Clean Code

1. Czym jest Clean Code?

- Książka Clean Code
  - Pochylenie się, bo gdy powstawała to narzędzia SCA były ograniczone
  - Teraz Clean Code jest chyba na wyższym poziomie - mamy gotowe konfiguracje, bierz i używaj
- Co to jest Clean Code?
  - Kod jest czytelny - intuicyjny
  - Kod jest rozszerzalno / zmienialny
- Kontekst - czyli ograniczenia, które posiadamy w projekcie. Nic nie jest uniwersalne.
  - Coś na zapas może nie być potrzebne
  - Coś co ma nastąpić, może się nie wydarzyć (zmiana scope)
  - Ograniczenia konstrukcyjne / językowe bo korzystamy z określonej wersji
- Ustalenie "wystarczająco dobry" kod w projekcie
  - Nie w każdym mamy te same zasady
  - Specyficzny dla zespołu / projektu

2. Jakie zasady i reguły przestrzegać?

- Wspólne standardy
  - Na każdym poziomie
  - Nazewnictwo, Struktura, Formaty, Protokoły etc.
- Komentarze
  - Usuwamy
  - Chyba, że skopiowałem kod z internetu i jest jakimś dziwnym fixem + link do odpowiedzi lub numer ticketa
  - Akceptujemy także @TODO
    - Można zmienić na ticket w narzędziu do zarządzania taskami
    - Szybkie przeszukanie swojego kodu, żeby sprawdzić czy coś jest do poprawy
- Prawidłowe nazewnictwo
  - Intuincyjnie zrozumiałe, ale bez przesady - znów trzeba zadbać o "to zależy"
  - Spójne nazewnictwa - konwencja wprowadzona w projekcie
- Zasada Skauta
  - Zostawiam lepiej posprzątane miejsce po sobie - np. po modyfikacji usprawniam nazewnictwo
  - Rozbij implementację i czyszczenie na osobne commity
- DRY*
  - Uzasadnione duplikaty istnieją - zła abstrakcja jest gorsza niż duplikacja kodu
  - Uwspólnianie kodu
    - Gdy mamy duplikację
    - Opieramy się na min. dwóch przypadkach
- Fajnie jest robić SOLID*
  - To są ciężkie zasady
  - Mogą być czasem pomijalne, bo nie przyniosą korzyści
  - Czasem mogą zaszkodzić wprowadzając złożoność i zbędną abstrakcję
- Magic number, string czy JS literał
  - Czym jest np. 12.5?
  - Wszystko co nie ma opisu czym jest
  - Opisanie złożonego typu, np. za pomocą klasy

3. Jak mierzyć Clean Code?

- Zasady muszą być proste i zrozumiałe - łatwo zrozumie i nauczy się ich Junior
- Fajnie gdy zasady są weryfikowane automatycznie przez skonfigurowane narzędzie
- Definicja wartości co chcemy mierzyć - co dla nas jest dobre, co daje nam korzyść
- Twarda Metryka - Ilość warningów itp. z SCA
- To Zależy - Czasem wydaje NAM się (w zespole), że jest to wystarczająco dobre
  - Uważać na nawyki - zrobię to w sposób A bo zawsze tak robiłem
  - Uważać na abstrakcję - np. nad bazą danych, bo przecież zmieniamy typ bazy danych w projekcie co 2 miesiące

4. Linki

- https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29 — Podsumowanie książki Clean Code
- https://github.com/kettanaito/naming-cheatsheet — Jak nazywać rzeczy
- https://devenv.pl/kod-nigdy-nie-klamie-komentarze-czasami/ — Komentarze w kodzie
- https://www.youtube.com/watch?v=6N_J7KnTErM&list=PLJhoIcqKni7Ov5uOul-w94HKtkxT-Kkm7 — SOLID
- https://devenv.pl/zasada-skautow/ — Zasada Skautów