



Warsztat Inżyniera Oprogramowania (bardzo mały wstęp)

Sztuka Wytwarzania Oprogramowania, w. 2

Konrad Grochowski

Instytut Informatyki, Politechnika Warszawska, 2023 ©





Sztuka Wytwarzania Oprogramowania

- › *Software craftsmanship*
- › Osobiście wolę tłumaczenie „rzemiosło”
- › Talent + *warsztat* (pełen narzędzi)
- › Dadaizm, prymitywizm, czy ogólnie rozumiana sztuka abstrakcyjna *nie powinna* trafiać do oprogramowania
- › Mini dygresja: ten *warsztat* **nigdy** nie będzie kompletny
 - bądź *zamrożony*
- › Standardy kodowania są jednym z narzędzi



Standardy kodowania

- › Cel: usystematyzowanie procesu wytwarzania kodu i samego kodu
- › Ujednolicenie – usprawnienie utrzymywalności (*maintainability*) i pracy w zespole
- › Zbiór reguł ma pomagać osiągnąć zakładaną jakość
- › *Coding conventions, coding standards, coding style* – stosowane często zamiennie, ale czasem ktoś rozróżnia *style* od *standard*
- › Bo *coding standard* to może być więcej niż głębokość wcięć...
 - Czy *tabs vs spaces*
- › Mini dygresja:
 - dostosowanie się do obowiązującego *coding standardu* stanowi test „dojrzałości”
 - usprawnianie standardów to cecha lidera i eksperta, ale konsensus jest kluczowy



Standard kodowania - Styl

- › Czyli: wcięcia, pozycje *klamerek*, białe znaki, białe linie etc.

```
int* a           vs  int *a
while( a==b )    vs  while (a == b)
if (x) {         vs  if (x)
                  {
```

- › Niektóre języki *proponują* style opracowane przez twórców
np. Python (PEP8), C# (C# Coding Conventions)
- › I na szczęście coraz częściej raczej się go *wybiera*, niż *wymyśla*
- › Wbrew pozorom *nie musi* być składnikiem standardu



Standard kodowania - Nazewnictwo

› Zarówno *notacja* jak i *semantyka*

Nazwa rzeczownik;	vs	LPSTR str1;
rzeczownik->czasownik();	vs	str1->m_len;
bool isManager;	vs	bool manager;
DługaNazwa	vs	długa_nazwa



Standard kodowania – co więcej?

- › Reguły komentowania kodu
- › Metryki kodu (długość funkcji, liczba parametrów etc.)
- › Wykorzystywane paradygmaty
(np. czy programujemy obiektowo w C)
- › Reguły bezpiecznego kodu
(np. nie używamy *gołych* wskaźników w C++,
wszystkie parametry funkcji powinny być *const*)
- › Reguły wynikające ze specyfiki przemysłu
(np. zakaz używania dynamicznej alokacji)
- › Wiele innych...



Standard kodowania – podsumowanie

- › Nie ma *jedynego słusznego poprawnego zawsze*
- › W idealnym świecie jest osiągany przez konsensus zespołu
- › Zespoły mogą używać różnych, a nawet ten sam zespół może używać wielu w różnych projektach
- › Rewolucje w inżynierii rzadko są lepsze od ewolucji (przestarzały standard trzeba powoli i konsekwentnie *przepychać* ku jasnej stronie mocy)
- › Istnieją *formalne* standardy przemysłowe, czasem wymagane w konkretnej branży (np. MISRA C), wtedy mniejszy wybór...
- › Istnieją standardy (reguły) przemysłowe *de-facto*, które można stosować do wielu zestawów: KISS, DRY, GRASP czy SOLID

Dziękuję za uwagę

Konrad.Grochowski@pw.edu.pl

