění na tom, že VFP bude pracovat s databázemi jako MySQL, MS-SQL nebo Oracle. Tak se vyhnete problémům plynoucím z dat v souborech DBF ve sdílených síťových adresářích. Později, když se vedoucí pracovníci zarazí nad licenčními nebo logistickými náklady souvisejícími s těmito databázemi SQL a když vám to připadá smysluplné, můžete vždycky vyrukovat s námětem ukládat data do souborů DBF. Většina programátorů, kteří vyvinuli úspěšný projekt pracující s úložištěm dat, má zájem u toho zůstat. Data VFP v celopodnikových aplikacích opravdu nejsou *nic úžasného*.

Malá zařízení

VFP nebude dobrou vývojovou platformou pro programování příručních zařízení, a nepřipadá ani v úvahu, aby se na těchto malých zařízeních provozovala. To ale neznamená, že VFP nemůže být hlavní součástí globálního systému, který malá zařízení používá.

Na vlastní oči jsem viděl několik impozantních řešení zahrnujících spolupráci VFP s malými zařízeními. Pro úspěch v této oblasti není nutné provozovat VFP přímo na malém zařízení. Stačí, když dokážete synchronizovat data v obou směrech, tedy na vstupu i na výstupu z malého zařízení.

Pokus mít jedno prostředí, které „všechno zvládne“, vás pravděpodobně zavede do slepé uličky. Zamyslete se: proč byste měli „automaticky“ vázat různorodé hardwarové platformy a prezentační sémantiku na totéž vývojové prostředí? Tato slepá víra si zaslouží pečlivou úvahu, protože rozhodnutí, která se nedají změnit, mohou podstatně omezit budoucí možnosti. Každý majitel příručního zařízení vám řekne, že tato zařízení mají své vlastní cykly „nových verzí“ a zůstat u dané technologie znamená koupit nové zařízení. Uvažte, že nová verze vývojového prostředí *může* vyžadovat upgrade Windows, což může a nemusí vyžadovat upgrade vývojářského PC, případně klientských stanic, a vy do té směsice chcete ještě přidat všechna ta malá zařízení a jejich operační systémy? Jinými slovy, zahrnutí malých zařízení do požadavků na aplikaci neznamená automaticky, že jedno vývojové prostředí „děvče pro všechno“ bude to nejlepší, jednotné a univerzální, i když na první pohled to může vypadat smysluplně.

Některé z nejlepších implementací malých zařízení zapojují VFP v překvapivé míře. Malá zařízení jsou programována nezávisle, s využitím frameworků, které se pro ně nejlépe hodí, a k integraci malých zařízení se zbytkem podniku slouží přímé kanály ošetřující logistiku na vstupu i na výstupu z malých zařízení.

Zahrnutí malých zařízení do požadavků nemusí tedy VFP automaticky diskvalifikovat. Zacházení s malými zařízeními v netriviálních aplikacích není žádná hračka, ať už děláte cokoli, a nejsou tu žádné „automatické“ volby.

Dostupnost odborníků

Už jsme se o tom zmínili: nic neděsí sponzory projektu víc než chabé vyhlídky na nalezení kvalifikované posádky. Naštěstí je komunita VFP velmi vstřícná a mimořádně dobře „přidrátovaná“, takže najít lidi je obvykle možné. Pokud je na tom ale vaše oblast bídně, měli byste udělat, co je ve vašich silách, a investovat čas a úsilí do propagace a podpory regionálního sdružení foxařů. Mějte na paměti, že jen malé procento programátorů je aktivní on-line. I když to třeba vypadá, že jich máte v okolí málo, neznamená to ještě, že tam nejsou. Jenom vám dá víc práce je najít.

Část IV – Dají se zde využít strategie koncovek? Stručný přehled

Koncovka: závěrečné stadium šachové hry, kdy už na šachovnici zbývá jen několik figur.

Když Microsoft před časem oznámil, že nebude žádná další verze Visual FoxPro, řada foxařů se vyjádřila, že se blíží jejich konec. Podle nich začala závěrečná fáze hry a někteří aktivně hledají jiné technologie, na něž by rychle přešli.

Dovolu, abych začal prohlášením, že nesouhlasím s názorem, že po ohlášení Microsoftu o neexistenci příští verze VFP je „konec na dohled“.

Je to složitý problém a v mnoha ohledech je do jisté míry „konec na dohled“ pro všechny současné softwarové technologie. Nemyslím si, že to, co dělá nebo nedělá s VFP Microsoft, bude nakonec znamenat podstatný rozdíl.

Než si vyrazíte o překot hledat svou novou hlavní technologii, možná by se vyplatila objektivní analýza vyhlídek vaší nynější hlavní činnosti.

Podklady k vyhodnocení

Předpokládejme však pro účely této diskuse scénář nejhoršího případu. Dejme tomu, že to, co Microsoft říká o příští verzi VFP, je pro foxaře opravdu důležité a navíc to přímo ovlivní šance vašeho podniku. Dejme tomu, že „žádná verze VFP 10“ znamená v podstatě *konec hry*.

V roce 1983 Kathryn Rudie Harrigan a Michael Porter z Harvard Business School publikovali klasický článek nazvaný „*Endgame Strategies for Declining Industries*“ (*Strategie koncovek pro upadající odvětví*). Je zde popsán všeobecný rámec pro přemýšlení o problémech odvětví, která mají na kahánku. Tento článek odolal zubu času a to, co bylo napsáno v roce 1983 o výzkumu probíhajícím v předchozích letech, platí i dnes.

Na všechny publikace Harvard Business School se vztahuje autorské právo, ale za 6 dolarů si můžete stáhnout tento článek ve formátu PDF z adresy <http://hbr.com>. Stačí vyhledat název „Endgame Strategies for Declining Industries“. Mimochodem, podle mého je to nejlepší online obchod, jaký jsem kdy na Internetu viděl.

Článek se zabývá 3 hlavními body:

- Faktory určující konkurenční prostředí
 - „podmínky na straně poptávky“
 - „překážky na ústupové cestě“ a
 - „faktory soupeření“
- Strategické alternativy pro upadající odvětví
- Výběr strategie pro pokles

Doplňky k článku tvoří:

- Studie strategií pro upadající odvětví
- Dodatek I Strukturální faktory ovlivňující přitažlivost prostředí upadajících odvětví
- Dodatek II Strategie pro upadající odvětví

Zde jsou reprodukce přehledů Exhibit I a II.

Structural factors	Environmental attractiveness	
	Hospitable	Inhospitable
Conditions of demand		
Speed of decline	Very slow	Rapid or erratic
Certainty of decline	100% certain predictable patterns	Great uncertainty, erratic patterns
Pockets of enduring demand	Several or major ones	No niches
Product differentiation	Brand loyalty	Commodity-like products
Price stability	Stable, price premiums attainable	Very unstable, pricing below costs
Exit barriers		
Reinvestment requirements	None	High, often mandatory and involving capital assets
Excess capacity	Little	Substantial
Asset age	Mostly old assets	Sizable new assets and old ones not retired
Resale markets for assets	Easy to convert or sell	No markets available, substantial costs to retire
Shared facilities	Few free-standing plants	Substantial and interconnected with important businesses
Vertical integration	Little	Substantial
"Single product" competitors	None	Several large companies
Rivalry determinants		
Customer industries	Fragmented, weak	Strong bargaining power
Customer switching costs	High	Minimal
Diseconomies of scale	None	Substantial penalty
Dissimilar strategic groups	Few	Several in same target markets

	Has competitive strengths for remaining demand pockets	Lacks competitive strengths for remaining demand pockets
Favorable industry structure for decline	Leadership or niche	Harvest or divest quickly
Unfavorable industry structure for decline	Niche or harvest	Divest quickly

Obrázek 3 Přehledy Dodatek I a II z článku „Endgame Strategies for Declining Industries“, který publikovali Kathryn Rudie Harrigan a Michael Porter a který se dá zakoupit a stáhnout na adrese <http://hbr.com>.

Z článku si lze odnést mnohé, například:

- Vyhodnocení probíhá asi takto:
 - Umožňuje struktura daného odvětví, aby byla fáze postupného zániku příznivá, případně i zisková (viz Dodatek I)?
 - Co brání významným konkurentům na ústupu z odvětví? Kdo rychle odejde a kdo zůstane?
 - Hodí se silné stránky vaší společnosti pro zbývající příležitosti na trhu?
 - Jak jsou na tom v tomto směru silné stránky vašich konkurentů? Jak se dají překonat jejich překážky na ústupové cestě?
- Pokud zůstanete u VFP, měli byste vzít v úvahu pár taktických manévřů. **Pozor:** Některé z nich jsme už v komunitě VFP zaznamenali.
 - **Postarejte se, aby se ostatní společnosti z daného odvětví rychle stáhly.** Příklad: H.I. Heinz a Gerber Products podnikli agresivní konkurenční kroky ve stanovení cen, marketingu a dalších oblastech, které zvyšují podíl na trhu, a jejich konkurenti se museli vzdát naděje, že by je porazili.

- **Odstraňte svým konkurentům některé překážky z ústupové cesty.** Příklady:
 - Společnost GTE Sylvania si vybuodovala pozici na trhu tím, že skupila produkci svých konkurentů za vyšší než běžnou cenu.
 - Společnost American Viscose zakoupila – a pak předčasně ukončila – výrobní kapacitu svých konkurentů. (Tento krok zaručuje, že tuto kapacitu nekoupí někdo jiný z daného odvětví.)
 - Společnost General Electric vyráběla náhradní díly pro produkty svých konkurentů.
 - Společnost Rohm & Haas převzala dlouhodobé kontrakty svých konkurentů.
 - Společnost Proctor-Silex vyráběla zboží pro své konkurenty pod svou značkou, takže ti mohli zastavit vlastní výrobu.
- **Vymyslete a rozšiřte přesvědčivé informace o trhu.** Upevníte tím jistotu jiných manažerů, že postupný zánik je nevyhnutelný.
- **Zvyšte sázky.** Přinutíte-li ostatní konkurenty, aby znovu investovali do nových produktů nebo do vylepšení procesů, bude pro ně nákladnější udržet se v podnikání.
- Jak předejít matu:
 - **Připusťte si možnost zániku odvětví:** Buďte objektivní. Je bolestné poznat špatná znamení a je pochopitelné, že lidé hledají ta dobrá.
 - **Ti, kdo jsou v řízení postupného zániku neobjektivnější, se také angažují v náhradním odvětví.** Zřetelněji vnímají vyhlídky náhradního produktu a realitu zániku.
 - Vyhněte se vyhlazovací válce.
- **Někdy mohou být upadající odvětví mimořádně výnosná pro hráče s dobrou pozicí,** jak odhalily GE a Raytheon v oboru elektronek. Společnosti, která dokáže vidět zánik odvětví jako příležitost a ne jen jako problém a umí udělat objektivní rozhodnutí, to může vynést slušný zisk.

Závěr

Navrhovat pro vývoj VFP vám někdy může připadat jako nerovný boj a často tomu tak opravdu je. Existují však systematické způsoby, jak k těmto otázkám přistupovat a jak budovat pádné argumenty pro technické a prodejní odůvodnění VFP. Doufám, že tato přednáška a tento dokument vám některé z vašich budoucích těžkých bitev a rozhodování trochu usnadní.

Odkazy

Vedle odkazů, které jsou uvedeny přímo v dokumentu, jsou tu i další zajímavé zdroje.

Odpovědi na nejčastější otázky o investicích do IT: http://www.rms.net/lc_faq_case_checklist.htm

Šablony prodejních odůvodnění společnosti ZDNet: <http://whitepapers.zdnet.co.uk/>

Ukázka obsáhlého (153stránkového) prodejního odůvodnění:

<http://www.show.scot.nhs.uk/itcontract/obc%20version%201.5%2026%20may%202005%20foi%20pdf%20v0.2.pdf>

Database Skills Availability: Critical to Your Selection of Database.

<http://uk.builder.com/whitepapers/0,39026692,60070482p-39000853q,00.htm>

<http://www.construx.com/docs/member/10MythsOfRapidDevelopment-Keynote.pdf>

Sedm smrtelných nebezpečí vývoje softwaru s využitím outsource/subdodavatelů

<http://accelerance.com/avoid-the-seven-dangers.htm> (povinná registrace).

