

XV Konferencja PLOUG
Kościelisko
Październik 2009

O ergonomiczności oprogramowania słów kilka

Tomasz Wyrozumski
BMS Creative, R-Data

tw@bms.krakow.pl

Abstrakt. Artykuł poświęcony jest problematyce ergonomiczności oprogramowania. Argumentujemy w nim, że na pewnym etapie projektu stopień owej ergonomiczności ma niezwykle istotne znaczenie, a czasami wręcz decyduje o powodzeniu całego przedsięwzięcia. Przeprowadzamy zatem analizę pojęcia ergonomiczności, a następnie, rozważając wszystkie etapy powstawania oprogramowania, szczególnie wyjaśniamy, dlaczego tak trudno jest zbudować ergonomiczny system. Z przedstawionych rozważań wynikają bezpośrednio pewne wskazówki dla analityków, projektantów, a także osób zarządzających projektami informatycznymi.

1. Sukces który okazuje się porażką

W niezliczonych publikacjach, wystąpieniach na konferencjach, dyskusjach panelowych i kulturalnych nieustannie rozmawia się o przyczynach niepowodzeń projektów informatycznych. Kомуś postronnemu mogłoby się nawet wydać to dziwne, jako że o wielkich i spektakularnych klapach – przynajmniej publicznie – mówi się dość rzadko. Odsetek podpisanych protokołów odbioru zdaje się również nie potwierdzać owego czarnego obrazu rzeczywistości. Jak więc jest naprawdę? W istocie, projekty na które wydano grube miliony, oficjalnie kończą się na ogół sukcesem, tzn. nagrodami, galami, okładką w *Computerworldzie*. Tak być musi, ponieważ ani zamawiający, ani wykonawcy nie lubią przyznawać się do porażek. Nieoficjalnie natomiast są poprawki, protezy, przybudówki i nieustanne narzekania użytkowników. Ileż to wdrożeń zintegrowanych systemów „do wszystkiego” okazało się później w rzeczywistości zaledwie stosunkowo prostymi, choć niewątpliwie mocno przepełnionymi, aplikacjami finansowo-księgowymi o ograniczonych możliwościach! Jest więc nad czym się pochylić podczas konferencji i niewątpliwie warto analizować przyczyny owych, często skrywanych niepowodzeń.

Dziś chcemy zająć się tylko jedną z nich i to pozornie nie najważniejszą. Ktoś mógłby wręcz powiedzieć – trzeciorzędą, choć naszym zdaniem powinna ona zdecydowanie stać w pierwszym szeregu. Dlaczego? Postaramy się to wyjaśnić, spoglądając z perspektywy praktyka i dzieląc się z czytelnikiem różnymi, czasem trochę luźnymi uwagami i spostrzeżeniami. A mowa będzie o ergonomiczności oprogramowania...

2. Krzesło a sprawa oprogramowania

Ze słowem *ergonomia* autorowi dane było spotkać się jeszcze w dzieciństwie, u początku lat siedemdziesiątych ubiegłego wieku, w czasach, gdy inne słowo, komputer, powszechnie kojarzyło się z ogromną, liczącą szafą, z którą można się było skomunikować za pomocą perforowanych kart i papierowych tasiemek przypominających serpentyny. Oczywiście nie w kontekście komputerów usłyszał on wówczas o ergonomii. Chodziło o meble, a konkretnie o krzesła. Otóż w jakimś popularnonaukowym programie telewizyjnym (było wtedy takich trochę w TVP!) pojawił się pewien sympatyczny pan i zademonstrował dość dziwne krzesło. Nie wydawało się ono ani ładne, ani wygodne, jednak pan ów stwierdził, iż jest „ergonomiczne” i bardzo dokładnie objaśniał, w jaki to sprytny sposób oparcie wpasowuje się w plecy siedzącego, z ewidentną korzyścią dla jego kręgosłupa. Dla autora stało się wówczas jasne, że o ergonomiczności można mówić wtedy, gdy na krześle będzie się siedzieć długo i to dzień w dzień. Omawiane pojęcie dotyczy zatem czynności powtarzalnych, jako że przy incydentalnych komfort nie odgrywa aż tak dużej roli. Jeśli na krzesło przyjdzie nam przysiąść tylko na chwilę, nasz kręgosłup nie ucierpi aż tak bardzo, nawet jeżeli oparcie okaże się skrajnie niewygodne.

Zajrzyjmy teraz do wszechwiedzącej i w pełni demokratycznej Wikipedii, by znaleźć tam następujący tekst:

Ergonomia (gr. *εργον* – praca + *νομος* – prawo) – nauka o pracy, czyli dyscyplina naukowa zajmująca się dostosowaniem pracy do możliwości psychofizycznych człowieka. Ma na celu humanizowanie pracy poprzez taką organizację układu: człowiek – maszyna – warunki otoczenia, aby wykonywana ona była przy możliwie niskim koszcie biologicznym i najbardziej efektywnie, co uzyskuje się m.in. poprzez eliminację źródeł chorób zawodowych. Ergonomia jest nauką interdyscyplinarną. Korzysta z dorobku takich nauk lub dziedzin naukowych jak: psychologia pracy, socjologia pracy, fizjologia pracy, higiena, medycyna pracy, organizacja pracy, antropometria oraz nauk technicznych, np. materiałoznawstwa, budowy maszyn.

Powyższa definicja wydaje się trochę zawężająca, jeśli pojęcie pracy rozumie się bardzo restrykcyjnie. Na pierwszy rzut oka nie kojarzy się bowiem z pracą jeżdżenie na nartach dla przyjemności, a przecież każdy dziś doskonale rozumie, co mamy na myśli mówiąc o ergonomicznych rączkach kijków narciarskich. Nie jest więc naprawdę istotne, czy chodzi o pracę wykonywaną w celach zarobkowych, ale o to, czy pewne czynności są powtarzalne, długotrwałe, częstotliwe. Jeśli tak, niewątpliwie leżą w kręgu zainteresowań ergonomii.

Jak zatem ma się to wszystko do systemów informatycznych? Jeżeli potraktować je jako maszyny oddziaływujące z człowiekiem, to sprawa wydaje się oczywista. Systemy składają się ze sprzętu i oprogramowania. W przypadku tego pierwszego ergonomia zajmie się takimi elementami jak monitory, klawiatury, myszki itp. Gdy zaś mowa o oprogramowaniu, to niemal z definicji w rachubę wchodzi tzw. interfejs użytkownika. I to właściwie jemu poświęćmy pozostałą część niniejszych rozważań.

Warto na początek odnotować, że od czasu pojawienia się pierwszych komputerów interfejs uległ ogromnym przeobrażeniom. W epoce kart perforowanych, tasiemek, a także wczesnych terminali znakowych cała komunikacja z komputerem odbywała się za pośrednictwem poleceń wydawanych maszynie w określonym języku oraz komunikatów drukowanych na papierze, bądź wyświetlanych przez nią na ekranie. Nie wszyscy dziś zdają sobie sprawę z tego, że w owych czasach, by wyciągnąć z bazy danych jakieś informacje, trzeba było znać język SQL! Uczyli się go więc nie tylko programiści, ale również tzw. użytkownicy, czyli np. sekretarki, księgowi, menedżerowie itp. Aż dziw bierze, że nikt nie narzekał wtedy na dyskomfort. Było oczywiste, że aby porozumieć się z maszyną, należało posługiwać się jej językiem, a kto tego nie potrafił, był po prostu mniej wartościowym pracownikiem.

Ułatwianie sobie życia leży jednak ewidentnie w naturze człowieka, a lenistwo jest – w jakimś sensie i do pewnego stopnia, oczywiście – motorem postępu technicznego. I tak zaczęto ułatwiać życie owym mniej pojętym użytkownikom komputerów, którzy wciąż zapominali, jak brzmią właściwe polecenia i jaka powinna być ich składnia. Na ekranach zaczęły więc pojawiać się podpowiedzi w postaci pytań, na które można było odpowiedzieć TAK lub NIE (Y/N), bądź szeregu ponumerowanych opcji do wyboru – wtedy wystarczyło wpisać właściwy numer – czy też innych bardzo podobnych rozwiązań. Przypomnijmy jeszcze dalszą ewolucję, której kolejnymi elementami były przesuwane po ekranie podświetlenia, rozwijane ramki i dialogi, ze słynną biblioteką Turbo Vision Borlanda, stanowiącą przedsmak ery okienek. Te ostatnie, o czym nie wszyscy pamiętają, zostały wymyślone przez firmę Apple, choć dziś słowo *Windows* kojarzy się zupełnie z kim innym.

I tak oto doszliśmy do Microsoftu. Firma ta, rozpropagowując okienka, zrobiła bardzo wiele dla ułatwienia kontaktu człowieka z maszyną. Nie ma sensu wymieniać tu szczegółowo wszystkich jej zasług, bowiem w kontekście poprzednich akapitów stają się one niemal oczywiste. Paradoksalnie, warto jednak zwrócić uwagę na kilka karygodnych błędów, które przy tej okazji popełniono.

Pierwszy z nich to wmówienie użytkownikowi, że on sam nie musi nic umieć, by korzystać z oprogramowania, a jeśli ma jakiegokolwiek problemy z jego obsługą, to cała wina leży po stronie producenta, który stworzył system „mało intuicyjny” (nie mylić z ergonomicznością!). Intuicyjność jest oczywiście pojęciem względnym. Doświadczony kierowca uzna prowadzenie samochodu za coś naturalnego i intuicyjnego, podczas gdy osoba po raz pierwszy w życiu siadająca za kierownicą będzie mieć zgoła inne odczucia. Podobne problemy napotka jednak wspomniany kierowca siadając za sterami awionetki, pilot awionetki w Boeingu lub F16 itd. Nie oczekujemy więc, że ktoś, kto bardzo sprawnie posługuje się Wordem, bez instruktażu zacznie korzystać z zaawansowanego systemu ERP!

Drugim błędem Microsoftu było ujednoczenie produktów dla amatorów i profesjonalistów (np. wersje *Windows XP Home* i *Professional* różnią się w istocie tylko nieznacznie). Oczywiście łatwo

zrozumieć motywy takiego postępowania, tzn. chęć redukcji kosztów, ale warto zauważyć, że to trochę tak, jakby dziecku do zabawy i zawodowemu stolarzowi oferować ten sam zestaw narzędzi. System operacyjny do pracy naprawdę nie musi mieć mnóstwa gadżetów, natomiast powinien być stabilny i bezpieczny. Gadżety w tym nie pomagają, natomiast z reguły przeszkadzają przy wykonywaniu czynności długotrwałych, bądź częstotliwych i z tego powodu po prostu bywają nieergonomiczne.

Jest jeszcze trzeci błąd, ale o nim za chwilę, bo stanowi niemal klasyczny przykład tego, w jaki sposób nie należy podchodzić do problemu ergonomiczności.

Podsumowując, stwierdzimy na razie, że Microsoft niewątpliwie bardzo rozpaskudził użytkowników, częściowo w złym, ale też częściowo w dobrym tego słowa znaczeniu. Z jednej strony sprawił, że producenci oprogramowania muszą przejmować się tym, czy jest ono wygodne w użyciu, z drugiej spowodował, że użytkownicy oceniają software zanim się go nauczą, a to nie ułatwia życia ani handlowcom, ani wdrożeniowcom. Użytkownicy oczekują ponadto, że profesjonalne systemy zawierać będą różnego rodzaju błyskotki, nie mając świadomości negatywnego wpływu tychże na wydajność, stabilność i bezpieczeństwo rozwiązań.

3. Straszny sen wdrożeniowca

Jeżeli system spełnia funkcjonalne oczekiwania klienta, a w dodatku interfejs wygląda przyjaźnie, to doprowadzenie do transakcji nie stanowi dla dobrego handlowca jakiegoś ponadprzeciętnego wyzwania. Podczas prezentacji z łatwością pokaże on, jak zrealizować poszczególne zadania, z gracją przechodząc będzie od formatki do formatki, umiejętnie klikając myszką w poszczególne przyciski. Zadowolony klient z uśmiechem spoglądać będzie z kolei na ekran, myśląc sobie: „To jest to!” Istna sielanka. Po standardowych negocjacjach cenowych podpisany zostaje kontrakt i zaczyna się wdrożenie, a wraz z nim schody... Oto bowiem teoretycznie wszystko jest w porządku, ale wykonanie pewnej typowej operacji (np. związanej z obsługą klienta, nadaniem biegu sprawie w urzędzie itp.) trwa cztery razy dłużej niż przedtem, gdy systemu nie było. Oczywiście, próbuje się argumentować, że tak naprawdę ten czas oszczędzany jest gdzie indziej itd., ale przez skórę wszyscy czują, że coś tu nie gra. Wydajność jest zgodna z założeniami, sprzęt nie zawodzi, więc co? Pozostają użytkownicy, których pracy zaczynamy się intensywnie przyglądać, podejrzewając, że być może niezbyt wprawnie posługują się systemem. I wtedy właśnie okazuje się rzecz najdziwniejsza: użytkownicy pracują dokładnie tak, jak należy, albo – ściślej mówiąc – tak, jak pozwala im na to oprogramowanie. Innymi słowy, męczą się, otwierając po wielokroć te same formatki, przewijając ekrany w górę i w dół, by zmienić wartości w poszczególnych polach, tudzież wykonują cały szereg innych czynności, z których żadna sama w sobie nie jest uciążliwa, ale jeśli zważyć, ile razy się je powtarza, w każdym, nawet najbardziej doświadczonym wdrożeniowcu rodzi się współczucie. Jeżeli któryś z czytelników jeszcze go nie doznał, niech wyobrazi sobie sekretarkę, która musi codziennie zarejestrować kilkaset pism, każdorazowo wprowadzając datę wpływu. System, zgodnie ze wszystkimi sensownymi założeniami stworzonymi na etapie analizy, bądź uzupełnionymi przez projektanta, podpowiada datę dzisiejszą. Jest tylko jeden drobny problem: listonosz przynosi pocztę dopiero około szesnastej i sekretarka, która o tej właśnie porze kończy pracę, **zawsze** rejestruje ją nazajutrz!

Gdyby ktoś chciał w tym momencie wypowiedzieć zdanie zaczynające się od słów: „Ale przecież można było...”, niech uzbroi się w cierpliwość. Wrócimy do tego przykładu w następnych częściach naszych rozważań i zastanowimy się, co i kiedy można było zrobić. Na razie poprzestańmy tylko na stwierdzeniu pewnego prostego faktu:

Elementarne mankamenty ergonomiczne oprogramowania bardzo często pozostają niezauważone aż do momentu, kiedy prawdziwi użytkownicy zaczynają je wykorzystywać w rzeczywistych zastosowaniach.

A to właśnie dzieje się na etapie wdrożenia i w dodatku zazwyczaj w jego późniejszych fazach. Jakie są skutki? Łatwo przewidzieć. Użytkownicy zaczynają kontestować oprogramowanie, nie- rzadko stosując przy tym strajk włoski i wykazują szefostwu, że z systemu tak na prawdę nie da się korzystać. W zależności od tego, jak zostały podpisane umowy oraz na ile racjonalnie zachowują się klient i dostawca, konsekwencje biznesowe mogą być różne. Ich szczegółowa analiza i kategoryzacja nie jest w ogóle przedmiotem niniejszego artykułu, choć zapewne zainteresowała- by prawników, którzy w tym miejscu udzieliliby obu stronom cennych rad na temat formułowania poszczególnych paragrafów kontraktu. Zwróćmy jedynie uwagę na to, że z punktu widzenia zarządającego projektem znajdujemy się właśnie na etapie, kiedy to ani harmonogram, ani budżet nie pozwala na żadne prace dostosowawcze, choć przecież właśnie ich tak naprawdę oczekuje klient. Co więcej, być może niektóre miałyby faktycznie charakter prostych poprawek krawiec- kich, lecz inne wymagałyby przeprowadzenia poważnych działań analitycznych, przeprojektowa- nia systemu, przepisania sporych fragmentów kodu, wykonania testów itd. Jednym słowem: w praktyce nierealne. Będziemy raczej łątać system, dobudowywać do niego jakieś dodatkowe aplikacje oraz... męczyć się. Może za jakiś czas (nie taki znów krótki) pojawi się nowa wersja, w której uwzględni się część doświadczeń zdobytych na etapie wdrożenia, a może za jakiś czas (nie taki znów długi) klient zacznie zastanawiać się nad wymianą systemu?

W kontekście wcześniejszego stwierdzenia, iż ergonomia dotyczy czynności powtarzalnych, długotrwałych i częstotliwych, jest rzeczą oczywistą, że ergonomiczne niedoróbki mogą nie ujawnić się w fazach analizy, projektowania, testów, bądź – to akurat dotyczy klienta – wyboru i odbio- ru oprogramowania. Powstaje jednak naturalne pytanie, czy tak właśnie musi być, względnie, co należy uczynić, by było inaczej?

4. Pierworodny grzech analityka

W świecie idealnym poprawnie przeprowadzona analiza, podczas której właściwie zidentyfi- kowano wszystkie potrzeby biznesowe użytkownika, przekłada się na doskonały projekt i bez- błędnie napisany kod. W rzeczywistości, jak wiadomo, jest trochę inaczej. Niemniej jednak należy dążyć do ideału, czyli przede wszystkim porządnie przyłożyć się do analizy.

Kim jest analityk? To człowiek dysponujący dużą wiedzą z zakresu informatyki, dobrze rozu- miąjący dziedzinę problemu, logicznie myślący, inteligentny, rzecz jasna itd. Ponadto jednak ce- chować go musi otwartość, łatwość nawiązywania kontaktów, dociekliwość, umiejętność wnikli- wego obserwowania otaczającej rzeczywistości, ale także asertywność, dzięki której nie będzie on akceptował bezkrytycznie wszystkich pomysłów klienta. Sherlock Holmes byłby zapewne dobrym kandydatem na to stanowisko, gdyby tylko Sir Arthur Conan Doyle uczynił go nieco mniej oschłym. Analityk jest też – o czym być może zapomnieliśmy powiedzieć – człowiekiem, a jako że *errare humanum est*, omylność stanowi jeszcze jedną istotną jego cechę. Błędy są wpisane w proces analizy dokładnie tak samo, jak w proces tworzenia kodu, tyle tylko, że błędy w kodzie znacznie łatwiej się poprawia.

Analityk koncentruje się na opisanu procesów biznesowych, zbudowaniu modelu zależności obiektowych, relacyjnego modelu danych itp. Ponieważ materiałem źródłowym są dla niego wy- wiady, przepisy prawa, różnego rodzaju dokumenty i obserwacje otaczającej rzeczywistości, zaś produktem diagramy UML lub coś bardzo podobnego, praca analityka w istocie polega na identy- fikowaniu struktury w tym co widzi, a więc abstrahowaniu od nieistotnych szczegółów i wych- wytywaniu wszystkiego, co stanowi informatyczną esencję analizowanego biznesu. I tu właśnie daje się dostrzec fundamentalny problem. Otóż w ergonomii właśnie chodzi o szczegóły! Udział analityka w procesie tworzenia (pierwszego) krzesła biurowego sprowadzi się do określenia jego rozmiarów, zadbania o to, by miało kółka, oparcie, może jakieś elementy regulacyjne, by było miękkie i się nie przewracało, jednak najprawdopodobniej umknie mu kwestia precyzyjnego profi-

lu podłokietników – czegoś, co da się zauważyć dopiero po wielu godzinach korzystania z powstałego mebla.

Nasz przykład z listonoszem obrazuje oczywiście sytuację zawinioną przez analityka. Popełnił on błąd zakładając, że poczta zawsze przychodzić będzie do biura rano. Czy można jednak mieć do niego o to wielkie pretensje? Nikt z użytkowników nie zasygnalizował mu tej kwestii. Co więcej, w większości lokalizacji problem w ogóle nie istniał! Oczywiście, zawsze da się powiedzieć, że badanie było nie dość wnikliwe, a uogólnienie nieuprawnione, jednak bądźmy sprawiedliwi: nasz analityk naprawdę musiał zajmować się znacznie poważniejszymi kwestiami, niż domyślna wartość daty na jednej z formatek!

Jest zresztą jeszcze jeden aspekt całej sprawy. Otóż użytkownik zawsze postrzeży system informatyczny przez pryzmat jego interfejsu. Kiedy systemu jeszcze nie ma, wyobraża go sobie na obraz i podobieństwo czegoś, co już gdzieś widział, czegoś z czego korzystał w poprzedniej pracy, albo czegoś, z czego właśnie korzysta, choć to coś jest bardzo proste funkcjonalnie i ma być wymienione na wymarzone, docelowe rozwiązanie. Gdy takie zaistnieje, użytkownik widzieć będzie dokładnie to, co wyświetli mu się na ekranie i nie zainteresuje się w ogóle serwerem bazy danych, serwerem aplikacji, czy organizacją warstwy logiki biznesowej. Nawiasem mówiąc, czy można mieć do niego o to żal? A kogo z nas, kierowców, naprawdę ciekawi, co ma pod maską? Moc, moment obrotowy, ilość punktów wtrysku – to dla laików! Przyznajmy, że znacznie bardziej obchodzi nas układ zegarów na desce rozdzielczej, niż właściwości stopu z którego wykonano głowicę silnika!

A jak widzi przyszły system analityk? Interfejs jest i musi być dla niego wtórny. Na początek zadowolony się stwierdzeniem: „okienkowy” lub „przeglądarkowy”, a ewentualne szkice formatek będą z pewnością ostatnim produktem jego pracy. Powstaje więc naturalny dysonans między analitykiem i użytkownikiem – zrozumieli i oczywisty, jednak owocujący całkowitym zlekceważeniem przez analityka tego, co dla użytkownika najistotniejsze. Efektem muszą być zaniedbania w sferze komfortu pracy, czyli właśnie przedmiotu zainteresowań ergonomii.

Ktoś mógłby argumentować, że swoistym remedium na błędy analityka jest prototypowanie. Przedstawianie użytkownikowi kolejnych prototypów, a więc wersji interfejsu, które potrafi zrozumieć i które do niego przemawiają, prowadzi bardzo często do pytań w rodzaju: „A gdzie tu jest to i to?”, albo: „A jak tym przeprowadzić taką, bądź inną operację?”. Padają na nie wtedy odpowiedzi typu: „No tak, zapomnieliśmy, musimy poprawić, uzupełnić...”. Prototypowanie na ogół jednak nie doprowadzi do szczegółowego zbadania interfejsu pod kątem ergonomiczności. Nie doprowadzi z dwóch powodów: po pierwsze dlatego, że nikt nie będzie z prototypem naprawdę pracował i nie oceni jego wygody, po drugie dlatego, że nawet gdyby użytkownik zaczął na tym etapie zgłaszać jakieś uwagi, czy zastrzeżenia, to i tak dowiedziałby się, że przecież nie o to w tej chwili chodzi, takie drobiazgi wygłodzi się później, a na razie należy skoncentrować się na kwestiach fundamentalnych, czyli na tym, czy nie brakuje jakiejś ważnej funkcjonalności, czy nie zapomniano o jakimś przypadku itp.

5. W poszukiwaniu winnych, czyli od projektanta do testera

O ergonomiczność oprogramowania powinien zatroszczyć się zatem jego projektant. Brzmi to jak prawda oczywista, choć równie oczywiste wydaje się, iż naszego problemu z listonoszem projektant nie rozwiąże poprawnie. Zakładając, że analityk w ogóle nie zastanowił się nad domyślną wartością daty, możemy przypuszczać, że projektant jednak nie pozostawi jej czystemu przypadkowi, lecz postanowi wypełnić domyślnie pole „w najbardziej naturalny sposób”. A za taki odruchowo uzna bądź to datę bieżącą, pobraną z zegara systemowego, bądź też ostatnio zapisaną datę. To drugie rozwiązanie, wymagające nieco więcej inwencji i pracy, niestety również nie okaże się poprawne! Jeżeli bowiem w większości oddziałów firmy korespondencja będzie rejestrowana w dniu jej przyścia, a tylko niekiedy zdarzy się, że dla jakiegoś listu będzie trzeba incydentalnie

przestawić datę, to osoba wprowadzająca, przyzwyczajona, że zawsze podpowiada się jej ta sama wartość, będzie niezadowolona, gdy po przestawieniu daty „na chwilę”, przy kolejnej rejestracji korespondencji podpowie się jej data zła (tj. poprzednia). Akademickie rozważania, marudzenie? Nie! Historia jak najbardziej autentyczna. Na tym właśnie polegają kłopoty z ergonomicznością!

Czy zatem istnieje w ogóle jakieś prawidłowe rozwiązanie przedstawionego problemu? Chyba najlepszym pomysłem byłoby wykonanie w systemie przełącznika umożliwiającego arbitralne ustawienie daty, która miałaby się podpowiadać. Oczywiście, w każdym oddziale firmy mogłaby to być inna wartość. I tu właśnie dochodzimy do kolejnego fundamentalnego konfliktu interesów. Jak bowiem dobrze wiadomo, każdy tego typu przełącznik, każda dopuszczalna parametryzacja, generuje dodatkowe narzuty na przetwarzanie, zaś oczywistym zadaniem projektanta jest dbałość o wydajność systemu. Będzie on więc żywotnie zainteresowany w usuwaniu tego rodzaju „zbędnych” opcji. Zresztą, nie używając naprawdę projektowanego oprogramowania, nie będzie w stanie wczuć się w rolę sekretarki, której przyjdzie rejestrować setki pism w takich, bądź innych warunkach. Nie będzie więc też umiał powiedzieć, które z elementów konfiguracji mogą mieć autentycznie istotny wpływ na komfort pracy. Ta sama uwaga dotyczy – nawet chyba w większym jeszcze stopniu – programisty. Mimo iż jest on de facto pierwszym użytkownikiem tworzonego przez siebie oprogramowania i jako pierwszy uruchamia każdą zbudowaną funkcjonalność, to jednak postawić się w sytuacji użytkownika „prawdziwego” będzie mu ekstremalnie trudno. Niech za przykład posłuży taki oto, z życia wzięty dialog pomiędzy programistą, tworzącym system do fakturowania, a jego szefem.

Szef: *Co teraz robisz?*

Programista: *Pracuję nad funkcjonalnością służącą do wygodnego usuwania większej ilości faktur.*

Szef: *Ale po co to?!*

Programista: *No bo gdyby ta pani, która wystawia faktury, chciała na raz usunąć ich więcej, to będzie się musiała strasznie męczyć.*

Szef: *Ale dlaczego miałyby usuwać faktury i to jeszcze w dużych ilościach?*

Programista: *No nie wiem, ale gdyby tak się zdarzyło, to będzie jej łatwiej.*

Szef: *Tak się nigdy nie zdarzy, tego nie ma w projekcie!*

Programista: *To prawda, ale taka funkcjonalność chyba nie zaszkodzi, a zawsze się może przydać, na wszelki wypadek...*

Dłuższą chwilę zajęło owemu szefowi zrozumienie rzeczywistych motywów kierujących programistą. W istocie sprawa była jednak całkiem prosta. Programista tworząc funkcjonalność fakturowania testował na bieżąco efekty swojej pracy i wystawiał duże ilości faktur, które z jakichś powodów mu potem przeszkadzały. Musiał więc je następnie usuwać, toteż aby ułatwić sobie życie, stworzył całkiem przyzwoitą, nawet ładną i funkcjonalnie dość zaawansowaną formatkę z bardzo poprawną obsługą wyjątków – tyle że użytkownikowi do niczego niepotrzebną.

Obrazuje to doskonale pewną fundamentalną prawdę: **Programiści piszą programy dla siebie!** Tak było jest i będzie. I nie można mieć do nich o to specjalnych pretensji, bo w końcu programowanie jest procesem twórczym, a każdy proces twórczy wymaga, by twórca zaangażował się emocjonalnie w to, co robi. Inaczej się więc nie da, chyba, że twórczość z założenia chcemy zastąpić lichym rzemiosłem. Zdecydowanie lepszym pomysłem jest jednak temperowanie i ograniczanie nadmiaru swobody wypowiedzi artystycznej, niż zrezygnowanie z niej całkowicie. Pamiętać przy tym należy, że sposób widzenia interfejsu przez programistów i użytkowników jest całkowicie różny! Programiści skażeni są niejako wyglądem narzędzi informatycznych i przez ich pryzmat widzą to, co sami tworzą. Nie przeszkadza im nadmiar przycisków, opcji, komunikatów, wręcz przeciwnie, uważają je za wartość świadcząca o elastyczności rozwiązań. Użytkownicy z kolei gubią się w takim gąszczu i nie rozumieją, po co im to wszystko, skoro wykonywać muszą załad-

wie jedną prostą czynność, tyle, że bardzo wiele razy... Czynności, które z kolei wiele razy wykonują programiści, to – poza stukaniem w klawiaturę, rzecz jasna – zapisywanie kolejnych wersji kodu i kompilowanie go. Spójrzmy jak łatwo zrobić jedno i drugie w dowolnym narzędziu programistycznym!

Jeśli chodzi o wygląd interfejsu, programiści nie są zbyt wymagający, i to nie tylko w stosunku do samych siebie, ale również wobec swoich kolegów po fachu. Można by określić to mianem swoistej zawodowej solidarności, lecz w istocie przyczyna jest głębsza. Oni rozumieją, jak naprawdę działają programy, więc potrafią ocenić to, co widzą nie tylko w postaci interfejsu. Potrafią dostrzec mechanizmy i technologie poza ekranem, zaś do samego ekranu podchodzą ze swego rodzaju profesjonalną nonszalancją, tą samą, która każe rasowym administratorom UNIX'ów korzystać wyłącznie z terminali znakowych (nawet jeśli otwiera się je w okienkach X'ów), księgowym – rzucać numerami kont, a prawnikom – paragrafami. Pamiętajmy jednak, że tzw. laik od prawnika oczekuje jasnej ekspertyzy, od księgowego – kilku zrozumiałych liczb, a od programisty... – ergonomicznego interfejsu, no bo przecież z jego punktu widzenia w systemie nie ma absolutnie nic więcej.

Dodajmy jeszcze, że ergonomiczność przybiera niekiedy w rękach programistów postać wielce zabawnej karykatury. Bardzo często owocem ich pracy są formatki, bądź komunikaty ewidentnie „przegadane”, zawierające informacje bądź to oczywiste, bądź redundantne. Ileż to razy trzeba im tłumaczyć, że zdanie: *Twoje dane zostały przez system poprawnie zapisane do bazy danych* niesie ze sobą z punktu widzenia użytkownika dokładnie tę samą treść, co jedno słowo: *Zapisano*. Ileż razy trzeba przekonywać, że na formatce nie należy opisywać ułożonych w pionie pól: *Data otrzymania, Data rejestracji, Data przekazania*, ale tylko raz napisać *Data*, a potem, z zachowaniem odpowiednich wcięć, dopisać, jakiej daty dotyczy odpowiednie pole. Dlaczego tak? Ponieważ każde słowo na ekranie zostanie w jakiś sposób zauważone przez użytkownika i przetworzone przez jego mózg. Jeżeli słowo nie niesie żadnej istotnej treści, przetwarzanie to jest zbędną pracą i tym samym narusza elementarne zasady ergonomiczności. Zapamiętajmy więc: ***Żadnych zbędnych tekstów w interfejsie użytkownika!***

Skąd jednak bierze się u programistów owa pokusa do naruszania tej skądinąd dość oczywistej zasady? Trudno powiedzieć. Ponieważ jest ona, jak się wydaje, dość powszechna, źródeł należy dopatrywać się chyba w samej psychologii procesu twórczego. Może po prostu jest tak, że programista, stworzywszy pewien fragment kodu, chce nieświadomie pochwalić się swoim sukcesem przed przyszłym użytkownikiem i oprawia ów kod w zbędne słowa (a czasem inne elementy graficzne) dokładnie tak, jak oprawia się w ramę obraz, by wyeksponować dzieło? Tyle tylko, że każdy dobry malarz wie, iż rama powinna stanowić jedynie akcent, podkreślenie, nie zaś przyćmiewać to, co znajduje się wewnątrz niej.

W naszym łańcuszku prowadzącym do powstania systemu zostaje jeszcze tester. W istocie ma on największe szanse dostrzec elementarne mankamenty ergonomiczne, ale i tu pojawiają się kłopoty. Po pierwsze, tester na ogół nie posiada zbyt dużych możliwości oddziaływania na proces tworzenia oprogramowania. Jego zadaniem jest sprawdzenie, czy dany element funkcjonuje poprawnie przy różnych możliwych zachowaniach użytkownika, rodzajach danych, itp., czyli szeroko rozumianych uwarunkowaniach zewnętrznych. Jeśli tester zacznie robić coś więcej, to albo dowie się, że to nie jego sprawa, albo... awansuje. Po drugie, jak już wielokrotnie wspominaliśmy, ewentualne problemy z ergonomicznością pojawiają się tam, gdzie mamy do czynienia z zachowaniami częstotliwymi, zaś te części testów, które dotyczą zachowań częstotliwych, zazwyczaj próbuje się automatyzować. Póki co, żaden skrypt scenariusza nie zaprotestuje, że musi po wielokroć w niewygodny sposób realizować zupełnie zbędne czynności. Może w dalekiej przyszłości...

6. Co robić, kiedy nic się nie da zrobić?

Podobno szwedzcy architekci wymyślił ongiś, by nie projektować chodników na nowych osiedlach mieszkaniowych, lecz dać ludziom nieco czasu na wydeptanie ścieżek i dopiero te zastąpić chodnikami. Ergonomiczność programów komputerowych bez wątpienia traktowana jest po macoszemu. Wynika to jednoznacznie z powyższych rozważań, a jeśli któryś z czytelników nie czuje się nimi przekonany, niech po prostu rozejrzy się wkoło i porozmawia z użytkownikami systemów. Powstaje jednak pytanie, czy w obecnie funkcjonujących modelach tworzenia oprogramowania da się zrobić cokolwiek, by poprawić sytuację? Być może da się jakoś wykorzystać ów pomysł z chodnikami? Niestety, wydaje się to bardzo trudne. Jedynym sposobem byłby bowiem z góry zaplanowany powrót do analizy (a co za tym idzie, projektowania i kodowania) na późnym etapie wdrożenia oraz współpraca twórców z prawdziwym użytkownikiem. Jeśli tego typu pozycję przewidzi się w harmonogramie i budżecie projektu, szanse na jego sukces znakomicie wzrosną. Z drugiej strony jakoś nie bardzo chce się wierzyć w upowszechnienie takiego podejścia. Dla klienta jest ono niezrozumiałe, no bo niby dlaczego płacić za poprawę czegoś, co powinno być od początku robione dobrze, dla wykonawcy – co najmniej niewygodne, bo w końcu trudno przewidzieć, co właściwie będzie do zrobienia oraz trudno z góry określić warunki, które trzeba będzie spełnić, by został podpisany protokół odbioru. Wydaje się więc, że jedna strona skazana jest na brak ergonomiczności, druga zaś na ustawiczne pretensje, które są owego braku wynikiem. W sytuacji, gdy oprogramowanie zostało odebrane, a faktura zapłacona, jedynym, co może zmotywować do dalszej, żmudnej pracy firmę informatyczną, jest perspektywa stworzenia i sprzedaży kolejnych wersji – być może również innym klientom. Jeśli firma będzie się umiała wsłuchać w narzekania użytkowników, jest szansa, że owe kolejne wersje (dopiero one!) okażą się jako tako ergonomiczne.

A zatem, co robić, kiedy nic się nie da zrobić? Jedyny sensowny pomysł, jaki przychodzi w tym miejscu na myśl, to włączenie do projektu doświadczonej osoby, odpowiedzialnej wyłącznie za kwestie ergonomiczności. Niech zajmuje się detalami, które umykają analitykom, niech zwraca uwagę projektantom na pewne pozornie nieistotne kwestie, niech recenzuje pracę programistów. Nawet jeśli wszystkiego nie wyłapie i nie poprawi, to i tak dzięki niej finalny efekt pracy całego zespołu będzie użytkownikowi przypominał system, nie zaś karykaturę systemu. Lepsze to niż nic.

Nasz *pełnomocnik d/s ergonomiczności* (nazywajmy go po prostu *ergonomistą*) musi bardzo uważać, by swojego zadania nie zrozumieć opacznie i nie pójść w kierunku tzw. intuicyjności. A o to nietrudno, bo niewiele wiedząc o przyszłej pracy rzeczywistego użytkownika z nieistniejącym jeszcze systemem, może jedynie wyobrażać sobie, co byłoby dla użytkownika wygodne. Przeprowadzi więc swoistą projekcję własnych oczekiwań, a jeśli nie jest specjalistą w dziedzinie, której dotyczy system, niewykluczone iż zapragnie, by program podpowiadał mu to, co dla fachowca jest skądinąd oczywiste. Bardzo łatwo tu o przesadę, o „przesłodzenie” interfejsu.

I w ten właśnie sposób doszliśmy do trzeciego poważnego błędu popełnionego ongiś przez Microsoft. Podobno firma testowała swego czasu interfejs Windows korzystając z pomocy dzieci, które bawiły się oprogramowaniem. Jedna kamera filmowała przy tym obraz na ekranie, a druga miny dzieci. W ten sposób badano, kiedy młodzi recenzenci uśmiechali się, a kiedy krzywili i wyciągano odpowiednie wnioski. Powstaje jednak zasadnicze pytanie: czy Windows ma być systemem dla dzieci? Wspomnieliśmy już, że naszym zdaniem firma pomyliła narzędzia z zabawkami. Pomyliła też ergonomię z intuicyjnością. Klasycznym przykładem takiej pomyłki jest menu programów należących do pakietu MS Office, które próbuje samo odgadywać preferencje użytkownika i dostosowywać się do nich – ostatnio używane opcje znajdują się na wierzchu, by sięgnąć do innych trzeba rozwinąć odpowiednią listę. Niby dobry pomysł, ale z ergonomicznością nie ma nic wspólnego! Wykonując prawdziwą pracę, a więc czynności długotrwałe, powtarzalne, częstotliwe, nabieramy pewnych odruchów i nawet jeśli nie są one wyćwiczone do tego stopnia, by używać samych gorących klawiszy, to i tak mniej więcej pamiętamy, gdzie co jest w menu. A tu menu

wciąż się zmienia! Próbuje w lot odgadnąć nasze potrzeby, tyle że na ogół robi to źle, bo niby skąd naprawdę mogłoby wiedzieć, jakiej opcji potrzebujemy użyć w danej chwili? W efekcie pozorna intuicyjność przekłada się na ewidentny regres w zakresie ergonomiczności. Można tylko mieć nadzieję, że Microsoft w końcu to zrozumie i zaprzestanie eksperymentów na dzieciach, zwłaszcza tam gdzie bynajmniej nie chodzi o zabawki.

Ergonomicznie poprawne podejście do konstruowania menu każe nie tylko zadbać o jego wizualną stabilność (przed opisanym powyżej wynalazkiem Microsoftu ten temat w ogóle nie istniał), ale również o to, by nie zawierało ono zbyt dużej ilości poleceń. Starożytni mawiali: *Omne trinum perfectum*. Różne badania psychologiczne pokazują, że ludzie w istocie najłatwiej liczą do trzech. Potem muszą już pokonać pewną barierę abstrakcji, a to kosztuje czas i wysiłek. Obiekty zgrupowane po trzy dostrzega się natychmiast i błyskawicznie przetwarza się zawartą w nich informację. Stąd właśnie zalecenia, by poszczególne trójki oddzielać od siebie jakimiś separatorami, stosować zagnieżdżenia itp., byle tylko na raz nie przekroczyć owej magicznej liczby. A jeśli już naprawdę trzeba ją przekroczyć, to chyba każdy przyzna, że łatwiej się połapać widząc na liście cztery pozycje, niż np. czternaście.

Od psychiki użytkownika przejdźmy teraz do psychiki ergonomisty. Naprawdę niełatwo nim być... Naturalne poczucie solidarności z kolegami programistami każe mu bronić produktu ich pracy wobec narzekającego użytkownika, zaś zgłaszane przez niego problemy traktować jako błahe i mało istotne. A przecież chodzi o coś dokładnie przeciwnego! Zdarzyło się kiedyś, że pewien ergonomista został wysłany do klienta, by kilka godzin spędzić wspólnie z osobami wykonującymi proste, ale żmudne prace. Cel wizyty był oczywisty. Szkoda mu było jednak czasu, więc rzucił tylko okiem na to, co robią owe osoby, zamienił z nimi parę słów, usłyszał, że się przyzwyczaiły i jakoś sobie radzą, a następnie zajął się innymi sprawami, jako że pełnił w projekcie jeszcze jedną, ważną funkcję. Po powrocie do firmy stwierdził, że wszystko jest zasadniczo w porządku i dopiero „przyciśnięty do muru” wyznał, iż pewien drobny problem polega na tym, że kilka osób w 95% przypadków kopiuje na jednej z formatek określoną wartość do pięciu pól. Wyobraźmy więc sobie to kopiowanie: CTRL+C oraz cztery razy CTRL+V po odpowiednich przejściach tabulatorem lub alternatywnie, myszką. I to wszystko kilkaset razy dziennie! Czy trzeba jeszcze komentarza?

Wróćmy jednak do tego, co ergonomista uczynić może na etapie samego projektowania interfejsu. Okazuje się, że ma on tu do dyspozycji potężne narzędzie – estetykę.

7. Ładne jest piękne

Przywołajmy raz jeszcze kwestię krzesła, choć tym razem spójrzmy na nią od nieco innej strony. Stanisław Wyspiański zaprojektował ongiś wystrój wnętrza kamienicy należącej do Towarzystwa Lekarskiego w Krakowie. Znajduje się ona przy ulicy Radziwiłłowskiej, niedaleko dworca kolejowego, ale już po zewnętrznej stronie Plant. Stanowi własność prywatną i nie jest udostępniana turystom, ale wrzucając kilka odpowiednich słów w wyszukiwarce Google, można z łatwością znaleźć bardzo ciekawe zdjęcia przepięknych witraży, czy klatki schodowej ze wspianymi roślinnymi ornamentami, zdobiącymi metalowe poręcze. W sali posiedzeń znajdują się z kolei krzesła, na pierwszy rzut oka bardzo ładne i jak na owe czasy nawet względnie wygodne, gdyby nie podłokietniki... Są półokrągłe, więc gdy położy się na nich ręce, natychmiast zsuwają się one w dół. Z drugiej strony nie da się rąk nie oprzeć, bo wtedy podłokietniki strasznie przeszkadzają. Złośliwi twierdzą nawet, że Wyspiański świadomie tak zaprojektował owe nieszczęsne meble, by posiedzenia Towarzystwa nie trwały zbyt długo.

Warto zadać sobie w tym miejscu pytanie o wzajemne relacje estetyki i ergonomiczności. W kontekście mebli wydają się one dosyć luźne, a nawet można odnieść wrażenie, że jedna wyklucza lub przynajmniej istotnie ogranicza drugą. Jak jednak dzieje się w przypadku oprogramowania, czy ściślej, jego interfejsu? Przecież na interfejs tak naprawdę tylko patrzymy, bo to czego

dotykamy, to sprzęt: klawiatura, myszka. Estetyce i ergonomiczności jest więc tu siłą rzeczy znacznie bardziej po drodze. Jeżeli część pracy wiąże się z samym patrzeniem, to bez wątpienia przyjemniej się pracuje patrząc na coś ładnego. Oczywiście, ktoś mógłby tu słusznie zauważyć, że kategorie estetyczne nie należą do obiektywnych, zaś *de gustibus est non disputandum*. W istocie, gust niemal z definicji jest tym, co ma każdy z nas, a czego ewentualnie nie mają inni (czy słyszano kiedy, by ktoś powiedział sam o sobie: *ja nie mam gustu?*). Tak więc interfejs, który spodoba się jednemu, wcale nie musi się podobać drugiemu, choć traktując rzecz statystycznie, są takie interfejsy, które większości osób się podobają i takie, które większości się nie podobają. Zapewne więc właśnie statystycznie należałoby całą rzecz potraktować i zadbać o to, by interfejs podobał się większości użytkowników. A jak to uczynić? Najlepiej poprosić o pomoc plastyka, ale prawdziwego, z talentem i odpowiednim wykształceniem, nie zaś osobę, która opanowała tajniki Corela lub Photoshopa. Przypominamy w tym miejscu, że malarzem nie jest bynajmniej ten, kto nauczył się sprawnie napinać płótno na blejtram, gruntować je, bądź czyścić pędzle terpentyną! W każdym razie, dobry plastyk, jeśli postawić przed nim odpowiednie zadanie, potrafi zaproponować elementy graficzne, które spodobają się zarówno maluczkim, jak i koneserom, choć zapewne nie będą nadawać się na wystawę i nie staną się wielkim dziełem sztuki. Co więcej, jeśli elementy te nie będą w oczywisty sposób przeszkadzać w pracy, to będą w niej pomagać! Jeżeli tylko np. ładne (samo w sobie) tło jakiejś formatki nie będzie na tyle agresywne, by utrudniać dostrzeżenie napisów na pierwszym planie, to na pewno uprzyjemni pracę patrzącemu, a więc samo w sobie będzie ergonomiczne! Z drugiej strony, jeśli będzie ono agresywne, to najprawdopodobniej nastąpi inny efekt: przytłoczy wspomniane napisy i przez to właśnie wyda się nieestetyczne, brzydkie – oczywiście na formatce, bo użyte w jakimś innym kontekście znów będzie ładne. Płyną z tego dwa wnioski. Po pierwsze, estetyka interfejsu podnosi ergonomiczność aplikacji, choć oczywiście nie jest remedium na wszystkie problemy (vide przykład z listonoszem). Po drugie, nasz plastyk musi bardzo ściśle współpracować z programistą tworzącym interfejs, by efekty swojej pracy mógł dostrzec natychmiast w postaci konkretnych elementów programu i od razu z łatwością nanosił na nie niezbędne korekty. Ergonomista z kolei będzie zapewne kierował pod adresem plastyka bardzo konkretne dezyderaty. Być może np. uzna za stosowne zaakcentować jakiś element na formatce, bądź raporcie, tak by od razu rzucił się on użytkownikowi w oczy. Głowa plastyka w tym, by zrobić to w sposób skuteczny i estetyczny, by uzyskać właściwy efekt, nie przesadzić i zarazem niczego nie zepsuć.

Warto w tym miejscu na marginesie zauważyć, że wielu projektantów interfejsów nie dostrzegło jeszcze, iż przeważająca większość użytkowników posługuje się dziś monitorami kolorowymi. Kolor zaś jest nie tylko środkiem wyrazu artystycznego, ale również ważnym nośnikiem informacji. Korzystamy z tego faktu codziennie, przechodząc lub przejeżdżając przez skrzyżowania, lecz jakoś nie zawsze potrafimy wykorzystać go podczas projektowania komunikacji komputer – użytkownik. Sensownie kolorowy (nie pstrokaty!) interfejs przyspiesza przyswajanie przez człowieka informacji wyświetlanej na ekranie, a tym samym zwiększa ergonomiczność aplikacji.

8. Nowe przeglądarki cesarza

Chwała firmie Apple za wymyślenie okienek, a firmie Microsoft – za ich rozpropagowanie. Wydaje się, że w zakresie interfejsów nie wynaleziono dotychczas nic lepszego. Wynaleziono natomiast coś bardzo oryginalnego, a tym czymś są przeglądarki internetowe. Poniżej parę słów o nich, kilka oczywistości, które, jak się wydaje, przestały być oczywiste...

Sama nazwa zdaje się wskazywać, że przeglądarki powinny służyć do przeglądania stron WWW i zapewne z takim zamysłem je ongiś stworzono. Bardzo szybko okazało się jednak, że dostawcy treści internetowych chcieliby posiadać pewien kanał zwrotny do komunikacji z użytkownikiem, a kanałem tym bynajmniej nie powinna być tylko poczta elektroniczna. Tak właśnie powstały pierwsze interfejsy przeglądarkowe, czyli strony interaktywne, na których użytkownik (dawniej zaledwie widz) mógł zaznaczać, bądź wybierać pewne wartości, czy wpisywać dane po

to, by następnie jednym kliknięciem przesłać je do dostawcy treści. Dostawca z kolei, czyli w istocie program komputerowy, mógł po przetworzeniu danych wyświetlić inne informacje itd. Wszystko odbywało się – dodajmy – w oknie przeglądarki, bez potrzeby instalowania po stronie użytkownika jakichkolwiek aplikacji.

Zawrotna kariera Internetu jako medium bardzo szeroko rozumianej komunikacji, biznesu itd. sprawiła, że rosło zapotrzebowanie na aplikacje internetowe, a wraz z nim na narzędzia do tworzenia tych aplikacji oraz na programistów, którzy tymi narzędziami będą umieli się posługiwać. I tu właśnie branża informatyczna postanowiła pokusić się o redukcję kosztów poprzez wmówienie użytkownikom, że interfejsy przeglądarkowe są lepsze od okienkowych, również wtedy, gdy nie mamy do czynienia z bliżej nieokreśloną liczbą korzystających, rozsiansych po całym świecie, lecz z bardzo konkretnymi operatorami, np. pracownikami przedsiębiorstwa lub urzędu. Na czym polegała redukcja kosztów – otóż na tym, że wystarczyło skoncentrować się na narzędziach i technologiach do owych „internetowych” zastosowań, porzucając zwykłe okienka. Oczywiście trzeba było użyć jakichś argumentów wobec klientów, więc wmówiono im, że interfejsy przeglądarkowe są nowocześniejsze (cokolwiek miałyby to oznaczać), że tylko one mają przyszłość i że łatwiej jest administrować systemem, bo nic poza przeglądarką nie instaluje się na komputerach użytkownika, w związku z czym całością zarządza się bardzo sprawnie z jednego miejsca.

Ile w tym prawdy? Niestety, bardzo niewiele. Jeśli odrzucić mało konkretny argument o nowoczesności, łatwość administrowania pozostaje kwestią wysoce dyskusyjną. Dotyczy ona praktycznie tylko aktualizacji wersji, a mianowicie tego, że nie trzeba chodzić z płytką od komputera do komputera, ale wystarczy podmienić oprogramowanie na serwerze. Tak się jednak składa, że automatyczną, zdalną aktualizację wymyślono całkiem niezależnie od przeglądarek, a bodaj najpopularniejszym programem, który potrafi się sam aktualizować jest ni mniej ni więcej, tylko MS Windows. Ponadto każdy ma dziś pełną świadomość, że skazanie użytkownika na aplikację wykorzystującą czysty HTML byłoby z punktu widzenia ergonomiczności oprogramowania po prostu nieludzkie. Potrzebne są więc wirtualne maszyny Javy, tudzież różnego rodzaju dodatki w postaci specjalnych wtyczek (plugins), które żyją swoim życiem, a więc mają własne wersje, niezależne od wersji przeglądarek. Tych zresztą też jest całkiem sporo – i wersji i samych przeglądarek – a każda z nich posiada jeszcze wiele różnych ustawień, którymi użytkownik niejedno może zepsuć. W efekcie stworzenie aplikacji, która będzie poprawnie działać na kilku różnych przeglądarkach, jest nie lada wyzwaniem, wobec czego i zamawiający i dostawcy świadomie się tu ograniczają. Znana jest historia, która przed laty wydarzyła się w jednym z polskich banków. Zamówiono tam aplikację przeglądarkową, która miała działać pod IE4. Po odebraniu i przetestowaniu owej aplikacji przeprowadzono podczas weekendu aktualizację Explorera z wersji 3 do 4, a następnie w poniedziałek rano okazało się, że przestała działać inna ważna aplikacja, która funkcjonowała jedynie pod IE3. Tak się składa, iż tzw. „downgrade” przeglądarki raczej nie wchodzi w rachubę...

Przeglądarka wraz ze swoim wtyczkowym otoczeniem staje się więc de facto powoli systemem operacyjnym, który uruchamia się na innym systemie operacyjnym i który pozwala na korzystanie z tzw. aplikacji przeglądarkowych. Te z kolei do przetwarzania danych wykorzystują serwery aplikacji, centralizując to, co dawniej z powodzeniem rozpraszano po komputerach użytkowników. Serwery aplikacji muszą być więc bardzo mocne, a więc i drogie, podczas gdy moc obliczeniowa stacji roboczych zazwyczaj po prostu się marnuje, bo na dwa systemy operacyjne, tj. ten podstawowy i przeglądarkę, jest jej na ogół aż nadto.

A jak ma się to wszystko do ergonomiczności? Otóż tak, że typowe interfejsy przeglądarkowe są z reguły znacznie mniej wygodne od okienkowych. Alternatywnie, by w przeglądarce zrobić coś naprawdę wygodnego, czyli przypominającego okienka, trzeba się nieźle napracować i od razu na wstępie zapomnieć o sympatycznych scenkach z pokazów dostawców narzędzi, którzy demonstrowali, jak to z modelu UML za jednym naciśnięciem guzika generuje się gotowa aplikacja – ku uciesze programisty i użytkownika, oczywiście.

Jak to więc możliwe, że technologia z punktu widzenia użytkownika ewidentnie lepsza wypierana jest przez gorszą? Egzemplifikacja prawa Kopernika-Greshama? Raczej nie. Mechanizm bliższy jest temu, który każe np. młodym dziewczynom chodzić podczas upałów w brzydkich i ciężkich buciorach. Moda! A moda kreowana być może przez producentów butów (aby klientki zmieniały je co sezon) lub przez producentów narzędzi informatycznych (aby ograniczyć ilość linii produktowych i zredukować koszty), ale w obu przypadkach jest modą i z korzyścią dla użytkownika końcowego, a więc i z ergonomicznością nie ma kompletnie nic wspólnego. Na szczęście mody także przemijają, więc przeminie zapewne również obsesja przeglądarek. Póki co, można jedynie krzyknąć: *Cesarz jest nagi!*

9. Que sera, sera...

Czy jednak okienka są rozwiązaniem najlepszym i ostatecznym? Zapewne nie. Zresztą twierdzenie, jakoby było inaczej, stanowiłoby szczyt nieostrożności. Wydaje się, że, niespełnionym jak na razie, marzeniem jest werbalna komunikacja z maszyną, oczywiście w języku naturalnym. Maszyna wykonywałaby wtedy polecenia człowieka jak dobry i zmyślny służący. Problem z konstrukcją takiego interfejsu nie polega jednak bynajmniej tylko na trudnościach technicznych związanych z przetwarzaniem przez maszynę tego, co usłyszy, ale raczej na umiejętności człowieka wyrażania się na tyle precyzyjnie, by jakakolwiek maszyna była w stanie pojąć jego rzeczywiste intencje. Więcej nawet, człowiek w ogóle ma problem ze sformułowaniem swych intencji na poziomie myśli, a więc nie tyle zwerbalizowaniem myśli, co nawet jej skonkretyzowaniem. Niektórzy zresztą uważają, że jest to jedno i to samo, bo tylko kategorie językowe umożliwiają sprecyzowanie myślenia. Jakby nie było, rzecz w tym, że w istocie ludziom bardzo trudno porozumieć się słowami między sobą – porozumienie się z maszyną może być tylko jeszcze bardziej skomplikowane.

Mimo to wierzymy, że interfejsy użytkownika będą się rozwijać i będą stawać się coraz bardziej ergonomiczne, nawet jeśli przejściowo wystąpią niewielkie regresy tego ogólnego trendu. I dobrze, bo użytkownicy mają prawo wymagać. Z drugiej jednak strony powinni mieć świadomość, że wymagania w stosunku do maszyn i systemów nie mogą być przesadne. System informatyczny to jednak tylko narzędzie ułatwiające pracę, a nie skrzyneczka z której wysypują się pieniądze. Dlatego spotykane niekiedy zarzuty typu: *Ale przecież tu trzeba wprowadzać dane! Kto to będzie robił?!* nie zasługują na obszerniejszy komentarz, jako że z ergonomicznością lub jej brakiem nie mają zupełnie nic wspólnego. U początku lat dziewięćdziesiątych w pewnej firmie pracownicy nieustannie narzekali na komputery i oprogramowanie, a czynili to w sposób tak irytujący, że któregoś dnia szef nakazał przynieść z piwnicy maszyny do pisania i zastąpić nimi wszystkie PC-ty. Pierwszą reakcją było niedowierzanie, następnie przyszedł szok, a potem do gabinetu owego szefa zapukała delegacja z prośbą, by zmienił swoją decyzję. Wydaje się więc, że w ciągłej walce o coraz większy komfort pracy warto czasem docenić to, co się już osiągnęło.

10. Dekalog Ergo-informatyka

Artykuł ten należałoby uznać za skrajnie nieergonomiczny dla czytelnika, jeśli ten musiałby go przeczytać np. dwa razy, by wydobyć najważniejsze treści. Postanowiliśmy więc zadbać o to, by do tego nie doszło i by – jako czytany tylko raz – w ogóle nie stał się przedmiotem zainteresowania ergonomii. Stąd też zebraliśmy na koniec pewne ogólne stwierdzenia i wskazówki, stanowiące de facto podsumowanie wcześniejszych rozważań. Przy niewielkim wysiłku autora mogłoby ich być nieco więcej lub mniej, ale jest dziesięć – głównie dlatego, że liczba ta dobrze się kojarzy i łatwo zapada w pamięć. Jeśli ktoś lubi pojęcie „intuicyjność”, o którym wspominaliśmy na początku, to owe „dziesięć” stanowi ukłon w jego stronę.

A oto i nasz dekalog:

1. O ergonomiczności czegokolwiek można mówić tylko wtedy, gdy przy pomocy tego czegoś wykonuje się czynności długotrwałe, wielokrotne, powtarzalne.
2. Problemy ergonomiczności systemów informatycznych zauważane są dopiero na późnych etapach wdrożeń i mają znaczny wpływ na rzeczywisty sukces lub porażkę projektów.
3. Ergonomiczność oprogramowania to ergonomiczność interfejsu użytkownika – sami użytkownicy uważają, że oprogramowanie składa się wyłącznie z interfejsu.
4. Ergonomiczności interfejsu w żadnym razie nie należy mylić z jego „intuicyjnością”, która sama w sobie jest pojęciem niejasnym i niezbyt dobrze określonym.
5. Kwestie ergonomiczności dotyczą najdrobniejszych detali interfejsu, wobec czego są i muszą być pomijane przez analityków.
6. Projektanci i programiści nie są rzeczywistymi użytkownikami tworzonego przez siebie oprogramowania, w związku z czym mają spalone pojęcie o jego ergonomiczności.
7. Dlatego w skład zespołu projektowego powinien wchodzić specjalny pracownik – ergonomista, który wcale nie ma łatwego zadania, za to łatwo może zejść na manowce.
8. W miarę możliwości ergonomista powinien obserwować rzeczywistych użytkowników, współpracować z nimi i wsłuchiwać się w ich opinie, a zwłaszcza narzekania.
9. Ergonomiczność i estetyka interfejsu idą ze sobą w parze, wobec czego w jego tworzenie powinien być również mocno zaangażowany profesjonalny plastyk.
10. W walce o ergonomiczność nie należy ulegać chwilowym modom, np. na przeglądarki, gdyż mody te kreowane są w interesie dostawców, nie zaś odbiorców oprogramowania.

Na koniec warto uświadomić sobie jeszcze jedną rzecz. Otóż bardzo często bywa tak, że poprawki ergonomiczne okazują się stosunkowo mało pracochłonne, zaś ich realne efekty są wprost nie do przecenienia. Nieodżałowany Stanisław Lem zdefiniował ongiś w *Cyberiadzie* jednostkę szczęścia: jeden hedon to szczęście, jakiego doznaje człowiek, który przeszedł kilometr z gwoździem w butcie, a następnie ten gwóźdź z buta wyciągnął. Miejmy to na uwadze i nie odkładajmy owych poprawek na później, gdyż 95% efektu za 5% pracy to dobry interes, a użytkownik uszczęśliwiony w tak prosty i przyziemny sposób, jak wyciągnięcie gwoździa z buta, wybaczy nam później niejedną poważną dysfunkcję systemu.