



**Design patterns and code smells.  
Relationships and impact on selected  
software quality metrics**

**Tarek Alkhaeir**

Faculty of Computing and Telecommunications  
Poznan University Of Technology

Supervisor

**Andrzej Jaskiewicz, Ph.D., Dr Habil, Assoc. Prof.**

Supporting supervisor

**Bartosz Walter, Ph.D.**

Poznan, 2021

## **Abstract**

Wzorce projektowe stanowią rekomendowane, ogólne rozwiązania powtarzalnych problemów związanych z projektowaniem oprogramowania o dobrze zbadanych konsekwencjach ich użycia. Stanowią one podsumowanie i uogólnienie wiedzy i doświadczenia wynikającego z zastosowania określonego rozwiązania. Z tego powodu wzorce projektowe nadal cieszą się dużą popularnością wśród programistów, a sama koncepcja ta doczekała się także wielu adaptacji w innych obszarach inżynierii oprogramowania.

Ze względu na swoją popularność i znaczenie dla produkcji oprogramowania wzorce projektowe stały się również przedmiotem zainteresowania badaczy, szczególnie pod kątem związku użycia wzorców w programach z różnymi właściwościami kodu źródłowego, np. czytelnością, pielęgnowalnością, częstością zmian czy gęstością efektów. Jednak publikowane wyniki badań nie są jednoznacznie: wprawdzie wiele prac wskazuje na pozytywną korelację obecności wzorców z wymienionymi charakterystykami programów, ale niektóre publikacje prowadzą do odmiennych wniosków. Jednym z możliwych wyjaśnień tego stanu rzeczy są czynniki kontekstowe, które w interakcji z wzorcami wpływały na badane wartości badanych własności.

W pracy zbadano znaczenie tzw. przykrych zapachów w kodzie programów (ang. code smells) jako czynnika kontekstowego w analizie wzorców projektowych. Przykre zapachy to wysokopoziomowe symptomy błędów projektowych, które mogą negatywnie wpływać na koszt pielęgnacji oprogramowania. Symptomy te są stosunkowo łatwe do wykrycia, jednak ich interpretacja i faktyczny związek z przyczyną problemu nie są jednoznaczne. Istnieje wiele różnych rodzajów przykrych zapachów, wskazujących na typowe anomalie związane z błędami projektowymi, architektonicznymi czy implementacyjnymi. Na przykład, klasy o zbyt dużym rozmiarze

wykazują często negatywne właściwości, np. są zmieniane częściej, z różnych powodów, a w konsekwencji charakteryzują się większą liczbą defektów. Istnieje jednak wiele możliwych przyczyn takiego stanu, w związku z czym ich identyfikacja oraz usunięcie tych przyczyn nastręcza dodatkowych problemów. Jednak z uwagi na popularność przykrych zapachów i ich intuicyjną interpretację, są one często wykorzystywane w praktyce podczas produkcji oprogramowania. Dlatego wybór przykrych zapachów jako potencjalnego czynnika kontekstowego wpływającego na efektywność stosowania określonych wzorców jest intuicyjnie zrozumiały: przykry zapach wskazuje na błąd w projekcie lub implementacji danego rozwiązania, a zatem może powodować, że wzorzec wykazujący takie symptomy również będzie obciążony wadą.

Przedmiotem pracy jest empiryczne zbadanie związku pomiędzy wzorcami projektowymi oraz przykrymi zapachami i ich wpływu na dwie istotne charakterystyki kodu programu: zmienność i gęstość defektów. Badania przeprowadzono na kilku średniej wielkości, dojrzałych systemach napisanych w języku Java, za pomocą nieparametrycznych testów statystycznych.

Uzyskane wyniki wskazują, że klasy uczestniczące we badanych wzorcach projektowych są obciążone przykrymi zapachami niż pozostałe klasy, jednak związek ten zależy od konkretnego wzorca. Co ważne, w toku ewolucji programu udział przykrych zapachów w klasach pełniących role we wzorcach nieznacznie, ale zauważalnie spada.

Ponadto, obecność przykrych zapachów jest czynnikiem wpływającym na zmienność kodu i gęstość defektów w we wzorcach projektowych. Zmiany w klasach obciążonych przykrymi zapachami są zmieniane częściej, niż pozostałe klasy, natomiast zmiany te charakteryzują się mniejszym rozmiarem. W odniesieniu do defektów, klasy uczestniczące we wzorcach projektowych częściej wykazują defekty, a liczba tych defektów jest większa niż w przypadku klas nieuczestniczące we wzorcach, jak i klas uczestniczące we wzorcach, ale pozbawionych zapachów.

Wyniki wskazują, że obecność przykrych zapachów jest istotnym czynnikiem kontekstowym przy stosowaniu wzorców projektowych w kodzie programu, który wyjaśnia zróżnicowanie wyników badań dotyczących tych wzorców. Uwzględnienie tego czynnika pozwala znacznie lepiej określić wartości wybranych charakterystyk kodu programu, co ma duże znaczenie dla praktyki inżynierii oprogramowania.