

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀◀

▶▶

◀

▶

Strona 1 z 45

Powrót

Full Screen

Zamknij

Koniec

Rozwój sektora produkcji oprogramowania

Tomasz Przechlewski

Bardzo krótka historia sektora produkcji oprogramowania

Aż do końca lat pięćdziesiątych ubiegłego stulecia program komputerowy był traktowany wyłącznie jako dodatek do komputera, na którym był uruchamiany a nie jako oddzielny produkt.

Pierwsze programy były projektowane na zamówienie [często rządowe].

Przedsiębiorstwa zajmujące się tworzeniem oprogramowania wytwarzające oprogramowanie w tym okresie można określić jako wykonawców projektów zleconych (*software contractors*). Nie sprzedawały one gotowych aplikacji, tylko oferowały usługi programistyczne.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 2 z 45

Powrót

Full Screen

Zamknij

Koniec

Przetłomem była seria komputerów IBM/360 zaprojektowana w połowie lat sześćdziesiątych przez amerykańską firmę IBM.

Decyzja IBM o oddzielnej sprzedaży sprzętu (*hardware*) i oprogramowania (oraz związanych z SI/TI usług), znana jest jako **software unbundling**.

Komputery IBM/360 są początkiem **prawdziwego** rynku oprogramowania.

W 1980 r. w USA było 6 tysięcy przedsiębiorstw sprzedających oprogramowanie.

Komputer osobisty [IBM], którego sprzedaż rozpoczęła się w 1981 r. wyróżniał się **otwartą architekturą** [jego konstrukcja nie stanowiła tajemnicy handlowej oraz nie istniały prawne bariery uniemożliwiające imitację, np. zastrzeżenia patentowe.]

To umożliwiło w szczególności szybko obniżającą cenę imitację sprzętu oraz wytwarzanie na masową skalę oprogramowania komplementarnego przez strony trzecie.

Komputer PC wyróżniał się **niską ceną** – był pomyślany jako komputer do zastosowań domowych. Początkowo oznaczało to niewielką funkcjonalność, wykluczającą zastosowania w przedsiębiorstwach.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 3 z 45

Powrót

Full Screen

Zamknij

Koniec

Szybki wzrost funkcjonalności komputera PC spowodował, że architektura ta stała się dominującą a rynek korporacyjny i rynek zastosowań domowych, dotąd rozwijające się oddzielnie, potoczyły się.

Firmy sprzedające oprogramowanie na rynku zastosowań domowych pojawiły się na rynku korporacyjnym (Microsoft) i odwrotnie (Oracle, IBM).

Wielkość sprzedaży oprogramowania dla komputerów PC wzrosła w latach 1981–1984 ze 140 milionów do 1,6 miliarda dolarów, tj. ponad 10 razy lub z przeciętnym rocznym tempem przewyższającym 300%.

Specyfiką branży był też prawie zupełny brak powiązań z istniejącą już branżą producentów oprogramowania korporacyjnego – sektor został zdominowany przez nowo założone firmy, inne też były sposoby tworzenia oprogramowania i jego dystrybucji. Przykładowo, zbędna była wiedza dotycząca tworzenia dużych i złożonych programów, ponieważ możliwości komputerów osobistych i tak nie umożliwiałyby ich wykonywania.

Oprogramowanie stało się masowym produktem rynkowym.

Gwałtowny rozwój branży taniej elektroniki użytkowej, za którego początek można umownie przyjąć lata 90. ubiegłego wieku spowodował ogromny rozwój sektora **oprogramowania wbudowanego**, tj. programów, które są częścią wielu urządzeń codziennego użytku, takich jak telefon,

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 4 z 45

Powrót

Full Screen

Zamknij

Koniec

telewizor, aparat fotograficzny, urządzenia audio/wideo rozmaitego zastosowania, odbiornik GPS, samochód itd.

Stosując perspektywę historyczną Campbell-Kelly dzieli **producentów oprogramowania** na trzy kategorie: **wykonawców**, **wydawców** [software **publishers**, nazwa nawiązuje do strategii rynkowej, która jest podobna do tej stosowanej przez „tradycyjnych” wydawców książek czy muzyki.] programowania korporacyjnego oraz oprogramowania masowego.

Oprogramowanie na zamówienie powstaje tak, jak tworzy się „tradycyjne” konstrukcje inżynierskie np. drogi czy budynki.

Rynek masowy z kolei jest bardziej podobny do rynku przemysłu rozrywkowego, co m.in. oznacza, że dużą rolę pełni na nim promocja i marketing, podczas gdy jakość oprogramowania ma mniejsze znaczenie.

Większość oprogramowania powstaje na zamówienie i jest wykonywana przez zewnętrznego wykonawcę albo przez pracowników organizacji. W tym drugim przypadku nie jest nawet przedmiotem wymiany na rynku

Oprogramowanie sprzedawane na rynku masowym nie jest ani dominującą ani nawet większą częścią oprogramowania powstającego w organizacjach. Raymond szacuje, iż oprogramowanie tworzone na zamówienie stanowi 90% produkcji oprogramowania światowego.

Nawet jeżeli szacunki Raymonda są zawyżone, to z drugiej strony utożsamianie przemysłu oprogramowania z działalnością „medialnych” firm z sektora oprogramowania masowego, takich jak Microsoft, Adobe czy Oracle jest dużym uproszczeniem

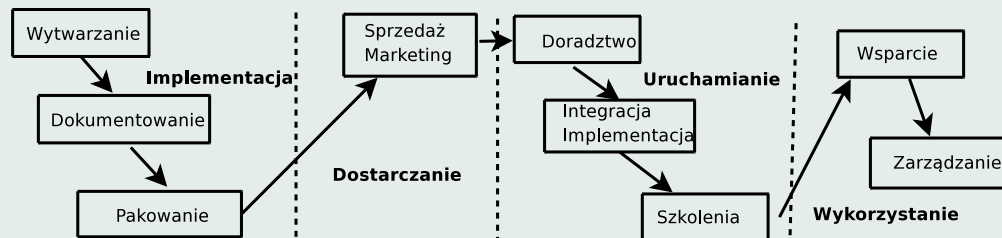
Struktura rynku SI/TI

Wiadomo, że komputer bez oprogramowania jest bezużyteczny. Współcześnie prawdą jest także, iż oprogramowanie bez towarzyszących mu usług jest również bezużyteczne.

Użytkownicy oprogramowania, zwłaszcza instytucjonalni, nie kupują oddzielnie sprzętu, oprogramowania i usług, ale decydując się na eksploatację SI/TI, traktują wyżej wymienione składniki łącznie.

Na rynku oprogramowania można wyróżnić **łańcuch wartości** składający się z następujących ogniw: **wytwarzanie, dostarczanie, uruchamianie i wykorzystanie**.

Szczegółowo fazy te przedstawiono na rysunku:



Usługi łańcucha wartości fazy wykorzystania (opracowano na podstawie Spiller i Wichmann (2002))

Współcześnie organizacje rzadko posiadają wiedzę i zasoby niezbędne do realizacji pierwszych dwóch faz. Oprogramowanie jest wytwarzane przez zewnętrznych – z punktu widzenia organizacji – **dostawców** a **dostarczane** przez **konsultantów** oraz **integratorów**.

Integrator kupuje, instaluje i testuje aplikacje oraz sprzęt, podczas gdy rolą konsultantów jest wyłącznie doradztwo.

Faza **dostarczania** obejmuje nie tylko zakup sprzętu i oprogramowania ale także **konsultację** usługi doradcze oraz **integrację** co oznacza instalację, testowanie, dostosowanie wewnętrznych procesów organizacji oraz niezbędne i szkolenia. Wynikiem jest system gotowy do działania.

Eksplotacja z kolei, to zapewnienie bezawaryjnej pracy systemu, w tym pomoc użytkownikom końcowym. Zwykle organizacja zatrudnia w tym celu wyszkolonych pracowników lub wynajmuje firmę zewnętrzną.

Program z reguły jest wykorzystywany na podstawie **umowy licencyjnej** [por. dalej], stąd przychody z fazy implementacji można utożsamiać z przychodami z tytułu udzielonych licencji.

Tabela pokazuje udział przychodów z tytułu opłat licencyjnych i usług w wybranych przedsiębiorstwach sektora SI/TI:

Nazwa firmy	Licencje ^a	Przychody ^b	Produkowane oprogramowanie
Adobe	98	1,7	DTP, grafika
Symantec	98	1,8	zabezpieczenie systemów SI/TI
Microsoft	94	31,5	system operacyjny, aplikacje
Oracle	79	10,1	bazy danych, aplikacje korporacyjne
CA	69	3,2	aplikacje korporacyjne
Siebel	36	1,3	aplikacje korporacyjne
SAP	31	9,7	aplikacje korporacyjne
Novell	26	1,1	system operacyjny, aplikacje
IBM	25	61,3	sprzęt, aplikacje korporacyjne

^a udział przychody z tytułu opłat licencyjnych w %; ^b roczny przychód w mld USD

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 8 z 45

Powrót

Full Screen

Zamknij

Koniec

Przychody z tytułu licencji i usług na przykładzie wybranych przedsiębiorstw sektora SI/TI w 2004 r. (na podstawie Valimaki (2005))

Oprogramowanie **Open Source** [por. dalej] w dużym stopniu wyklucza strategię rynkowe oparte na przychodach z tytułu opłat licencyjnych.

Rozwój systemu operacyjnego Microsoft Windows

Założone w 1975 r. przedsiębiorstwo Microsoft, w 1980 r. **zatrudniało 38 pracowników i wykazało 8 mln dolarów przychodu.**

W tymże 1980 r. otrzymało zlecenie od firmy IBM na opracowanie systemu operacyjnego dla komputerów PC – nowej klasy komputerów, o niewielkich możliwościach, do którego brak było oprogramowania, w tym systemu operacyjnego.

Microsoft kupił system – nazwany MS-DOS – od niewielkiej firmy, udoskonalił go i sprzedawał licencje użytkownikom komputerów PC.

System w wersji 1.0 zawierał **4 tys. wierszy kodu**, w wersji 2.0 – **20 tys wierszy kodu.**

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 9 z 45

Powrót

Full Screen

Zamknij

Koniec

W 1985 r. udział komputerów PC wyposażonych w system MS-DOS przekroczył 40% **(po 5 latach)**.

W 1989 r. Microsoft rozpoczął pracę nad nowym systemem, bardziej zaawansowanym, który mógłby być sprzedawany na rynku korporacyjnym. W tym celu Microsoft „wykupił” programistów z firmy DEC, którzy stanowili trzon zespołu projektującego.

Mowy system, nazwany **Windows NT**, został opublikowany w 1993 r. **Zespół projektowy liczył 340 osób, a liczba wierszy kodu – około 5 mln.** Opracowując Windows NT, kontynuowano rozwój systemu MS-DOS wzbogacając go o **graficzny interfejs użytkownika**. Ta linia produktów oznaczana min. nazwami Windows 1.0, Windows 3.0, Windows 95, Windows 98 ostatecznie została zakończona około 2000 r.

Późniejsze wersje systemu **stanowiły udoskonalenie Windows NT: Windows 2000 (1999 r. – ponad 3000 projektantów, około 30 mln wierszy kodu), Windows XP (2002 roku, ponad 4000 projektantów, około 40 mln wierszy kodu), Windows Server 2003 (wyłącznie na rynek korporacyjny, opublikowany w 2003 roku, 50 mln wierszy kodu), Windows Vista (2007 roku, ponad 50 mln wierszy kodu).**

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 10 z 45

Powrót

Full Screen

Zamknij

Koniec

Campbell-Kelly i Garcia-Swartz szacują, że **koszty ogółem rozwoju systemu Windows w latach 1975–2003 (prawie 30 lat) na 25–30 tysięcy osobołat, co odpowiada według nich 5–6 miliardom dolarów.**

Microsoft jest firmą, która odniosła największy sukces na rynku oprogramowania masowego. Jej strategia rynkowa, podstawą której są dwa produkty: system operacyjny Windows oraz pakiet biurowy Office jest typowa dla wielu firm, które wzorując się na Microsoftzie powtórzyły ten sukces, aczkolwiek w mniejszej skali. Firmy te w większości sprzedają oprogramowanie komplementarne z MS Windows i MS Office – dominującą platformą w segmencie komputerów PC.

Rozwój oprogramowania *Open Source*

Za początek „ruchu” **Open Source** zwykle uważa się **projekt GNU** oraz założenie organizacji **Free Software Foundation (FSF)**, co miało miejsce **w końcu lat siedemdziesiątych ubiegłego wieku.**

Powstanie GNU oraz FSF wiąże się z działalnością **Richarda Stallmana**, amerykańskiego programisty, autora kilku ważnych programów narzędziowych, przede wszystkim kompilatora języka C, gcc.

Struktura rynku SI/TI

Rozwój systemu . . .

Rozwój . . .

System Linux

Ochrona prawna . . .

Ekonomiczne . . .

Oprogramowanie . . .

Strona główna

Strona tytułowa



Strona 11 z 45

Powrót

Full Screen

Zamknij

Koniec

Stallman opracował ramy prawne w postaci **licencji GPL** oraz przygotował **Manifest GNU**, w którym postulował swobodę dystrybucji i modyfikacji oprogramowania.

W tym samym czasie na **Uniwersytecie California w Berkeley** ulepszono system Unix (*Berkeley Software Distribution*, BSD), opracowany w firmie AT&T, ostatecznie w 1980 r. rozpowszechniony na „licencji BSD” – jednej z pierwszych licencji **Open Source**.

Na początku lat 90. **Linus Torvalds** – student **Uniwersytetu w Helsinkach** opublikował w sieci Internet pierwszą wersję jądra **systemu operacyjnego Linux** [decyzją Torvaldsa miał być rozpowszechniany na **licencji GPL**.]

W 1993 zarówno GNU/Linux jak i BSD były stabilnymi platformami, którym wszakże brakowało oprogramowania aplikacyjnego.

Od tego czasu BSD rozwinął się w grupę systemów (NetBSD, FreeBSD, and OpenBSD), a Linux stał się podstawą rozlicznych wariantów GNU/Linux (Slackware, Debian, Red Hat, Suse, Mandrake i inne).

W latach 1990 powstało liczne oprogramowanie: Apache (serwer WWW), MySQL (baza danych), XFree86 (implementacja środowiska graficznego X11 na platformę PC), GNOME i KDE (okienkowy interfejs ułatwiający pracę z systemem użytkownikom), Mozilla (przeglądarka WWW), oraz

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 12 z 45

Powrót

Full Screen

Zamknij

Koniec

OpenOffice (zestaw programów biurowych, odpowiednik pakietu Microsoft Office) itd.

Oprogramowanie **Open Source** zaczęło być wytwarzane/dostarczane przez firmy komercyjne, w tym duże przedsiębiorstwa.

Oficjalna decyzja **firmy IBM** o zaangażowaniu się w rozwój Linuksa oraz serwera Apache legitymizowała oprogramowanie **Open Source** traktowane do tej pory często „z przymrużeniem oka” przez użytkowników korporacyjnych z uwagi na jego prywatno-hobbystyczną proveniencję

Firma IBM na rynku serwerów sprzedaje oprócz komputerów opartych na platformie Wintel, także trzy w dużym stopniu niekompatybilne typy komputerów: **komputery mainframe** [system zOS], **duże stacje robocze** [OS/400] oraz **małe stacje robocze** [AIX].

Zaangażowanie IBM w rozwój Linuksa i innych projektów **Open Source** [np. serwer Apache] ma trzy cele.

Pierwszy to ograniczenie kosztów związanych z pielęgnacją systemów operacyjnych poprzez zastąpienie ich wspólną platformą.

Drugi to promocja sprzedaży sprzętu i oprogramowania komercyjnego [np. system zarządzania bazami danych DB2], poprzez zwiększenie jego

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 13 z 45

Powrót

Full Screen

Zamknij

Koniec

użyteczności dzięki udostępnieniu taniego i wysokiej jakości oprogramowania komplementarnego (np. Apache).

Trzeci cel, to strategia osłabienia konkurentów [Microsoft, Sun Microsystems], poprzez ograniczenie rentowności rynku systemów operacyjnych.

Własnościowe projekty **Open Source**, tj. programy rozpowszechniane na podwójnej licencji [patrz dalej].

System Linux

Odpowiednikiem systemu Windows w systemie Linux jest **jądro systemu**.

Jądro systemu w wersja 2.6.30 z czerwca 2009 r. składa się z 11,5 mln wierszy kodu i jest on dziełem 1150 indywidualnych wykonawców oraz 240 przedsiębiorstw sponsorujących.

W porównaniu z wersją 2.6.11 z marca 2005 roku, liczba wykonawców wzrosła prawie 3 razy a liczba przedsiębiorstw ponad 3,5 razy.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 14 z 45

Powrót

Full Screen

Zamknij

Koniec

Pomimo dużej liczby zaangażowanych wykonawców większość kodu, bo 25% jest dziełem wąskiego grona 30 osób. Linus Torvalds – pomysłodawca i twórca Linuksa nie mieści się już w pierwszej trzydziestce wykonawców, jego **udział w tworzeniu systemu jest mniejszy niż 1%**.

Kod źródłowy Linuksa przed dodaniem do wersji „produkcyjnej” podlega procesowi „recenzji”, która jest dokonywana przez jednego z około setki opiekunów poszczególnych modułów składających się na kompletny system. Dopiero po akceptacji opiekuna owego modułu zmiana jest dodawana do programu. **Tordvalds akceptuje około 3% całości kodu.**

Kompletny system, użyteczny z punktu widzenia użytkownika końcowego składający się z jądra oraz zbioru setek aplikacji nosi nazwę **dystrybucji**.

Tworzenie dystrybucji obejmujące takie czynności jak integracja, testowanie, dokumentowanie i pakowanie, to etap dostarczania w łańcuchu wartości na rynku oprogramowania.

Ten etap może mieć charakter komercyjny lub niekomercyjny. Debian jest projektem niekomercyjnym, ale są także takie przedsiębiorstwa jak RedHat czy Novell, które sprzedają licencje na komercyjne wersje własnych dystrybucji. *De facto* sprzedawane są w ten sposób usługi wsparcia związane z fazą uruchamiania/wykorzystania oprogramowania.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 15 z 45

Powrót

Full Screen

Zamknij

Koniec

Debian – jedna z dystrybucji Linuksa – jest największym projektem **Open Source**.

Szacowana wielkość wersji 4.0 wynosi ponad **280 milionów wierszy kodu**. Amor wykorzystując metodę COCOMO szacowania kosztów ocenił, iż wykonanie Debiana „od zera” **wymagałoby 73 tysięcy osobolat** i kosztowałoby **10 mld dolarów**.

Linux i Debian są przykładami programów o wielkości porównywalnej z największymi programami komercyjnymi, które zostały wyprodukowane w zupełnie inny od nich sposób.

Ochrona prawna programów komputerowych

Prawo autorskie i licencje o udostępnianiu

Przedmiotem prawa autorskiego jest każdy przejaw działalności twórczej o indywidualnym charakterze, ustalony w jakiejkolwiek postaci, niezależnie od wartości, przeznaczenia i sposobu wyrażenia.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 16 z 45

Powrót

Full Screen

Zamknij

Koniec

Rozdział 8 ustawy [PAPP1994] zawiera szczegółowe przepisy dotyczące programów komputerowych, które m.in. wyznaczają bardzo wąski zakres tzw. dozwolonego użytku.

W szczególności modyfikowanie programu bez zgody autora jest możliwe tylko w ograniczonym zakresie, mianowicie wyłącznie w celu uzyskania kompatybilności z innym programem.

Zmiana zwiększająca funkcjonalność albo naprawa usuwająca błąd jest zatem bez zgody autora niemożliwa. Także zabronione jest wykonanie kopii zapasowej chyba, że jest ona „niezbędna od korzystania z programu”

Nota Bene: Ustawa nie precyzuje ani pojęcia **program** ani co oczywiste **kopii zapasowej programu**. Wydaje się, że samo znaczenie słowa **kopia** i do tego zapasowa wyklucza iż coś takiego jest niezbędne do korzystania z programu. Na tym przykładzie widać, że priorytetem ustawodawcy było przede wszystkim **zagwarantowanie interesu właściciela programu**.

W porównaniu do „tradycyjnych” dóbr intelektualnych **ochrona praw do programów jest zatem znacznie bardziej surowa**.

Ochrona z tytułu prawa autorskiego powstaje z chwilą stworzenia utworu, a do jej ustanowienia nie są wymagane żadne procedury formalne (jak w przypadku ochrony patentowej).

Struktura rynku SI/TI

Rozwój systemu . . .

Rozwój . . .

System Linux

Ochrona prawna . . .

Ekonomiczne . . .

Oprogramowanie . . .

Strona główna

Strona tytułowa



Strona 17 z 45

Powrót

Full Screen

Zamknij

Koniec

Dla przyznania ochrony prawa autorskiego nie jest konieczne utrwalenie utworu, a więc zapisanie go na jakimkolwiek trwałym nośniku.

Należy zaznaczyć, że ochroną objęty może być wyłącznie sposób wyrażenia, nie są zaś nią objęte odkrycia, idee, procedury, metody i zasady działania oraz koncepcje matematyczne [art. 1 PAPP1994].

W przypadku programów oznacza to w praktyce, że nie jest chroniony sposób działania programu (ten ewentualnie może zostać zastrzeżony patentem) a podstawową korzyścią ochrony prawno-autorskiej jest zabezpieczenie programu przed nieuprawnionym zwielokrotnieniem.

Z tego powodu producenci zwykle rozpowszechniają programy w postaci skompilowanej, zabezpieczając się przed imitatorami przy wykorzystaniu tajemnicy handlowej.

Umowy licencyjne

Licencja jest umową, w której właściciel własności intelektualnej zezwala na korzystanie z niej innym stronom.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 18 z 45

Powrót

Full Screen

Zamknij

Koniec

Licencjonowana może być każda forma własności intelektualnej: autorskie prawa majątkowe, patenty, znaki towarowe a nawet tajemnica handlowa.

Licencje (umowy o udostępnienie programu komputerowego) stanowią przeważającą formę wykorzystywaną przy obrocie oprogramowaniem, podczas gdy obrót „tradycyjnymi” dobrami intelektualnymi, jak na przykład książkami, odbywa się zwykle za pomocą umowy sprzedaży.

Kluczowe postanowienie umów licencyjnych dotyczą: **odpowiedzialności producenta za wady oprogramowania** oraz **dozwolonego zakresu wykorzystania**, w tym modyfikowania oprogramowania oraz jego zwielokrotniania.

W przypadku sprzedaży programu wraz z nośnikiem, bez dodatkowych zobowiązań licencyjnych nabywca byłby ograniczony jedynie przepisami prawa autorskiego. Mógłby zatem swobodnie odsprzedać, wynająć lub pożyczyć program [Ustawa „O prawie autorskim...” *explicite* zakazuje użyczenia i najmu [PAPP1994], ale ustawodawstwo innych państw może być bardziej liberalne], zmienić program, próbować odtworzyć kod źródłowy czy też użytkować go w takim zakresie, w jakim jest to technicznie wykonalne.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 19 z 45

Powrót

Full Screen

Zamknij

Koniec

Licencje własnościowe w przeciwieństwie do **licencji publicznych** znacznie ograniczają zakres dozwolonego wykorzystania programu, w szczególności zakazane jest zwielokrotnianie (nawet na potrzeby użytku osobistego) oraz modyfikowanie programu.

Drugim zasadniczym celem umów licencyjnych [zarówno własnościowych jak i publicznych] jest ograniczenie zakresu i wielkości odpowiedzialności licencjodawcy za szkody związane z użytkowaniem programu.

Producent nie odpowiada za ewentualne szkody związane z użytkowaniem wadliwego programu, np. w oparciu o surowe reguły odpowiedzialności z **tytułu rękojmi za wady rzeczy**.

Oprogramowanie masowe oferowane jest użytkownikom na zasadach (własnościowych) **licencji standardowych** zwanych potocznie **umowami celofanowymi** (*shrink-wrap* albo *click-wrap licences*).

W przypadku gdy nabywca nie zgadza się z postanowieniem licencji to przysługuje mu prawo oddania programu sprzedawcy i zwrotu zapłaty. Treść licencji sprowadza się zaś do następujących postanowień: **licencja obejmuje korzystanie dla użytku osobistego, program nie może być odsprzedany lub wypożyczony, niedozwolona jest modyfikacja oraz odtwarzanie programu źródłowego.**

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 20 z 45

Powrót

Full Screen

Zamknij

Koniec

Przykładem może być **licencja użytkownika końcowego (EULA) firmy Microsoft**, obowiązująca indywidualnych użytkowników systemu Windows, która zezwala na zainstalowanie programu wyłącznie na jednym komputerze.

Komputer nie może być wyposażony w więcej niż **dwa procesory**.

Liczbę podłączonych innych komputerów do komputera z systemem – nie może być większa niż 10. [system nie może być wykorzystywany jako serwer. Jest to technicznie możliwe ale niezgodne z umową licencyjną.[różnicowanie produktów]]

Jeżeli w czasie 30 dni użytkownik nie dokona **obowiązkowej** aktywacji **licencja wygasa**.

Podczas aktywacji użytkownik jest zobowiązany przekazać dane osobowe, których treść **nie jest precyzyjnie określona w licencji**.

System może **przestać funkcjonować** z powodu bliżej nieokreślonych zmian w konfiguracji sprzętu i w takiej sytuacji może wymagać powtórnej aktywacji [odszkodowanie nie przysługuje].

Jeżeli użytkownik zakupi kolejną wersję programu (*upgrade*), to **traci prawo** do wykorzystania wersji wcześniejszej.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 21 z 45

Powrót

Full Screen

Zamknij

Koniec

Tylko pierwszy licencjobiorca ma **prawo do odsprzedaży** licencji – kolejni nie mają tego prawa.

Zabroniona jest dzierżawa, najem czy pożyczanie oprogramowania.

Użytkownik **zrzeka się roszczeń** powstałych wskutek błędów w działaniu programu, **przekraczających cenę za którą program został nabyty**.

Zapisy dotyczące zrzeczenia się roszczeń są bardziej szczegółowo opisane w kilkunastu zastrzeżeniach i postanowieniach licencji. Głównym celem owych uściśleń jest jeszcze większe ograniczenie ewentualnych roszczeń.

Wniosek: licencja jest skrajnie niekorzystna dla licencjobiorcy.

Jak zauważają Barta i Markiewicz „Co do ważności całości lub przynajmniej części postanowień charakteryzowanej umowy istnieją duże zastrzeżenia [prawne]”.

W szczególności umowa ta stoi w sprzeczności z artykułem 355 Kodeksu Cywilnego (umowy wiążą drugą stronę tylko wówczas, gdy znając ich treść wyraziła zgodę na włączenie ich do treści umowy) oraz artykułu 4 „Ustawy o przeciwdziałaniu praktykom monopolistycznym”, który zakazuje narzucania uciążliwych warunków umowy przynoszących podmiotowi gospodarczemu nieuzasadnione korzyści.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 22 z 45

Powrót

Full Screen

Zamknij

Koniec

Teoretyczne wątpliwości nie są jednak poparte praktyką sądowniczą – uciążliwe klauzule są stosowane powszechnie i jak dotąd nie są kwestionowane przez sądy.

Licencje otwarte

Oprogramowanie Open Source jest rozpowszechniane na zasadach licencji otwartych.

Przyjmuje się za otwartą licencję, taką w której dozwolone jest:

nieograniczone wykorzystanie, [liczba użytkowników, system/platforma, komercyjne/niekomercyjne wykorzystanie];

bezpłatne zwielokrotnianie i publikowanie, co wyklucza dla oprogramowania rozpowszechnianego zgodnie z licencją otwartą jakiegokolwiek strategię rynkowe oparte na dochodach z opłat licencyjnych;

bezpłatne modyfikowanie programu, co nie wyklucza innych warunków dotyczących modyfikowania. Warunkiem *sine qua non* modyfikowania jest dostępność kodu źródłowego, która z kolei eliminuje tajemnicę handlową jako potencjalną strategię rynkową producenta.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 23 z 45

Powrót

Full Screen

Zamknij

Koniec

Licencje OS podzielić z punktu widzenia prawa do modyfikowania kodu źródłowego na dwie podstawowe kategorie: **restrykcyjne** oraz **dopuszczające**.

Licencje restrykcyjne, zwane także **wirusowymi** lub **żywotnymi** nakazują, iż modyfikacje oraz dzieła pochodne muszą być licencjonowane na identycznej licencji jak program oryginalny [warunek/obowiązek wzajemności].

Licencje zezwalające nie nakładają na dzieła pochodne warunku wzajemności.

Historycznie najstarsza i najbardziej rozpowszechniona **licencja GNU** jest licencją żywotną.

Zgodnie z warunkami tej licencji jest rozpowszechniane przykładowo jądro systemu Linux, program do grafiki GIMP, zestaw aplikacji biurowych OpenOffice lub system zarządzania danych PostgreSQL.

System BSD, będący alternatywą dla Linuksa, przeglądarka WWW Firefox czy też serwer WWW Apache są rozpowszechniane na warunkach zezwalających na zmianę licencji [**dopuszczająca**].

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 24 z 45

Powrót

Full Screen

Zamknij

Koniec

Licencje podwójne

Umowy licencyjne są także skutecznym narzędziem różnicowania oprogramowania i segmentacji rynku.

Podwójne licencjonowanie (PL, **dual licensing**) to strategia rynkowa polegająca na rozpowszechnianiu programu zarówno na zasadach licencji otwartej oraz jako program własnościowy.

W przypadku oprogramowania własnościowego **właścicielem całości programu** jest producent. W przypadku wielu programów **Open Source** własność jest **rozproszona**, co oznacza, że każdy współtwórca jest autorem jedynie fragmentu programu [konsekwencje prawne jak i dotyczące sposobu organizacji i rozwoju projektu.]

W przypadku współautorstwa problemem jest **kompatybilność licencji** – przykładowo licencje żywotne oraz własnościowe są **niekompatybilne**.

Nie można dołączyć programu udostępnionego na licencji żywotnej do programu własnościowego.

Podwójne licencjonowanie jest możliwe w praktyce jeżeli prawo własności do oprogramowania nie jest rozproszone, stąd społecznościowy model

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 25 z 45

Powrót

Full Screen

Zamknij

Koniec

tworzenia oprogramowania i podwójne licencjonowanie wzajemnie się wykluczają.

Rozwój projektów PL jest **scentralizowany** a udział stron trzecich ogranicza się do zgłaszania przez użytkowników poprawek i błędów.

W modelu PL właściciel, starając się zmniejszyć zagrożenie ze strony potencjalnych imitatorów wynikające z ujawnienia kodu źródłowego, najczęściej udostępnia program na zasadach **licencji niekompatybilnych** – zwykle licencją otwartą jest licencja żywotna. [program konkurencyjny **nie może** już być podwójnie licencjonowany, co ogranicza możliwość czerpania ekonomicznych korzyści przez imitatora.]

Oczywiście teoretycznie mógłby na podstawie programu udostępnionego w modelu podwójnej licencji powstać konkurencyjny projekt społecznościowy, ale w praktyce nie ma to miejsca [jazda na gapę].

Jeżeli wszakże projekt zostałby porzucony, zaniechany lub rozwijał się niezgodnie z oczekiwaniami użytkowników, to społeczność ma pełne prawo oraz techniczne możliwości do jego kontynuowania.

W przypadku oprogramowania własnościowego producent często nie jest skłonny ujawnić kod źródłowy nawet aplikacji porzuconej, z uwagi na to, że jest on wykorzystywany także w innych programach.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 26 z 45

Powrót

Full Screen

Zamknij

Koniec

Strategia PL jest stosowana w projektach dotyczących oprogramowania strukturalnego lub **middleware** [MySQL].

Ochrona patentowa

Patent to prawny monopol przyznany przez państwo, przyznający jego posiadaczowi wyłączność wykorzystania, sprzedawania lub importowania **wynalazku** na terenie kraju, który patentu udzielił.

„Przez uzyskanie patentu nabywa się prawo wyłącznego korzystania z wynalazku w sposób zarobkowy lub zawodowy na całym obszarze Rzeczypospolitej Polskiej” [Art. 63 PWP].

Kryteria przyznawania ochrony patentowej wynalazkom są określone w ustawie w sposób bardzo ogólny: „patenty są udzielane na wynalazki, które są **nowe**, posiadają **poziom wynalazczy** i nadają się do **przemysłowego stosowania**.”

Ustawa wyklucza też wybrane innowacje z możliwości ochrony za pomocą patentów, w szczególności Art. 28 stanowi, że „za wynalazki [...] nie uważa się w szczególności odkryć, teorii naukowych i metod matematycznych [oraz] programów do maszyn cyfrowych”.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 27 z 45

Powrót

Full Screen

Zamknij

Koniec

Podobnie **Europejska Konwencja Patentowa** wyłącza spod swego zakresu oprogramowanie (w artykule 52). [Nie licząc się wszakże ze obowiązującym stanem prawnym Europejski Urząd Patentowy (EPO) przyznał do tej pory tysiące patentów na oprogramowanie].

Amerykański urząd patentowy (**Patent and Trade Office** – PTO) od lat dziewięćdziesiątych ubiegłego wieku przyznaje masowo patenty na oprogramowanie.

Liczba przyznanych patentów **jest szacowana na setki tysięcy**.

Dodatkowo **porozumienie TRIPS** [Porozumienie w sprawie Handlowych Aspektów Praw Własności Intelektualnej, **Trade Related Aspects of Intellectual Property Rights**], według niektórych interpretacji nakłada obowiązek na sygnatariuszy przyznawania patentów także w dziedzinie oprogramowania [„patents shall be available for any inventions, whether products or processes, **in all fields of technology**, provided that they are new, involve an inventive step and are capable of industrial application”].

W przeciwieństwie do ochrony z tytułu prawa autorskiego, która nie wymaga żadnych procedur formalnych, ochrona patentowa jest kosztowna: wynalazca musi przygotować wniosek i wnosić stosowne opłaty.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 28 z 45

Powrót

Full Screen

Zamknij

Koniec

W Polsce Urząd Patentowy pobiera opłaty jednorazowe oraz okresowe w związku z ochroną wynalazków, wzorów użytkowych, wzorów przemysłowych, znaków towarowych (art. 222).

W roku 2008 opłata jednorazowa za zgłoszenie wynalazku lub wzoru użytkowego wynosiła 550 zł, opłaty okresowe za okres 20 lat ochrony to zaś około 17 tys. złotych, przy czym opłaty okresowe rosną.

Pierwszeństwo do uzyskania patentu, oznacza się według daty zgłoszenia wynalazku a czas trwania patentu wynosi 20 lat od daty dokonania zgłoszenia wynalazku w Urzędzie Patentowym (art. 63).

Ekonomiczne podstawy produkcji oprogramowania

Korzyści skali i zakresu

Punktem wyjścia wielu modeli jest założenie, iż **oprogramowanie jest dobrem informacyjnym**, którego „koszty wytworzenia są znaczne, ale za to jest ono bardzo tanie w dystrybucji [...] dla produktów informacyjnych koszty stałe są wysokie, a **koszty krańcowe niewielkie**”.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 29 z 45

Powrót

Full Screen

Zamknij

Koniec

Innymi słowy koszt **krańcowy wytworzenia** jest równy bliskiemu zera **kosztowi zwielokrotnienia**, przy czym znaczne koszty stałe są w większości **kosztami utopionymi**, niemożliwymi do odzyskania w przypadku niepowodzenia przedsięwzięcia.

Wytwarzanie oprogramowania jest związane z **korzyściami skali**, a w konsekwencji, wyznaczanie ceny oprogramowania w oparciu o koszt krańcowy nie jest właściwe – cena bliska zeru nie gwarantuje nawet odzyskania przez producenta poniesionych nakładów.

Dane dotyczące struktury kosztów producentów oprogramowania wskazują, iż jest to założenie mocno upraszczające.

Przykładowo udział kosztów bezpośrednio związanych z wytworzeniem oprogramowania w przypadku firmy Microsoft wynosi około 15%.

Największy udział w łącznych kosztach stanowią zaś **wydatki na reklamę (35%)**. Udział reklamy w kosztach wytworzenia/wypromowania pierwszej wersji arkusza kalkulacyjnego Lotus 1-2-3, **sięgał 70%**.

Istnieje różnica pomiędzy takimi dobrami informacyjnymi jak muzyka, filmy wideo, dokumenty elektroniczne (ogólnie: **dane**) a oprogramowaniem. Wartość oprogramowania zwiększa się znacząco w fazach dostarczania,

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 30 z 45

Powrót

Full Screen

Zamknij

Koniec

uruchamiania i eksploatacji, podczas gdy łańcuch wartości w przypadku **danych** jest znacząco krótszy.

Perens wyróżnia cztery sposoby organizacji produkcji oprogramowania:
– **kontrakt**, – **wydawanie oprogramowania**, – **konsorcjum przedsiębiorstw** oraz **oprogramowanie Open Source**.

Powyższe paradygmaty różnią się: – sposobem podziału kosztów i ryzyka pomiędzy interesariuszy, – **sprawnością ekonomiczną** rozumianą jako udział nakładów przeznaczonych bezpośrednio na produkcję **użytecznego oprogramowania**. w cenie programu oraz – możliwością wykluczenia stron trzecich z użytkowania oprogramowania.

Oprogramowanie kupione i **faktycznie używane** to dwa różne pojęcia.

Zakupione oprogramowanie jest nie używane, albo używane nie w takim zakresie jakiego spodziewał się użytkownik, albo wiele funkcji programu jest niepotrzebnych z punktu widzenia potrzeb określonego użytkownika, a niektórych z kolei może brakować.

Zaniechanie produkcji programu często wymusza jego „przedwczesną” zmianę na inny, bez możliwości odzyskania poniesionych nakładów (koszty utopione, po stronie użytkownika).

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 31 z 45

Powrót

Full Screen

Zamknij

Koniec

Perens szacuje, że **50% wydatków na oprogramowanie masowe jest marnowane z powyższych powodów**, co wydaje się i tak bardzo ostrożnym szacunkiem.

Paradygmat wydawania oprogramowania charakteryzuje się **skrajnie niską sprawnością**, na którą składa się wysoki udział kosztów pośrednich i zysku (85%) oraz niski stopień wykorzystania.

Perens szacuje **sprawność tego modelu produkcji oprogramowania na zaledwie 5%**, co – jak słusznie zauważa – implikuje, iż **pewne kategorie oprogramowania nie mogą być produkowane w ramach tego paradygmatu** ponieważ albo nie istnieje dostatecznie duży rynek gwarantujący zwrot poniesionych nakładów i zysk, albo – jeżeli nawet rynek jest dostatecznie duży – inwestor może ocenić iż z takiego lub innego powodu nie będzie w stanie odzyskać nakładów [nie istnieje infrastruktura albo imitacja jest łatwa].

Przykładem nieefektywności modelu **wydawania oprogramowania** jest sieć WWW, która powstała jako oprogramowanie **Open Source** także dlatego, że żaden **wydawca** nie potrafił opracować sensownego sposobu odzyskania poniesionych nakładów.

Z produkcją programów związane są także duże **korzyści zasięgu**.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 32 z 45

Powrót

Full Screen

Zamknij

Koniec

Efekt zasięgu występuje gdy produkcja dobra A zmienia koszty produkcji dobra B i wiąże się zwykle z fazą dystrybucji i strategiami takimi jak różnicowanie jakości produktu, sprzedaż wiązana, czy sprzedaż w pakiecie. **Nota bene** jeżeli faktycznie program byłby „tani w dystrybucji”, to korzyści zasięgu powinny być niewielkie, a jest dokładnie odwrotnie.

Raz napisane programy mogą być wykorzystane wielokrotnie w innych programach.

Zewnętrzne efekty sieciowe i koszty zmiany

Często wskazywaną cechą rynku oprogramowania są **efekty sieciowe**.

Wyróżnia się pośrednie i bezpośrednie efekty sieciowe, przy czym **efekty bezpośrednie**, to wzrost użyteczności dobra bezpośrednio związany z fizycznym dołączeniem kolejnego użytkownika sieci (wysłanie listu elektronicznego jest użyteczne jeżeli jest go **do kogo** wysłać).

Efekty pośrednie występują zaś wtedy, gdy zwiększająca się liczba konsumentów danego dobra powoduje zwiększenie liczby oraz obniżenie ceny dóbr do niego komplementarnych.

Struktura rynku SI/TI

Rozwój systemu . . .

Rozwój . . .

System Linux

Ochrona prawna . . .

Ekonomiczne . . .

Oprogramowanie . . .

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 33 z 45

Powrót

Full Screen

Zamknij

Koniec

Istnienie efektów sieciowych, zwłaszcza pośrednich, wzmacnia rolę oczekiwań konsumentów względem wyborów dokonywanych przez innych uczestników rynku.

Efekty sieciowe na rynku dóbr informacyjnych są jednym z ważniejszych przyczyn jego nieefektywności.

Liebowitz i Margolis negują znaczenie efektu sieci [większość obserwowanych zależności można objaśnić obniżającym ceny i zwiększającym popyt postępem technicznym oraz korzyściami skali].

Masa startowa (*critical mass*), pojęcie oznaczającą minimalną liczebność sieci, przy której uczestnicy sieci skorzystają z dołączenia do niej. [masa startowa jest funkcją ceny].

Subsydiowanie uczestników jest jedną z możliwych strategii w początkowym okresie rozwoju sieci. Alternatywą dla subsydiowania są **wysokie wydatki na promocję i marketing**.

Wielkość sieci jest **mniej potrzebna użytkownikom** a bardziej **producentom na odzyskanie nakładów**: „projekt SI/TI jest zarzucany nie dlatego, że użytkownicy nisko oceniają korzyści wynikające z efektu sieci, tylko dlatego, że inwestor ocenił iż jego kontynuowanie nie daje szans na zwrot nakładów.”

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 34 z 45

Powrót

Full Screen

Zamknij

Koniec

Platforma [„system współzależnych ze sobą sobą standardów, pozwalający na modularną wymianę zasobów, takich jak oprogramowanie i sprzęt”] jest tą kategorią oprogramowania, dla której efekty sieciowe są istotne [ale bez przesady – Mac OS].

Platforma pośredniczy pomiędzy **rynkami dwustronnymi** (*two-sided markets*) określanymi także jako **sieci dwustronne** (*two-sided networks*), tj. rynkami, na których można wyróżnić dwie odrębne grupy podmiotów.

System operacyjny łączy użytkowników i producentów aplikacji, konsole do gier graczy i producentów gier, wyszukiwarka Google użytkowników sieci WWW oraz reklamodawców. Platformami są także „tradycyjne” produkty/usługi takie jak karty kredytowe, gazety/TV.

Eisenmann wyróżnia następujące strony występujące na rynkach związanych platformą pośredniczącą: – **stronę popytową** (użytkownicy końcowi), – **stronę podażową** (producenci dóbr komplementarnych), – **dostawcy platformy**, – **sponsorzy**.

Microsoft i Intel są sponsorami i dostawcami platformy Wintel.

Dostęp do platformy może być **otwarty** jeżeli ograniczenia, co do jej wykorzystania są „sensowne” (np. dotyczą zgodności z obowiązującymi

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 35 z 45

Powrót

Full Screen

Zamknij

Koniec

standardami lub sprowadzają się do innych wymagań technicznych) i niedyskryminujące, co oznacza, że jednakowo dotyczą wszystkich stron.

Dostęp może być **otwarty** lub **zamknięty** na każdym z czterech poziomów. Z uwagi na kluczowy dla platformy dostęp w roli sponsora, dzielimy platformy na **własnościowe** i **otwarte**.

Decyzja sponsora o „otworzeniu platformy” jest zwykle kompromisem pomiędzy wielkością sieci a możliwością osiągnięcia ekonomicznych korzyści.

Sukces rynkowy platformy nie jest wprost związany z otwartością – jeżeli w ogóle taki związek istnieje – bowiem można podać wiele przykładów platform o różnym stopniu otwartości dostępu, które go osiągnęły.

W przypadku **dóbr informacyjnych**, których **krańcowy koszt wytworzenia** wynosi 0 producent może sprzedawać dobro po cenie zero w celu szybszego zawłaszczenia rynku. Strategia taka różni się od subsydiowania sprzedaży związanej w rodzaju rozdawania lamp naftowych w celu zwiększenia sprzedaży oliwy (lampy/oliwę kupuje ci sami konsumenci).

W przypadku oprogramowania, jeżeli **platformy są niekompatybilne**, to wykonanie programu kompatybilnego z wieloma platformami jest zwykle

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 36 z 45

Powrót

Full Screen

Zamknij

Koniec

znacznie kosztowniejsze dla wytwórcy oprogramowania niż przygotowanie go w wariantcie „jednoplatformowym”.

Powoduje to wytworzenie się **samonapędzającego się zjawiska**: platforma dominująca mocniej „przyciąga” aplikacje komplementarne gdyż wielu wytwórców oceniając, iż koszt opracowania programu kompatybilnego z platformą niszową jest wyższy od [spodziewanych] korzyści rezygnuje z tego.

Bresnahan i Greenstein w oparciu o teorię endogenicznych kosztów utopionych Suttana [Endogeniczne koszty utopione to koszty utopione, co do których czynione są następujące założenia: – podnoszą wyłącznie wartość przedsiębiorstwa, które je ponosi, – są nieograniczone, tj. na każdym poziomie zawsze można wydać więcej, – wzrost kosztów zwiększa popyt potencjalnych nabywców] dowodzą, że rywalizacja platform **nieuchronnie prowadzi do wysokiej koncentracji rynku po stronie podaży**.

Empiryczne szacunki wielkości efektu sieciowego są nieliczne. Trudne jest już samo **zdefiniowanie rynku**, z uwagi na to, że producenci oprogramowania powszechnie stosują różnicowanie, sprzedaż w pakietach, itp. **Spośród wielu aplikacji, różniących się funkcjonalnością i stopniem kompatybilności, to które z nich tworzą pewien rynek, a które już nie jest decyzją w dużej mierze arbitralną.**

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 37 z 45

Powrót

Full Screen

Zamknij

Koniec

Koszty zmiany

Zmiana programu na inny lub lepszy zwykle wiąże się ze znacznymi kosztami, określanymi jako **koszty zmiany**.

Koszt uruchomienia nietrywialnej aplikacji zwykle jest znacznie wyższy niż wydatki na licencje, obejmuje bowiem także zintegrowanie i dopasowanie nowego systemu do istniejącej już infrastruktury oraz **procesów biznesowych**, w tym **szkolenie użytkowników**.

Znaczne koszt zmiany powodują, że użytkownik zostaje uzależniony od dostawcy; w rezultacie kontrakt, który formalnie jest umową jednorazową lub dotyczącą jednego produktu, *de facto* staje się relacją długookresową, z czasem obejmującą także inne oprogramowanie, kompatybilne z pierwotnie zakupioną aplikacją.

Rezultatem jest monopol *ex-post* dla firm konkurujących *ex-ante*.

Koszty zmiany połączone z efektem sieciowym mogą powodować silny efekt **inercji**; nowa, lepsza technologia nie jest akceptowana, z powodu obaw iż użytkownicy sieci, którzy dokonają zmiany mogą na tym skorzystać dopiero wtedy, jeżeli inni postąpią tak samo z uwagi na korzyści wynikające z kompatybilności.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 38 z 45

Powrót

Full Screen

Zamknij

Koniec

Niechęć do zmiany blokująca nową lepszą technologię jest określana jako **nadmierny bezwład**. Z kolei zjawisko nieefektywnego (zbyt wczesnego) przejścia na nowy standard określane jako **nadmierny pęd** oznacza, że stracili ci, którzy zbyt wcześnie weszli do „osieroconej” sieci, zanim osiągnęli korzyści.

Wytwórcy i dostawcy oprogramowania zwiększają koszty zmiany jako część swojej strategii poprzez zmuszanie lub zachęcanie do związania aplikacji z innymi aplikacjami, w tym specyficznymi dla określonej platformy oraz stosowaniem formatu danych, który stanowi tajemnicę handlową, co utrudnia lub uniemożliwia ich wykorzystanie przez inne programy.

Inna strategią, *nota bene* skarżoną w sporach antymonopolowych są **grabieżcze zapowiedzi (predatory preannouncements)**, mające na celu zniechęcenie konsumentów do zmiany dostawcy.

Firma Microsoft obawiając się przejścia użytkowników systemu DOS na funkcjonalnie dużo lepszy system OS/2, w sytuacji gdy sama nie oferowała jeszcze porównywalnego jakościowo z OS/2 systemu Windows 95 zastosował tego rodzaju strategię pokazując prawie-gotowy system kilka miesięcy wcześniej niż był on dostępny w sprzedaży.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 39 z 45

Powrót

Full Screen

Zamknij

Koniec

Proponenci oprogramowanie **Open Source** wskazują, że ten **wyścig po rynek** skutkuje niższą jakością programów. **Nie ten wygrywa kto sprzedaje lepszy jakościowo program, tylko ten, kto pierwszy zajmie rynek.**

Także zakazaną praktyką są **grabiężcze ceny (predatory prices)**, polegające na okresowej lub ograniczonej do pewnego segmentu rynku obniżce cen. Dla powstrzymania konkurencji wystarczy **sama zapowiedź takiego działania.**

Aby zmobilizować nabywców wcześniejszych wersji programu do powtórnego zakupu nowej wersji, wydawcy oprogramowania z jednej strony oferują niższą cenę za *upgrade*, a z drugiej stosują strategię częściowej, **jednokierunkowej kompatybilności.**

Nowa wersja programu jest kompatybilna z poprzednią, ale nie odwrotnie. Ponieważ producent zaprzestaje sprzedaży starej wersji, jej użytkownicy szybko odnajdują się w sieci, która „nie do końca” jest kompatybilna z nową wersją.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 40 z 45

Powrót

Full Screen

Zamknij

Koniec

Komodyzacja SI/TI

Carr dzieli technologie na **własnościowe** oraz **infrastrukturalne** i utrzymuje, że te pierwsze – pod warunkiem iż są chronione przed imitatorami – są źródłem długoterminowej przewagi konkurencyjnej, podczas gdy wartość tych drugich wzrasta gdy są wspólne.

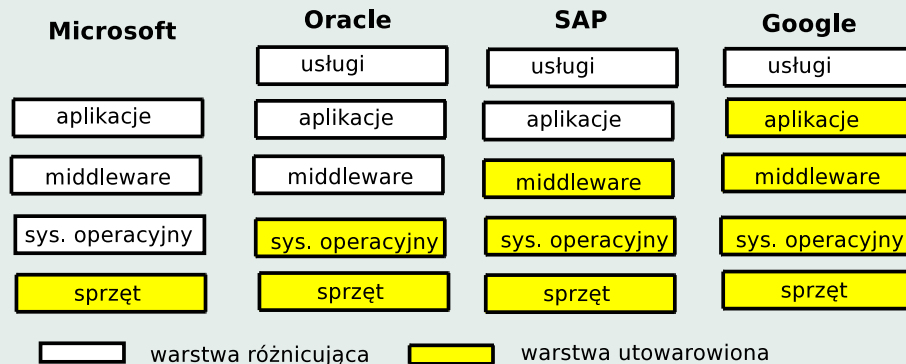
Ponieważ podstawowa funkcjonalność SI/TI – przetwarzanie, przechowywanie, przesyłanie danych stały się tanie i powszechnie dostępne, zatem SI/TI jest **technologią infrastrukturalną** – SI/TI jest niezbędna do prowadzenia działalności gospodarczej, ale nic więcej.

Podobnie jak miało to miejsce z innymi technologiami, także SI/TI zostanie **utowarowiona**. Perens szacuje, że w przypadku SI/TI 90% oprogramowania ma charakter infrastrukturalny, co więcej te 90% – z uwagi na warstwowy charakter SI/TI – jest podstawą, na podstawie której powstaje **technologia różnicująca**.

Większa dostępność, niższe koszty, lepsza jakość infrastruktury pozwala przeznaczyć większe środki na tę kategorię oprogramowania, która umożliwia uzyskanie przewagi na rynku.

Firmy działają na różnym poziomie stosu SI/TI, stąd co jest infrastrukturą, a co technologią różnicującą jest względne i zależy od przyjętej przez nie strategii rynkowej.

Przykłady przedstawiono na rys:



Komodyzacja SI/TI a strategia rynkowa wybranych producentów oprogramowania
(opracowano na podstawie West 2007)

Przykładowo infrastruktura firmy Google składa się z dziesiątków tysięcy komputerów PC wyposażonych w system operacyjny Linux – oba tanie i powszechnie dostępne. To co wyróżnia SI/TI Google, to opracowane

Struktura rynku SI/TI

Rozwój systemu . . .

Rozwój . . .

System Linux

Ochrona prawna . . .

Ekonomiczne . . .

Oprogramowanie . . .

Strona główna

Strona tytułowa

◀

▶

◀

▶

Strona 42 z 45

Powrót

Full Screen

Zamknij

Koniec

przez pracowników tego przedsiębiorstwa oprogramowanie, które jest pilnie strzeżoną tajemnicą.

Komodyzacja SI/TI w przypadku sprzętu stała się faktem: kupuje się komputer o określonej funkcjonalności a nie komputer firmy IBM czy komputer *Thinkpad*.

Zakładając, że oprogramowanie **nie jest** czymś szczególnym w historii ekonomii, należy oczekiwać, że komodyzacja obejmuje teraz wyższe warstwy stosu SI/TI, w tym system operacyjny.

Z tego punktu widzenia oprogramowanie **Open Source** jest czymś zupełnie normalnym i naturalnym

Inny scenariusz przyszłości, to model ASP, tj. usługi świadczone przez różnych dostawców w oparciu o wspólną infrastrukturę SI/TI zbudowaną z programów **Open Source**. Przykładem może być firma Google i jej usługi wyszukiwania zasobów Internetu.

Wracając do sposobów produkcji SI/TI Perensa, kontrakt – sposób produkcji oprogramowania, w którym użytkownik ponosi pełne koszty i ryzyko projektu SI/TI – jest jedynym sposobem tworzenia oprogramowania różnicującego.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 43 z 45

Powrót

Full Screen

Zamknij

Koniec

Tworzenie oprogramowania w ramach konsorcjum rozkłada koszty i ryzyko na członków konsorcjum i jest sprawniejszym ekonomicznie sposobem produkowania oprogramowania infrastrukturalnego.

Tworzenie oprogramowania **Open Source** jest podobne do „paradygmatu” konsorcjum, tyle że mniejszy stopień formalizacji zmniejsza koszty i ryzyko projektów. Projekt **Open Source** rozwija się ewolucyjnie [koszty i ryzyko są rozłożone w czasie]. Jeżeli projekt okazuje się nieudany, to straty są przeciętnie mniejsze, niż w przypadku wydawania oprogramowania

Oprogramowanie *Open Source*

Otwartość i zarządzanie projektem *Open Source*

We wczesnych opisach i analizach wytwarzanie oprogramowania **Open Source** było przedstawiane jako nieformalne, rozproszone geograficznie i oparte na dobrowolnym zaangażowaniu **wielu** wykonawców.

Taki sposób wytwarzania oprogramowania, który Raymond nazywa bazarowym porównywano z tradycyjnym sposobem produkcji

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 44 z 45

Powrót

Full Screen

Zamknij

Koniec

oprogramowania, który z kolei jest scentralizowany, sformalizowany i oparty o zaangażowanie wynikające *stricte* z pobudek komercyjnych.

Postulowano – obserwując takie projekty programistyczne jak Linux, Debian i Apache – iż wysoka jakość oprogramowania wynika z zastosowania nieformalnego sposobu zarządzania projektami.

W literaturze przedmiotu powszechny jest pogląd, że większa jakość programu wymaga większych **początkowych** nakładów. Jakościowo lepsze programy z kolei są tańsze w utrzymaniu i pielęgnacji.

W przypadku oprogramowania własnościowego jakość może zostać poświęcona dla osiągnięcia innych celów krótkookresowych, takich jak: termin ukończenia, mniejsze nakłady początkowe i/lub prymat strategii marketingowej nad zasadami sztuki inżynierskiej.

Badania empiryczne wskazują, iż identyfikowanie oprogramowania **Open Source** z **bazarowym** stylem produkcji, a programów własnościowych z **katedralnym** nie odpowiada rzeczywistości.

Po pierwsze, większość projektów **nie jest dziełem społeczności ale wąskiego grona wykonawców**. Po drugie, nawet w przypadku projektów bazarowych większość wykonawców jest wynajęta i opłacana przez organizacje komercyjne zainteresowane rozwojem projektu.

Struktura rynku SI/TI

Rozwój systemu...

Rozwój...

System Linux

Ochrona prawna...

Ekonomiczne...

Oprogramowanie...

Strona główna

Strona tytułowa



Strona 45 z 45

Powrót

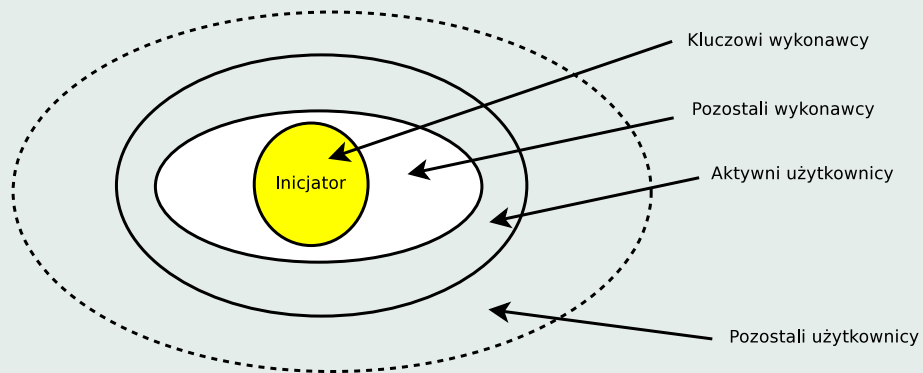
Full Screen

Zamknij

Koniec

Po trzecie wreszcie, całkiem sporo projektów **Open Source** ma charakter **projektów komercyjnych**.

W zależności od typu projektu **Open Source** struktura społeczności jest różna, co za tym idzie różna jest też aktywność poszczególnych interesariuszy.



Struktura interesariuszy projektu OSS (opracowano na podstawie Crowston i Howison 2005)