

Narzędzia statystycznej kontroli procesów – część 2

 intermediate

Autor: *Piotr Piotrowski*

O Autorze:

Pracownik wiedzy, Inżynier, Badacz, Działacz.



Inżynier Testów w dziale Industrial R&D przedsiębiorstwa Tieto Poland, gdzie zajmuje się m. in. czynnościami testowymi w przemyśle motoryzacyjnym.

Działacz w organizacjach non-profit od regionalnych do międzynarodowych w zakresie edukacji oraz nauki i technologii.

Posiada kilkuletnie doświadczenie w branżach: telekomunikacja, badania i rozwój, informatyka oraz liczne certyfikaty z zarządzania i języka angielskiego.

Ukończył edukację w obszarze zarządzania na niezależnym opanowaniu treści programowych Doctor of Business Administration, natomiast w obszarze techniki na trzecim roku doktoratu z telekomunikacji na Politechnice Warszawskiej.

Autor ponad 20-tu publikacji oraz dwóch zgłoszeń patentowych.

Streszczenie

Artykuł stanowi rozwinięcie pracy „Wprowadzenie do statystycznej kontroli procesów testowych” [Piot12], która pojawiła się w ósmym numerze polskiej wersji czasopisma c0re, w zakresie narzędzi statystycznej kontroli procesów. W publikacji zostaną przedstawione popularne techniki statystyczne, opisane przykładowo w [Lewi09], takie jak: wykres Pareto, wykres rozrzutu, diagram przyczyn i skutków oraz schemat blokowy.

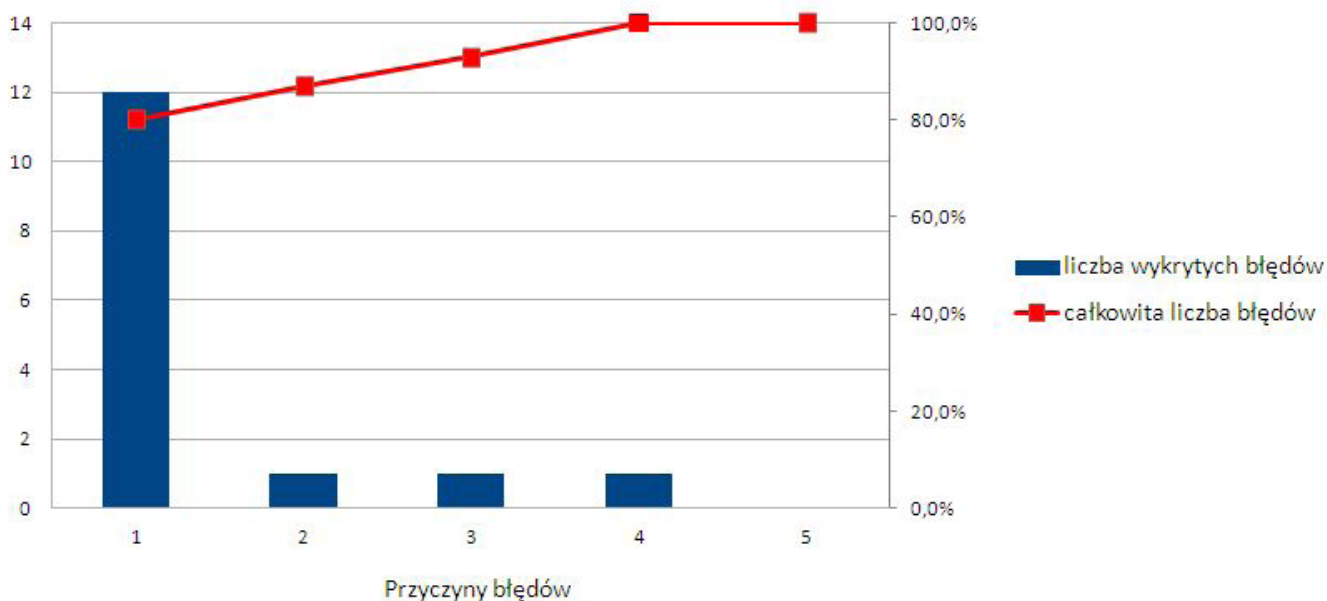
Wykresy Pareto

W poprzednim artykule dotyczącym zastosowania statystyki w kontrolowaniu procesów zaprezentowano wykresy kontrolne, wykresy przebiegu w czasie oraz histogramy.

Kolejnym narzędziem statystycznym są wykresy Pareto (ang. Pareto charts), które składają się ze słupków oraz krzywej. Słupki są coraz niższe przy przechodzeniu od lewej do prawej strony wykresu, natomiast krzywa reprezentuje sumę wszystkich wartości. W dziedzinie testowania oprogramowania słupki mogą określać liczby znalezionych błędów dla poszczególnych przyczyn, z kolei krzywa – całkowitą liczbę błędów. Rys. 1 pokazuje także, że 20% procent przyczyn – jeden słupek spośród 5-ciu (gdzie 5-ty słupek ma wartość zero) - wywołało dwanaście spośród wszystkich piętnastu błędów, czyli 80%. Jest to odzwierciedlenie zasady Pareto, która mówi, że około 20% przyczyn wywołuje około 80% skutków. Jednocześnie pierwszy, najwyższy słupek reprezentuje najważniejszą przyczynę, w tym przypadku przyczynę pojawiania się błędów w oprogramowa-



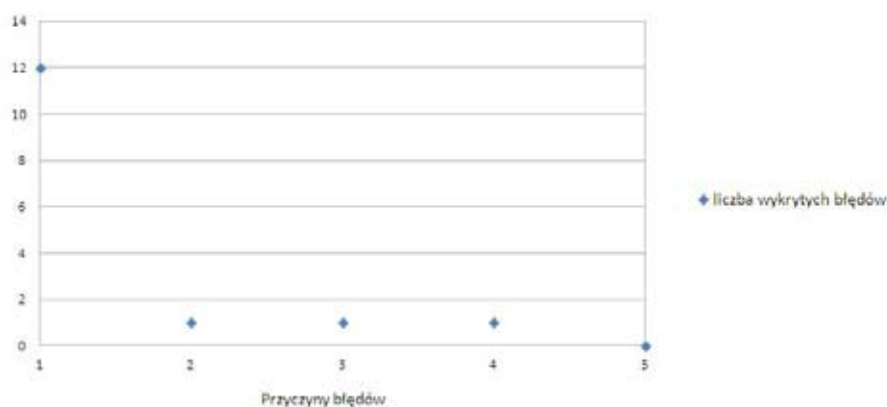
Wykres Pareto



Rysunek 1 Wykres Pareto – opracowanie własne

niu. Zasada Pareto jest pewnym przybliżeniem – stosunek przyczyna-skutek nie musi wynosić dokładnie 20 do 80. Ponadto może ona nie być zachowana w konkretnych przypadkach. Podsumowania zastosowania wykresów Pareto w testowaniu oprogramowania dokonano przykładowo w [TtR11] wskazując na wykorzystanie wymienionych w artykule technik statystycznych do raportowania testów. Odpowiednio zilustrowana analiza Pareto może być pomocna w analizie przyczyn błędów, czy wyznaczaniu priorytetów w pracy testera.

Wykres rozrzutu



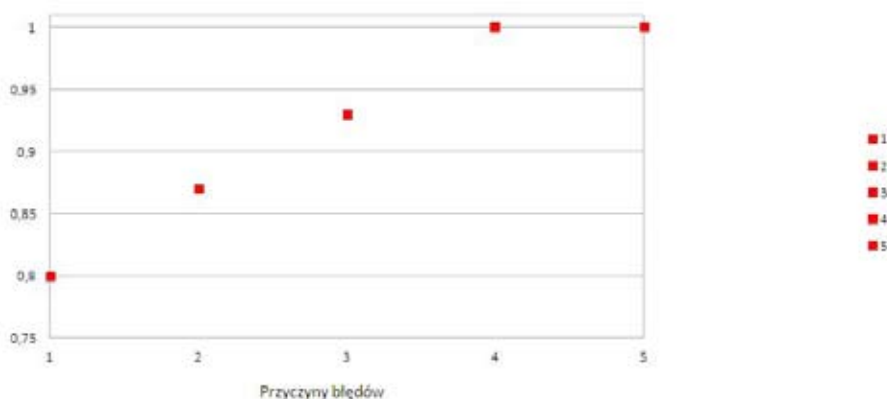
Rysunek 2 Wykres rozrzutu z ujemną zależnością między zmiennymi – opracowanie własne

Wykresy rozrzutu (punktowe)

Wykresy rozrzutu (ang. scatter plots) znane są również pod nazwą wykresów punktowych, z uwagi na to, że przedstawiają punkty, których pozycja na wykresie zależy od wartości dwóch zmiennych – jedna zmienna określa położenie punktu względem osi X, druga położenie punktu w stosunku do osi Y. Innymi słowami, wykres rozrzutu pokazuje zależności pomiędzy dwoma zmiennymi.

Podobnie do analizy Pareto, wykres ten wskazuje, w jaki sposób określona przyczyna wpływa na skutek. Korelacja jest pozytywna, jeśli wraz ze wzrostem wartości x, wzrasta wartość y, a negatywna w przeciwnym przypadku. Gdy punkty wykresu nie skupiają się w pewnym obszarze wykresu, to zależności nie ma lub jest ona względnie słaba.

Wykres Rozrzutu



Rysunek 3 Wykres rozrzutu z dodatnią zależnością między zmiennymi – opracowanie własne

Rys. 2 pokazuje negatywną, początkowo silną zależność, natomiast Rys. 3 dodatnią korelację, która na końcu słabnie. Przy okazji warto wspomnieć, iż zyski z wytworzenia oprogramowania są zależne od kosztów znalezienia i usunięcia błędów, o których mogą świadczyć takie zmienne, jak na przykład liczba znalezionych błędów i waga poszczególnych przyczyn wystąpienia błędów.

Przykład użycia wykresu rozrzutu w inżynierii oprogramowania jest zawarty w [Aina09], gdzie zamieszczono, wyrażoną tego typu wykresem, analizę zależności pomiędzy złożonością programu, określoną przez indeks złożoności McCabe'a a liczbą defektów. Jednak dla każdego z opisywanych w artykule narzędzi statystycznych można podać zapewne wiele zastosowań na polu testowania, czy całej inżynierii oprogramowania, toteż autorzy zajmujący się tematyką narzędzi statystycznych skupiają się głównie na poprawnym zrozumieniu danej techniki.



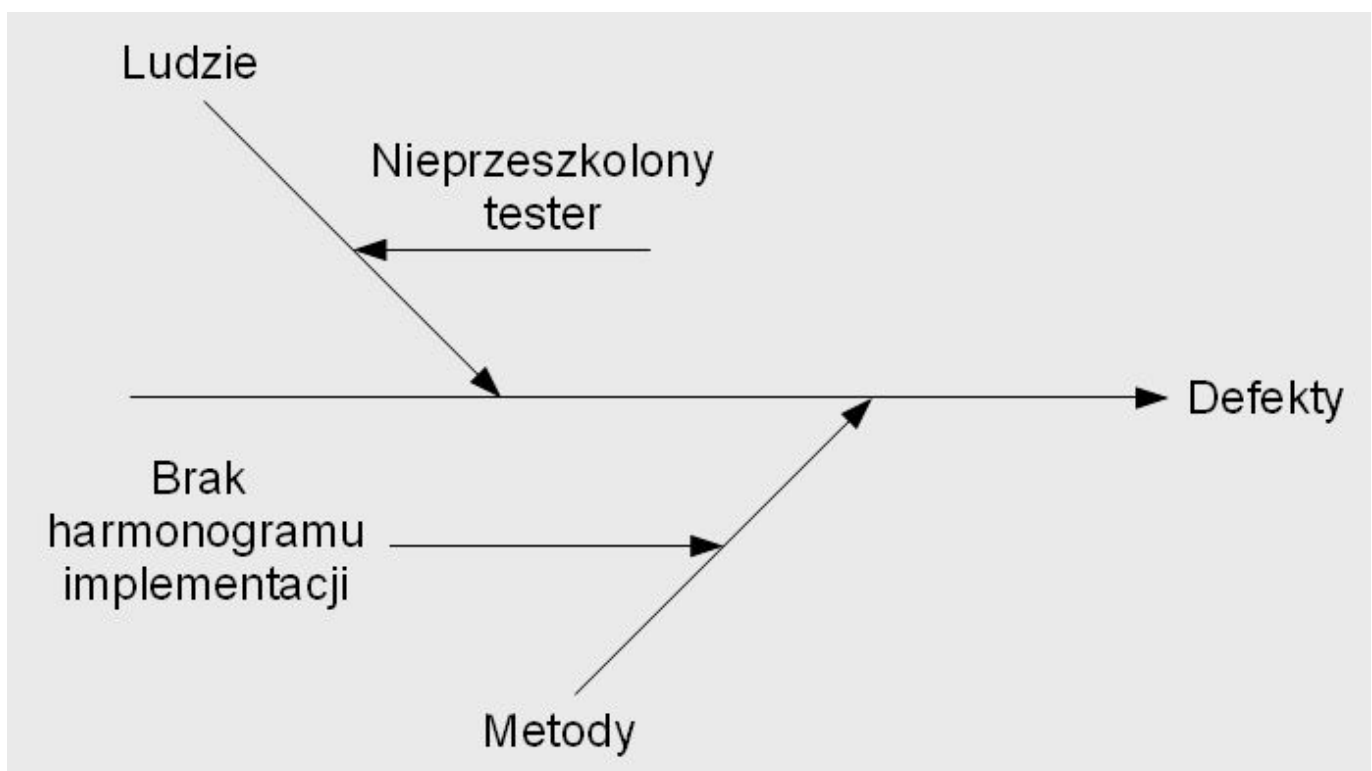
Diagramy przyczyn i skutków (Ishikawy)

Diagram przyczyn i skutków (ang. cause and effect diagram), zwany od nazwiska twórcy diagramem Ishikawy, służy również do powiązania przyczyn problemów z ich skutkami oraz do nadania priorytetów przyczynom oraz zbadania związków między przyczynami.

Przyczyny dzieli się na główne składowe, którymi są często: ludzie, metody, pomiary, materiały, sprzęt i środowisko, a także zarządzanie czy pielęgnacja.

Na diagramie przyczyn i skutków (Rys. 4), skutek reprezentowany jest przez linię poziomą, natomiast kategorie przyczyn – linie skierowane ukośnie do tejże głównej, poziomej linii. Poszczególne przyczyny dołączane są natomiast do

kategorii przyczyn w formie krótszych linii poziomych. Do określonych przyczyn można dołączać kolejne pod-przyczyny, czyli rozbudowywać diagram przyczyn i skutków. Z drugiej strony czasami potrzebna jest mniej złożona analiza lub jej część, toteż wprowadza się modyfikację lub fragment diagramu przyczyn i skutków. Prostsza wersja narzędzia pokazano na Rys. 4.



Rysunek 4 Diagram przyczyn i skutków – opracowanie własne

Schematy blokowe (organizacyjne)

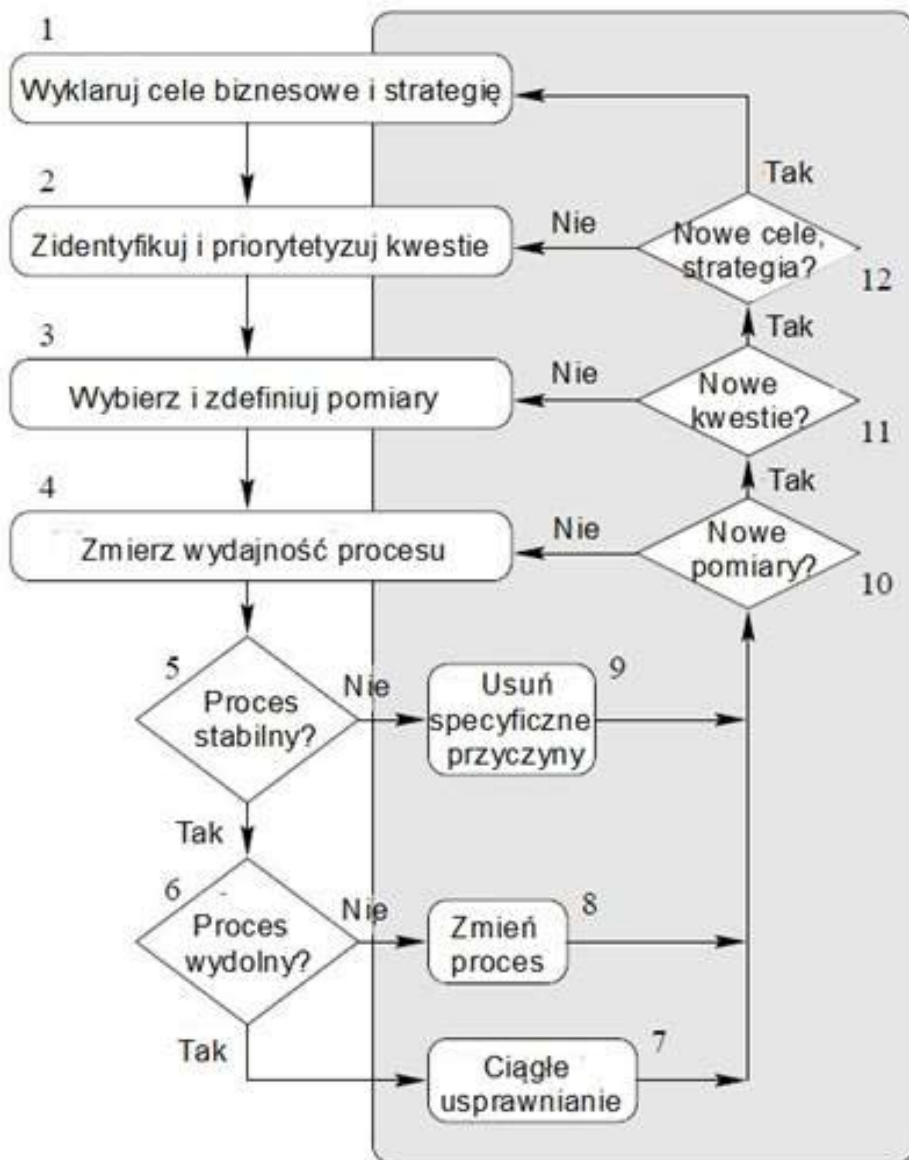
Schemat blokowy (ang. flow chart) jest popularnym narzędziem wyrażającym graficznie rozwiązanie jakiegoś problemu. Jest często wykorzystywane w algorytmice lub zarządzaniu procesami, może na przykład przedstawiać pracę programu komputerowego, czy działanie jakiegoś procesu związanego choćby z fazą testowania oprogramowania.

Rys. 5, pochodzący z [CDM03], jest właśnie przykładem schematu blokowego, który wyjaśnia kroki zarządzania procesami uwzględniające statystyczną kontrolę procesów. Innym popularnym zastosowaniem schematów blokowych może być modelowanie faz rozwijania oprogramowania. Innymi zastosowaniami tego i pozostałych narzędzi statystycznych opisanych w niniejszym artykule jest bardzo wiele.

Podsumowanie

W artykule opisano podstawy niektórych narzędzi statystycznych stosowanych w kontroli procesów. Użycie elementów statystyki w zarządzaniu procesami testowymi zostanie szerzej objaśnione także w obszarze procesu 5.2 poświęconym kontroli jakości, wchodzącym w skład najwyższego poziomu dojrzałości modelowania procesów testowych według TMMi.

Podstawowe informacje o modelu TMMi (w zakresie wybranych elementów należących do niższych poziomów dojrzałości) zostały przedstawione przez Autora na łamach magazynu c0re. ■



Rysunek 5 Schemat blokowy z ([Piot12] za) [CDM03] przedstawiający algorytm zarządzania procesami z użyciem metod statystycznych

Literatura

[Aina09] Ainapure B.S.: Software testing and quality assurance. Technical Publications 2009.

[CDM03] Cangussu J. W., DeCarlo R.A., Mathur A. P.: Monitoring the Software Test Process Using Statistical Process Control: A Logarithmic Approach. ACM 2003.

[Piot12] Piotrowski P.: Wprowadzenie do statystycznej kontroli procesów testowych. Magazyn c0re, nr 8 styczeń 2012.

[Ttr11] Tools to Report your Testing Activity Part – 1 <http://www.softwaretestingportal.com/reporting-tools-software-testing/> w Software Testing Portal. 18 listopad 2011.

[Lewi09] Lewis W. E.: Chapter 5. Quality through Continuous Improvement Process w Software Testing and Continuous Quality Improvement. Third Edition by CRC Press, Taylor & Francis Group, LLC 2009.