

Nie stać Cię... aby nie testować

Testowanie oprogramowania pisanego w językach C/C++ od zawsze narażało wiele problemów. Problemy te szczególnie eksponowane są w przypadku systemów wbudowanych (embedded systems), które zwykle mają znacznie okrojone możliwości diagnostyczne. W większości przypadków jakość procesu testowania i co za tym idzie – całego kodu – w dużej mierze zależy od kwalifikacji osób odpowiedzialnych za finalną jakość produktu. Taki stan rzeczy pociąga za sobą wiele negatywnych konsekwencji, począwszy od niskiej powtarzalności, poprzez dużą wrażliwość na zmiany w zespole, a kończąc na wysokich kosztach procesu. W takiej sytuacji dobrym rozwiązaniem jest zastosowanie zautomatyzowanych narzędzi do testowania kodu C/C++.

Nowoczesne narzędzia do analizy statycznej kodu, czy do wykonywania testów jednostkowych, pozwalają na formalizację trudnych i złożonych zagadnień technicznych, przez co ogromnie obniżają koszty procesu zapewnienia jakości. Czynią go również bardziej wiarygodnym, dzięki eliminacji czynnika ludzkiego ze zmuśnych i czasochłonnych czynności. Dobrym przykładem mogą tu być typowe dla systemów wbudowanych „pułapki” przy operacjach bitowych, konwersji typów danych, czy też działaniach arytmetycznych. Problemy te są trudne do wychwycenia przez programistę, zwłaszcza niedoświadczonego, a bardzo łatwe do zlokalizowania przez analizator statyczny.

Wieleletnie doświadczenia zebrane przez firmę Parasoft przy wdrażaniu systemów kontroli jakości procesu tworzenia oprogramowania dla systemów wbudowanych wskazują, że szczególnie dobre rezultaty uzyskiwane są przy jednoczesnym zastosowaniu wielu technologii kontroli jakości. I tak:

Analiza statyczna. Pozwala na wyeliminowanie niebezpiecznego kodu, jak i zapewnienie jego zgodności ze standardem. Szczególnie pomocna

The screenshot displays the Parasoft C++Test environment. The main window shows C++ source code for a sorting algorithm. A 'Coverage' window in the foreground provides a detailed report on test results:

Component	Coverage	Executable Lines
Sorts	75%	46/61
main	38%	9/24
check_order	100%	5/5
insert_sort	100%	12/12
qs_string_compare	100%	1/1
randomise	100%	6/6
shell_sort	100%	13/13

The 'Test Case Explorer' at the bottom shows a list of test cases, with 'test_insert_sort_10' selected. The 'Run' window indicates that 28/42 tests succeeded.

jest analiza statyczna z kontrolą przepływu danych w programie, która pozwala zlokalizować ewidentne błędy, takie jak: dzielenie przez zero, użycie wyzerowanego wskaźnika, niezwolnienie przydzielonych zasobów.

Testy jednostkowe. Radykalnie wpływają na jakość kodu poprzez zapewnienie zgodności implementacji ze specyfikacją. Zaobserwowano również, że spójne stosowanie testów jednostkowych znacząco obniża koszty utrzymania produktu wprowadzonego na rynek. Nie bez znaczenia jest też możliwość analizy pokrycia kodu uzyskanego z testów jednostkowych. Metryka ta pozwala zweryfikować jakość testów jednostkowych i jest coraz częściej wymagana dla celów certyfikacyjnych, jak i bywa również wymogiem narzuconym przez kontrakt biznesowy. Na rynku są narzędzia, które pozwalają na wykonywanie testów jednostkowych, bezpośrednio na urządzeniu docelowym, pozwalając tym samym podnieść wiarygodność testów poprzez ominięcie warstwy symulacyjnej. Przykładem takiego narzędzia może być C++test firmy Parasoft.

Inspekcje Kodu. W wielu przypadkach są one jedyną drogą do zlokalizowania problemów algorytmicznych. Nie do pominięcia jest też aspekt wzajemnego podnoszenia kwalifikacji przez programistów i poprawa ogólnej znajomości kodu.

Podsumowując warto zaznaczyć, że opisane techniki powinny być implementowane w procesach tworzenia oprogramowania jako praktyki wymuszane w sposób automatyczny, a uzyskiwane wyniki powinny być monitorowane w czasie (trendy rezultatów). Obserwowanie trendu w rezultatach testów, np. ilości naruszeń standardów kodowania zgłoszonych przez analizator statyczny, pozwala na reakcję ze strony osoby zarządzającej projektem stosowną do sytuacji, co jest kluczowe przy stabilizacji kodu źródłowego produktu. Poprzez analogię do teorii automatyki można powiedzieć, że trendy rezultatów dają możliwość zachowania się jak regulator „PID”, podczas gdy pojedyncze raporty z sesji testowej zmuszają nas do regulacji typu „P”. Konsekwencje są oczywiste. Otrzymujemy albo stabilny proces kontroli jakości, albo szereg inicjatyw podejmowanych ad-hoc, które w dłuższej perspektywie czasowej nie prowadzą do niczego.

Miroslaw Zieliński

Dodatkowe informacje

Parasoft S.A.
ul. Kielkowskiego 9, 30-703 Kraków
Tel.: 012 290 91 01, fax: 012 290 91 02
e-mail: info-pl@parasoft.com