

Tomasz Barbaszewski

Uzależnienie od dostawcy (Vendor Lock-In) - czym grozi i jak mu zapobiec?



Steve Jobs jako „Szatan-Kusiciel” w twórczej przeróbce obrazu „Upadek” - Hugo van der Goes, Kunsthistorisches Museum, Wiedeń – modyfikacja „eater”

Celowo wybrano jako „vendra” firmę Apple, aby nie podgrzewać w Polsce zbędnych emocji związanych ze znanym potentatem działającym na rynku oprogramowania...

Co to jest efekt „Locked-In” ?

W działalności komercyjnej bardzo istotne jest skuteczne dotarcie do Klienta. Na wszystkich kursach „Marketingu i Zarządzania” naucza się, że „Best Customer is Returning Customer”, ponieważ koszt dotarcia do już raz obsłużonego klienta jest najmniejszy. Związaniu klienta z firmą służą różne programy lojalnościowe, punkty „za benzynę” itp., itd...

Oczywiście nie ma w tym nic złego, jeśli stosowane metody nie ograniczają swobody klienta. Jednak „zawodowi macherzy od losu” nie byliby sobą, gdyby „kreatywnie” nie starali się „wzmocnić oddziaływania na klienta”. Wymyślane przez nich mechanizmy są różne – najlepiej zilustrować choć jeden z nich na przykładzie:

Zakupiłem kiedyś (podczas stypendium za granicą) jeden z modeli aparatu „Polaroid”. Był on prosty i stosunkowo niedrogi (ok. 50 USD). Ładunek materiału światłoczułego do fotografii natychmiastowej (10 zdjęć kolorowych) kosztował 9 USD. Oczywiście musiałem go kupować od jedynie od firmy Polaroid.

W bardzo krótkim czasie za kasety zapłaciłem znacznie więcej, niż za aparat!

Dodatkowo po ok. 1,5 roku firma wprowadziła „nowy, lepszy...” i droższy model „wykorzystujący unikalną technologię opracowaną przez naszych inżynierów”. Nawet łaskawie zaproponowano mi 10% upustu na zakup nowego aparatu równocześnie znacznie ograniczając dostępność do starych kaset z materiałem światłoczułym. Okropnie się zdenerwowałem i nie kupiłem. Stary aparat trzymam w domu jako groźne „memento”.

Co najweselsze – mam w domu aparat „Rolleicord” z lat 1920-1930. I filmy do niego są nadal dostępne!!!



„Vendor Locked-In” - POLAROID

Wszystko od jednego „zaufanego” dostawcy!!!
„nie będziesz miał kłopotu, że coś nie pasuje!”



„No Vendor Locked-In” - RolleiCord

Swobodny wybór dostawców – tak krytykowany a pomimo swobody w ramach standardu działa!

Jest to dość typowy przykład efektu „Vendor Locked-In”. Firma Polaroid skorzystała ze swej monopolistycznej pozycji na rynku fotografii natychmiastowej, której zresztą zazdrośnie broniła głównie metodami prawnymi, wstrzymując np. produkcję „Kodak Instant Camera” (wygrała proces o naruszenie praw z tym potentatem). Nie uchroniło to jej jednak przed bankrutem...

Może po przedstawieniu tego przykładu odpowiednio potraktujecie Państwo następujące opowieści „przedstawicieli handlowych” -



„Nasza firma, o ugruntowanej pozycji rynkowej, światowy lider technologiczny dostarczy Państwu kompletne rozwiązanie – co gwarantuje w 100% właściwą współpracę wszystkich jego elementów. Nie będziecie Państwo mieli kłopotów ze zwalaniem winy przez dostawców – jeden na drugiego. Za wszystko odpowiadamy MY – Światowy Lider!!!”.

**Zatrzymajcie się Państwo, o ile jeszcze możecie!
Zmierzacie wprost w objęcia PEŁNEGO
uzależnienia się od dostawcy – on już będzie
wiedział jak to wykorzystać!**

Aby nie było żadnych nieporozumień – efekt „Vendor Lock-In” może wystąpić (choć ze znacznie mniejszym prawdopodobieństwem) także dla Wolnego Oprogramowania!!! Znów posłużę się prawdziwym przykładem z mojej praktyki:

W jednym z większych banków pracował znakomity specjalista „od Linuksa” - pan Jacek (imię celowo zmieniłem). Opracował on wiele systemów automatycznie wspomagających pracę banku. Oczywiście całość oprogramowania była dostępna z kodami źródłowymi.

Pewnego dnia przyjechał do mnie szef Jacka i z troską spytał: „Panie Tomku, co będzie, jak przyjdzie mi „zbackupować” Jacka? Te programy są znakomite, w kodach jest nawet sporo komentarzy, ale ile zajmie analiza działania tych programów osobie, która ich nie pisała?”

Nie potrafiłem znaleźć konkretnej odpowiedzi...

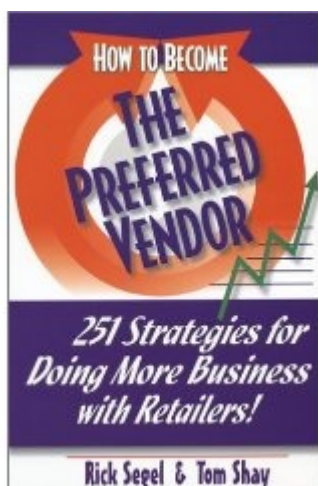
Tym niemniej pocieszyłem Pana Szefa, że jego sytuacja jest i tak lepsza, niż w przypadku eksploatowanego przez bank systemu Finansowo-Księgowego, który został dostarczony „As Is” przez „czołowego dostawcę systemów bankowych w Polsce” (oczywiście bez kodu źródłowego).

Wpadnięcie w pułapkę „Vendor Lock-In” jest bardzo groźne dla użytkownika. Może nawet dojść wręcz do dobrowolnej akceptacji takiej sytuacji (sic!), gdy użytkownik w pełni akceptuje to uzależnienie brnąc konsekwentnie w sprytnie zastawioną pułapkę. Nieodparcie kojarzy mi się to z działaniem sekt, których członkowie dobrowolnie poddają się woli GURU.

Znów podkreślam, że takie zjawiska dają się zauważyć zarówno wśród „wyznawców” Wolnego Oprogramowania (FOSS), jak i Oprogramowania Komercyjnego (COTS). Stopień wewnętrznego pogodzenia się z sytuacją może zajść bardzo daleko – np. jedna z osób odpowiedzialnych za wdrażanie oprogramowania w dużej instytucji sektora publicznego stwierdziła na otwartym spotkaniu bez żenady:

„Nie mogę wymagać żadnych zmian od wielkich dostawców oprogramowania „z półki” - zarówno w ich właściwościach, jak i w umowach licencyjnych. Świadomie też nie korzystamy ze wsparcia technicznego takich firm, bo się na ich produktach znamy i możemy to robić we własnym zakresie. Kupujemy od nich standardowe licencje na zaproponowanych przez nich warunkach.

Natomiast w stosunku do firm krajowych, które dostarczają nam własne oprogramowanie stawiamy o wiele wyższe wymagania i wprowadzamy do umów odpowiadające nam klauzule – np. prawo do powielania instalacji i dostępu do kodu źródłowego.”



Chwileczkę, a gdzie gwarantowana ustawowo równość podmiotów???

Ja dotychczas myślałem, że to dostawca ma się dostosować do potrzeb klienta, a nie klient do wymagań dostawcy (zwłaszcza, jeśli klient wydaje pieniądze podatnika, a więc nie swoje własne!).

Ale cóż, widocznie się myliłem, i tak jak u Orwella - „wszyscy dostawcy są równi, ale wśród nich są RÓWNIEJSI.”

Czym konkretnie grozi uzależnienie od dostawcy:

- **dyktatem warunków z jego strony przy zakupie.**

Zawierając umowę o dostawę rozwiązania Zamawiający nie ma żadnej możliwości negocjacji jej warunków, lub możliwości te są bardzo ograniczone (np. do ceny oferowanej przez pośrednika). Znow konkretny przykład:

Znakomita większość oprogramowania COTS (Commercial of the Shelf) jest sprzedawana w systemie sprzedaży pośredniej trójstopniowej PRODUCENT -> DYSTRYBUTOR -> RESELLER (DILER) -> klient. Wielu producentów nadaje sprzedawcom końcowym ładnie brzmiące tytuły (najczęściej związane z drogocennymi kruszcami) np. „Platinum Partner” albo „Diamond Award”. Tak naprawdę związane są one jedynie z wartością sprzedaży!

Ponieważ klient kupuje u resellera, to najczęściej umowa licencyjna, warunki wsparcia technicznego itp. są przyjmowane standardowo. Polscy klienci (zwłaszcza z sektora publicznego) przyzwyczaili się, że mogą jedynie negocjować cenę (a i to w bardzo ograniczonym zakresie – kupujesz w segmencie A wg. ścieżki A12 i tyle). Wiele się mówi o „wartości dodanej” - ale tak naprawdę to nikt nie zwraca na nią uwagi i nie negocjuje np. warunków wsparcia technicznego z „Value Added Resellerem” i kupuje oprogramowanie „As Is” wstawiając słynne standardowe wymaganie w SIWZ (kryteria oceny oferty: 100% cena). Tak lubią prawnicy z „Działów Zamówień Publicznych” zgodnie z zasadą najmniejszego działania.

- **pełnym uzależnieniem od dostawcy w okresie eksploatacji systemu.**

Argumenty są na ogół bardzo proste – „wydaliśmy już 100.000,00 zł, 500.000,00 zł, czy też 2 mln zł i nie jest uzasadnione ekonomicznie wycofywanie się z tego rozwiązania”. Brnie się więc dalej uzasadniając zaciskanie się pętli „Vendor Lock-In”, kupuje się nowe licencje, uaktualnienia (bo przecież szkoda już wydanych środków) itd.

Występuje tu oczywiste pomylenie skutków i przyczyn. Kompetentni lekarze dobrze wiedzą – **jeśli grozi rak, to należy ciąć!** I to jak najwcześniej! Leczenie zachowawcze doprowadzi w takim przypadku do rychłej śmierci pacjenta. W Internecie krąży taki dowcip:

Pytanie: „To jak długo można wdrażać ten nasz nowy system R5 u Klienta?”

Odpowiedź: „Tak długo, dopóki Klient ma jeszcze jakieś pieniądze!!!”.

- **„zarażaniem” innych eksploatowanych programów.**

Jeśli w naszej instalacji wystąpi przypadek „Locked-In” będzie miał on zapewne tendencję do rozprzestrzeniania się (zjawisko to jest do przerzutów nowotworów). Załóżmy, że efekt wystąpił dla oprogramowania biurowego. Ale przecież w urzędzie czy firmie nie tylko pisze się pisma. Wielokrotnie zachodzi potrzeba pobrania danych z naszej bazy lub wypełnienia określonych formularzy ze sprawdzaniem poprawności (lub kolizji) wprowadzanych pól. Najlepiej, aby proces ten został zautomatyzowany.

I co się dzieje – oczywiście stworzymy odpowiednie instrukcje złożone (najczęściej makra w naszym systemie biurowym. I tu wpadliśmy – bo na ogół są one nie w pełni nieprzenośne (np. standardowy problem MS Office – OpenOffice). Tak więc bardzo szybko wpadniemy w uzależnienie i będziemy argumentować z pełnym wewnętrznym przekonaniem, że to była nasza przemyślana decyzja... A przejście na inny system makr będzie trudne mentalnie (bo nie technicznie). Komu się chce podejmować dodatkową pracę? Prościej kupić kolejną licencję... I pętla się zaciska, bo nasz system baz danych (i middleware) zostały już skutecznie „zarażone” określonym rozwiązaniem. I im lepiej zautomatyzujemy nasz system – tym odwrót będzie trudniejszy.

Także przy zamawianiu nowych produktów pojawi się w naszych dokumentach SIWZ „całkowicie uzasadnione i oczywiste wymaganie” zgodności nowego rozwiązania z określonym – systemem operacyjnym, oprogramowaniem biurowym, silnikiem lub aplikacją bazodanową... **Cel osiągnięty – przerzuty już wystąpiły!!!**

Już choćby tylko te trzy wymienione powyżej zagrożenia uzasadniają przedsięwzięcie jak najdalej idących środków ostrożności w celu zapobieżenia wystąpieniu zjawiska „Vendor Lock-In”.

I należy spróbować odpowiedzieć sobie na proste pytanie: Jak to zrobić ???

Wbrew wielu obawom i powoływaniu się na „ogólną niemożność” nie jest to wcale takie trudne, choć oczywiście wymaga sporego zaangażowania emocjonalnego (bez czego każda działalność jest „nieuzasadniona i zbędna”) oraz niewielkiego (naprawdę!) wysiłku.

Najlepszą bronią jest wyeliminowanie możliwości „przerzutów choroby”, a to można osiągnąć stosując dwie podstawowe zasady:

1. Podzielić system na możliwie niewielkie moduły logiczne, które mogą być wymieniane bez konieczności modyfikacji innych elementów systemu,
2. Zadbać o dokładne zdefiniowanie komunikacji pomiędzy poszczególnymi modułami – czyli określenie i udokumentowanie właściwości ich interfejsów.

Oba wymagania zapewne spowodują zapewne silną kontr-reakcję ze strony dostawców wykorzystujących mechanizmy „Lock-In”:

Jak to!? Nasz system to zamknięta całość np. z przeglądarką WWW i odtwarzaczem multimedialnym. To w pełni zintegrowany, przyjazny dla użytkownika system!

Obrona jest nadzwyczaj prosta:

Nie należy się tym czczym gadaniem zupełnie przejmować – ale bezwarunkowo także nie należy wykorzystywać modułów, które nie mogą być niezależnie wymieniane!!! I naprawdę należy się przejść tym zaleceniem, zgodnie z ostrzeżeniem „Starego Górala”: „*jageś roz go dotknął – toś przepod!*”.

Drugi punkt jest pozornie nieco trudniejszy do spełnienia – choć widoczne są pozytywne zmiany wymuszane przez użytkowników. Właściwie jedynym (lecz za to w 100% skutecznym!) zabezpieczeniem przez „Vendor Lock-In” jest bezwzględne wymaganie dostarczenia opisu działania interfejsów, formatów wymienianych danych (należy z oczywistych powodów preferować standardy otwarte) oraz zestawu akceptowanych komend, sygnałów sterujących typu sekwencja ESC itp. I powinniśmy uzyskać formalne potwierdzenie poprawności i kompletności tych danych przez producenta / dostawcę.

Takie podejście likwiduje następujące problemy: „nasz program „X” w pełni poprawnie współpracuje z najpopularniejszym systemem biurowym (przeglądarką itp.). Nie przewidujemy obecnie obsługi innych systemów... Takie podejście nie powinno nas w żadnym przypadku zadowolić – grozi poważne uzależnienie!!! Bowiem co z tego, że współpracuje – ale JAK WSPÓŁPRACUJE? W jaki sposób wymienia dane? Jak są przekazywane rozkazy? Jeśli otrzymamy kompletny opis interfejsów programu – nie musimy się obawiać „Vendor Lock-In”. Konwersja dowolnych danych nie jest problemem dla współczesnego komputera. Zrobi to w locie i to tak szybko, że nikt nawet nie zauważy.

Ale aby utworzyć tablicę konwersji (czyli tak naprawdę słownik umożliwiający „dogadywanie się” programów posługujących się różnymi językami) trzeba wiedzieć, co mamy konwertować na co! Jeśli taką informację otrzymamy – to współpraca programu „X” z dowolnym systemem obsługi biura nie będzie problemem. Wystarczy nam zaprogramowanie odpowiedniego słownika.



To też można odczytać i zaprogramować!

Trudności możemy natomiast się spodziewać, jeśli dostawca odmówi nam dostarczenia kompletnego opisu interfejsów. Podawane będą zapewne różne argumenty. Najczęściej dostawcy starają się przywołać względy bezpieczeństwa czyli dalej króluje mit „Security by Obscurity” lub „tajemnicę handlową firmy” - a co to znowu za bzdura – na tej samej zasadzie zakres fal oraz rodzaj modulacji akceptowane przez odbiornik radiowy powinny stanowić „tajemnicę handlową firmy!”

Sprawdzając rzecz to absurdu – czy kupując opony zadowolilibyście się Państwo deklaracją sprzedawcy - „pasują, pasują na pewno!” i uznalibyście Państwo, że średnica i szerokość osadzenia, maksymalna prędkość i nośność oraz rodzaj bieżnika stanowią „tajemnicę handlową firmy”?

Tajemnica może dotyczyć jedynie szczegółów budowy opony (np. chemiczny skład mieszanki gumowej) – ale na pewno nie jej interfejsów!

Co więc uczynić, jeśli dostawca konsekwentnie odmawia dostarczenia takiej specyfikacji? Odpowiedź jest jednoznaczna – nie kupować i nie stosować (nawet za darmo), bo ryzyko wpadnięcia w „Lock-In” jest bardzo znaczne!

Przecież nikt rozsądny nie zakupi profesjonalnej maszyny produkcyjnej (a system informacyjny to linia technologiczna do przetwarzania informacji!) bez kompletnej Dokumentacji Techniczno-Ruchowej!

Pragnę jeszcze raz zwrócić Państwa uwagę – eliminowanie zjawiska „Vendor Lock-In” nie oznacza niczyjej dyskryminacji! Sprowadza jedynie relację Dostawca – Użytkownik do właściwej równowagi, która obecnie jest na tyle naruszona, że o dostęp do specyfikacji technicznej (sic!) formatu wymiany danych toczą się procesy sądowe!!! (Przykład: ZUS – Płatnik / Janosik). Nie dajmy się zwariować!

Jeśli mam ochotę wydać swoje własne pieniądze na produkt pozbawiony praktycznie gwarancji i rękojmi „**w maksymalnym zakresie dozwolonym przez właściwe prawo**” (cytat dosłowny!) – **ależ to cudo „nowomowy”!** i zapłacić jedynie za prawo do korzystania (i to bardzo ograniczonego) z tego oprogramowania i dodatkowo nie otrzymuję kompletnej specyfikacji formatów wymienianych danych to jest to moja sprawa – bo moje są też pieniądze i moja fanaberia.

Jeżeli jednak są to pieniądze publiczne (podatnika!), nad którymi opiekę mi powierzono, to sprawa jest całkiem inna. Moim zadaniem jest dopilnowanie, aby zakupiony system informacyjny dobrze służył jego użytkownikom i nadążał w sposób ciągły za rozwojem technologii światowej.

Konieczne jest więc dostosowanie systemu do zmian (Design for Change), których rodzaju dziś jeszcze nie znamy. Czy to w ogóle jest możliwe – oczywiście, tak. Pod warunkiem, że nie znajdziemy się w pułapce „Vendor Lock-In”!

Remedium - System Otwarty !

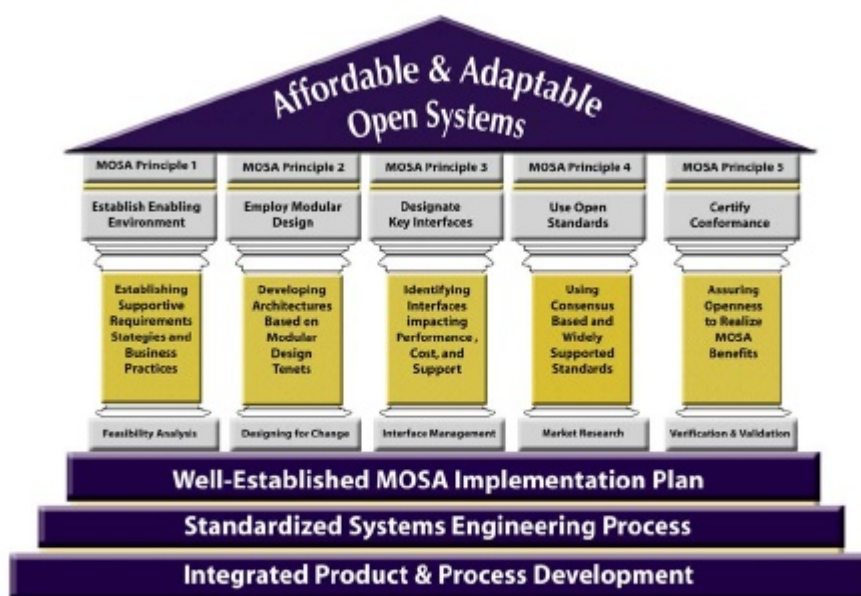


Figure 3: The MOSA Framework

Źródło:

Col. Kenneth Flowers and Cyrus Azani – Proceedings of Defense Industrial Association Systems Engineering Conference, October 2004, Dallas, Texas

Problem „Vendor-Lock In” dotyka wielu – nawet największych odbiorców najbardziej zaawansowanych technologii. Departament Obrony USA (DoD) zareagował opracowaniem unikalnej strategii techniczno/biznesowej obowiązującej przy zakupach nowych systemów uzbrojenia lub modernizacji istniejących zasobów obronnych USA.

Strategia ta nazwana w skrócie MOSA (Modular Open System Approach) jest objęta formalną dyrektywą DoD D5000.1 i w zasadzie dotyczy jedynie systemów militarnych. Jednak ponieważ warto się uczyć choćby od diabła (w końcu projektem DARPA zarządzanym przez DoD zawdzięczamy TCP/IP i tym samym Internet) postanowiłem dostosować zasady określone przez MOSA do analizy procesu akwizycji systemów komputerowych ze szczególnym uwzględnieniem sektora publicznego.

Przejdźmy więc do szczegółów:

Analiza dostępnego środowiska

W tym etapie niezbędne jest przeprowadzenie studium wykonalności zadania w ramach dostępnych aktualnie rozwiązań technicznych. Powinna ona obejmować:

- inwentaryzację aktualnych potrzeb,
- dyskusję alternatywnych metod ich zaspokojenia.
Należy przy tym określić podstawowe ich właściwości takich jak otwartość, powszechność stosowania, możliwość adaptacji nowych technologii, liczbę producentów/dostawców produktów konkurencyjnych, niezawodność, stosowane cykle technologiczne, organizację oraz poziom wsparcia technicznego, a także preliminowane koszty charakteryzujące poszczególne analizowane rozwiązania z estymacją TCO (Total Costs of Ownership) włącznie,
- analizę poziomu interoperacyjności branż pod uwagę rozwiązań ze szczególnym uwzględnieniem obowiązujących praktyk i standardów (w tym także *de-facto*),
- stosowane przez producentów modele biznesowe – a w szczególności warunki dostępności poszczególnych produktów wraz z technologią realizacji sprzedaży zarówno produktów, jak i usług z nimi związanych,

Wprowadzenie środowiska modułowego

Należy wprowadzić podział projektowanego systemu na niezależne, wymienne moduły. Taka praktyka powinna być bezwarunkowo stosowana także w przypadku rozbudowy lub modernizacji istniejących systemów. Definiowane moduły:

- powinny mieć dokładnie określoną funkcjonalność,
- ich działanie nie powinno w żadnym przypadku uzależnione od instalacji, działania lub właściwości innych modułów,
- powinny mieć dokładnie określone zasady współpracy z innymi modułami systemu, a w szczególności łącza (interfejsy) umożliwiające współpracę ze środowiskiem zewnętrznym modułu.

Powyższe właściwości mają na celu realizację wymagania „Design for Change”, która powinna zapewnić możliwość wymiany każdego z modułów niezależnie od innych. Dzięki podejściu modułowemu możliwa staje się ciągła modernizacja systemu oraz uzupełnianie jego funkcjonalności o technologie, które nie były jeszcze znane lub powszechnie wdrożone w czasie projektowania systemu.

Zdefiniowanie kluczowych interfejsów

Modularna budowa systemu, która bardzo skutecznie zabezpiecza przez uzależnieniem się od dostawców wymaga starannego określenia łączy (interfejsów) pomiędzy poszczególnymi modułami. W procesie projektowania systemu korzystne jest wprowadzenie podziału interfejsów na dwie grupy:

- Interfejsy kluczowe (Key Interfaces), do których należy zaliczyć interfejsy o zasadniczym znaczeniu dla pracy całego systemu,
- Interfejsy pomocnicze – które realizują dodatkowe funkcje komunikacyjne pomiędzy modułami wchodzącymi w skład systemu i zakłócenia w ich ewentualnej pracy nie powodują unieruchomienia systemu, a jedynie ograniczenia jego dodatkowych funkcjonalności.

Interfejsy kluczowe powinny być zarządzane przez użytkownika systemu przez cały cykl jego życia – od projektu koncepcyjnego do ostatecznego wycofania całego systemu (nie pojedynczych modułów!) z eksploatacji.

Podczas projektowania systemu projektant powinien bezwarunkowo określić, które interfejsy są kluczowe oraz przygotować ich specyfikację zarówno koncepcyjną, jak i kompletną specyfikację techniczną. Dokumentacja kluczowych interfejsów powinna być dostępna dla zainteresowanych przez cały okres eksploatacji systemu. Stanowi ona także bardzo ważny element Specyfikacji Istotnych Warunków Zamówienia przy ew. wymianach modułów systemu na nowsze lub przy wprowadzaniu nowych modułów (warunek konieczny dla zachowania Designing for Change).

Wykorzystywanie otwartych standardów

Przy projektowaniu systemu należy zwrócić uwagę, aby jego kluczowe interfejsy wykorzystywały standardy otwarte. Za standard otwarty należy uznawać jedynie standard, który spełnia cztery warunki otwartości zdefiniowane przez Unię Europejską, według których za Standard Otwarty uważa się standard, który:

- *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.*

Wykorzystanie Standardów Otwartych dla kluczowych interfejsów systemu jest jednym z najlepszych sposobów zabezpieczenia użytkownika przed wystąpieniem efektu „Vendor Locked-In”, ponieważ zapewniają w praktyce realizację zasad wolnej konkurencji dostawców.

Standardy zamknięte, których specyfikacja nie została opublikowana lub jest dostępna jedynie za dodatkową opłatą stawiają w uprzywilejowanej pozycji wybranych dostawców / producentów. Co gorzej, pozycja ta najczęściej okazuje się bardzo trudno „naruszalna” w przyszłości, ponieważ oczywiste jest, że nowe moduły muszą współpracować z już wykorzystywanymi. Ogranicza to wybór ich dostawców jedynie do tych podmiotów, które dysponują dostępem do odpowiednio szczegółowej dokumentacji.

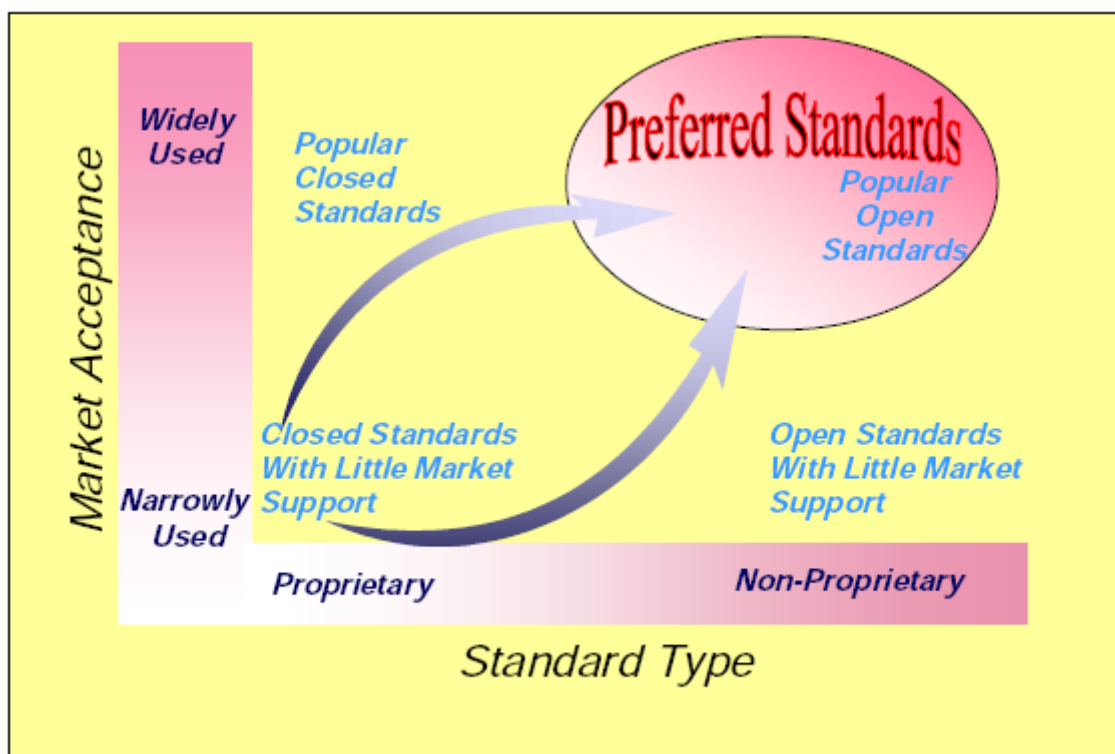


Figure 3: Preferred Standards

Cyrus H. Azani – ITEA Journal, September.October 2001 Vol.22 No.3

Przy projektowaniu systemu lub podczas jego modernizacji należy używać preferowanych standardów (patrz rysunek). Kryteria uznania standardu za preferowany to:

- Otwartość (patrz Warunki Otwartości),
- Powszechność wykorzystywania.

Należy natomiast unikać zarówno standardów zamkniętych (proprietary), jak i mało popularnych. Grozi to wystąpieniem efektu „Locked-In”!

Closed System Characteristics	Open System Characteristics
Use of closely held, private interfaces, languages, data formats and protocols (government or vendor unique standards)	Use of publicly available and widely used interfaces, languages, data formats and protocols
critical importance is given to unique design and implementation	critical importance is given to interfaces management and widely used conventions
less emphasis on modularity	heavy emphasis on modularity
vendor and technology dependency	vendor and technology independence
minimization of the number of implementations	minimization of the number of types of interfaces
difficult and more costly integration	easier and more cost effective integration
difficulty with portability, connectivity interoperability and scalability	high degree of portability, connectivity, interoperability, and scalability
use of sole-source vendor	use of multiple vendors
expansion and upgrading usually requires considerable time, money and effort	easier, quicker and less expensive expansion and upgrading
higher total ownership cost	lower total ownership cost
slower and more costly technology transfer	technology transfer is faster and less costly
components, interfaces, standards, and implementations are selected sequentially	components, interfaces, standards, and implementations are selected interactively
systems with shorter life expectancy	systems with longer life expectancy
use of individual company preferences to set and maintain specifications	use of group consensus process to maintain interface specifications
less adaptable to change in threats and technologies	more adaptable to evolving threats and technologies
focusing mostly on development cost and meeting present mission	focusing on total costs of ownership, sustainment, and growth
user as the producer of systems	user as the consumer of components
rigid and slow system of influence and control	real time and cybernetic system of influence and control
adversarial relationship with prime contractors/suppliers/vendors	Symbiotic relationship with prime contractors/suppliers/vendors
mostly confined to traditional suppliers	non-traditional suppliers can compete
simple conformance testing	very challenging conformance testing

Table 1: Open versus Closed Systems

Cyrus H. Azani – ITEA Journal, September/October 2001 Vol.22 No.3

O ile dla kluczowych interfejsów stosowanie otwartych standardów powinno być wręcz obligatoryjne, a odstępianie od tej zasady może być uznane za zasadne jedynie w przypadku, gdy nie istnieje standard otwarty, który jest w stanie spełnić wymagania stawiane przez projektowaną obsługę interfejsu – o tyle interfejsy pomocnicze mogą być obsługiwane przy użyciu standardów nie spełniających w pełni warunków otwartości. Moim zdaniem nie należy jednak w żadnym przypadku rezygnować z przygotowania kompletnej dokumentacji obsługi takiego interfejsu.

Jest ona niezbędna przynajmniej z 2 powodów:

- dla celów serwisu systemu, niezależnie od tego, czy jest on realizowany wewnątrz, czy też przez podmiot zewnętrzny (outsourcing),
- w przypadku planowanej rozbudowy lub modernizacji systemu umożliwia opracowanie nowych modułów w taki sposób, aby mogły one współpracować poprawnie z resztą systemu.

Przykład:

W poniższej Tabeli zamieszczono fragment specyfikacji interfejsu obsługi klawiatury w systemie POLTAX. Specyfikacja ta została zamieszczona w Specyfikacji Istotnych Warunków Zamówienia na dostawę bezdyskowych końcówek sieciowych.

Jest ona intersubiektywna i interkomunikowalna (oczywiście dla specjalistów). Dzięki zamieszczeniu tej specyfikacji zostały zachowane warunki swobodnej konkurencji pomiędzy dostawcami końcówek, ponieważ każda firma dysponująca odpowiednim zapleczem programistycznym mogła przygotować odpowiedni produkt oraz sprawdzić, czy spełnia on wymagania Zamawiającego.

Nie wystąpiło więc zjawisko „Vendor Lock-In”.

Fragment dokumentacji interfejsu klawiatury – załącznik do SIWZ:

Klawisze i kombinacje klawiszy	Wysyłana sekwencja HEX (tryb 8 bit)	Wysyłana sekwencja HEX (tryb 7 bit)
F1	8F50	1B4F50
CTRL/F1	1B1B8F50	1B1B1B4F50
ALT/F1	1B8F50	1B1B4F50
F2	8F51	1B4F51
CTRL/F2	1B1B8F51	1B1B1B8F51
ALT/F2	1B8F51	1B1B8F51
F3	8F52	1B4F52
SHIFT/F3	9B32357E	1B5B32357E
CTRL/F3	1B9B32357E	1B1B5B32357E
CTRL/SHIFT/F3	1B1B9B32357E	1B1B1B5B32357E
ALT/F3	1B8F52	1B1B4F52

W pierwszym etapie wykorzystywania systemu POLTAX stosowano terminale znakowe obsługiwane przez łącza szeregowo RS232 i sieciowe koncentratory terminali.

Wprowadzając nowocześniejsze terminale graficzne podłączane bezpośrednio do sieci Ethernet o znacznie poszerzonej funkcjonalności należało zapewnić poprawną obsługę systemu POLTAX – podstawowej aplikacji produkcyjnej Urzędów Skarbowych. Gdyby Ministerstwo Finansów RP nie dysponowało tą specyfikacją zadanie modernizacji systemu byłoby znacznie utrudnione – konieczny byłby np. kontakt techniczny z firmą BULL, która mogłaby zażądać dodatkowych opłat za wydanie powyższej dokumentacji.

Dzięki temu, że pełny opis interfejsu klawiatury i ekranu (z dodatkowymi wymaganiami) był w dyspozycji użytkownika wdrożenie końcówek (opartych o Wolne Oprogramowanie – system operacyjny LINUX) w liczbie około 18000 (osiemnaście tysięcy!) w całej Polsce przebiegło bez najmniejszych problemów!



Testowanie rozwiązania i sprawdzenie jego zgodności z założeniami

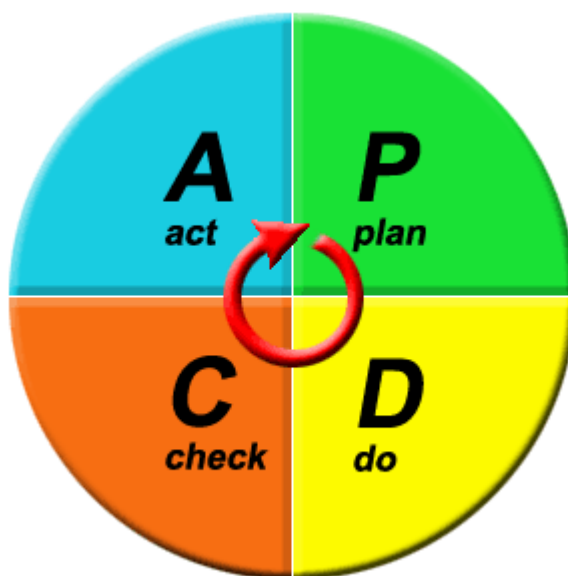
Współczesne metody projektowania zasługujące na miano TQM (Total Quality Management czyli Kompleksowe Zarządzanie przez Jakość) wymagają formalnego sprawdzenia zbudowanego systemu lub modernizowanych elementów. Przy sprawdzaniu należy wziąć pod uwagę następujące elementy:

- Czy zastosowane technologie i standardy odpowiadają aktualnym trendom technicznym?
- Czy są dostępne produkty spełniające założenia projektowe i wykorzystujące zaproponowane do stosowania standardy? Ilu producentów oferuje takie produkty? Czy zgodność oferowanych produktów z przyjętymi standardami jest pełna?
- Czy organizacje zarządzające standardami dostarczają odpowiednie narzędzia lub udostępniają serwery do sprawdzenia zgodności ze standardem (np. popularny HTML Validator na stronach www.w3.org)? Pozwoli nam to na znaczne przyspieszenie i redukcję kosztów testowania systemu.
- W jaki sposób i przez kogo są zarządzane wybrane standardy?

Proszę o zwrócenie uwagi, że stosowanie standardów oraz sprawdzenie poprawności ich implementacji redukuje do minimum potrzebę prowadzenia testów na stanowiskach pracy. Takie (wciąż niestety popularne...) testy mogą być porównywane ze stosowanym w Związku Radzieckim w trakcie drugiej wojny światowej systemem „rozpoznanania bojem”. Był on zawsze związany z bardzo dużymi stratami własnymi.

Obecnie największą wagę przywiązuje się do testów w prowadzonych w procesie produkcji, których poprawne przeprowadzenie praktycznie gwarantuje szybkie i bezproblemowe wdrożenie systemu bez potrzeby angażowania instalacji i pracowników produkcyjnych.

Przedstawione zasady są w pełni zgodne z systemami TQM opisanymi przez tzw. „koło Deminga”:



Strategia MOSA respektuje także słynne „10 zasad” Kaizen, które stanowią podstawę praktyki nowoczesnego zarządzania. Myślę, że warto je przytoczyć:

- 1. Problemy stwarzają możliwości**
- 2. Pytaj 5 razy „Dlaczego?”**
- 3. Bierz pomysły od wszystkich**
- 4. Myśl nad rozwiązaniami możliwymi do wdrożenia**
- 5. Odrzucaj ustalony stan rzeczy**
- 6. Wymówki, że czegoś się nie da zrobić są zbędne**
- 7. Wybieraj proste rozwiązania – nie czekając na te idealne**
- 8. Użyj sprytu zamiast pieniędzy**
- 9. Pomyłki koryguj na bieżąco**
- 10. Ulepszanie nie ma końca**

***Kaizen** (z japońskiego: "kai" - zmiana, "zen" - dobry, czyli ciągle doskonalenie) - filozofia wywodząca się z japońskiej kultury i praktyki zarządzania. Jej zastosowanie nie sprowadza się tylko do tego, by produkt, czy usługa odpowiadały określonym wymaganiom jakości. W myśl tej filozofii jakość sprowadza się do stylu życia - niekończącego się procesu ulepszania. Podstawową regułą tej filozofii jest ciągle zaangażowanie oraz chęć ciągłego podnoszenia jakości firmy i produktu.*

dr Tomasz Barbaszewski

Przewodniczący Rady Ekspertów FWiOO

tomasz.b@fwioo.pl

Rada Ekspertów Fundacji Wolnego i Otwartego Oprogramowania

Celem statutowym Fundacji (www.fwioo.org) jest wspieranie rozwoju i popularyzowanie wykorzystywania Wolnego Oprogramowania. Cel ten jest między innymi realizowany poprzez Radę Ekspertów, której zadaniem jest zarówno przygotowywanie materiałów informacyjnych poświęconych wszelkim zagadnieniom (technicznym, prawnym itp.) związanym z Wolnym Oprogramowaniem.

Rada Ekspertów wykonuje także projekty koncepcyjne związane z wdrażaniem Wolnego Oprogramowania, organizuje szkolenia specjalistyczne ze szczególnym uwzględnieniem seminariów dla Kadry Zarządzającej oraz prowadzi pełny zakres usług doradczych i konsultingowych w zakresie związanym z wykorzystywaniem Wolnego Oprogramowania.

Wszystkich zainteresowanych zapraszamy do nawiązywania bezpośredniego kontaktu z nami!

