



Arkadiusz Kałkus

# Programista

Przewodnik po zawodzie

Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Grzegorz Krzystek  
Projekt okładki: Studio Gravite

Helion S.A.  
ul. Kościuszki 1c, 44-100 Gliwice  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [onepress@onepress.pl](mailto:onepress@onepress.pl)  
WWW: <http://onepress.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!  
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres  
<http://onepress.pl/user/opinie/proppz>  
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-7375-4

Copyright © Arkadiusz Kalkus 2021

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

# SPIS TREŚCI

|   |            |
|---|------------|
| <b>Wstęp</b> .....                          | <b>7</b>   |
| <b>Rozdział 1. Programowanie</b> .....      | <b>15</b>  |
| Na czym polega programowanie? .....         | 17         |
| Języki programowania .....                  | 46         |
| Bazy danych .....                           | 65         |
| Serwery i chmura .....                      | 76         |
| Narzędzia pracy programisty .....           | 82         |
| Testowanie oprogramowania .....             | 93         |
| <b>Rozdział 2. Branża IT</b> .....          | <b>101</b> |
| Programiści – piechota branży IT .....      | 103        |
| Dlaczego programiści mają dużo pracy? ..... | 114        |
| Zarobki programistów .....                  | 130        |
| Dochodowość branży IT .....                 | 135        |
| <b>Rozdział 3. Praca programisty</b> .....  | <b>141</b> |
| Na czym polega zawód programisty? .....     | 143        |
| Dzień pracy programisty .....               | 148        |
| Produkcja oprogramowania .....              | 151        |
| Specjalizacje .....                         | 165        |
| Slang .....                                 | 182        |
| Kariera .....                               | 185        |
| Zalety i wady zawodu .....                  | 193        |
| Odpowiedzialność .....                      | 208        |

|  |            |
|--|------------|
| <b>Rozdział 4. Droga do zawodu .....</b> | <b>221</b> |
| Predyspozycje .....                      | 223        |
| Matematyka .....                         | 227        |
| Studia .....                             | 235        |
| Bootcampy i kursy .....                  | 247        |
| Przebranżowienie .....                   | 251        |
| Poszukiwanie pracy .....                 | 258        |
| Rozmowa kwalifikacyjna .....             | 261        |
| Pierwsza praca .....                     | 265        |
| <b>Rozdział 5. Przyszłość .....</b>      | <b>269</b> |
| Programowanie jutro i pojutrze .....     | 271        |
| Ostatni zawód przed końcem pracy .....   | 276        |
| <b>O autorze .....</b>                   | <b>281</b> |

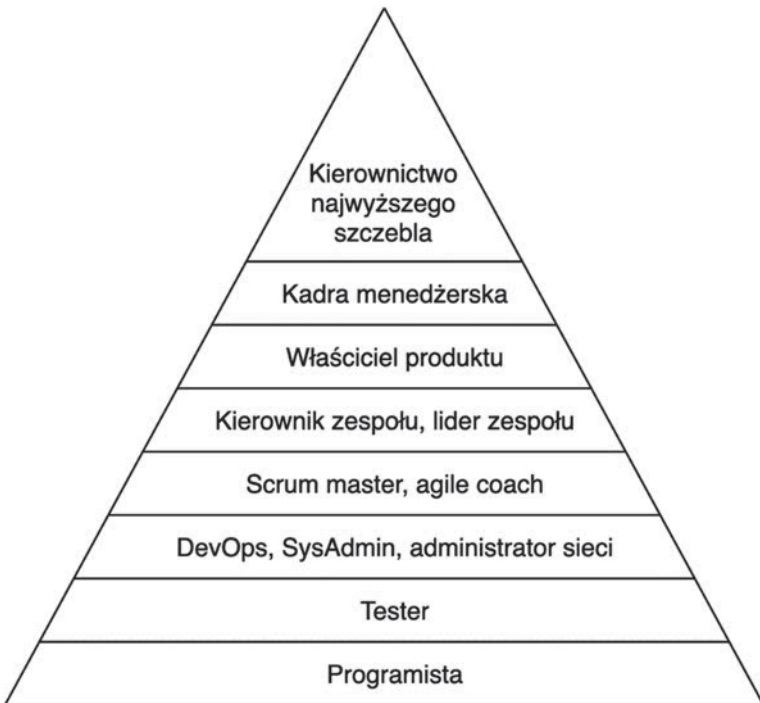
Rozdział 2.

# Branża IT



# PROGRAMIŚCI – PIECHOTA BRANŻY IT

Wszyscy znamy piramidy. Te egipskie, jak i te naukowe – potrzeb Masłowa oraz żywnościową. Gdybyśmy chcieli zrobić podobną, dzielącą branżę IT na specjalizacje, wyglądałaby mniej więcej tak:



Programistów w IT jest po prostu najwięcej. To oni oprogramowanie wytwarzają, poprawiają, aktualizują i modyfikują. Programiści tworzą narzędzia dla innych programistów (oraz administratorów, menedżerów itd.), programiści przygotowują

to, co testują testerzy, to ich efektami pracy zarządzają administratorzy, ich pracę nadzorują menedżerowie i optymalizują scrum masterzy i agile coachowie. Jak na dawnym polu bitwy — programiści są piechotą. Jest ich najwięcej, a wszyscy inni albo nimi sterują, albo wspomagają ich w pracy.

Nie znaczy to oczywiście, że są oni lepsi czy ważniejsi. Ważni są wszyscy, bo wszystkie elementy systemu muszą działać sprawnie, by całość prawidłowo funkcjonowała. Nie da się jednak nie zauważyć, że programiści są główną siłą, największą masą, esencją branży IT. Pozostałe zawody związane z wytwarzaniem oprogramowania opisuje kolejny podrozdział zatytułowany „Nie tylko programiści”.

## NIE TYLKO PROGRAMIŚCI

Programiści stanowią najliczniejszą grupę specjalistów biorących udział w procesie produkcji oprogramowania. Ról w branży IT jest jednak o wiele więcej. Każdy inżynier spotyka się i współpracuje codziennie z innymi ekspertami, których rola w tworzeniu produktu nie jest mniej istotna od roli programisty. Część z nich odpowiada za zarządzanie projektem, część monitoruje jakość tworzonego oprogramowania, a jeszcze inni pomagają je wdrożyć i dostarczyć klientom. Poniżej omówienie szeregu tych specjalizacji, z którymi programista ma do czynienia w codziennej pracy.

## TESTER/QUALITY ASSURANCE ENGINEER (QA)

Tester, jak wskazuje na to nazwa, odpowiedzialny jest za testowanie oprogramowania bądź — co sugeruje angielska wersja nazwy — za zapewnienie jego (dostatecznie wysokiej) jakości. Mówiąc

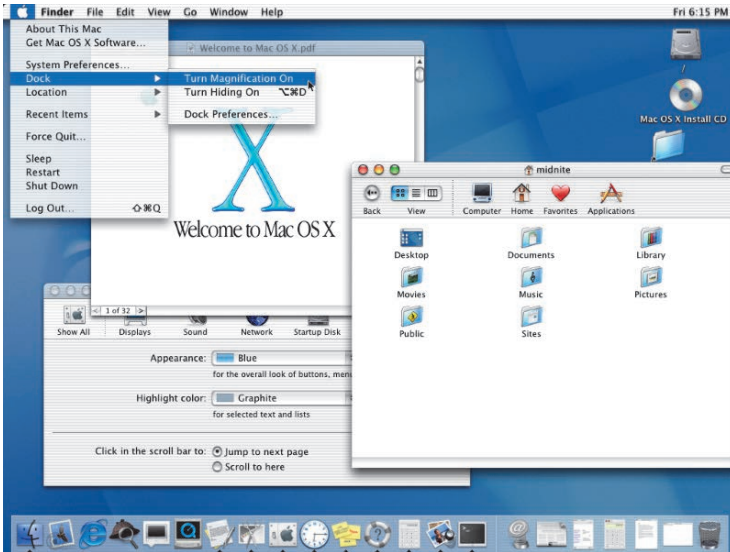


krótko — szuka błędów i zgłasza je programistom. Czasami poszukiwanie to odbywa się ręcznie, i często ciężko wykonać to zadanie inaczej, niemniej coraz częściej szukanie błędów odbywa się automatycznie. Testerzy tworzą scenariusze testowe, a następnie używając odpowiednich narzędzi, programują zautomatyzowane procedury testowe. Takie automatyczne testy są szczególnie wartościowe w przypadku testów regresyjnych (o których mowa w rozdziale „Testowanie oprogramowania”).

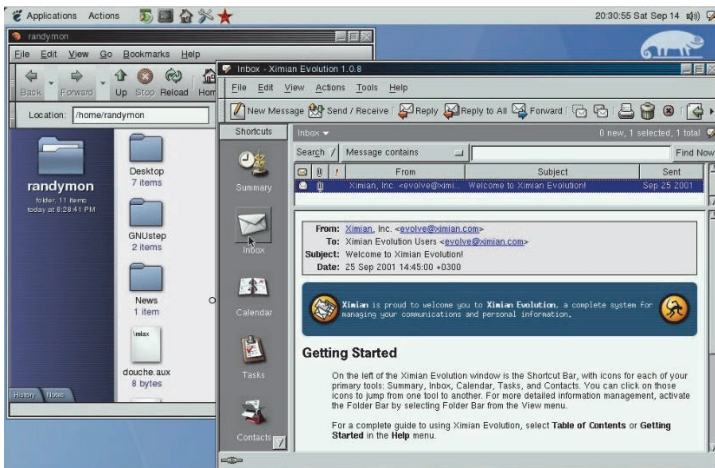
Tester to bez wątpienia najbliższy współpracujący z programistą członek zespołu (nie licząc oczywiście innych inżynierów oprogramowania). Praca testerów, choć bywa lekceważona, jest szalenie ważna. Dobry tester jest w stanie zapobiec wizerunkowej kompromitacji firmy, wyłapując w odpowiednim momencie poważny błąd.

## PROJEKTANT UI

Projektanci UI tworzą interfejs graficzny produktu — kształt i rozmieszczenie przycisków, ich kolor, obramowanie, wybierają paletę barw, czcionki i układ przestrzenny. Ich obecność lub brak, kunszt lub nieudolność może ogromnie wpłynąć na projekt. Wystarczy porównać wygląd MacOS X oraz Linuksa z okolic 2001 roku, by zrozumieć, jak imponujące mogą być efekty pracy projektantów UI (patrz grafika poniżej).



Źródło: [https://en.wikipedia.org/wiki/Mac\\_OS\\_X\\_10.0#/media/File:MacOSX10-0screenshot.png](https://en.wikipedia.org/wiki/Mac_OS_X_10.0#/media/File:MacOSX10-0screenshot.png), MacOS X, 2001



Źródło: <http://therandymon.com/index.php?/archives/345-Linux-in-2001-as-good-as-it-ever-got.html>, SuSE Linux 8.1, około 2001 r.

## SPECJALISTA UX

Specjaliści UX (ang. *user experience*, pol. doświadczenie użytkownika) pomagają projektować programy w taki sposób, by były wygodne w użyciu. Im łatwiej, szybciej i bardziej intuicyjnie możemy korzystać z aplikacji, tym więcej pracy musiał w nią włożyć jakiś ekspert UX. Można powiedzieć, że *user experience* jest odpowiednikiem ergonomii w świecie produktów cyfrowych.

Przykładem genialnego wręcz rozwiązania jest przycisk Start w systemach operacyjnych Windows. Ten doskonale widoczny i bardzo charakterystyczny element nawigacyjny pozwalał w Windows 95 jednym kliknięciem przejść do listy najważniejszych czynności w zarządzaniu komputerem: listy programów, ustawień, dokumentów i okienka pozwalającego wyłączyć lub zrestartować komputer. Prostota i wygoda świadczą o geniuszu tego rozwiązania.

O ironio, przykładem fatalnego UX był Windows 8, który przyciskowi Start próbował się pozbyć. Bardziej konkretnie, katastrofą *user experience* był sposób wyłączania komputera w tym systemie operacyjnym. Otóż przycisku Start nie było, ale kiedy przesunęliśmy mysz w lewy dolny róg ekranu pojawiał się tam prostokąt symbolizujący Menu Start. Jego naciśnięcie otwierało nową wersję menu, opartą na kafelkach. Każdy chyba użytkownik szukał tam — zgodnie z tradycją Windows — przycisku do wyłączania komputera. Na próżno. By wyłączyć komputer w Windows 8, trzeba było przesunąć mysz w prawy górny róg ekranu, w menu bocznym kliknąć ikonę Ustawienia, a następnie Zasilanie, po czym wybrać z menu Zamknij. Było to rozwiązanie nieintuicyjne, niewygodne i nadmiernie skomplikowane, z którego zresztą szybko zrezygnowano.

Rola specjalistów UX jest ogromnie istotna, bo nawet najlepiej wyglądająca, bezbłędnie napisana aplikacja, która będzie niewygodna w użytkowaniu, może przegrać z konkurencją.

## ADMINISTRATOR

Administrator to dość szerokie obecnie pojęcie, w którym mieszczą się różni specjaliści (np. administrator sieci, administrator baz danych, administrator serwerowni). W ogólnym sensie oznacza ono informatyka, który zajmuje się administracją pewnym fragmentem infrastruktury — czyli na przykład konfiguruje sieć i dba o jej poprawne działanie albo zarządza serwerownią. Charakterystyką pracy administratora jest dbałość o niezawodność. Im mniej przerw w optymalnym działaniu tego, czym administrator zarządza, tym lepiej.

Choć admini wydają się być nieco niewidzialni, są niezwykle istotni. O tym, jak ważna jest ich praca, zwykle przypominają sobie wszyscy, gdy coś, czym zarządzają, przestaje działać...

## DEVOPS

Do niedawna programiści i administratorzy byli od siebie dość mocno odseparowani. Ci pierwsi tworzyli aplikację, a następnie przygotowywali paczkę instalacyjną dla tych drugich, których zadaniem było wdrożyć ją na serwery lub komputery klientów. Powodowało to niestety sporo problemów. To, co działało na maszynach programistów, niekoniecznie działało tak samo i nie zawsze funkcjonowało poprawnie na serwerach i komputerach klientów. Programiści nie mieli wpływu na infrastruk-

ture, a admini nie wiedzieli omal nic o tym, co na serwerach instalują. Brak częstej komunikacji i współpracy utrudniał szybkie i sprawne dostarczanie produktu do klientów.

Z powyższych problemów zrodziło się nowe podejście, nazywane *DevOps*, łączące dwa słowa: *development* i *operations*, czyli zbliżające programistów i administratorów.

Inżynier DevOps to zwykle doświadczony programista, który posiada wiedzę o sieciach i wdrażaniu aplikacji, lub administrator, któremu nie jest obce programowanie i problemy programistów<sup>1</sup>. DevOps tworzy pomost pomiędzy programistami a administratorami. DevOpsi zajmują się takimi rzeczami jak automatyzacja wdrożeń (narzędzia to np. Jenkins, AzureDevops, Octopus Deploy, Drone.io), konfigurowanie infrastruktury jako kodu (Terraform) czy monitorowanie aplikacji. W chwili obecnej rośnie zapotrzebowanie na tego typu specjalistów, a popyt przerasta podaż.

## SPECJALISTA DS. BEZPIECZEŃSTWA

Choć często nie zdajemy sobie z tego sprawy, a jeszcze częściej zapominamy o tym, oprogramowanie pełni w naszym życiu wiele ważnych funkcji i przetwarza mnóstwo naszych prywatnych, wrażliwych danych. Płacimy za zakupy w sklepie internetowym, realizujemy e-receptę w aptece, robimy zdjęcie, którego nie chcielibyśmy pokazać publicznie, zamykamy bezprzewodowym kluczykiem samochód — dzień jak co dzień. Co jednak, gdyby po transakcji zniknęły pieniądze z naszego konta, gdyby ktoś ukradł nasze dane medyczne, gdyby wstydlive zdjęcie, które trzymamy w chmurze, zostało opublikowane, gdyby nasze auto pozostało otwarte?

---

<sup>1</sup> <https://puppet.com/blog/what-a-devops-engineer>.

Tym wszystkim katastrofom starają się zapobiec specjaliści do spraw bezpieczeństwa. Przed programistami stoi wiele zadań, a hackerom wystarczy czasami jeden błąd, niewielkie przeoczenie kodera, by włamać się do systemu. Z tego powodu eksperci od bezpieczeństwa wspierają zwykłych programistów. Przeprowadzają tzw. testy penetracyjne, projektują procedury w przedsiębiorstwach, kładąc nacisk na skanowanie oprogramowania pod kątem tzw. podatności (ang. *vulnerabilities*), ograniczają swobodę programistów w szczytnym celu uczynienia efektów ich pracy bezpieczniejszymi.

## SCRUM MASTER

Rola scrum mastera polega na pomocy zespołowi programistów w efektywnym wykorzystywaniu metodyki Scrum. Scrum master usuwa przeszkody, na jakie napotyka zespół, pomaga mu w komunikacji z właścicielem produktu, a także chroni programistów przed osobami spoza zespołu, które mogą próbować odciągać zespół od pracy nad konkretnymi zadaniami stanowiącymi cel sprintu (prosząc na przykład o dodatkową pracę niezwiązaną ze sprintem).

Scrum master organizuje i nadzoruje przebieg spotkań charakterystycznych dla metodyki Scrum, takich jak planowanie sprintu, retrospektywa czy daily scrum. Pomaga on zespołowi w efektywnym planowaniu pracy, a także szacuje jego wydajność, pomagając określić, kiedy praca nad danym oprogramowaniem może się zakończyć.

Więcej o roli scrum mastera w rozdziale „Produkcja oprogramowania”.

## AGILE COACH

Agile coach to osoba doświadczona w pracy z metodykami zwinnymi, która doradza przedsiębiorstwom. W przeciwieństwie do scrum mastera nie skupia się głównie na zespole i jedynie na metodyce Scrum, ale raczej na całej organizacji i na wspomaganii jej w budowaniu tak zwanej zwinności. Agile coachowie często pomagają w przejściu od starego modelu, w jakim firma tworzyła oprogramowanie, do nowego, opartego na agile<sup>2</sup>.

O metodykach zwinnych więcej w rozdziale „Produkcja oprogramowania”.

## KIEROWNIK ZESPOŁU/TEAM MANAGER (TM)

Kierownik zespołu pełni rolę formalnego menedżera. Uczestniczy w rozmowach rekrutacyjnych i bierze udział w decyzji o zatrudnieniu nowych osób, przeprowadza rozmowy okresowe z pracownikami, akceptuje podania o urlop, rozwiązuje konflikty międzyludzkie, motywuje i wspiera swoich ludzi oraz dba o harmonię w ramach zespołu. To rola odpowiedzialna i wymagająca umiejętności współpracy z ludźmi. Bywa, że łączy się ona ze stanowiskiem lidera programistów lub kierownika projektu.

---

<sup>2</sup> <https://porzadnyagile.pl/015-kto-to-jest-agile-coach/>,  
<https://www.cio.com/article/3294700/agile-coach-role-defined.html>.

## KIEROWNIK PROJEKTU/PROJECT MANAGER (PM)

Project manager jest pracownikiem, na którego barkach spoczywa odpowiedzialność za projekt. PM planuje, organizuje i nadzoruje projekt. Innymi słowy – rozpoczyna, przeprowadza i kończy przedsięwzięcie. Zawiera się w tym znalezienie i zaangażowanie udziałowców, zabezpieczenie budżetu, planowanie zakresu projektu, definiowanie jego celów, monitorowanie i nadzorowanie, a także jego zakończenie i zamknięcie<sup>5</sup>.

## ANALITYK BIZNESOWY/BUSINESS ANALYST (BA)

Analitycy biznesowi odpowiedzialni są za rozpoznawanie i definiowanie wymagań biznesowych, identyfikują problemy i możliwości w firmie oraz sugerują ich rozwiązanie, pozwalają też osiągać firmie cele biznesowe<sup>4</sup>. Analitycy pośredniczą między wymaganiami biznesu a programistami, przekształcają oczekiwania firmy na konkretne elementy projektu, dzięki którym pracownicy przedsiębiorstwa będą mogli sprawniej działać. Programiści nie mają bowiem zwykle za wiele rozeznania w problemach biznesowych firmy, dla której tworzą projekt. Innymi słowy – analitycy, nomen omen, analizują, czego firma

---

<sup>5</sup> <https://www.pmi.org/learning/library/pm-role-lean-agile-world-9350>, <https://www.wrike.com/project-management-guide/faq/what-is-an-it-project-manager/>.

<sup>4</sup> <https://www.roberthalf.co.nz/our-services/finance-accounting/business-analyst-jobs>.



potrzebuje, co usprawni biznes, czego trzeba, by działać sprawniej, i te wymagania przedstawiają właścicielowi produktu i zespołowi deweloperskiemu. Mówiąc krótko, tłumaczą potrzeby biznesu na wymagania biznesowe związane z produktem.

## WŁAŚCICIEL PRODUKTU/PRODUCT OWNER (PO)

Product owner lub po polsku właściciel produktu jest jedną z ról w metodyce Scrum. Jest to osoba reprezentująca użytkowników końcowych oraz inwestorów czy też interesariuszy lub sponsorów projektu<sup>5</sup>. Mówiąc krótko, jest to pracownik, który odpowiada za produkt. PO zarządza listą zadań dla zespołu deweloperskiego, decyduje o priorytetach, planuje sprinty (patrz podrozdział „Produkcja oprogramowania”) i, najogólniej rzecz ujmując, czuwa nad tym, by prace nad produktem przebiegały sprawnie i w odpowiednim kierunku. Product owner ten kierunek nadaje. Właściciel produktu często negocjuje również budżet i etapy projektu.

## SPECJALISTA DS. ZASOBÓW LUDZKICH/ HUMAN RESOURCES SPECIALIST (HR)

We współczesnym świecie (nie licząc może Japonii) nikt nie pracuje w jednej firmie przez dekady. Pracownicy przychodzą i odchodzą, nieustannie poszukując lepszych możliwości rozwoju, ciekawszych projektów, lepiej płatnych stanowisk. Dział HR wraz z zewnętrznymi agencjami rekrutującymi pracuje nad

---

<sup>5</sup> [https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)#Product\\_owner](https://en.wikipedia.org/wiki/Scrum_(software_development)#Product_owner).

tym, by odpływ pracowników nie przewyższył napływu nowych. Chodzi tu zarówno o dbanie o pracowników już zatrudnionych, badanie ich satysfakcji, rozwiązywanie problemów, konfliktów, próby realizowania ich postulatów, jak i rekrutowanie nowych. Z perspektywy programisty dział HR zajmuje się głównie zatrudnieniem: rekrutuje, przedstawia ofertę pracy, negocjuje ją i finalizuje. Po dołączeniu do firmy natomiast od czasu do czasu bada satysfakcję, spełnia zadania administracyjne oraz przy zakończeniu współpracy organizuje tzw. exit interview, czyli pyta o przyczyny odejścia z firmy.

## DLACZEGO PROGRAMIŚCI MAJĄ DUŻO PRACY?

Rzadko zastanawiamy się nad tym, jak bardzo sztuczny jest świat, w którym żyjemy. Od momentu, gdy otwieramy oczy po przebudzeniu, otoczeni jesteśmy przedmiotami wykonanymi przez ludzi. Chyba jedynie rośliny doniczkowe, odrobina warzyw i owoców oraz zwierzęta domowe w naszym mieszkaniu są czym naturalnym. Wszystko inne, na co byśmy nie spojrzeli, jest dziełem ludzkich rąk i pomysowości.

Spoglądamy na sufit i ściany — ludzki wynalazek. Meble ze sklejk — stworzone przez ludzi. Telefon, zegarek, buty, łóżko, ubrania — wszystkie te rzeczy wymyślił człowiek i ludzie pracują codziennie godzinami, by obywatele wszystkich krajów na Ziemi mieli je w swoich domach i żyli wygodnie. Wychodzimy z domu (zakładając, że żyjemy w mieście), a tam chodniki z betonu, ulice z asfaltu i metalowo-plastikowo-szklane samochody na nich.

Kiedy przyjrzymy się dobrze naszemu otoczeniu, gdy zaczniemy patrzeć na zachowania swoje własne oraz innych osób, zacznie do nas powoli docierać krajobraz ekonomiczny. Za każdym praktycznie fragmentem naszego życia, za każdym przedmiotem, stoi jakaś firma, jakaś branża i pewna ilość technologii, które pozwoliły na jego wyprodukowanie. Zastanówmy się, jak mniej więcej przebiegał na przestrzeni lat rozwój techniki<sup>6</sup>, a potem pomyślimy, gdzie jesteśmy obecnie i jak się to ma do pracy programistów...

## PIERWSZE KROKI

Najstarszymi „technologiami”, które zmieniły los ludzkości, są ogień i kamienne narzędzia. Ten pierwszy dał człowiekowi przewagę nad innymi zwierzętami pozwalającą przechowywać żywność dłużej i jeść to, czego wcześniej, bez obróbki termicznej, spożywać nie mogliśmy. Narzędzia z kolei pozwoliły nam ścierać się z pozostałymi gatunkami na równej stopie. Bezbronny człowiek był przerażająco słaby w starciu z tygrysami, nosorożcami i innymi zwierzętami. Uzbrojony w nóż czy toporek — mimo że jedynie kamienny — stał się konkurencyjny.

Około 10 000 lat temu z powodu końca epoki lodowcowej i postępujących zmian klimatycznych nastąpiła rewolucja neolityczna<sup>7</sup>. Nasi przodkowie porzucili koczowniczy tryb życia, w którym łowili zwierzyne i zbierali pożywienie, a zamiast tego zajęli się życiem osiadłym, opartym na rolnictwie i hodowli zwierząt. Wydawać by się mogło, że uprawa roli nie jest technologią, bo przywykliśmy do nazywania tym mianem spraw

---

<sup>6</sup> [https://en.wikipedia.org/wiki/Technological\\_revolution](https://en.wikipedia.org/wiki/Technological_revolution).

<sup>7</sup> [https://en.wikipedia.org/wiki/Neolithic\\_Revolution](https://en.wikipedia.org/wiki/Neolithic_Revolution).

związanych z automatyzacją, informatyką, elektroniką i innymi nowoczesnymi osiągnięciami naszej cywilizacji, ale tak naprawdę nią jest. Zgodnie ze słownikową definicją technologia to metoda produkcji lub przetwórstwa. Rolnictwo i hodowla są więc technologiami produkcji żywności. Technologami, dodajmy, o wiele wydajniejszymi niż łowiectwo i zbieractwo.

Ta wydajność nowo odkrytego sposobu pozyskiwania żywności zapoczątkowała spektakularny rozwój naszego gatunku. Obfitość jedzenia poskutkowała eksplozją demograficzną, a osiadły tryb życia wraz ze wzrostem populacji pozwolił na rozwinięcie podwalin cywilizacji. Pojawiły się domy, naczynia, tkaniny, a także handel, usługi i polityka. Polując na zwierzynę, podążaliśmy za nią, tropiliśmy, wybijaliśmy i przemieszczaliśmy się na inny teren, gdzie było jej więcej. Uprawiając rolę, mogliśmy w końcu wybudować domy, wsie i miasta. Między osadami trzeba było handlować. Hodowla i uprawa zajmowały mniej czasu, więc część osób mogła się zająć wytwarzaniem dóbr polepszających jakość życia — stąd garmcarstwo i tkactwo. W miastach potrzebna była organizacja społeczna, więc powstała polityka...

Przez kolejne wieki rozmaite cywilizacje dokonywały istotnych odkryć. Wynaleziono proch, papier, druk, kompas, pieniądze, udoskonalano metalurgię, rozwijano chemię, wszystko to jednak nie miało charakteru epokowej zmiany. Aż do czasu rewolucji naukowej.

## REWOLUCJA NAUKOWA

Za początek rewolucji naukowej<sup>8</sup> uznaje się często publikację (w 1543 roku) pracy Mikołaja Kopernika *O obrotach sfer niebieskich*. Samych przełomów było w nauce, rzecz jasna, wiele, natomiast okres kilkuset lat od końca renesansu do kresu XIX wieku był szczególny.

Wiek XV przyniósł upadek Konstantynopola, a wraz z nim kres cesarstwa wschodniorzymskiego. W wyniku tych wydarzeń rozpoczęła się emigracja bizantyjskich uczonych na zachód<sup>9</sup>, w szczególności w region obecnych Włoch. Doprowadziło to do odrodzenia się w Europie wiedzy o osiągnięciach naukowych starożytnej Grecji i Rzymu, co z kolei zapoczątkowało renesans.

To, co wydarzyło się potem, było wielkim, triumfalnym pochodem nauki. Idee antyku zastąpiono nowymi teoriami. Nastąpił przewrót kopernikański, a Galileusz i później Newton położyli fundamenty pod spektakularny rozwój fizyki. Filozofowie oświecenia myśleli inaczej od swych poprzedników. Pojawił się nacisk na naturalizm, czyli skupienie na rzeczywistości, a nie zjawiskach nadprzyrodzonych. Kolejną tendencją był racjonalizm – myślenie chłodne, trzeźwe, logiczne, zamiast intuicyjnego. Empiryzm, zapoczątkowany przez Francisca Bacona, filozofa końca epoki odrodzenia, rozwijał się w oświeceniu m.in. dzięki Johnowi Locke’owi. Nauka stała się doświadczalna. Z wolna zaczęto metodycznie poszukiwać dowodów na stawiane tezy.

---

<sup>8</sup> [https://en.wikipedia.org/wiki/Scientific\\_Revolution](https://en.wikipedia.org/wiki/Scientific_Revolution).

<sup>9</sup> [https://en.wikipedia.org/wiki/Greek\\_scholars\\_in\\_the\\_Renaissance](https://en.wikipedia.org/wiki/Greek_scholars_in_the_Renaissance).

Owe kilkaset lat doprowadziło do stworzenia współczesnych instytucji naukowych, refleksji nad tym, jak należy prowadzić badania, by dochodzić do prawdy, oraz do powstania metody naukowej<sup>10</sup>. Poczęto obserwować, mierzyć, eksperymentować i dowodzić. Rzetelnie zbadana wiedza zaczęła być gromadzona, a następnie wykorzystywana praktycznie. Nauka podzieliła się na dziedziny, o których do dziś uczymy się w szkołach, a odkrycia, jakie zostały poczynione w okresie kilkuset zaledwie lat, zmieniły świat w sposób dla naszych przodków niewyobrażalny. I co najważniejsze — umożliwiły powstanie cywilizacji naukowo-technicznej.

## REWOLUCJA PRZEMYSŁOWA

Rewolucja przemysłowa była kolejnym z największych przełomów w historii ludzkości. Być może obok rewolucji naukowej największym. Przed jej nadejściem przytłaczająca większość ludzi żyła na wsi, a wszystkie przedmioty wytwarzane były ręcznie przez rzemieślników. Szacunki mówią, że rolnictwem mogło zajmować się nawet 80% populacji<sup>11</sup> (dla porównania obecnie w Polsce jest to 12%, a w UE około 4%<sup>12</sup>). Innymi słowy, ogromna większość ludzi produkowała żywność, a zaledwie jedna piąta mogła zająć się wytwarzaniem ubrań, ceramiki, mebli i innych dóbr, w tym absolutna mniejszość miała czas i pieniądze na naukę i sztukę.

---

<sup>10</sup> [https://pl.wikipedia.org/wiki/Metoda\\_naukowa](https://pl.wikipedia.org/wiki/Metoda_naukowa).

<sup>11</sup> <https://webs.bcp.org/sites/vcleary/ModernWorldHistoryTextbook/IndustrialRevolution/IREffects.html>.

<sup>12</sup> <https://obserwatorgospodarczy.pl/komentarze-i-analizy/587-w-polsce-ciagle-jest-znaczny-odsetek-osob-pracujacych-w-rolnictwie>.

Rewolucja przemysłowa w przeciągu zaledwie mniej więcej 70 lat od 1760 do około 1830 roku dosłownie zmiotła z powierzchni Wielkiej Brytanii — gdzie się rozpoczęła — stary ład. W kolejnych zaś dekadach odmieniła świat nie do poznania.

Najlepszym przykładem tych zmian z naszego własnego podwórka jest Łódź. W roku 1820 – gdy zdecydowano o rozpoczęciu procesu industrializacji — miejscowość liczyła około 800 mieszkańców. Osiemdziesiąt lat później, w roku 1900, liczba ta wynosiła 300 000<sup>13</sup>. Przez te osiemdziesiąt lat wybudowano około 2 km<sup>2</sup> tego, co dziś stanowi centrum miasta, choć na początku w Łodzi nie istniał nawet jeden murowany obiekt, a pierwszy piętrowy budynek z cegieł zbudowano w 1835 roku (przy ul. Piotrkowskiej 243<sup>14</sup>). Praktycznie wszystkie kamienice, które możemy dziś podziwiać w śródmieściu Łodzi, powstały do roku 1920, czyli w przeciągu stu lat od początku industrializacji miejscowości. Bez rewolucji przemysłowej nie byłoby wielopiętrowych, ceglanych budynków stanowiących symbol miasta — w tym słynnej Manufaktury, czyli wzniesionej w latach 1872 – 1892 fabryki tekstylnej Izraela Poznańskiego. Nie byłoby pałaców fabrykantów, nie byłoby być może w ogóle takiego miasta na mapie Polski. Na pewno nie byłoby w każdym razie znaczącego ośrodka miejskiego.

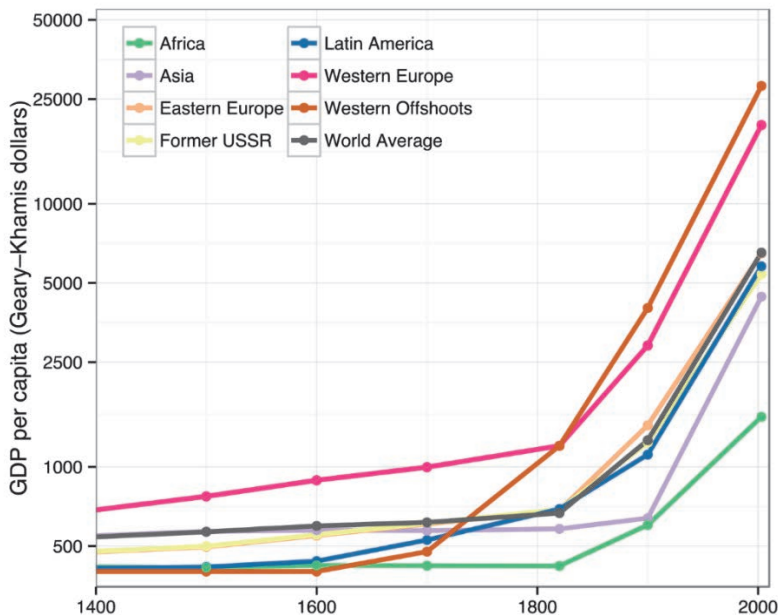
Historia Łodzi, zobrazowana przez Władysława Reymonta w *Ziemi obiecanej*, była właściwie powtórzeniem tego, co działo się kilkadziesiąt lat wcześniej w Wielkiej Brytanii. To tam właśnie pierwszy raz w historii więcej ludzi zamieszkało w miastach niż na wsi. W książce naszego noblisty również widzimy migrację do miasta, wielkie zmiany w życiu bohaterów, ogromne szanse biznesowe i nagły przyrost bogactwa wśród właścicieli fabryk.

---

<sup>13</sup> <https://pl.wikipedia.org/wiki/Łódź>.

<sup>14</sup> [https://pl.wikipedia.org/wiki/Kamienica\\_Gottlieba\\_Beera\\_w\\_Łodzi](https://pl.wikipedia.org/wiki/Kamienica_Gottlieba_Beera_w_Łodzi).

Szacunki wzrostu PKB na osobę pokazują, jak niebywały wpływ na produktywność miała rewolucja przemysłowa:



Źródło: [https://en.wikipedia.org/wiki/Industrial\\_Revolution#/media/File:Historic\\_world\\_GDP\\_per\\_capita.svg](https://en.wikipedia.org/wiki/Industrial_Revolution#/media/File:Historic_world_GDP_per_capita.svg)

Przez setki lat produktywność przeciętnego człowieka omalże stała w miejscu. Od czasu rewolucji przemysłowej zaczęła się zwiększać w piorunującym tempie. W przeciągu pierwszych kilkudziesięciu lat omal trzykrotnie. Razem z kolejnymi przełomami technologicznymi na przestrzeni dwustu lat produkt krajowy brutto na osobę wzrósł w Europie Zachodniej trzydziestokrotnie.

Czym była rewolucja przemysłowa? Była przejściem od produkcji ręcznej do zautomatyzowanej.



Pierwszą branżą, w której nastąpiły zmiany, było włókiennictwo. Początkowo pojawiały się urządzenia usprawniające pracę, takie jak czółenko szybkobieżne (ang. *flying shuttle*) czy przędąca Joasia (ang. *spinning Jenny*). Dużym przełomem stała się maszyna przędzalnicza napędzana energią wodną (konkretnie przez użycie koła wodnego). Wynalazek ten oddzielił produkcję od siły ludzkich — czy też zwierzęcych — mięśni. W roku 1712 Thomas Newcomen zbudował silnik parowy, który udoskonalony potem przez Jamesa Watta stał się maszyną parową i ikoną rewolucji przemysłowej. Nazwisko Watta zaś zostało symbolem mocy w fizyce. Co ważne, Wielka Brytania posiadała znaczące zasoby węgla, który mógł być użyty do zasilania nowo wynalezionych maszyn.

W krótkim czasie okazało się, że zautomatyzowana produkcja tekstylna zasilana maszyną parową opalaną brytyjskim węglem jest w stanie wyprodukować więcej i szybciej. Dziesiątki razy więcej i szybciej...

Mechanizacja, raz odkryta, rozprzestrzeniła się po innych gałęziach gospodarki. Zaczęto budować parowe statki, parowozy. Powstała kolej. Dynamicznie rozwijał się transport. Wzrosło zapotrzebowanie na stal. Pod koniec pierwszej rewolucji przemysłowej na horyzoncie pojawiły się niespotykane perspektywy rozwoju.

## DRUGA REWOLUCJA PRZEMYSŁOWA

Druga rewolucja przemysłowa, zwana też rewolucją technologiczną, trwała od około 1870 roku do początku pierwszej wojny światowej. Był to niezwykle czas pełen wynalazków i zmian w życiu każdego Europejczyka i Amerykanina, choć przemian w mniejszym stopniu doświadczyły również Japonia i Chiny oraz

inne kraje. Powstały wówczas urządzenia tak przełomowe jak silnik gazowy, silnik spalinowy, dynamit, telefon, lampa naftowa, żarówka, odkurzacz, radio, aparat fotograficzny, tramwaj, metro, samochód i samolot.

Okres drugiej rewolucji przemysłowej przyniósł nowe źródło energii — ropę — oraz przełom w sposobie zasilania urządzeń — elektryczność. Pierwsze maszyny napędzał głównie węgiel — źródło ciepła i pary. Kilka dekad później to elektryczność zasilala rozmaite urządzenia, a silniki spalinowe napędzały większość pojazdów. Pierwszy silnik spalinowy, który odniósł komercyjny sukces, stworzył w 1860 roku Étienne Lenoir, zaś to, co można uznać za pierwszy współczesny silnik spalinowy, skonstruował w 1876 roku Nicolaus Otto<sup>15</sup>. On też jest twórcą cyklu Otta stanowiącego teoretyczny opis zjawisk zachodzących w tłokowych silnikach spalinowych o zapłonie iskrowym. Ojcami kolejnych istotnych konstrukcji byli inżynierowie, których nazwiska zna każdy fan motoryzacji, choć niekoniecznie z uwagi na ich wkład w budowę silników. We współpracy z Otto Gottlieb Daimler oraz Wilhelm Maybach stworzyli i opatentowali czterosurowy silnik, w którym paliwo ulegało sprężeniu. W 1879 roku Karl Benz opatentował dwusurowy silnik benzynowy. Silniki wysokoprężne natomiast swoją nazwę wzięły od Rudolfa Diesela, który w roku 1892 zbudował pierwszy z nich. Przewagi nowych konstrukcji doprowadziły do wieloletniej dominacji silników spalających ropę i benzynę, kładąc kres epoce maszyny parowej.

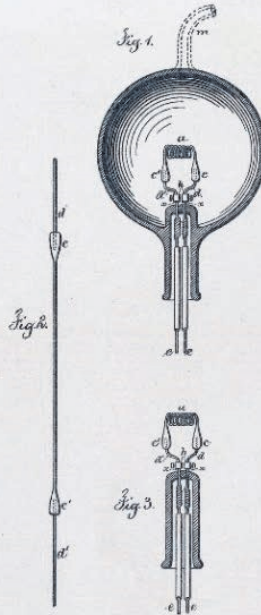
---

<sup>15</sup> [https://en.wikipedia.org/wiki/Internal\\_combustion\\_engine](https://en.wikipedia.org/wiki/Internal_combustion_engine).

T. A. EDISON.  
Electric-Lamp.

No. 223,898.

Patented Jan. 27, 1880.



Witnesses  
Charles Smith  
Geo. P. Mahony

Inventor  
Thomas A. Edison

for  
Lemuel W. Perrell  
att'y

THE MERRILL STORE CO. PHOTO-LITHO. WASHINGTON, D. C.

Źródło: [https://en.wikipedia.org/wiki/Second\\_Industrial\\_Revolution#/media/File:Light\\_bulb\\_Edison\\_2.jpg](https://en.wikipedia.org/wiki/Second_Industrial_Revolution#/media/File:Light_bulb_Edison_2.jpg)

Nie mniej przełomowe były wynalazki związane z energią elektryczną. W 1880 roku Thomas Edison opatentował żarówkę, a zaledwie siedem lat później Nikola Tesla otrzymał patent na silnik elektryczny zasilany prądem przemiennym.

Podwaliny pod praktyczne zastosowania elektryczności położył kilka dekad wcześniej Michael Faraday. Fundamentalne znaczenie dla rozwoju przemysłu miało odkryte przez niego w 1831 roku zjawisko indukcji elektromagnetycznej. Do dziś jest ono podstawową metodą wytwarzania prądu elektrycznego, a także stanowi podstawę działania licznych urządzeń, takich jak prądnice, silniki, transformatory czy kuchenki indukcyjne.

Elektryczność okazała się doskonałym źródłem energii dla nowo opracowanych urządzeń automatyzujących ludzką pracę. Silnik parowy wymagał spalania węgla na miejscu — do urządzeń elektrycznych energię można było dostarczyć kablami, a także transportować ją na wielkie odległości. Nie bez znaczenia było też zmniejszenie ilości pożarów.

## REWOLUCJA CYFROWA

Rewolucja cyfrowa, tak samo jak przemysłowa, stała się dla nas jakby niewidoczna. Każdego dnia korzystamy z jej owoców. Urządzenia i technologie te są w codziennym użyciu, nie wyobrażamy sobie życia bez nich do tego stopnia, że wręcz nie zwracamy na nie uwagi. Przynajmniej do chwili, do której coś przestaje działać...

Najogólniej rzecz ujmując, rewolucja cyfrowa to przejście z urządzeń kompletnie mechanicznych oraz elektroniki analogowej na elektronikę cyfrową. Transformacja ta umożliwiła rzeczy, które wcześniej były kompletnie poza zasięgiem. Najwyraźniej uwidocznili się to w postępującej miniaturyzacji.

Znakomitym jej przykładem jest ewolucja telewizora — od wielkiego, lampowego, awaryjnego, analogowego pudła do niewielkiego płaskiego ekranu, z omal nieskończoną ilością kanałów, a nawet integracją z internetem. Urządzenia nie tylko stały się mniejsze. Stały się też wielofunkcyjne. Tradycyjny zegarek mechaniczny, choćby najdroższy, najbardziej dopracowany, będący arcydziełem szwajcarskiego zegarmistrzostwa, ustępuje pod względem ilości funkcji nawet dość prostym elektronicznym mechanizmom. Za przykład niech posłuży Casio CA-53W. Poza wskazywaniem godziny ten kwarcowy zegarek z 1984 roku posiada stoper, alarm i... kalkulator.



Źródło: [https://commons.wikimedia.org/wiki/File:Casio\\_CA-53W,\\_1.jpg](https://commons.wikimedia.org/wiki/File:Casio_CA-53W,_1.jpg)



Źródło: [https://en.wikipedia.org/wiki/Patek\\_Philippe\\_SA#/media/File:Patek-Philippe\\_MG\\_2583.jpg](https://en.wikipedia.org/wiki/Patek_Philippe_SA#/media/File:Patek-Philippe_MG_2583.jpg)

Jedna jeszcze rzecz wydarzyła się za sprawą cyfrowej rewolucji. Z pozoru nieistotna, ale w istocie przełomowa. Digitalizacja umożliwiła błyskawiczne powielenie informacji, stworzenie kopii idealnie naśladującej pierwowzór. W dodatku tanio. Wręcz za darmo. Przypomnijmy, że na początku pisma trzeba było ręcznie przepisywać książki. Prasa drukarska obniżyła znacząco ten koszt, umożliwiając masowe czytelnictwo. Cyfrowa reprodukcja poszła krok dalej. Pozwoliła wykonywać idealne, szybkie i tanie kopie tekstów i nagrań muzycznych oraz filmowych. Informacja została oddzielona od nośnika — bo na każdym stała się taka sama. Dwie płyty gramofonowe są mimo wszystko nieco różne, a dwa pliki z albumem muzycznym mogą być zawsze idealnie takie same.

Ta redukcja omal do zera kosztu kopiowania miała potężne skutki dla każdego z nas. W pierwszej dekadzie XX wieku, wraz z coraz tańszym i lepszym dostępem do internetu, ogromnie rozpowszechniło się piractwo. W odpowiedzi na nie zaczęły się upowszechniać tanie serwisy streamingowe — Spotify, Tidal, Netflix, HBO Go. W chwili obecnej w zasięgu kilku kliknięć i kilkudziesięciu złotych miesięcznie — właśnie dzięki tej niskiej cenie kopiowania plików — mamy przepastny ogrom muzyki i książek.

\*\*\*

W centrum rewolucji cyfrowej na głównej jej scenie mimo całej reszty usprawnień i technologii stoją jednak komputery.

Od lat 50. do 70. dokonano przełomów w elektronice, od produkcji krzemowych półprzewodników, po układy scalone. W 1971 roku Intel wypuścił na rynek pierwszy mikroprocesor w całości zintegrowany w jednym chipie — Intel 4004. Siedem lat później do produkcji wszedł 16-bitowy Intel 8086. Dał on początek architekturze x86, która przez kolejne lata stanowiła standard zestawu instrukcji. Procesory Intela albo ich kopie lub ulepszenia były sercem zarówno mikrokomputerów — takich jak popularne w Polsce Atari czy ZX Spectrum — jak i komputerów klasy PC. Poza tym oczywiście znalazły inne zastosowania — użytkowane były na przykład w programowalnych elektronicznych kalkulatorach Basicom.

Przewagi technologii cyfrowych oraz perspektywy ich rozwoju zaczęły wypierać urządzenia analogowe.

W porównaniu z technologią analogową urządzenia cyfrowe mają następujące zalety:

- są tańsze,
- eliminują problem szumu/zakłóceń sygnału,
- są łatwe w produkcji jako układy scalone,
- pozwalają lepiej przechowywać dane,
- mogą być programowalne.

Tym, co najłatwiej było zauważyć osobom mającym okazję doświadczyć przechodzenia z urządzeń cyfrowych na analogowe, było zwiększenie jakości dźwięku i obrazu. Magnetowidy VHS i odtwarzacze kasetowe oraz płyty winylowe traciły powoli jakość wraz z wielokrotnym odtwarzaniem i upływem czasu. Płyta CD i DVD natomiast zawsze grała tak samo.

Warto zwrócić uwagę na ostatni punkt z listy zalet — programowalność. Urządzenia cyfrowe pozwalają na programowanie. Umożliwia to po pierwsze obniżenie kosztu, bo zaprojektowanie programu jest z reguły tańsze od zaprojektowania układu elektronicznego, a po drugie pewne zmiany i poprawki błędów możliwe są już po wyprodukowaniu urządzenia. Choćby w przypadku smartfonów — proces ten trwa nieustannie i regularnie otrzymujemy nowe wersje oprogramowania i aktualizacje usprawniające pracę telefonu.

\*\*\*

Przejsie z technologii analogowej na cyfrową stworzyło gigantyczny rynek dla usług programistów. Przytłaczająca większość współczesnych urządzeń zawiera elementy, które wymagają



oprogramowania. Programy znajdziemy w telefonach, telewizorach, samochodach, samolotach, dronach, pociągach, na statkach, w drukarkach, skanerach, kamerach, aparatach fotograficznych, instrumentach cyfrowych, odkurzaczach automatycznych, pralkach, zmywarkach, mikrofalówkach, płytach indukcyjnych, satelitach, bezprzewodowych głośnikach, ciśnieniomierzach, maszynach do szycia, zabawkach, czytnikach e-booków, bankomatów, urządzeniach do rezonansu magnetycznego i USG, satelitach, czołgach, radarach, konsolach do gry, klimatyzatorach oraz oczywiście tabletach, laptopach i innych komputerach. Słowem: wszędzie.

Powyższa lista odpowiada na tytułowe pytanie podrozdziału. Dlaczego programiści mają dużo pracy? Ponieważ prawie każde współczesne urządzenie wymaga ich pracy, by mogło powstać.

Dla pełnego obrazu trzeba jednak zauważyć jeszcze jedno. O ile od wielu dekad trwa już informatyzacja, to od ponad dwóch dziesięcioleci następuje internetyzacja. Nie przez przypadek amerykańskie spółki technologiczne, które kojarzą się z erą internetu — takie jak Facebook, Amazon, Apple, Netflix i Google — symbolizują najwyższe zyski i wzrost cen akcji. Internet stał się na tyle nieodłącznym elementem współczesnego życia, tak kluczowym fragmentem biznesu większości firm, że samo już utrzymanie i tworzenie oprogramowania związanego z obecnością online daje programistom fenomenalne perspektywy zawodowe. Pod koniec 2019 roku samo Google zatrud-

niało około 120 000 pracowników, nie wliczając w to podwykonawców. W przypadku Microsoftu to około 160 000 osób, Facebooka — 45 000, a Netfliksa — omal 9000.

Ostatnim elementem tworzącym znaczącą ilość miejsc pracy dla specjalistów od oprogramowania są militaria. Współczesny sprzęt wojskowy jest pełen nie tylko elektroniki, ale i oprogramowania. Od GPS, przez systemy rakietowe i antyrakietowe, autonomiczne drony, wyposażenie czołgów, odrzutowców i okrętów, po systemy świadomości sytuacyjnej — przemysł zbrojeniowy potrzebuje programistów. W samym Lockheed Martin pracuje ponoć 60 000 inżynierów. Sądzę, że przynajmniej 1/10 z nich to programiści.

Reasumując, programiści mają dużo pracy, bo w każdej niemal dziedzinie gospodarki jest ona potrzebna. W rozdziale „Przyszłość” spróbuję odpowiedzieć na pytanie, jak może wyglądać perspektywa kolejnych dekad.

## ZAROBKI PROGRAMISTÓW

Zarobki programistów w Polsce obrosły legendą. W sieci znajdziemy przynajmniej kilkadziesiąt artykułów poruszających ten temat — zwykle w sensacyjnym tonie, zazwyczaj wymieniających kwoty będące wielokrotnością średniej krajowej. W powszechnej opinii wynagrodzenia w branży IT są wysokie. Część osób uważa wręcz, że są one zbyt duże, i spodziewa się „pęknięcia bańki” i ich spadku.

Jakie są zatem zarobki programistów? Przede wszystkim ogromnie zróżnicowane. Przepaść między stawkami oferowanymi początkującym koderom a tymi, na jakie mogą liczyć uta-

lentowani inżynierowie z wieloletnim doświadczeniem (i nie-raz doktoratem), jest taka mniej więcej jak pomiędzy młodym prawnikiem w małym mieście a najlepszym adwokatem w stolicy albo jak między internistą z prowincji a znanym na cały kraj chirurgiem. Słowem, ogromna. W praktyce oznacza to tyle, że w najgorszym przypadku programista może liczyć na minimalne wynagrodzenie. Najwyższe stawki są praktycznie nieznanne. Największe z publicznie wymienianych oscylują na chwilę obecną<sup>16</sup> w granicach 40 000 – 50 000 PLN netto na miesiąc przy umowie B2B. Daje to szacunkowo 30 000 – 40 000 PLN na rękę miesięcznie<sup>17</sup>.

Warto pamiętać, że tak wysokie wynagrodzenia dotyczą jedynie niewielkiego odsetka uzdolnionych i niezwykle pracowitych programistów. Wiążą się one także z odpowiedzialnością, presją na wyniki, ambitnymi i złożonymi zadaniami, a także podróżami służbowymi. Najlepsze finansowo oferty zazwyczaj dotyczą zleceń dla klientów zagranicznych i konieczna jest w tym przypadku doskonała znajomość języków obcych (zazwyczaj angielskiego, ale zdarza się również niemiecki, francuski czy nawet japoński, a w przyszłości pewnie i chiński). Poza umiejętnościami technicznymi istotna jest również znajomość domeny, branży, umiejętności miękkie i marka osobista.

Jeśli chodzi o przeciętne wynagrodzenia, to zmieniają się one dynamicznie, a także różnią się znacząco w zależności od miasta, wybranej specjalizacji, a nawet stosu technologicznego.

---

<sup>16</sup> Na początku 2021 roku.

<sup>17</sup> W zależności oczywiście od konkretnego przypadku, wybranej formy opodatkowania itd.

W przybliżeniu można powiedzieć, że młodszy programista zarabia zazwyczaj poniżej 5000 PLN miesięcznie na rękę, jego bardziej doświadczony kolega, mid developer 5000 – 10 000 PLN, zaś starszy programista pomiędzy 10 000 a 15 000 PLN. W praktyce jednak trzeba obserwować rynek pracy w okolicy.

Zarobki w programowaniu są na tle innych profesji w Polsce bardzo jawne. Wiele osób rozmawia o wynagrodzeniu ze swoimi współpracownikami, rekruterzy bardzo często pytają o oczekiwaną stawkę już na początkowym etapie rekrutacji, a oferty pracy zwykle zawierają widełki płacowe.

Popularne portale z ofertami pracy dla programistów, na których możemy znaleźć oferowane wynagrodzenia, to:

- <https://justjoin.it/>,
- <https://nofluffjobs.com/pl>,
- <https://bulldogjob.pl/>,
- <https://it.pracuj.pl/>.

Sytuacja programistów na tle innych zawodów w Polsce jest specyficzna. Statystyki GUS i inne badania sytuują koderów wysoko w rankingu zawodów. Może się więc wydawać, że o wiele lepiej — z finansowego punktu widzenia — być programistą niż na przykład lekarzem, nie mówiąc już o hydraulikach czy elektrykach. Trzeba jednak wziąć pod uwagę kilka powszechnie ignorowanych kwestii...

Po pierwsze w przeciwieństwie do wielu profesji wynagrodzenia programistów są niezwykle jawne. Jest tajemnicą poliszynela, że nie za każdą usługę dostaniemy w Polsce paragon fiskalny, więc nie wszystkie zarobki są ujmowane w statystykach. W przypadku programistów — są. Praca odbywa się tu za-

zwyczaj nie dla indywidualnych klientów, ale dla dużych, międzynarodowych firm. Tworzy to iluzoryczny rozdział między wynagrodzeniami programistów i w innych zawodach.

Po drugie — i to o wiele istotniejsze — inżynierowie oprogramowania osiągają wysokie zarobki w bardzo krótkim czasie. To, rzecz jasna, dobrze. Tyle że w krajach takich jak USA czy Chiny ich kariery po czterdziestce częstokroć zaczynają dobiegać końca. To, jak będzie w Polsce, nie jest jeszcze pewne, ale ogólny trend jest jednak taki, że programiści w podeszłym wieku nie są tak szanowani jak na przykład doświadczeni lekarze (więcej na ten temat w sekcji „Długość kariery” w podrozdziale „Zalety i wady zawodu”). Rzecz trzecia to możliwość dodatkowego zarobku. Wielu medyków pracuje w licznych przychodniach i szpitalach, nauczyciele dorabiają, udzielając korepetycji, a taksówkarze potrzebujący dodatkowych pieniędzy mogą zwykle poświęcić na pracę weekendy i święta. Programiści mają pod tym względem nieco gorzej. Z uwagi na charakter tej pracy mało kto angażuje koderów do projektów informatycznych na  $\frac{1}{2}$  czy  $\frac{1}{4}$  etatu. Nie jest to niespotykane, ale jest o wiele rzadsze niż w przypadku wielu innych zawodów. Co więcej — ponieważ jest to praca wymagająca sprawności intelektualnej — część firm bardzo potępia lub wręcz próbuje zabraniać takich praktyk, walcząc o to, by ich programiści byli wypoczęci i skupieni na aktualnym projekcie.

Kolejnym istotnym tematem jest rozdział pomiędzy pozycją początkujących i doświadczonych programistów. Od kilku lat napływ dużej ilości osób zainteresowanych pracą w IT sprawił, że o jedno stanowisko młodszego programisty walczy nieraz kilkaset osób. Oznacza to oczywiście jedno — bardzo słabą pozycję negocjacyjną junior developerów. Co gorsza, brak rąk do

pracy połączony z dużą rotacją pracowników sprawia, że wielu pracodawców nie jest zainteresowanych szkoleniem początkujących programistów. Obawiają się, że po przyuczeniu pracownik taki zniknie, a jedyne, co po nim pozostanie, to koszt. Ponadto młodszemu programiście muszą poświęcić swój czas seniorzy, a przez to praca nad projektem zostanie spowolniona. Winduje to wynagrodzenia seniorów i podkopuje pozycję juniorów na rynku pracy. Warto o tym pamiętać w tym sensie, że rozpoczynając przygodę z IT, będziemy zapewne musieli spędzić kilka lat w swego rodzaju czyścicu, zanim otworzą się przed nami wrota niebieskie seniorskich zarobków i pasjonujących projektów.

Specyficzną dla naszego kraju i regionu cechą jest przewaga firm zagranicznych w ofertach pracy dla programistów. Napływ kapitału (głównie z Europy Zachodniej) zwiększył popyt na programistów i podniósł cenę ich pracy. Kwestią dyskusyjną jest, jak długo sytuacja ta potrwa, ale na pewno jest korzystna dla polskiej gospodarki w tym sensie, że eksportujemy nasz talent techniczny, otrzymując w zamian, mówiąc językiem PRL-u, dewizy — czyli zagraniczne waluty, a także przy okazji importujemy zachodnie praktyki zarządzania projektami, doświadczenie organizacyjne i nawiązujemy kontakty biznesowe z bogatszymi krajami wspólnoty europejskiej. Sukces Polski (oraz szerzej — Europy Środkowo-Wschodniej) wynika z wysokiego poziomu edukacji, bliskości kulturowej z krajami Zachodu, niewielkiej odległości geograficznej, niezłej znajomości angielskiego oraz oczywiście niższych wynagrodzeń. Na chwilę obecną sytuacja wygląda tak, że kluczowe projekty odbywają się w centrach korporacji, te mniej istotne w krajach takich jak Polska czy Rumunia, a te najbardziej kosztowo zoptymalizowane

— w Indiach. Ulotki dla inwestorów prezentują Europę Środkowo-Wschodnią jako kompromis między jakością i ceną. W kontekście zarobków warto o tym pamiętać.

Wynagrodzenia programistów bardzo szybko poszły w górę, ale ponieważ większość przedsiębiorstw stanowi kapitał zagraniczny, może on bardzo szybko zniknąć z Polski, o ile tylko znajdzie się gdzieś w przyszłości grono równie dobrych koderów skłonnych pracować za mniejsze stawki. Zdarzenie takie spowodowałoby bezprecedensowy spadek zarobków w branży. Oczywiście dyskusyjne jest, czy jakiś region świata rozwinie się na tyle, by konkurować z Europą Środkowo-Wschodnią, ale mimo wszystko warto o tym ryzyku pamiętać.

## DOCHODOWOŚĆ BRANŻY IT

Oprogramowanie zjada świat — napisał w swoim głośnym artykule o wpływie technologii informacyjnej na biznes Marc Andreessen, współautor pierwszej przeglądarki internetowej, który inwestował w firmy takie jak Facebook czy Skype, zanim odniosły sukces. Wyliczał on w nim kolejne branże, które zostały zrewolucjonizowane przez oprogramowanie. I nie sposób odmówić mu racji...

## RENTOWNOŚĆ

Przedsiębiorstwa, o których w ostatnich latach najgłośniej, te, których wyceny rynkowe rosną najbardziej, a usługi i nazwy w ciągu kilku lat od ich założenia docierają do większości krajów na Ziemi, to niemal wyłącznie firmy technologiczne. Innymi słowy, to biznesy oparte na oprogramowaniu. Wszyscy je znamy — Google, Facebook, Amazon, Spotify, Skype, Uber, Netflix i inne. Każda

z nich w zaskakujący sposób zdominowała pewien rynek i każda z nich jest firmą stricte informatyczną. Uber nie ma swoich taksówek, Amazon nie ma swoich sklepów, Skype nie ma swoich centrali telefonicznych. Za sukcesem — spektakularnym sukcesem — każdej z nich stoi zwykle grupa kilkudziesięciu lub kilkuset pierwszych programistów. Na przykład Spotify, założone w 2006 roku<sup>18</sup>, liczyło w 2011 roku zaledwie 311 pracowników<sup>19</sup>. Co prawda firmom tym dojście do rentowności zajmuje nieraz wiele lat (na przykład Amazon zaczął przynosić zyski w 2001 roku, a powstał w 1994), lecz ich wpływ na rynek jest tak duży, że kiedy już zaczynają wykazywać zyski, są one duże. Bardzo duże.

Za przykład może posłużyć przynoszący przez pierwsze lata straty Facebook, który w piątym roku zarobił prawie 7 miliardów dolarów<sup>20</sup>. W roku 2020 było to już 29,15 miliarda dolarów, przy zatrudnieniu rzędu zaledwie 52,5 tysiąca pracowników<sup>21</sup>. Oznacza to, że średnio jeden pracownik przyniósł firmie 555 238 USD, czyli o ponad dwa miliony złotych rocznie więcej, niż wydano na jego wynagrodzenie i pozostałe koszty działalności firmy! Warto przy tym zauważyć, że w 2019 roku PKB na osobę w USA wynosił około 65 tysięcy USD. Tymczasem w tym samym roku sieć supermarketów Dino osiągnęła 410,91 miliona PLN zysku netto<sup>22</sup>, zatrudniając

---

<sup>18</sup> <https://en.wikipedia.org/wiki/Spotify>.

<sup>19</sup> <https://www.statista.com/statistics/245130/number-of-spotify-employees/>.

<sup>20</sup> <https://www.businessinsider.com/tech-companies-worth-billions-unprofitable-tesla-uber-snap-2019-11?IR=T>.

<sup>21</sup> [https://en.wikipedia.org/wiki/Facebook,\\_Inc](https://en.wikipedia.org/wiki/Facebook,_Inc).

<sup>22</sup> <https://forsal.pl/artykuly/1459752,dino-polska-mialo-41091-mln-zl-zysku-netto-7254-mln-zl-ebitda-w-2019-r.html>.



17 372 osoby<sup>23</sup>. Zatem zysk Facebooka to 177 PLN na osobę miesięcznie, a w Dino — 1971 PLN. Niemal 90 razy więcej w firmie Marka Zuckerberga.

Żeby jeszcze bardziej uzmysłwić sobie skalę różnicy, dodajmy, że w Dino zarobki to zapewne około 5 tysięcy PLN brutto miesięcznie, a w Facebooku to jakieś 10 tysięcy USD na miesiąc. Przy wspomnianych zyskach Facebook mógłby podnieść wynagrodzenie do jakichś 55 tysięcy USD na miesiąc i wyjść na zero. Tymczasem w przypadku sieci marketów możliwe byłoby zaledwie około 7 tysięcy PLN.

Nie znaczy to w żadnym wypadku, że Dino jest źle zarządzane. Wręcz przeciwnie. Spółka z roku na rok notuje coraz wyższe zyski, a jej cena na giełdzie wzrosła w ciągu czterech lat sześciokrotnie. Porównanie to pokazuje jednak, że nowoczesny biznes oparty na oprogramowaniu może wygenerować oszałamiająco większe zyski niż przedsiębiorstwa prowadzące tradycyjną działalność.

## PRZYCZYNY

Przyczyny tak wysokiej rentowności przedsiębiorstw technologicznych wynikają głównie z trzech faktów:

- informatyzacja pozwala na automatyzację i redukcję zatrudnienia;
- technologia informacyjna umożliwia szybki rozwój biznesu na skalę globalną;
- oprogramowanie czyni możliwym rzeczy wcześniej niewykonalne.

---

<sup>23</sup> <https://www.wiadomoscihandlowe.pl/artikel/dino-zwiekszylo-przez-rok-zatrudnienie-o-niemal-3-7-tys-osob>.

Przykład do punktu pierwszego — swego czasu, by sporządzić raport sprzedaży w jakimkolwiek sklepie, trzeba było usiąść przed stosem kartek i wykonać dziesiątki, setki operacji matematycznych. Dziś wystarczy kilka kliknięć, by otrzymać wartości i wykresy, a to wszystko w kilka sekund. Jeśli w jakimś przedsiębiorstwie istniała osoba zajmująca się przygotowaniem raportów (będących powtarzalnymi operacjami przetwarzania danych), można ją zwolnić. Raz opracowane narzędzie do generowania raportów posłuży tysiącom przedsiębiorstw, zaś wykonanie i aktualizacja tych narzędzi zajmuje o wiele mniej czasu i wymaga mniej etatów, niż wynosiła liczba tych, które dzięki owemu narzędziu stały się zbędne.

Tego typu przykładów można znaleźć setki, jeśli nie tysiące. Automatyzacja od czasu rewolucji przemysłowej przynosiła spektakularne wyniki biznesowe (patrz podrozdział „Dlaczego programiści mają dużo pracy?”). W chwili obecnej większość tego typu usprawnień wymaga do działania oprogramowania.

Rozwój biznesu na globalną skalę możliwy był już od dawna. Nigdy jednak nie był równie szybki i tani. Infrastruktura informatyczna i sięgający na każdy kontynent (a już niedługo dzięki sieci satelitów Starlink w każde miejsce na Ziemi) internet umożliwił dostarczanie produktów cyfrowych dosłownie z prędkością światła. Koszt dotarcia na przykład Forda lub Toyoty na polski rynek był w porównaniu z wydatkami technologicznych gigantów ogromny. Firmy motoryzacyjne musiały zbudować sieć dystrybucji, wypromować swoje produkty, dostarczyć części zamienne do serwisów, nauczyć mechaników naprawiać ich auta. Kiedy Google czy Facebook wchodziły do Polski, musiały w zasadzie wyłącznie przetłumaczyć teksty na swoich stronach i zadbać o kwestie prawne. Tradycyjne biznesy

uważnie analizują dane rynkowe i decydują o wejściu na dany obszar gospodarczy. Firmy technologiczne docierają w zasadzie tam, gdzie pozwoli internet (i rządy).

Ostatnie, co decyduje o wielkich nadziejach pokładanych w oprogramowaniu to możliwości. Technologie informacyjne pozwalają na to, czego nie umożliwiło do tej pory nic innego. Kilka sprytnych algorytmów i nagle samochody zaczynają samodzielnie parkować, odtwarzacz muzyki podpowiada nam kolejne piosenki, a telefon nawiguje nas najlepszą trasą przez zakorkowane miasto. Przywykliśmy do tego i z wolna nie wydaje nam się to niczym szczególnym, ale tak nie jest. Są to fantastyczne usprawnienia. I to te mniej ambitne.

Wszystko, co czyni możliwym rzeczy wcześniej niewykonalne — o ile, rzecz jasna, są to rzeczy warte uwagi — generuje zyski i daje władzę. I to właśnie robi technologia informacyjna. Algorytmy uczenia maszynowego pozwalają wygrywać wybory — można dzięki nim analizować społeczeństwo i przedstawiać mu się jako polityk bardziej atrakcyjny. Jak pisze Pedro Domingos w *The Master Algorithm*, zarówno Demokraci, jak i Republikanie korzystają z oprogramowania do osiągnięcia lepszych wyników w kampaniach wyborczych. Wiele krajów pracuje obecnie nad bronią autonomiczną<sup>24</sup>. Tego typu urządzenia (drony, samoloty, statki) mają rozpoznać cel ataku (na przykład osobę ze zdjęć) i wyeliminować go bez wsparcia czy decyzji ludzi (tzn. nie wymagać potwierdzenia oddania strzału). Poza wojskowymi i politycznymi zastosowaniami spektakularne mogą być rozwiązania cywilne. Wystarczy wyobrazić sobie wpływ na rynek transportu, ale też na rynek pracy, firm, które opracują

---

<sup>24</sup> [https://en.wikipedia.org/wiki/Lethal\\_autonomous\\_weapon](https://en.wikipedia.org/wiki/Lethal_autonomous_weapon).

i wdrożą autonomiczne ciężarówki i taksówki. Nie sięgając w przód wyobraźnią, można natomiast zastanowić się, jak wyglądał rynek reklamy 20 lat temu i jak wygląda obecnie. Jaki w nim udział ma Google i Facebook...

## SKUTKI

Skutkiem możliwości oferowanych przez oprogramowanie jest to, że firmy technologiczne wpływają na coraz większą liczbę branż (np. reklamową, fotograficzną, muzyczną), a tradycyjne biznesy coraz więcej inwestują w IT. Zwiększa to popyt na programistów i podnosi presję płacową. Co jednak najważniejsze – po prostu się opłaca. Mądre, przemyślane inwestycje w informatyzację podnoszą wydajność, rentowność i jakość świadczonych usług. Rzecz dotyczy zresztą nie tylko biznesu. Również państwa zaczynają między sobą konkurować w tym względzie. Znana z polskiego podwórka możliwość składania deklaracji podatkowych przez internet, e-recepta i inne usługi tego typu podnoszą jakość życia obywateli. Kto z nas chciałby jeszcze jeździć do banku, by wykonać przelew, albo do urzędu, by złożyć PIT?

Informatyzacja to wydajność, a wydajność to rentowność. Nic dziś nie dostarcza lepszych narzędzi do ich podnoszenia niż oprogramowanie. Dlatego dochodowość branży IT jest wysoka, a zarobki i popyt na programistów rosną.

# PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA  
**Helion** 

# KARIERA KODERA

Na listach najlepiej opłacanych zawodów koder, developer i programista zawsze plasują się wysoko. Specjaliści do spraw administrowania sieciami komputerowymi, frontend i backend developerzy, analitycy systemów i architekci oprogramowania są stale poszukiwani. A ponieważ głodny nowoczesnych technologii rynek tak bardzo ich pożąda, oferuje programistom warunki płacy i pracy, o jakich przedstawiciele wielu innych zawodów mogą tylko pomarzyć. Zarobki grubo powyżej średniej krajowej? Tak! Praca zdalna z dowolnego miejsca na świecie? Bardzo często! Duża samodzielność? Jasne! W dodatku, jeśli wolisz, jako programista możesz założyć firmę i działać w stu procentach na własnych zasadach.

Brzmi jak bajka?

Oczywiście. I jest to bajka, której bohaterem możesz się stać, jeśli ukończysz odpowiednie studia i nabierzesz wprawy jako stażysta, a potem junior w firmach z branży IT. Nim jednak zdecydujesz o wyborze zawodu programisty, upewnij się, że właśnie TO chcesz robić w życiu:

- **Przekonaj się, na czym tak naprawdę polega programowanie**
- **Poznaj ramowy plan dnia programisty**
- **Zorientuj się, z kim na co dzień współpracuje developer**
- **Dowiedz się, w jaki sposób organizuje się projekty w branży IT**
- **Zajrzyj w przyszłość i sprawdź, co czeka developerów w nadchodzących latach**

**Arkadiusz Kałkus** — programista i konsultant z wieloletnim doświadczeniem. Pracował dla banków, dla polskich i międzynarodowych organizacji, w tym dla skandynawskiego sektora publicznego i francuskiej firmy farmaceutycznej. Jest autorem bloga *Programista też człowiek* ([kalkus.dev/pl/](http://kalkus.dev/pl/)).

**Helion** 



helion.pl



HELION SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel.: 32 230 98 63  
helion@helion.pl

Sprawdź nasze szkolenia!



AKADEMIA IT & BUSINESS

HELIONSZKOLENIA.PL

**KOD KORZYŚCI**

Sięgnij po więcej! ▶



ISBN 978-83-283-7375-4



9 788328 373754

INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 69,00 zł