Old Man GURU Magazine

Wychodzi bardzo nieregularnie, kiedy wydaje mi się,
że mam coś ciekawego lub pożytecznego do napisania...

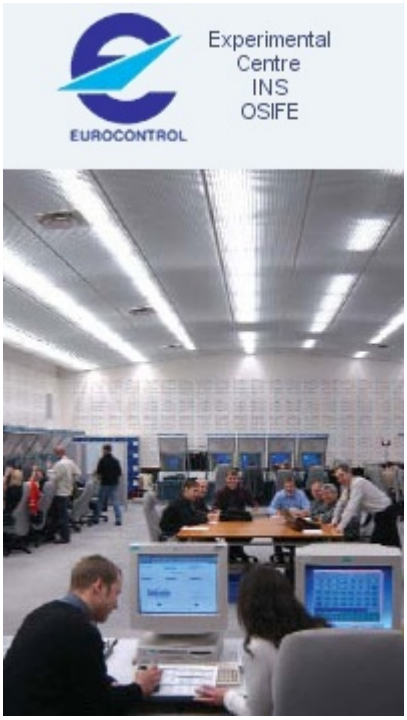
Numer 3/2009 **7 wrzesień 2009 r.**

Numer specjalny!

Wydany z okazji ukazania się "REKOMENDACJI" Udzielania Zamówień Publicznych na Systemy informatyczne opracowanych przez Urząd Zamówień Publicznych

LINUX i "Gęba"
czyli trochę o projektowaniu systemów informatycznych

Odwiedził mnie wczoraj znajomy programista pracujący w dużej firmie. Znam dość dobrze jego możliwości z tak zwanych "dawnych lat" i ceniłem go zawsze za sprawność i elegancję jego kodów tworzonych głównie w języku C. Nie byłbym sobą, gdybym się nie zapytał o wykorzystywanie Wolnego Oprogramowania w firmie, która obecnie go zatrudnia. Odpowiedział bez chwili wahania:
"no co ty, nasz prezes zdecydowanie sprzeciwia się temu. Twierdzi, że będziemy kupować oprogramowanie jedynie od dużych firm komercyjnych. LINUX? Open Office? Nie ma mowy! Nasza firma nie może polegać na programach tworzonych przez zasmarkanych uczniaków podczas przerw w szkole i pomiędzy przegryzaniu kawałków pizzy. To niepoważne!"



Ta rozmowa świadczy dobitnie o tym, jaką "GĘBĘ" (w sensie gombrowiczowskim) udało się "dorobić" Linuksowi i całemu Wolnemu Oprogramowaniu. Gdyby Pan Prezes wiedział, że lecąc samolotem nad Europą powierzył swoje życie "programowi stworzonemu przez nieodpowiedzialne dzieciaki" pewnie nie wsiadłby do samolotu, bo system wyznaczający strefy bezpieczeństwa w przestrzeni powietrznej zwany NoGoZone wspomagający kontrolerów ruchu lotniczego wykorzystywany (choć na razie próbnie) przez EUROCONTROL pracuje pod systemem LINUX... Łatwo to sprawdzić...
Co więcej - oprogramowanie to można nawet pobrać ze stron Sourceforge jako otwarty projekt na licencji GPL v2.
NoGoZone opracował Ośrodek Rozwojowy EUROCONTROL - pracujące tam osoby (patrz zdjęcie) nie wyglądają raczej na "zasmarkanych uczniaków"...

Podobne przykłady można mnożyć...

Jeszcze bardziej można się uśmiać z głośno wygłaszanych stwierdzeń - "*no tak, ale Linux nie jest skalowalny - i nie sprawdza się w dużych instalacjach*".

Wyznawcy takich poglądów chyba nigdy nie zaglądnęli na listę najszybszych maszyn Świata - www.top500.org bo stwierdziliby, że ponad 88% maszyn z pierwszej "pięćsetki" (dokładnie 443) pracuje pod Linuksem. Warto również dodać, że w pierwszej "pięćsetce" znajduje się zaledwie 5 maszyn pracujących po ulubionym przez wielu MS Windows.
Pierwsza dziesiątka najszybszych komputerów to wyłącznie maszyny pracujące pod kontrolą systemu LINUX, który charakteryzuje się ogromną skalowalnością (obsługa wielu tysięcy procesorów, Terabajtów pamięci operacyjnej, złożonych systemów klastrowych itp.).

I kto tu mówi o braku skalowalności...

Choćby tylko te dwa podane powyżej fakty (a nie zasłyszane "opinie specjalistów") świadczą o tym, że Linux (oraz inne Wolne Oprogramowanie) należy traktować jako pełnowartościowe rozwiązanie mogące z powodzeniem stawić czoła systemom komercyjnym - zarówno w największych systemach o ogromnej mocy przetwarzania, jak i w systemach klasy "Mission Critical".

Jeśli jest tak dobrze - to dlaczego jest tak źle?

Dlaczego wiele specyfikacji przetargowych jest w Polsce przygotowywanych w taki sposób, aby z góry wykluczyć rozwiązania oparte o Wolne Oprogramowanie? Czy to tylko efekt intensywnych działań marketingowych? A może jednak istnieją jeszcze jakieś inne przyczyny?

Moim zdaniem jest ich co najmniej kilka:

Jako pierwszą wymieniłbym przeświadczenie, że w przypadku oprogramowania komercyjnego dokonujemy jego zakupu jako pewnej trwałej wartości. Tak również traktują to aktualne przepisy księgowe - "oprogramowanie" o "wartości" powyżej 3500 PLN podlega przecież nawet amortyzacji...
Dlaczego napisałem słowa "oprogramowanie" i "wartość" w cudzysłowiu? Ponieważ tak naprawdę bardzo rzadko mamy do czynienia z rzeczywistym zakupem oprogramowania. Tak naprawdę otrzymujemy odpłatnie jedynie prawo do używania "zakupionego" programu i to zazwyczaj w dość ograniczonym zakresie. No, ale kto czyta dołączane do oprogramowania dokumenty prawne? Firmy komercyjne stawiające sprawę jasno i uczciwie należą również jak choćby Network Instruments (NI) do rzadkości - w ich cennikach mamy następujące pozycje:

1. Base Kit (nośnik, dokumentacja, klucz sprzętowy),
2. Right to Use - czyli to co my potocznie nazywamy "licencją", a tak naprawdę jest dokładnie tym, co znaczy czyli "prawem do wykorzystywania" programu,
3. Maintenance Contract - a więc umowa o świadczenie wsparcia technicznego. W przypadku Netowork Instruments jest to umowa opcjonalna.

Na czym polega różnica pomiędzy oprogramowaniem "Komercyjnym z półki" (COTS - Commercial of the Shelf) a Wolnym Oprogramowaniem. Z punktu widzenia przeciętnego użytkownika przede wszystkim na tym, że prawo do użytkowania (Right to Use) otrzymujemy nieodpłatnie (dla porównania w przypadku programu Expert Observer NI ta pozycja to 3650 EUR), przy cenie Base Kit 100 EUR, i rocznej umowy Maintenance Contract 750 EUR.

A więc cena praw do wykorzystywania to ponad 80% ceny "zakupu" oprogramowania Expert Observer z rocznym wsparciem technicznym.

Z dużym prawdopodobieństwem można przypuszczać, że inni dostawcy oprogramowania komercyjnego stosują podobną politykę. A stąd wynika, że oszczędności z tytułu wyboru oprogramowania, dla którego nie musimy płacić za prawo do korzystania wyniosą około 80%.

Proszę zauważyć, że także dla Wolnego Oprogramowania opłacamy koszty nośnika i dokumentacji oraz zawieramy umowę rocznego wsparcia technicznego! A więc argument o braku wsparcia technicznego dla Wolnego Oprogramowania nie ma uzasadnienia - za taką samą cenę otrzymamy zapewne bardzo podobne warunki jego świadczenia. Zawarcie odrębnej umowy o świadczenie wsparcia technicznego jest zawsze korzystne - zarówno w przypadku komercyjnego, jak i wolnego oprogramowania, ponieważ po prostu dokładnie wiadomo, czego w ramach tego wsparcia możemy oczekiwać!

Aby zakończyć rozważania dotyczące opłat za prawo do korzystania z oprogramowania należy zauważyć, że zdecydowana większość licencji otwartych (a w szczególności GPL) nie stawia żadnych ograniczeń dotyczących wykorzystywania oprogramowania oraz jego kopiowania. Oznacza to, że możemy korzystać z dowolnej liczby kopii oprogramowania na dowolnej liczbie komputerów!

Jest to dość oczywiste, ponieważ jeśli w ogóle zrezygnowano z opłat za jego wykorzystywanie, to nie ma znaczenia liczba kopii oraz maszyn, bo przecież 0 x cokolwiek = 0.

Oczywiście nie dotyczy to dostarczanej liczby nośników i dokumentacji oraz umów wsparcia technicznego, dla których obowiązują takie same reguły, jak dla produktów komercyjnych.

Dlaczego więc "lekką rączką" płacimy o 80% więcej za prawo do korzystania z programów komercyjnych o właściwościach porównywalnych z programami, których twórcy (lub dostawcy) nie pobierają opłat za samo ich użytkowanie?

Oprócz dość powszechnego przeświadczenia, że prawdziwą wartość można otrzymać jedynie za pieniądze, które w przypadku "udzielenia praw do korzystania" nie ma żadnego logicznego uzasadnienia spore znaczenie ma fakt. że dostawcy programów komercyjnych twierdzą, że obejmują je "troskliwą opieką". Może i w wielu przypadkach jest to prawda - ale klient powinien wiedzieć, na co konkretnie może w ramach tej "troskliwej opieki" liczyć. A z tym bywa już znacznie gorzej. W większości przypadków nie otrzymujemy umowy określającej obowiązki dostawcy lub producenta oprogramowania, a jedynie dość długą listę wyłączeń odpowiedzialności "w maksymalnych granicach dozwolonych przez prawo".

Prawda jest dość bolesna - użytkownik może liczyć tylko na takie wsparcie, jakie zostało mu zagwarantowane w konkretnej umowie - wszelkie inne deklaracje mają niewielkie znaczenie.

Druga przyczyna ma charakter bardziej techniczny niż ekonomiczny. Spróbuję to wyjaśnić na przykładzie:

Rozwój komputerów osobistych datuje się od lat osiemdziesiątych ubiegłego wieku. Dla wielu osób komputer PC to synonim komputeryzacji w ogóle, a przecież (uważany wówczas za przyjazny dla użytkownika) system UNIX to rok 1969, a X Window - 1984. Warto przy tym pamiętać, że X Window od początku swego istnienia było systemem sieciowym, że MS Windows 1.0 zaprezentowane po raz pierwszy w 1985 r. jedynie nakładką graficzną na system DOS. Pierwsze sprawnie działające MS Windows 3.0 to dopiero rok 1990, zaś obsługę protokołów rodziny TCP/IP włączono dopiero do MS Windows 95 w drugiej połowie lat dziewięćdziesiątych.

System MS Windows "rósł" więc w naturalny sposób wraz z rozwojem sprzętu komputerów PC - lecz przez bardzo długi czas był tworzony głównie pod kątem potrzeb indywidualnego użytkownika. Projektanci firmy Microsoft przyjęli więc dwa słuszne założenia:

- użytkownik komputera jest zarazem administratorem jego oprogramowania, bo jest to komputer osobisty,
- od przeciętnego użytkownika komputera nie można wymagać zaawansowanej wiedzy technicznej.

Wpływ tych założeń jest do dziś bardzo dobrze widoczny - technologia Plug&Play, różnego typu kreatory, ograniczenie do minimum wpisywania danych z klawiatury na korzyść wyboru z zakładek oraz list itp.

Z Uniksem (a później także z Linuksem) było akurat odwrotnie. System był projektowany jako przeznaczony dla wielu użytkowników (komputery o wystarczającej mocy obliczeniowej były dość drogie) - a to wymuszało powołanie administratora. Założono, że jest to fachowiec dysponujący na tyle sporą wiedzą, że można my nadać uprawnienia zgodne z dwoma regułami:

1. ROOT ma zawsze rację i wie co robi.
2. W przypadkach wątpliwych - patrz 1.

I znów - nawet w popularnych "domowych" dystrybucjach Linuksa ta filozofia jest widoczna do dzisiaj. Tak duże uprawnienia (nieograniczony dostęp i możliwość modyfikacji każdego pliku) administratora mają jednak określone konsekwencje. Administrator (lub ktoś inny, który uzyskał jego uprawnienia) może bardzo łatwo zniszczyć system, który po prostu "nie broni się" przed swym administratorem (co często dość skutecznie robią Windowsy). Plusem są za to praktycznie nieograniczone możliwości konfiguracyjne.

I właśnie to jest główny powód, dla którego MS Windows "zakupione" bez dodatkowego kontraktu wsparcia technicznego najprawdopodobniej będą działać (przynajmniej w zakresie podstawowej funkcjonalności) o tyle Linux bez zapewnionego wsparcia technicznego będzie stwarzał pewnie więcej problemów. Jeśli jednak mu zapewnimy wsparcie techniczne na przyzwoitym poziomie okazuje się często systemem nawet bardziej przyjaznym dla użytkowników niż popularne MS Windows!

Niestety - wielu decydentów stwierdza - "przecież LINUX jest darmowy!". Dlaczego mamy płacić za jakieś "wsparcie techniczne"? Jak za darmo - to za darmo! Na nieśmiałe próby wyjaśnienia, jak jest naprawdę reagują wytrychami typu: "ja nie chcę wchodzić w szczegóły...".
Do tego dochodzi propaganda "Na MS Windows i MS Office zna się każdy". Wystarczy przeglądnąć przysyłane do firmy CV. Ale jak "biegła" jest to znajomość? Z tym jest naprawdę bardzo różnie.

A niestety duży system komputerowy to nie jeden PC w domu - i przy kilkuset maszynach w MS Windows w sieci okazuje się, że jednak wsparcie techniczne jest potrzebne i tak czy owak trzeba zatrudnić administratorów...

W efekcie i tak musimy sobie zafundować wsparcie techniczne (własne lub zewnętrzne), które rzekomo otrzymaliśmy "zakupując" oprogramowanie komercyjne czyli opłacając prawo do jego użytkowania.

W tym miejscu dochodzimy do kolejnego oblicza "GĘBY", którą przyprawia się Linuksowi i całemu Wolnemu Oprogramowaniu:

"*Owszem, być może LINUX to dobry system, ale aby z niego korzystać trzeba być specjalistą! A z typowych programów komercyjnych może bez problemów korzystać każdy*"

Otóż jest to całkowita nieprawda!

Współczesny użytkownik komputera nie korzysta bezpośrednio z systemu operacyjnego. Wykorzystuje oprogramowanie użytkowe - uniwersalne (systemy obsługi biura, procesory tekstu i grafiki, przeglądarki sieciowe, pocztę elektroniczną itp.) lub programy specjalistyczne - najczęściej powiązane z obsługą baz danych (systemy księgowe, przygotowania produkcji, programy inżynierskie itp.).
Jeśli stanowisko pracy użytkownika umożliwia sprawne korzystanie z potrzebnych mu do pracy programów, to jest to całkowicie obojętne jaki system operacyjny wykorzystuje zarówno samo stanowisko pracy, jak i pod jakim systemem i na jakim komputerze uruchamiany jest program, z którego aktualnie korzysta.

Najistotniejszy dla użytkownika jest jego interfejs wykorzystywany do uruchamiania niezbędnych programów. Jeżeli np. "Pani od ZUSu" potrzebny jest na jej stanowisku słynny program "Płatnik" to na jej pulpicie powinna być dostępna odpowiednia ikonka, której "kliknięcie" spowoduje, że na ekranie pojawi się okno z tym programem. Tylko tyle - i aż tyle!

Z punktu widzenia "Pani od ZUSu" jest absolutnie obojętne, czy "Płatnik" działa lokalnie na jej komputerze, czy też jest dostępny w sieci w systemie terminalowym czy też klient-serwer. Ma on po prostu działać, zapewniać sprawne korzystanie z danych, umożliwiać tworzenie różnych deklaracji oraz wysyłać je gdzie trzeba...

A sposób obsługi współczesnego stanowiska komputerowego jest już właściwie całkowicie zestandaryzowany. Różni dostawcy systemów operacyjnych zostali de-facto zmuszeni do stosowania "pulpitów" z ikonkami, przycisku uruchamiającego rozwijalny system Pop-Up menu (słynny przycisk "Start", który należy nacisnąć aby wyłączyć komputer), paska zadań itp. Zasadnicza struktura interfejsu użytkownika właściwie nie ulega zmianom od prawie 20 lat - jest on tylko uatrakcyjniany graficznie oraz wprowadzane są różnego rodzaju średnio przydatne gadżety, którymi entuzjazmują się zwłaszcza "małolaty" (a to bardzo ważna grupa docelowa dla przemysłu komputerowego!).

Podstawowe funkcje interfejsu użytkownika (uruchamianie programów i zarządzanie oknami) jednak się nie zmieniają - podobnie, jak od wielu lat nie zmienia się zasadniczo interfejs użytkownika samochodu osobowego i wprowadzenie GPS nie spowodowało przynajmniej jak na razie usunięcia kierownicy.

Ale przyprawiona "GĘBA" działa na tyle skutecznie, że wielu moich studentów widząc na ekranie pulpit systemu LINUX pyta - "Jakie ma Pan ładne Windowsy - jak sobie można takie zrobić?". I w wielu przypadkach są to studenci wydziałów politechnicznych.

W firmach, które traktują swoje instalacje komputerowe profesjonalnie niezależnie od tego, jakie rozwiązania stosują, użytkownicy nie mają praw do konfigurowania swoich komputerów (lub prawa te są mocno ograniczone) oraz do instalowania jakiegokolwiek oprogramowania we własnym zakresie. Za utrzymanie sprawnego działania infrastruktury komputerowej odpowiedzialny jest dział IT.

Jest to konieczne z wielu względów - ale przede wszystkim chodzi o zapewnienie ciągłości działania organizacji, bezpieczeństwa danych (w tym również wymaganego ustawowo - na przykład danych osobowych), zabezpieczenia kopii awaryjnych i innych działań niezbędnych dla minimalizacji skutkom awarii lub zdarzeń losowych itp.

Zadania te są również niezależne od rodzaju wykorzystywanego systemu oraz oprogramowania. Zaryzykuję nawet stwierdzenie, że są one łatwiejsze, bardziej efektywne i tańsze w systemach opartych o Wolne Oprogramowanie niż w wykorzystujących głównie oprogramowanie komercyjne typu COTS. Jest to prosta konsekwencja opisanej w pierwszej części niniejszego opracowania ewolucji tych rozwiązań - systemy klasy UNIX, z których narodziło się Wolne

Oprogramowanie od samego początku były bowiem projektowane do pracy z dużą liczbą użytkowników równocześnie, a nie dopiero później dostosowywane do takiej pracy.

Po tym dość przydługim wstępie chciałbym przejść do omówienia konkretnej metodyki przygotowywania projektu systemu informatycznego. Metodyka ta jest oparta o prace związane z zarządzaniem projektami obronnymi w USA (Cyrus H. Azani i inni) i została sprawdzona także w inżynierii oprogramowania.
Wykorzystuje ona koncepcję systemu otwartego (Open System). Pojęcia Open System nie należy mylić z Open Source, ponieważ Open System może zawierać zarówno oprogramowanie dostępne na zasadach komercyjnych, jak i oprogramowanie, które spełnia wszystkie wolności "stalmanowskie" i co za tym idzie należy do kategorii Wolnego Oprogramowania.

Zadaniem Open System jest przede wszystkim spełnienie wymagań użytkownika przy zachowaniu kilku dodatkowych atrybutów:

- możliwości rozwoju systemu w miarę pojawiania się nowych możliwości technologicznych i potrzeb użytkownika - zarówno planowanych, jak takich, których nie można było przewidzieć podczas projektowania i realizacji systemu.
- zapewnienie możliwości współpracy systemu z otoczeniem - zarówno obecnie, jak i w dającej przewidzieć się przyszłości.
- gwarancję możliwości wyboru technologii i rozwiązań najkorzystniejszych dla użytkownika bez ogranicza tego wyboru do grupy technologii i rozwiązań zarówno podczas realizacji systemu, jak i w przyszłości.
- minimalizację kosztów utrzymania systemu w stanie odpowiadającym poziomowi dostępnych rozwiązań technologicznych przez możliwie najdłuższy czas eksploatacji.

W moim przekonaniu zapoznanie PT Czytelników z zasadami projektowania takiego systemu i co za tym idzie profesjonalnego przygotowania Specyfikacji Istotnych Warunków Zamówienia (słynny SIWZ) pomoże nam wszystkim uzyskać rzeczywisty sukces w realizowanych projektach informatycznych - a nie zamieniać je w "Marsz ku klęsce" (świetna książka Edwarda Yourdona), co się niestety dość często zdarza - i to nie tylko w Polsce...

Określ swoje potrzeby

Każda organizacja - niezależnie od jej charakteru wykorzystuje (lub przynajmniej powinna wykorzystywać) swój system informatyczny do wspomagana lub wręcz realizacji procesów, które są celem jej działania.
Mogą to być procesy biznesowe (w przedsiębiorstwie), administracyjne (w urzędzie), edukacyjne (w szkole), informacyjne i służące rozrywce (portale, telewizja itp.) itd.

Podejście procesowe jest powszechnie stosowane w zarządzaniu przez jakość - w szczególności przez popularne normy ISO rodziny 9000 oraz coraz częściej wdrażane Systemy Zarządzania Bezpieczeństwem Informacji (SZBI) zgodne z normami rodziny ISO/IEC 27000.

Zastosowanie podejścia procesowego jest także bardzo pomocne przy projektowaniu lub modernizowaniu systemu informatycznego w każdej organizacji. Dzięki analizie procesów można bowiem precyzyjnie określić wymagania, jakie muszą spełniać poszczególne stanowiska pracy w systemie oraz jakie zasoby powinny być im udostępniane. Możliwie precyzyjne określenie rzeczywistych potrzeb znacznie ułatwia także opracowywanie Polityki Bezpieczeństwa Systemu - a w konsekwencji także wdrożenie Systemu Zarządzania Bezpieczeństwem Informacji (SZBI) a w perspektywie także uzyskania odpowiednich Certyfikatów - np. ISO 27001.

Należy zdecydowanie podkreślić, że na etapie określania potrzeb nie należy w żadnym przypadku dokonywać wyboru środków technicznych, które te potrzeby mają zaspokoić. Dotyczy to zarówno sprzętu, jak i oprogramowania.
Potrzeby powinny odpowiadać zadaniom, które realizowane są na danym stanowisku - dla stanowiska fakturowania będzie to dostęp do odpowiednich modułów programu FK, możliwość importu danych z bazy klientów, dla stanowiska sekretarskiego istotna będzie możliwość wykorzystywania procesorów tekstu, marketingu - systemu klasy CRM, produkcji - TPP itd.
Bezwzględnie należy się wystrzegać na tym etapie uwzględniania tak zwanych "ogólnych życzeń" typu: "ja koniecznie potrzebuję w moim komputerze nagrywarki DVD". Potrzeby tego typu będą uwzględniane w następnych etapach - i jedynie w przypadku, gdy ich spełnienie jest niezbędne dla realizacji procesów prowadzonych w organizacji.

Określane na tym etapie potrzeby nie powinny być przeszacowywane (określane "z zapasem") - powinno się uwzględniać jedynie aktualne i rzeczywiste wymagania, które powinien spełnić system informatyczny. Jednym z następnych etapów będzie bowiem przygotowanie systemu na zmiany, które mogą wystąpić zarówno w wyniku rozwoju technologii biznesowej oraz zmianach w realizowanych przez organizację procesach.

Dokonaj podziału systemu na moduły funkcjonalne

Dysponując arkuszem potrzeb możemy przystąpić do podziału systemu na moduły funkcjonalne. W przypadku systemu, który jest modernizowany należy starać się również określić, jakie jego części mogą być potraktowane jako oddzielne moduły - np. serwer bazy danych, serwer plików i aplikacji użytkowników, obsługa połączeń zewnętrznych itp.

Głównym celem tego etapu jest takie zaprojektowanie systemu (lub zmian w nim dokonywanych), aby była możliwa wymiana poszczególnych jego modułów lub wprowadzanie modułów niezbędnych do wspomagania nowych zadań stawianych organizacji.

Takie podejście powinno umożliwić wprowadzanie modernizacji systemu na bieżąco w miarę pojawiających się potrzeb lub nowych rozwiązań technologicznych.

Również na tym etapie nie dokonujemy doboru konkretnych produktów lecz jedynie technologii, które są aktualnie możliwe do wykorzystania - oczywiście uwzględniając już eksploatowane elementy systemu.

Na tym etapie najistotniejsze jest określenie funkcji, które mają realizować poszczególne moduły oraz interfejsów pomiędzy poszczególnymi modułami. Kluczowe interfejsy w systemie powinny wykorzystywać jedynie standardy otwarte (zgodnie z definicją stosowaną przez Unię Europejską 2004):

Standard, aby mógł być nazwany otwartym, spełniać musi łącznie cztery warunki:

- jest przyjęty i zarządzany przez niedochodową organizację, a jego rozwój odbywa się w drodze otwartego procesu podejmowania decyzji (konsensusu, większości głosów, itp.), w którym mogą uczestniczyć wszyscy zainteresowani,
- jest opublikowany, a jego specyfikacja jest dostępna dla wszystkich zainteresowanych bezpłatnie lub po kosztach sporządzenia kopii oraz możliwa dla wszystkich do kopiowania, dystrybuowania i używania również bezpłatnie lub po kosztach operacyjnych,
- wszelkie związane z nim prawa autorskie, patenty i inna własność przemysłowa są nieodwołalnie udostępnione bez opłat
- nie ma żadnych ograniczeń w jego wykorzystaniu

Tylko takie podejście umożliwi bezproblemową wymianę modułów - a więc spełnienie warunku "design for change". Zabezpieczy nas to również przed ryzykiem uzależnienia się od jednego dostawcy (Vendor Lock-In), które szczegółowo opisałem w opracowaniu "Zjawisko Lock-In i związane z nim zagrożenia dla użytkownika".

Przestrzeganie zasady wykorzystywania jedynie otwartych standardów dla kluczowych interfejsów w systemie informatycznym daje możliwość łączenia w nim modułów różnych dostawców, ich poprawne współdziałanie, a także umożliwia łączenie w ramach jednego systemu różnych typów oprogramowania - Wolnego (Open Source), standardowego komercyjnego (COTS) oraz opracowywanego na indywidualne zamówienie. Dzięki temu możemy dokonać wyboru najkorzystniejszej oferty uwzględniając wszelkie jej aspekty - zarówno ekonomiczne, jak i techniczne.

W przypadku, gdy ze względów technicznych musimy zastosować rozwiązanie interfejsu, dla którego nie jest dostępny standard otwarty należy bezwarunkowo sporządzić i starannie przechowywać pełną dokumentację techniczną takiego interfejsu. Dokumentacja powinna umożliwiać opracowanie produktu w pełni obsługującego opisany przez nią interfejs przez osoby lub inne podmioty dysponujące odpowiednią wiedzą techniczną, a nie zaangażowane w realizację konkretnego projektu (warunek intersubiektywności i interkomunikowalności).

Po zakończeniu wprowadzania struktury modularnej, określeniu funkcjonalności poszczególnych modułów oraz wyboru standardów interfejsów z preferencją dla standardów otwartych możemy przejść do analizy rynkowej dostępności niezbędnych nam produktów i usług oraz analizy ryzyka związanego z wyborem określonych rozwiązań.

System zamknięty	***System Otwarty***
Wykorzystuje zazwyczaj zamknięte, niedostępne lub opisane w udostępnianych za znaczną opłatą specyfikacjach standardy interfejsów, formaty danych, protokoły oraz narzędzia programistyczne (np. języki).	Wykorzystuje popularne, publicznie dostępne i powszechnie zaakceptowane interfejsy, języki programowania, formaty danych i protokoły.
Dążenie do unikalności produktu oraz jego implementacji w celu uzyskania przewagi konkurencyjnej.	Rozwój w kierunku zapewnienia akceptacji poprzez powszechną aprobatę w istniejącym środowisku - wykorzystywanie powszechnie przyjętych konwencji, otwarte interfejsy oraz sposób zarządzania.
Najczęściej unikanie modularności - tendencja do tworzenia dużych zamkniętych produktów, których elementy nie mogą być odłączane lub wymieniane.	Powszechne stosowanie rozwiązań o dużym stopniu modularności oraz wymienności elementów.
Uzależnienie od dostawcy (producenta) oraz wykorzystywanej technologii.	Niezależność od dostawcy (producenta) oraz wykorzystywanej technologii.
Trudności w realizacji interoperacyjności oraz ograniczone możliwości współpracy z produktami innych dostawców.	Wysoki stopień interoperacyjności, dostępność rozwiązań umożliwiających korzystanie z produktów innych dostawców.
Dostępność uaktualnień na zasadach określonych przez dostawcę - zazwyczaj odpłatnie.	Uaktualnienia powszechnie dostępne.
Krótki "czas życia" rozwiązania, ograniczone możliwości prowadzenia ciągłej modernizacji systemu.	Długi czas życia. Możliwość prowadzenia ciągłej modernizacji systemu poprzez wymianę modułów.
Wysokie opłaty za prawa do wykorzystywania rozwiązania.	Niskie opłaty związane z wykorzystywaniem rozwiązania lub wręcz ich brak.
Wsparcie techniczne realizowane poprzez sieć producenta (ograniczony dostęp do narzędzi, procedur diagnostycznych itp.).	Możliwość realizacji usług technicznych przez podmioty trzecie ze względu na powszechny dostęp do narzędzi oraz otwartość kodu oprogramowania.
Niewielkie możliwości adaptacji do indywidualnych potrzeb.	Bardzo duże możliwości adaptacji do indywidualnych potrzeb.
Bezpieczeństwo oparte głównie na zaufaniu do dostawcy / producenta. Ograniczone możliwości niezależnych testów.	Bezpieczeństwo oparte na otwartości oraz powszechnej akceptacji zastosowanych mechanizmów i implementacji. Możliwość pełnego audytu mechanizmów bezpieczeństwa.
Możliwość przekazania czynności administracyjnych użytkownikom.	Przekazywanie czynności administracyjnych użytkownikom wiąże się z dużym ryzykiem awarii systemu.
Ograniczenie konkurencyjności.	Szerokie możliwości konkurowania dostawców.

UDZIELANIE ZAMÓWIEŃ PUBLICZNYCH NA SYSTEMY INFORMATYCZNE

REKOMENDACJE

Krótkie omówienie 33 stronicowego dokumentu Urzędu Zamówień Publicznych

O różnych nieprawidłowościach i "minach" umieszczanych w tak zwanych SIWZ przez zamawiających od dawna opowiadano sobie dowcipy. Trzeba przyznać, że firmy przystępujące do przetargów też podejmowały co najmniej dziwne działania - np. znany mi jest protest przeciwko wpisaniu do specyfikacji słowa LINUX. Protestujący stwierdził, że w ten sposób nastąpiło zabronione jednoznaczne wskazanie producenta...

Pomimo pozornie obiektywnych procedur przetargowych i deklarowanego zachowania warunków otwartej konkurencji pomiędzy potencjalnymi dostawcami, jakoś tak dziwnie się składało, że zamawiający bardzo często wykazywali zadziwiającą lojalność wobec producentów i dostawców rozwiązań informatycznych. Nawet jeśli wszyscy wiedzieli, że na rynku jest dostępne rozwiązanie co najmniej równoważne technicznie oferowane w atrakcyjnej cenie, to bardzo często zamawiający dokonywali zakupu "z wolnej ręki" lub wprowadzali do SIWZ zapisy, które w oczywisty (choć zakamuflowany) sposób preferowały jedno rozwiązanie - tłumacząc to pewnością, że wybór jednego dostawcy gwarantuje właściwą współpracę elementów systemu, a tym samym jego sprawne wdrożenie i bezproblemową eksploatację.

W ten sposób zamawiający bardzo szybko wpadał w zastawioną pułapkę uzależnienia od dostawcy (Vendor Lock-In) i z każdym kolejnym krokiem stawał się coraz bardziej ubezwłasnowolniony - aż wreszcie osoby odpowiedzialne za procedury przetargowe stwierdzały, że najwygodniej jest po prostu udzielać dalsze zamówienia "z wolnej ręki", gdyż i tak produkty i usługi może dostarczyć tylko jeden podmiot... I liczba takich zamówień zaczęła niepokojąco rosnąć.

Oczywiście spowodowało to ograniczenie konkurencyjności i umożliwiło "preferowanym dostawcom" podniesienie cen.

A jednak w końcu zauważono problem i Urząd Zamówień Publicznych (UZP) wydał dokument zatytułowany "Udzielanie zamówień publicznych na systemy informatyczne - Rekomendacje".

Dokument ten jest udostępniony na stronie UZP i można mieć nadzieję, że jego publikacja będzie co najmniej powodem do refleksji podmiotów dokonujących zakupów przetargowych na ogólnie rozumianym rynku IT.

Autorzy dokumentu podkreślają we wstępie, że głównym powodem przygotowania "Rekomendacji" jest rosnąca liczba zamówień na systemy informatyczne w trybie "z wolnej ręki", jednakże zaraz po tym znajdujemy stwierdzenie, że

"***przyczyną tego zjawiska jest powstanie uzależnienia zamawiającego od pierwotnego wykonawcy systemu lub producenta sprzętu lub oprogramowania gotowego uniemożliwiające nabycie niezbędnych usług lub dostaw w trybach konkurencyjnych. Uzależnienie to jest w dużej mierze konsekwencją niewłaściwego przygotowania postępowania i udzielenia zamówienia publicznego***"

A więc wreszcie zwrócono uwagę na efekt uzależnienia od dostawcy, znany na całym Świecie po nazwą "Vendor Lock-IN" - polecam Państwu moje opracowanie "Zjawisko Lock-In i związane z nim zagrożenia dla użytkownika" będące pokłosiem Konferencji "Locked-In European Parliament, Bruksela, kwiecień 2008, w której miałem okazję uczestniczyć.

Lepiej późno, niż wcale - warto więc zajrzeć do tych zawartych na 33 stronach "Rekomendacji", ponieważ zawarte w nich tezy nie tylko wynikają z analizy polskich "realiów przetargowych", lecz zawierają także odniesienia do przepisów i orzecznictwa UE.

Każda z grup rekomendacji poprzedzona jest analizą błędów, popełnianych przez zamawiających podczas prowadzenia procedury przetargowej. Autorzy "Rekomendacji" wskazują, że to właśnie błędy zamawiających są główną przyczyną powstawania uzależnienia od dostawcy - i w konsekwencji ograniczenia konkurencji procedur przetargowych.

Oczywiście nie mam ambicji przeprowadzenia na kilku stronach analizy zaleceń technicznych i prawnych zawartych w "Rekomendacjach", jednak sam fakt ich ukazania się, a co ważniejsze ich treść należy powitać z radością. Oto ich króciutkie streszczenia:

Rekomendacja 1 - planowanie rozwoju

Zaleca zamawiającym opracowanie Polityki Rozwoju Systemów Informatycznych. Zgodnie z rekomendacją powinna ona uwzględniać możliwość rozwoju systemu w warunkach postępu technologicznego ("Design for Change"), w jasny sposób rozwiązywać zagadnienia związane z prawami autorskimi, dokumentacją systemu i dostępem do kodu oprogramowania.

Uwzględnienie wymagań tej rekomendacji powinno pozwolić zamawiającemu na zachowanie warunków konkurencyjności we wszystkich etapach powstawania, eksploatacji i modernizacji systemu - a więc likwidować uzależnienie od pojedynczego dostawcy rozwiązania lub produktu.

Rekomendacja 2 - standaryzacja procedur

Poświęcono ją szeroko pojętemu zarządzaniu systemem podkreślając, ze powinno być one oparte o istniejące standardy (ITIL, ISO20000, COBIT...). Rekomendacja zaleca, aby procesy związane z zarządzaniem zmianami (np.

prowadzenie prac rozwojowych) były realizowane oddzielnie od procesów związanych z bieżącym utrzymaniem systemu. Usługi związane z bieżąco eksploatacją systemu i zarządzaniem jego rozwojem powinny być uwzględnione w zamówieniu podstawowym lub zamawiane w oddzielnym postępowaniu z zachowaniem konkurencji.

Rekomendacja 3 - serwis

Zwraca uwagę na celowość zamawiania usług serwisowych już podczas postępowania podstawowego na podstawie jasno określonych kontraktów serwisowych.

Rekomendacja 4 - ocena jakości

Obiektywizuje procesy związane z oceną jakości systemu już w toku realizacji zamówienia. Zaleca wprowadzenie szeregu testów - modułowych, interoperacyjności, akceptacji itp. Zamówienie na bieżącą analizę jakości realizowanego systemu oraz samego procesu jego realizacji powinien oczywiście otrzymać niezależny podmiot.

Rekomendacja 5 - moduły i interfejsy

W moim przekonaniu ma ona absolutnie kluczowe znaczenie. Zaleca ona podział systemu na współpracujące pomiędzy sobą moduły funkcjonalne połączone za pomocą udokumentowanych interfejsów.

Stosowanie się do tego zalecenia umożliwia kontynuowanie rozwoju systemu z zastosowaniem aktualnie dostępnych technologii, ewolucyjny rozwój systemu poprzez wymianę poszczególnych modułów oraz skutecznie zapobiega powstaniu uzależnienia zamawiającego od dostawcy.

Rekomendacja zwraca także uwagę na otwartość, interoperacyjność oraz jawność definicji zaimplementowanych protokołów komunikacyjnych.

Rekomendacja 6 - zarządzanie wiedzą o systemie informatycznych

W myśl zaleceń UZP do budowy systemów należy stosować udokumentowane technologie informatyczne. Zapewnia to możliwość kontynuowania rozwoju systemu z zachowaniem konkurencyjności. W takim przypadku należy zachować możliwość przejęcia przez nowego wykonawcę wiedzy o konstrukcji systemu.

Rekomendacja 7 - prawa autorskie

Zarówno SIWZ, jak i zawierana umowa powinny zawierać klauzule gwarantujące sprawną współpracę wykonawcy z dostawcami innych, niezależnych (luźno powiązanych) podsystemów.

SIWZ oraz Umowa powinny regulować kwestie związane z Majątkowymi Prawami Autorskimi do zakupywanego systemu, jego modułów i jednoznacznie określać pola eksploatacji i warunki ich udzielenia.

UZP zaleca, aby w stosunku do nowoutworzonego systemu przeniesienie praw powinno dotyczyć również kodów źródłowych oraz jego swobodnej modyfikacji. Jednocześnie zamawiający powinien zobowiązać w umowie wykonawcę do wydania kodów źródłowych oraz dokumentacji technicznej systemu najlepiej z chwilą jego odbioru przez zamawiającego.

W zawieranej umowie powinny być uregulowane sprawy związane z przeniesieniem licencji lub sublicencji programów komputerowych niezbędnych

do prawidłowego funkcjonowania systemu. Warunki te nie powinny stanowić barier do dokonania modyfikacji lub rozbudowy oprogramowania.

Ukazanie się "Rekomendacji" witam z prawdziwą radością. Podczas prowadzonych zajęć na Uczelniach, wykładach na Konferencjach i publikowanych materiałach wielokrotnie apelowałem o stosowanie struktury modularnej, przygotowania systemu do nieuniknionych zmian, zwracanie szczególnej uwagi na kluczowe interfejsy oraz staranne ich dokumentowanie. Projekty realizowane według powyższych zasad kończyły się sukcesami - i wiele systemów, które powstały w ich wyniku pracuje bez problemów przez wiele lat. Wielu "niekwestionowanych liderów rynku" zarówno polskiego, jak i światowego w tym czasie po prostu zniknęło. Systemy informatyczne zbudowane według strategii otwartego systemu modularnego przetrwały próbę czasu.
Pozostaje mieć tylko nadzieję, że "Rekomendacje" będą powszechnie stosowane przez zamawiających.

Tomasz Barbaszewski

Dodatkowe informacje:

http://openparliament.eu/content/conference-at-european-parliament

http://www.uzp.gov.pl/aktualnosci/akt-rekomendacje-uzp-syst-inf-25-08-09

http://www.aba.krakow.pl/Download/Artykuly/locked_in.pdf