

# Jacek Unold

---

## Web 2.0 w tworzeniu globalnej platformy społeczeństwa informacyjnego

---

Ekonomiczne Problemy Usług nr 67, 631-639

---

2011

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej [bazhum.muzhp.pl](http://bazhum.muzhp.pl), gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

JACEK UNOLD

Uniwersytet Ekonomiczny we Wrocławiu

## WEB 2.0 W TWORZENIU GLOBALNEJ PLATFORMY SPOŁECZEŃSTWA INFORMACYJNEGO

### Wprowadzenie

W odróżnieniu od wczesnych ujęć, kojarzących cyberprzestrzeń z wirtualną rzeczywistością, w dzisiejszym rozumieniu cyberprzestrzeń jest utożsamiana z rzeczywistością cyfrową stworzoną w globalnej sieci WWW. Nadejście ery Web 2.0 mocno ugruntowało tę zależność pojęciową i otworzyło nowe, nieznane wcześniej możliwości w zakresie wykorzystania platform technologicznych. Jedno z podstawowych haseł programowych Web 2.0 brzmi: „Web jako platforma” (*Web as a platform*). W ślad za tym następuje wyraźne przesunięcie akcentów z aplikacji desktopowych, ograniczonych do indywidualnego komputera osobistego, na aplikacje sieciowe czy szerzej – webowe.

W niniejszym artykule przedstawiono te aspekty funkcjonowania drugiej generacji WWW, tzw. Web 2.0, które decydują o tworzeniu globalnej platformy technologicznej na potrzeby komunikacyjne współczesnego społeczeństwa informacyjnego.

### 1. Konceptyjne wzorce drugiej generacji WWW

Idea Web 2.0 jest charakteryzowana przez osiem zasadniczych wzorców konceptyjnych, wzajemnie powiązanych i warunkujących aplikacyjny sukces tej drugiej generacji Globalnej Sieci<sup>1</sup>:

1. Wykorzystanie zbiorowej inteligencji (*Harnessing Collective Intelligence*).

---

<sup>1</sup> J. Musser: *Web 2.0: Principles and Best Practices*, O'Reilly Media, Inc. 2007.

2. Dane jako następny *Intel* w środku (*Data is the Next „Intel Inside”*).
3. Innowacja poprzez remiks (*Innovation in Assembly*).
4. Bogate doświadczenia użytkownika (*Rich User Experiences*).
5. Oprogramowanie łączące urządzenia (*Software Above the Level of a Single Device*).
6. Ustawiczny stan beta (*Perpetual Beta*).
7. Wykorzystanie tzw. długiego ogona (*Leveraging the Long Tail*).
8. Lekkie modele i opłacalna skalowalność (*Lightweight Models and Cost-Effective Scalability*).

Spośród nich aż trzy można odnieść bezpośrednio do motywu „Web jako platforma”. Wzorzec nr 5, „oprogramowanie łączące urządzenia”, wskazuje na konieczność tworzenia takich kategorii oprogramowania, które łączą różne urządzenia w Sieci i tworzą nową wartość, opartą na rosnącym znaczeniu doświadczeń *on-line* i łączeniu ich ze strategiami desktopowymi. Jest to zatem naturalne rozwinięcie idei Web 2.0 jako platformy. Jeśli zaś chodzi o praktyczną aplikację tej koncepcji, to powoduje ona, że oprogramowanie przestaje być produktem sprzedawanym w pakietach, a staje się usługą, dostępną w sieci i bezpłatną. Hasło „oprogramowanie jako usługa” (*Software as a Service – SaaS*) stało się kolejnym motywem przewodnim ery Web 2.0.

Szósty wzorzec to „ustawiczny stan beta”. W informatyce stan beta oznacza etap rozwoju oprogramowania bądź systemu. Koncepcja *Perpetual Beta* zakłada odejście od tradycyjnych modeli rozwoju softwaru na rzecz modeli *on-line*, nieustannie udoskonalanych, w których oprogramowanie traktowane jest jako usługa. Niezwykle istotny jest tu współudział użytkownika w procesie projektowania i doskonalenia.

Wreszcie, ósmy wzorzec rozwoju Web 2.0, „lekkie modele i opłacalna skalowalność”, to wykorzystanie tzw. lekkich modeli biznesu i rozwoju oprogramowania, w których istotna jest m.in. skalowalność. Wzorzec ten pozwala na szerokie wykorzystanie kolejnej wiodącej koncepcji Web 2.0 – „innowacja w montażu”.

## 2. Nowe koncepcje rozwoju oprogramowania

Cyberprzestrzeń jest najodpowiedniejszym miejscem do praktycznej aplikacji koncepcji „oprogramowania ponad pojedynczym urządzeniem” oraz „oprogramowania jako usługi”. Każda aplikacja webowa jest realizacją pierwszej z tych idei, gdyż nawet najprostsze rozwiązanie angażuje co najmniej dwa komputery, jeden pełniący rolę serwera, drugi – klienta (np. przeglądarki lub wyszukiwarki). Co więcej, rozwój modelu komunikacji P2P (*peer-to-peer – równy z równym*) powoduje, że obie strony procesu komunikują się na równych prawach, a każdy komputer w sieci może jednocześnie pełnić funkcję klienta, jak i serwera. Obecnie najpopular-

niejszą formą implementacji P2P są programy do wymiany plików. Taka wymiana jest prowadzona bez pośrednictwa centralnego serwera, a każdy komputer użytkownika jest modelowym węzłem sieci. Jako serwer komputer przyjmuje połączenia od innych użytkowników sieci i udostępnia zgromadzone dane, a jako klient – pobiera dane z innych komputerów. Sieć P2P posiada płynną i dynamiczną strukturę, zmieniającą się w zależności od liczby zalogowanych użytkowników i zachodzących aktualnie procesów wymiany danych. „Oprogramowaniem łączącym urządzenia” jest tu zatem oprogramowanie do wymiany plików.

Bezpośrednim skutkiem aplikacji zasady „wszechobecnego oprogramowania” jest możliwość potraktowania „programu jako usługi” (*SaaS*). *SaaS* jako sieciowy model dystrybucji aplikacji programowych, w zasadniczy sposób przyczynił się do rozwoju syndykacji w cyberprzestrzeni. W *SaaS* twórcą jest najczęściej programista piszący oprogramowanie biznesowe lub niezależny handlarz oprogramowaniem, a nie sam konsument, bloger, użytkownik YouTube’a czy innej tego typu witryny. Taki twórca rozwija webową aplikację, obsługuje ją i udostępnia. Czyni to albo samodzielnie, albo korzystając z pomocy innych. A sam syndykator (dystrybutor) jest najczęściej dostawcą platformy oferującej *SaaS*, na przykład *Salesforce.com*. *Salesforce.com* jest dziś czołowym dostawcą *SaaS*, wielu wręcz uważa tę witrynę za symbol aplikacji traktującej oprogramowanie jako usługę.

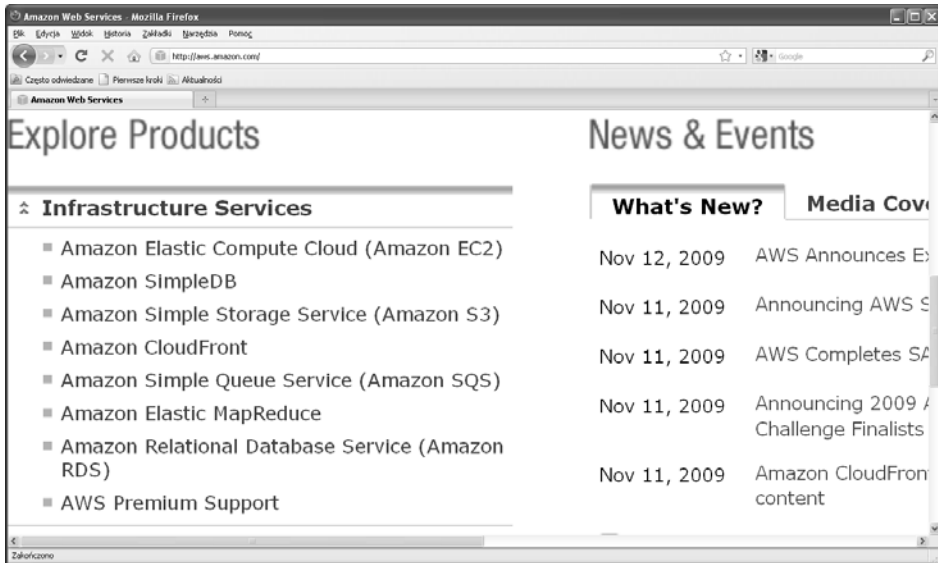
Co najistotniejsze, użytkownik płaci za używanie aplikacji tylko wtedy, gdy jej rzeczywiście potrzebuje (*service-on-demand*), zamiast kupować dane oprogramowanie w formie produktu (pakietu programowego) bądź licencji. To zasadniczo redukuje koszty wstępne i dlatego jest szczególnie atrakcyjne dla użytkowników indywidualnych oraz dla małego biznesu. Ponieważ w aplikacjach *SaaS* płaci się za faktyczne korzystanie, identyfikuje się tu silny pozytywny efekt sieciowy.

### 3. Rozwój usług webowych

W obszarze *SaaS*, oprócz aplikacji zapewniających określoną funkcjonalność, pojawiła się inna kategoria usług infrastrukturalnych. Są to na przykład usługi *software’owe*, które oferują przestrzeń magazynową w cyberprzestrzeni. Jednym z najbardziej znanych dostawców jest tu Amazon S3 (*Simple Storage Service*). Jest to usługa *on-line*, która umożliwia składowanie danych na przestrzeni dyskowej serwerów Amazon. W zależności od potrzeb dane mogą być składowane w USA bądź w Europie. Usługa S3 jest wyposażona w interfejs programowania aplikacji API, dzięki któremu użytkownik może w prosty sposób operować na składowanych i udostępnianych zasobach informacyjnych. Amazon S3 wycenia swoją usługę na 15 centów za gigabajt na miesiąc. Innym dostawcą podobnej usługi jest Omnidrive.

Kolejną kategorią usług związanych z możliwością użytkowania infrastruktury dzięki *SaaS* oraz API jest dzierżawa mocy obliczeniowej. W praktyce oznacza to

możliwość wynajmowania serwerów na godziny, również niemal za symboliczną opłatą. Najbardziej znanym dostawcą jest także Amazon, tym razem w postaci Amazon EC2 (*Elastic Compute Cloud*). Na rysunku 1 przedstawiono fragment strony Amazon z pełną ofertą infrastrukturalnych usług webowych opartych na API.



Rys. 1. Webowe serwisy infrastrukturalne Amazon.com

Źródło: <http://aws.amazon.com>

Pozytywne doświadczenia z interfejsem oprogramowania aplikacji sprawiły, że obecnie zaleca się, aby każda usługa oferowana w cyberprzestrzeni była wyposażona w API<sup>2</sup>. W erze komputerów stacjonarnych interfejsy użytkownika były dostarczane przez sprzedawców systemów operacyjnych, tj. przez Microsoft, Apple lub RedHat. Obecnie, w erze Web 2.0, API pochodzą od Amazon.com, Ebay albo Google. API są podstawą działania każdej skutecznej strategii platformy programowej, gdyż sprzyjają innowacjom dokonywanym przez osoby trzecie. Najlepsze strategię „platformerskie” łączą zalety zasadniczego biznesu oferowanego przez daną platformę z możliwościami remiksu dokonywanego przez innych. Na przykład Amazon.com skojarzył swój API z programem stowarzyszonym (afiliowanym). Oznacza to, że za każdym razem, kiedy dochodzi do sprzedaży za pośrednictwem afiliowanej strony, zysk jest dzielony między Amazon.com i właściciela tej stowarzyszonej strony.

<sup>2</sup> J. Unold: *Teoretyczno-metodologiczne podstawy przetwarzania informacji w cyberprzestrzeni*, Wydawnictwo Naukowe Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław 2011.

Strategia platformy wraz z możliwościami technicznymi oferowanymi przez API stały się też podstawą sukcesu Flickr.com. Firma bardzo szybko znalazła się w samym centrum ekosystemu cyfrowej fotografii w cyberprzestrzeni, ekosystemu złożonego z partnerów, aplikacji wykonywanych przez osoby trzecie, blogerów i twórców mashupów. W strategii tej oparto się na wykorzystaniu API typu „od odczytu do zapisu” (*read-write*), syndykacji danych, zewnętrznej integracji i szerokim zastosowaniu urządzeń przenośnych, głównie cyfrowych aparatów fotograficznych.

#### 4. Otwartość i nieustający stan beta

Kiedy urządzenia i oprogramowanie zostają podłączone do Internetu, aplikacje przestają być produktami programowymi, stają się nieustającą usługą. Ma to zasadniczy wpływ na proces rozwoju oprogramowania. Zamiast tworzyć gotowe i co jakiś czas uaktualniane pakiety, regularnie dodawane są nowe możliwości, będące wynikiem bogatego doświadczenia użytkowników. Użytkownicy są zaangażowani w charakterze osób testujących oprogramowanie w czasie rzeczywistym, a informacje dotyczące rzeczywistego wykorzystania oferowanej aplikacji programowej są na bieżąco wykorzystywane w procesie stałego jej doskonalenia. Rzeczywiste zachowania realnych użytkowników dostarczają znacznie dokładniejszych założeń modelowych do oceny oferowanego produktu czy usługi niż badania marketingowe, prototypy itd. W rezultacie oprogramowanie dostępne *on-line* znajduje się w „ustawicznej fazie beta”, a nienaganne funkcjonowanie aplikacji staje się podstawowym wymogiem kompetencyjnym. Rozwój koncepcji Web 2.0 nieuchronnie przybliży ostateczny koniec tradycyjnego cyklu adopcji oprogramowania. Nikt dziś nie pyta, która to wersja Google’a. Oczekuje się, że tego typu serwis zawsze jest na miejscu i automatycznie się doskonali z upływem czasu. Nie ma kolejnych wersji, instalacji ani uaktualnień. Utrzymywanie ustawicznej i otwartej dla innych fazy beta pozwala milionom użytkowników pracować nad oprogramowaniem, które zmienia się i doskonali w sposób ciągły.

Charakterystyczny jest tu przykład firmy *Slim Devices*, producenta sprzętu do nagrywania i odtwarzania muzyki w formacie MP3. Kiedy spostrzeżono, że hakerzy włamują się do oprogramowania i dokonują w nim różnych zmian, dostrzeżono w tym szansę. *Slim Devices* zainicjował powstanie otwartej społeczności programistów, którzy codziennie dokonywali uaktualnień oprogramowania, oferowanego następnie w Sieci do ściągnięcia. W dalszym etapie utworzono listy dyskusyjne, umożliwiające współpracę personelu z klientami, a nawet formularz zgłoszeń dostrzeżonych wirusów.

Podobnie, kiedy hakerzy zaatakowali witrynę *Google Maps*, jeszcze przed udostępnieniem oficjalnego API, firma, zamiast działań zabezpieczających czy odwetowych, po prostu otworzyła tę aplikację. Jednym z wielu skutków był bardzo

wyraźny efekt psychologiczny – użytkownicy i zainteresowani programiści, często niedawni hakerzy, od razu nabrali olbrzymiego zaufania do firmy i w zdecydowanej większości zaczęli pracować dla jej dalszego rozwoju.

## 5. Skalowalność, innowacyjność i rozwój lekkich modeli biznesu

Kolejny wzorzec technologiczny Web 2.0 wskazuje na wykorzystanie „lekkich modeli biznesu i rozwoju oprogramowania”, co pozwala na optymalizację skalowalności oraz rozwój „innowacji w montażu”. Innowacja taka bazuje na istnieniu wielu komponentów, które można zestawić w różnorodnych, nieznanych wcześniej konfiguracjach, tworząc w ten sposób zupełnie nową wartość. W epoce komputerów osobistych taką strategię zastosował Dell w obszarze sprzętu, zestawiając komplety PC zgodnie z wymogami indywidualnych klientów. Pokonał w ten sposób konkurencję, skupioną raczej na innowacyjności w procesie rozwoju produktu. W erze Web 2.0 ta zasada odnosi się do oferowanych usług, a ściślej do wykorzystania i integracji usług oferowanych przez innych w celu wytworzenia zupełnie nowej usługi.

Jednym z przykładów udanej aplikacji zasady *Innovation in Assembly* jest coraz wyraźniejszy trend powstawania nowej generacji witryn internetowych – witryn, które w zasadzie nie posiadają... własnej witryny. Wykorzystuje się tu komponenty platform internetowych dostarczanych przez inne jednostki, a służących na przykład do składowania danych czy do dokonywania transakcji, strona logiczna zaś takiej usługi jest udostępniana poprzez kontekst danej przeglądarki. Najczęściej stosuje się tu technologię tworzenia aplikacji internetowych Ajax, w której interakcja użytkownika z serwerem odbywa się bez przeładowywania całego dokumentu, w sposób asynchroniczny. W rezultacie takie witryny istnieją bez tradycyjnej infrastruktury serwerowej. Do pionierskich rozwiązań w tym zakresie należy *Eventsites*.

Jednym z pierwszych przedsięwzięć w cyberprzestrzeni, kiedy przekształcono witrynę internetową w platformę webową, był Amazon.com. Powstała cała strategia platformy, obejmująca takie przedsięwzięcia, jak program afiliacyjny *Amazon Associate's*, *zShops* czy *Amazon Marketplace*. Jednym z ciekawszych przykładów jest oferta transkrypcji, na przykład z nagrania audio na tekst, oferowana przez *CastingWords.com*. Klient dostarcza cyfrowy podkast audio lub wideo i otrzymuje transkrypcję tekstową. Podcasting, podcast (*iPod + broadcast*), spolszczone: podkasting, podkast, jest formą internetowej publikacji dźwiękowej lub filmowej, przygotowywanej przy zastosowaniu technologii RSS. Podkast jest udostępniany najczęściej w postaci regularnych odcinków i przyjmuje formę bloga, audycji radiowej,

kursu językowego itd. Wiele amerykańskich uczelni udostępnia w takiej formie wybrane wykłady, muzykę lub audycje radia studenckiego<sup>3</sup>.

## 6. Web jako platforma

Hasło programowe „Web jako platforma” może być aplikowane w różnym zakresie i z różnym powodzeniem. Jednym z lepszych przykładów ilustrujących odmienne rozumienie pojęcia „platforma” może być porównanie filozofii rozwoju swojego produktu dwóch firm: Netscape i Google. Netscape rozumiał ideę „Web jako platforma” w kategoriach starego paradygmatu oprogramowania, tzn. ich sztandarowym produktem była przeglądarka zaprojektowana jako aplikacja dla desktopu. Strategia zaś ograniczała się do utrzymania dominującej pozycji na rynku przeglądarek w celu wykreowania popytu na drogie produkty serwerowe. Netscape miał nadzieję na zapewnienie sobie kontroli nad standardami prezentacji zawartości i aplikacjami wyszukiwarek, co miało zapewnić pozycję porównywalną z Microsoftem na rynku komputerów osobistych. Netscape promował zatem ideę „webtopa”, w miejsce dotychczasowego „desktopa” (komputera stacjonarnego). Podejście to porównywano czasami do działań we wczesnej erze motoryzacji, kiedy auto było projektowane w zasadzie jako „bezkonny powóz”. Z biegiem czasu jednak, zarówno przeglądarki webowe, jak i serwery stały się bardzo popularnym produktem, a prawdziwą wartość zaczęto odnajdywać w usługach dostarczanych na platformie webowej. A zatem nie tyle sama platforma technologiczna, co usługi na niej oferowane. Wraz z pojawieniem się nowych przeglądarek typu Firefoks czy Opera, pozycja Nawigatora Netscape zaczęła wyraźnie słabnąć, aż w grudniu 2007 roku ówczesny właściciel marki, AOL (*America Online*), ogłosił wstrzymanie prac nad rozwojem aplikacji internetowych<sup>4</sup>.

Z kolei wyszukiwarka Google powstała od razu jako aplikacja webowa i nigdy nie była typowym pakietowym *software'em*. Jest typowym przykładem SaaS, czyli oprogramowania funkcjonującego jako usługa, z rzeszą klientów płacących, bezpośrednio lub pośrednio, za korzystanie z tej usługi. Zamiast kolejnych wersji, nieustającego procesu doskonalenia, zamiast licencjonowania czy sprzedaży, po prostu użytkowanie. Co istotne, Google jest wyposażony w funkcję, której Netscape nigdy nie potrzebował: zarządzanie bazą danych. Google, oprócz tego, że stanowi zestaw narzędzi programowych, jest bowiem wysoce specjalistyczną bazą danych. Bez tych danych wszelkie narzędzia stałyby się bezużyteczne; choć również bez narzę-

---

<sup>3</sup> A. Shuen: *Web 2.0: A Strategy Guide*, O'ReillyMedia, Inc. 2009.

<sup>4</sup> B. Palser: *Give Patch a chance: controversy swirls around AOL's ambitious hyperlocal venture*, „American Journalism Review” vol. 32, iss. 4, s. 62(1), University of Maryland, December 22, 2010.



dzi dane stałyby się bezużyteczne. A zatem wartość oprogramowania staje się proporcjonalna do ilości i dynamiki danych.

## Podsumowanie

Zarysowana w artykule problematyka pozwala na sformułowanie podstawowych zaleceń odnośnie do efektywnej aplikacji zasady „Web jako platforma technologiczna”:

- oferowane produkty i usługi powinny integrować to, co najlepsze z aplikacji desktopowych i *on-line*,
- oferowane oprogramowanie powinno być dostępne w postaci serwisów webowych bądź API,
- należy oferować zewnętrzne API,
- zapisywanie się użytkownika do API powinno przebiegać samoobsługowo,
- należy nagradzać użytkowników za innowacyjność,
- oferowany produkt lub serwis powinien znacząco wykraczać poza przeglądarkę,
- należy rozwijać architekturę partycypacji, aby móc osiągnąć jak największy zasięg w cyberprzestrzeni,
- należy oferować mechanizmy wspierające deweloperów (np. fora, wiki, blogi, społeczności),
- należy stosować strategię rozszerzalnej platformy,
- należy dążyć do uniezależniania danych od miejsca, urządzenia itp.,
- należy wykorzystywać użytkowników jako testujących aplikację w czasie rzeczywistym,
- należy stosować modele afiliacyjne,
- należy dokonywać *outsourcingu* infrastruktury, funkcji i wiedzy,
- należy dążyć do tego, aby niektóre obszary działalności zaczęły być dostarczane w formie usługi przez strony trzecie.

## Literatura

1. Musser J.: *Web 2.0: Principles and Best Practices*, O'Reilly Media, Inc. 2007.
2. Unold J.: *Teoretyczno-metodologiczne podstawy przetwarzania informacji w cyberprzestrzeni*, Wydawnictwo Naukowe Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław 2011.
3. Shuen A.: *Web 2.0: A Strategy Guide*, O'ReillyMedia Inc. 2009.

4. Palser B.: *Give Patch a chance: controversy swirls around AOL's ambitious hyper-local venture*, „American Journalism Review” vol. 32, iss. 4, s. 62(1), University of Maryland, December 22, 2010.
5. [www.salesforce.com](http://www.salesforce.com)
6. <http://aws.amazon.com>
7. <http://home.zcubes.com>
8. <http://www.eventsites.co.uk>
9. <http://castingwords.com>

## **WEB 2.0 IN THE CREATION OF A GLOBAL PLATFORM OF THE INFORMATION SOCIETY**

### **Summary**

Contrary to previous approaches, modern take on cyberspace assumes that this area is created by the World Wide Web. “Web as a platform” is the leading catchphrase of Web 2.0, and it assumes the shift from users’ desktop applications to the idea of “software above the level of a single device”. The derivative notions include “software as a service”, “innovation in assembly”, “perpetual beta”, and “lightweight models and cost-effective scalability”.

*Translated by Jacek Unold*