

Charakterystyki jakości oprogramowania

Przeczytaj poniższy tekst i zastanów się nad swoim produktem i jego cechami. Określ, jaki jest kontekst Twojej aplikacji, a następnie przekształć listę, by była zgodna z przedmiotem Twojej pracy.

Zdolność. Czy produkt ma wartościowe funkcje?

- *Kompletność:* wszystkie ważne funkcje potrzebne użytkownikom końcowym są dostępne.
- *Dokładność:* dowolny wynik lub kalkulacja w produkcie są poprawne oraz prezentowane z użyciem dokładnych wartości.
- *Efektywność:* produkt wykonuje swoje czynności wydajnie (nie robiąc rzeczy, których nie powinien robić).
- *Współpraca:* różne funkcje współdziałają ze sobą w możliwie najlepszy sposób.
- *Wielozadaniowość:* produkt może wykonywać wiele zadań prowadzonych równolegle.
- *Agnostycyzm danych:* produkt wspiera wszystkie możliwe formaty danych i radzi sobie z zakłóceniami.
- *Rozwijalność:* produkt umożliwia klientom lub osobom trzecim dodawanie nowych funkcji lub zmienianie zachowań produktu.

Niezawodność. Czy możesz zaufać produktowi w wielu trudnych sytuacjach?

- *Stabilność:* produkt nie może powodować awarii, nieobsługiwanych wyjątków, ani błędów skryptów.
- *Odporność:* produkt obsługuje przewidziane i nieprzewidziane błędy we właściwy sposób.
- *Obsługa stresowych sytuacji:* jak system radzi sobie w przypadku przekraczania różnych limitów?
- *Odtwarzalność:* istnieje możliwość odzyskania danych i dalszego korzystania z produktu po wystąpieniu błędu krytycznego.
- *Spójność danych:* wszystkie typy danych pozostają niezmiennie w całym produkcie.
- *Bezpieczeństwo:* produkt nie spowoduje żadnych strat w ludziach lub w mieniu.
- *Odtworzenie po katastrofalnej awarii:* co, jeśli stanie się coś bardzo, bardzo złego?
- *Zaufanie:* czy zachowanie produktu jest spójne, przewidywalne i godne zaufania?

Użyteczność. Czy produkt jest łatwy w użyciu?

- *Zachęcenie:* produkt zachęca do odkrywania jego możliwości.
- *Intuicyjność:* czy jest łatwo zrozumieć i wyjaśnić do czego służy ten produkt?
- *Minimalizm:* nie ma zbędnych elementów w zawartości i wyglądzie produktu.
- *Uczenie się:* można szybko i łatwo nauczyć się obsługi produktu.
- *Zapamiętywanie:* jeśli raz nauczyłeś się, jak coś robić, nie zapomnisz tego.
- *Odkrywanie:* informacje i potencjał produktu mogą być odkryte poprzez eksplorację interfejsu użytkownika.
- *Operowanie:* doświadczony użytkownik może wykonywać standardowe akcje bardzo szybko.
- *Interaktywność:* produkt ma łatwe do zrozumienia stany oraz możliwe interakcje wewnątrz aplikacji (poprzez GUI i API).
- *Kontrola:* użytkownik powinien mieć poczucie kontroli nad działaniem programu.
- *Klarowność:* czy wszystko jest przekazane w sposób szczegółowy, zrozumiały i niepodlegający wątpliwości?
- *Błędy:* komunikaty o błędach powinny być pomocne; trudno popełnić błąd, a po jego popełnieniu łatwo go naprawić.
- *Spójność:* zachowanie oraz wygląd są takie same w całym produkcie.
- *Dopasowanie:* domyślne ustawienia i zachowania mogą być elastycznie zmieniane.
- *Dostępność:* produkt może być użyty przez możliwie największą grupę użytkowników i jest zgodny ze standardami dostępności.
- *Dokumentacja:* dostępna jest Pomoc, która pomaga i pokrywa zakres funkcjonalności.

Charyzma. Czy produkt ma „to coś”?

- *Unikalność:* produkt wyróżnia się na tle innych i ma coś, czego nie mają inne produkty.
- *Satysfakcja:* jakie są Twoje wrażenia po skorzystaniu z tego produktu?
- *Profesjonalizm:* czy produkt ma właściwy poziom profesjonalizmu i sprawia wrażenie odpowiedniego do realizacji celu?
- *Atrakcyjność:* czy wszystkie aspekty produktu oddziałują na wzrok i inne zmysły?
- *Ciekawość:* czy użytkownicy będą na tyle zainteresowani, by sprawdzić, co mogą osiągnąć korzystając z produktu?
- *Zaciekawienie:* czy użytkownik „połknął haczyk”, czy korzystał z produktu z przyjemnością, będąc w pełni zaangażowany?
- *Nowinki:* czy produkt używa najnowszych i najlepszych technologii oraz idei?
- *Oczekiwania:* produkt wyprzedza oczekiwania oraz zaspakaja potrzeby, których użytkownicy nawet nie byli świadomi.
- *Perspektywa:* czy produkt i informacje w nim zawarte są poprawnie skonturowane pod kątem języka i stylu?
- *Kierunkowość:* czy aplikacja robi dobre (pierwsze) wrażenie?
- *Historia:* czy istnieją interesujące historie o produkcie, jego powstaniu, konstruowaniu i użytkowaniu?

Bezpieczeństwo. Czy produkt jest zabezpieczony przed niepożądanym użyciem?

- *Autentykacja:* produkt identyfikuje użytkownika.
- *Autoryzacja:* obsługa tego, co użytkownik może zobaczyć i zrobić w produkcie
- *Prywatność:* zdolność do nieujawniania chronionych danych nieautoryzowanym użytkownikom.
- *Luki bezpieczeństwa:* w produkcie nie ma luk podatności socjotechnicznych.
- *Tajność:* produkt w żadnym wypadku nie ujawnia informacji o technologiach leżących u podstawy systemu.
- *Nienaruszalność:* zdolność stawiania oporu próbom penetracji.
- *Brak wirusów:* produkt nie ma wirusów i nie jest identyfikowany jako wirus.
- *Odporność na piractwo:* brak możliwości nielegalnego kopiowania i dystrybucji oprogramowania lub kodu.
- *Zgodność:* przestrzeganie obowiązujących standardów.

Wydajność. Czy produkt jest wystarczająco szybki?

- *Zdolność:* różne ograniczenia w produkcie dla różnych okoliczności (np. niska przepustowość)
- *Utylizacja zasobów:* właściwe zarządzanie pamięcią, przestrzenią dyskową i innymi zasobami.
- *Reagowanie:* szybkość wykonania (lub postrzegania wykonania) zadania.
- *Dostępność:* system jest dostępny do użycia wtedy, kiedy powinien.
- *Przepustowość:* zdolność produktu do przetwarzania wielu rzeczy.
- *Wytrzymałość:* produkt może wytrzymać obciążenie przez długi czas?

- *Informacja zwrotna*: czy informacja zwrotna wysyłana przez system w odpowiedzi na działanie użytkownika jest odpowiednia?

- *Skalowalność*: jak dobrze produkt skaluje się dla wzrastającego i malejącego obciążenia?

IT-alność. Czy produkt jest łatwy do zainstalowania, utrzymania i wspierania?

- *Wymagania systemowe*: zdolność do uruchomienia produktu na wspieranych konfiguracjach i obsługa różnych środowisk lub brakujących komponentów.

- *Instalowalność*: produkt może zostać zainstalowany na zdefiniowanej platformie oraz na odpowiedniej przestrzeni.

- *Aktualizacje*: łatwość zaktualizowania do nowszej wersji bez utraty konfiguracji i ustawień.

- *Dezinstalacja*: czy wszystkie pliki (za wyjątkiem plików użytkownika lub plików systemowych) i inne zasoby zostają usunięte podczas dezinstalacji?

- *Konfiguracja*: czy instalacja może być konfigurowana na różne sposoby i do różnych lokalizacji, co wspiera użytkowanie produktu przez użytkownika?

- *Dostarczenie*: dział IT może dostarczyć produkt użytkownikom różnego typu oraz dla różnych środowisk.

- *Utrzymywanie*: czy produkt i jego składowe mogą być łatwo utrzymane i wspierane dla klienta?

- *Testowalność*: jak dostarczony produkt może być efektywnie testowany przez klienta?

Kompatybilność. Jak dobrze produkt współdziała z innym oprogramowaniem i środowiskami?

- *Kompatybilność sprzętowa*: produkt może zostać użyty na odpowiednich konfiguracjach sprzętowych.

- *Kompatybilność względem systemu operacyjnego*: produkt może zostać uruchomiony na wyspecyfikowanych wersjach systemu operacyjnego i zachowuje się w typowy sposób.

- *Kompatybilność aplikacyjna*: produkt i jego dane współpracują z innymi aplikacjami, których klient może używać.

- *Kompatybilność konfiguracyjna*: zdolność oprogramowania do rozpoznania konfiguracji środowiska.

- *Kompatybilność wsteczna*: czy produkt robi wszystko, co potrafiła jego wcześniejsza wersja?

- *Kompatybilność przyszła*: czy produkt będzie zdolny do użycia artefaktów oraz interfejsów przyszłych wersji?

- *Zrównoważony rozwój*: wpływ na środowisko np. wydajność energetyczna, wyłączanie, tryb oszczędzania energii, telepraca.

- *Zgodność ze standardami*: produkt jest zgodny z odpowiednimi standardami, regulacjami, prawem lub etyką.

Wewnętrzne charakterystyki jakości oprogramowania

Poniższe charakterystyki nie są bezpośrednio odczuwane przez końcowego użytkownika, jednak mogą być równie ważne dla powodzenia produktu.

Wsparcie. Czy użytkowanie i problemy klienta mogą być wspierane?

- *Identyfikacja*: czy można łatwo zidentyfikować części produktu, ich wersje lub konkretne błędy?

- *Diagnostyka*: czy znalezienie szczegółów sytuacji klienta jest możliwe?

- *Rozwiązywalność problemów*: czy można łatwo wskazać błędy (np. w logach) i uzyskać pomoc?

- *Debugowanie*: czy w razie potrzeby można zaobserwować wewnętrzne stany oprogramowania?

- *Wszechstronność*: możliwość użycia produktu na więcej sposobów niż wstępnie zaprojektowano.

Testowalność. Czy można łatwo sprawdzić i przetestować produkt?

- *Śledzenie*: produkt loguje akcje na właściwym poziomie i w użytecznym formacie.

- *Kontrolowalność*: zdolność do niezależnego definiowania stanów, obiektów oraz zmiennych.

- *Obserwowalność*: zdolność do obserwowania elementów, które powinny zostać poddane testom.

- *Monitorowanie*: czy produkt podpowiada co się z nim dzieje?

- *Izolacja*: zdolność do testowania pojedynczych części produktu.

- *Stabilność*: zmiany oprogramowania są kontrolowane i niezbyt częste.

- *Automatyzacja*: czy w oprogramowaniu są publiczne lub ukryte interfejsy programistyczne, z których można skorzystać?

- *Informacja*: umożliwienie testerom nauczenia się tego, co trzeba...

- *Audyt*: czy produkt i to, co produkt tworzy może być walidowane?

Utrzymanie. Czy produkt może zostać utrzymywany i rozszerzany niskim kosztem?

- *Elastyczność*: zdolność do modyfikacji produktu na żądanie klienta.

- *Rozszerzalność*: czy będzie łatwo dodawać w przyszłości nowe funkcje?

- *Prostota*: kod nie kod nie jest bardziej skomplikowany niż jest to konieczne; nie utrudnia projektowania, uruchomienia i ewaluacji.

- *Czytelność*: kod jest właściwie opisany, łatwy do odczytania i zrozumienia.

- *Przejrzystość*: czy można łatwo zrozumieć strukturę bazową?

- *Modułowość*: kod jest podzielony na dające się kontrolować elementy.

- *Refaktorowalność*: czy testy jednostkowe są według Ciebie wystarczająco dobre?

- *Analiza*: zdolność do znalezienia przyczyn defektów lub innego interesującego nas kodu.

Przenośność. Czy przenoszenie oprogramowania do innych środowisk i języków jest możliwe?

- *Ponowne użycie*: czy części produktu mogą być ponownie użyte w innym miejscu?

- *Adaptacja*: czy można łatwo zmieniać produkt, by wspierać inne środowiska?

- *Kompatybilność*: czy produkt współgra ze standardowymi interfejsami lub oficjalnymi standardami?

- *Umiejętność*: czy można łatwo przetłumaczyć produkt?

- *Lokalizacja*: czy części produktu można dopasować dla spełnienia potrzeb docelowych kultur lub państw?

- *Odporność interfejsu użytkownika*: czy produkt będzie wyglądał tak samo dobrze po przetłumaczeniu?



Rikard Edgren, Henrik Emilsson i Martin Jansson - thetesteye.com v1.1

Translated to Polish by Radosław Smilgin

Niniejsza praca objęta jest licencją Creative Commons Attribution-No Derivative License inspirowaną przez James Bach CRUSSPIC STMP, ISO 9126-1, Wikipedia:Ilities i więcej...