

## Wartość automatyzacji testów

---

Ponieważ tematem tej książki jest automatyzacja testów, powinniśmy w zasadzie zacząć od jej definicji. Jednak bez nakreślenia właściwego kontekstu definicja taka może nie być wystarczająco przejrzysta i może bardziej prowadzić do dezorientacji niż zrozumienia. Jest to na tyle szeroki i zróżnicowany temat, że trudno jest tu podać taką definicję, która będzie jednocześnie dokładna, przejrzysta i obejmie wszystkie istniejące rodzaje automatyzacji testów. Gdybym jednak miał teraz przytoczyć jakąś definicję, mogłaby ona wyglądać tak: „Używanie oprogramowania w celu ułatwienia testowania innego oprogramowania”, ale nie jestem do końca pewien, na ile jest ona przydatna. Dlatego też zamiast skupiać się na formalnych definicjach, w pierwszej części książki szczegółowo analizuję ten obszerny temat, starając się w ten sposób wyjaśnić, czym tak naprawdę jest automatyzacja testów, a także – co równie istotne – czym ona nie jest!

### Dlaczego potrzebujemy automatyzacji testów?

Gdy pytam moich klientów, czego spodziewają się uzyskać dzięki automatyzacji testów, najczęściej udzielaną odpowiedzią jest skrócenie czasu potrzebnego na przetestowanie oprogramowania przed jego wydaniem. Z jednej strony, choć jest to niewątpliwie ważny cel, to w zakresie korzyści, jakie możemy uzyskać dzięki automatyzacji testów, stanowi on jedynie wierzchołek góry lodowej. Ponadto osiągnięcie celu w postaci skrócenia cykli testów manualnych zajmuje zwykle sporą ilość czasu. Z drugiej strony, dużo wcześniej możemy zacząć zauważać pozostałe korzyści. Ale najpierw zobaczmy, dlaczego ten prosty cel w postaci skrócenia czasu trwania cyklu testowego stał się w ostatnich latach tak istotny.

## Od modelu kaskadowego do zwinnego tworzenia oprogramowania

Mimo że niektóre organizacje korzystają z automatyzacji testów już od dekad, jednak zaczęła być ona powszechnie stosowana dopiero w ostatnich latach. Jest wiele powodów, dla których tak się stało, ale bez wątplenia można powiedzieć, że wzrost zapotrzebowania na automatyzację testów zawdzięczamy w dużej mierze odejściu od tradycyjnego modelu kaskadowego (*waterfall*) na rzecz programowania zwinnego (*Agile software development*). W tradycyjnym podejściu kaskadowym projekty oprogramowania postrzegane były jako coś jednorazowego, podobnie jak budowanie mostu. Najpierw planujemy i projektujemy oprogramowanie, potem je budujemy, a na końcu testujemy i sprawdzamy jakość końcowego produktu, naprawiając przy tym pomniejsze błędy, które znaleźliśmy. Opieramy się tu na założeniu, że jeśli fazy planowania i budowy zostały przeprowadzone poprawnie, to poza pewnymi drobnymi pomyłkami programistycznymi, które możemy bardzo łatwo naprawić, wszystko powinno działać zgodnie z planem. Takie podejście sprawia, że proces weryfikowania, czy rezultat końcowy zachowuje się zgodnie ze specyfikacją, musimy przeprowadzić tylko raz. Ponowne wykonanie testu powinno mieć miejsce tylko w przypadku wykrycia jakiegoś błędu i przygotowania dla niego odpowiedniej poprawki, a następnie jej sprawdzenia. Jeśli każdy test wykonywany jest tylko raz lub dwa razy, to w wielu przypadkach znacznie taniej i łatwiej będzie wykonywać je ręcznie niż je automatyzować.

Po latach stało się jasne, że w większości przypadków podejście kaskadowe nie spełnia swoich obietnic. Większość projektów oprogramowania była już na tyle skomplikowana, że zaplanowanie i domknięcie wszystkich technicznych szczegółów w początkowej fazie tworzenia było niemożliwe. Nawet w tych przypadkach, w których było to wykonalne, do czasu ukończenia takiego projektu (trwającego zwykle kilka lat) zmieniały się zarówno sama technologia, jak i potrzeby biznesowe, czyniąc takie oprogramowanie mniej adekwatnym niż miało być początkowo. Z tych właśnie powodów szybkie reagowanie na opinie klientów stało się cenniejsze od sztywnego trzymania się początkowego planu. Wraz z upływem czasu większość przemysłu oprogramowania odeszła od tych jednorazowych projektów, rezygnując z cyklicznego wydawania nowych wersji tego samego oprogramowania co kilka lat na rzecz szybkich cykli wydawniczych. Dzisiaj niektóre z największych firm działających w sieci Web dostarczają nowe funkcje i poprawki dla swojego oprogramowania po kilka razy dziennie, a niekiedy nawet kilka razy na minutę!

### MANIFEST PROGRAMOWANIA ZWINNEGO

W 2001 roku 17 liderów z obszaru rozwoju oprogramowania sformułowało **Manifest programowania zwinnego**, którego treść jest następująca<sup>1</sup>:

Odkrywamy nowe metody programowania dzięki praktyce w programowaniu i wspieraniu w nim innych. W wyniku naszej pracy zaczęliśmy bardziej cenić:

**Ludzi i interakcje** od procesów i narzędzi

**Działające oprogramowanie** od szczegółowej dokumentacji

---

<sup>1</sup> Tłumaczenie treści manifestu pochodzi ze strony: <http://agilemanifesto.org/iso/pl/manifesto.html> (przyp. tłum.)