



KOMPUTEROWE PRZETWARZANIE WIEDZY

**Kolekcja prac 2012/2013
pod redakcją Tomasza Kubika**



SPIS TREŚCI

1	Audyt w systemach informatycznych	3
1.1.	Sposoby pomiaru jakości pracy programisty	3
1.2.	Ocena jakości wygenerowanego kodu	4
1.3.	Dostępne systemy pomiaru pracy	4
1.4.	Proponowane rozwiązanie	4
	Bibliografia	4

AUDYT W SYSTEMACH INFORMATYCZNYCH

M. Nowak, G. Maj

Jednym z podstawowych problemów profesjonalnego zarządzania firmą w branży informatycznej jest zarządzanie jakością. Jakość można definiować i mierzyć na wielu poziomach. Bardzo dużo uwagi poświęcane jest jakości produktów, choć równie ważnym elementem jest ocena jakości pracy pracowników oraz ich efektywności. Audyt w systemach informatycznych powinien zawierać również elementy oceny jakości pracy i służyć usprawnieniu działania organizacji.

W rozważaniach na temat oceny jakości pracy i wydajności pracowników skupiono się na firmach informatycznych, których przedmiotem działalności jest wytwarzanie oprogramowania. Kluczowym pracownikiem biorącym udział w tworzeniu końcowego produktu, czyli programu jest programista. Ocena pracy programisty jest zadaniem złożonym i wymaga znajomości wielu aspektów procesu wytwarzania oprogramowania. W dzisiejszych czasach zarządzanie jakością oprogramowania jest prawdziwym wyzwaniem dla dużych firm o zasięgu globalnym zatrudniających tysiące programistów z całego świata.

Podstawowym pytaniem, które postawiono w tym rozdziale jest: jak efektywnie mierzyć jakość i efektywność pracy programisty? W niniejszym artykule autorzy postarają się przybliżyć problem, przedstawić dostępne na rynku rozwiązania oraz opisać prosty system kontroli efektywności pracy.

1.1. Sposoby pomiaru jakości pracy programisty

Problem oceny jakości pracy programisty nie jest problemem łatwym. Dostyc prosto można zaproponować podstawowe kryteria jednak nie zawsze oddają one faktyczny nakład pracy.

- czas pracy pracownika,
- ilość rozwiązanych zadań,
- efektywność mierzona stosunkiem predyktywności do wykonanych zadań,
- ocena wygenerowanego kodu:
 - ilość wygenerowanego kodu,
 - rozkład kodu w plikach,

1. Audyt w systemach informatycznych

- przejrzystość kodu,
- jakość testów,
- jakość dokumentacji,
- nazewnictwo funkcji i klas,
- analiza statyczna kodu,

1.2. Ocena efektywności pracy

1.3. Ocena jakości wygenerowanego kodu

1.4. Dostępne systemy pomiaru pracy

1.4.1. Systemy kontroli wersji

Przykładami systemów kontroli wersji są najbardziej popularne darmowe systemy takie jak:

- Git,
- Svn,
- Mercurial.

Dają one możliwość śledzenia postępu prac poczynionych przez poszczególnych pracowników.

1.4.2. Systemy zarządzania projektem

- Redmine,
- Jira,
- GitHub.

1.4.3. Systemy do statycznej analizy kodu

- Klocwork (C++),
- Pylint (Python),
- CodeSonar (Java).

1.5. Proponowane rozwiązanie

Literatura

- [1] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter. *Podstawy biologii komórki. Wprowadzenie do biologii molekularnej*. Wydawnictwo Naukowe PWN, 1999.