

Wymagania pozafunkcjonalne

Projektowanie interfejsu użytkownika

Wymagania pozafunkcjonalne

Jak wspomniano na poprzednich zajęciach, wymagania funkcjonalne dotyczą funkcjonalności tworzonego systemu i odpowiadają na pytanie "co" dane oprogramowanie ma wykonywać. W przypadkach użycia, które opisują różne funkcje aplikacji, kładzie się nacisk na demonstrację zasady realizacji danego wymagania, natomiast celowo pomija się szczegóły techniczne, aby nie sugerować rozwiązań, które potem mogą okazać niedostatecznie dobre (a przez to zmiana przypadków użycia wymagałaby większej ilości czasu). Musi istnieć jednak sposób na przedstawienie tego "jak" powinien działać system.

Na to pytanie odpowiadają **wymagania pozafunkcjonalne** (ang. *non-functional requirements*, NFR), które dotyczą wszelkich rzeczy związanych z tworzonym oprogramowaniem, ale bardziej od strony jakościowej. Mogą wynikać one z wymagań funkcjonalnych oraz ogólnych założeń architektonicznych, ich źródłem zwykle są klienci, użytkownicy oraz osoby, które będą utrzymywać oprogramowanie. Czynniki brane pod uwagę przy specyfikowaniu NFRów lepiej można zobrazować na przykładzie głównych kategorii standardu ISO 25010:

- **Funkcjonalne dopasowanie** (funkcjonalna kompletność, poprawność, np. jak dokładny ma być system w różnych kwestiach)
- **Wydajność** (charakterystyka czasowa, zużycie zasobów, np. jak szybko mają być wyświetlani użytkownicy)
- **Kompatybilność** (współistnienie w różnych środowiskach, np. z jakimi systemami operacyjnymi aplikacja będzie współpracować)
- **Użyteczność** (łatwość korzystania, ochrona przed błędami, estetyka, np. jak długo zajmie nauka korzystania z oprogramowania)
- **Niezawodność** (dostępność techniczna, odporność na wady, odtwarzalność, np. jak często dopuszczamy brak możliwości korzystania z serwisu)
- **Bezpieczeństwo** (poufność, identyfikowalność, np. jak dokładnie będziemy zapamiętywać historię operacji w systemie)
- **Łatwość utrzymania** (łatwość analizy, modyfikacji, testowania, np. jaka dokumentacja techniczna powinna być przygotowana)
- **Przenośność** (łatwość adaptacji, instalacji, aktualizacji)

Wymagania pozafunkcjonalne (oraz ich opis) powinien być zgodny z pięcioma cechami układającymi się w skrót SMART:

- **Simple & Specific** (proste) - krótkie, proste, trafiające w sedno
- **Measureable** (mieralne) - istnieje metoda za pomocą której podczas odbioru pokażemy, że wymaganie jest spełnione
- **Achievable** (osiągalne) - możliwe do realizacji w naszym kontekście

- **Relevant** (istotne) - rzeczywiście wymaganie jest potrzebne, aby system miał wartość biznesową (tutaj również stosujemy system priorytetów)
- **Traceable** (możliwe do śledzenia) - łatwo je "zlokalizować" (w dokumentacji lub innym źródle) wraz z uzasadnieniem biznesowym

Ta część analizy systemu jest potrzebna z uwagi na jej implikacje w kolejnych fazach projektowania oprogramowania - tworzeniu architektury czy prototypu interfejsu. Zwykle specyfikuje się przeglądając każdą kategorię (i podkategorie) po kolei, starając się przy tym odkryć cechy, które są odpowiednie dla tworzonego systemu. Sposób opisu każdego wymagania pozafunkcjonalnego powinien wyglądać następująco:

- Identyfikator - podobnie jak w przypadkach użycia
- Opis wymagania - jednozdaniowa definicja wymagania
- (Pod)kategoria - typ wymagania zgodny z ISO 25010 (lub innymi standardami)
- Priorytet - jak ważne jest dane wymaganie (skala H, M, L)

Przykłady poprawnego opisu wymagań pozafunkcjonalnych:

Identyfikator	NFR04
Opis	System musi podawać wartości pieniężne z dokładnością do drugiego miejsca po przecinku.
Kategoria	Funkcjonalna poprawność
Priorytet	H

Identyfikator	NFR11
Opis	Maksymalny czas pomiędzy pobudzeniem przez użytkownika a odpowiedzią systemu nie powinien być dłuższy niż 3 sekundy.
Kategoria	Charakterystyka czasowa
Priorytet	H

Identyfikator	NFR12
Opis	System nie powinien blokować działania zewnętrznych źródeł danych przez wielokrotne wykonywanie zapytań.
Kategoria	Interoperacyjność
Priorytet	H

Identyfikator	NFR16
Opis	System powinien wyraźnie zaznaczać pola obowiązkowe w formularzu.
Kategoria	Ochrona użytkownika przed błędami
Priorytet	M

Identyfikator	NFR19
Opis	System nie powinien wyświetlać tekstu napisanego czcionką mniejszą niż 12 pikseli.

Kategoria	Dostępność personalna
Priorytet	L

Identyfikator	NFR26
Opis	System ma działać na następujących przeglądarkach internetowych: Internet Explorer 8.0+, Google Chrome 17.0+, Mozilla Firefox 10.0+, Opera 12.0+.
Kategoria	Kompletność kontekstowa
Priorytet	H

Identyfikator	NFR28
Opis	System powinien przechowywać historię przeprowadzanych operacji w formie pliku tekstowego.
Kategoria	Kompletność kontekstowa
Priorytet	M

Identyfikator	NFR29
Opis	System musi przysyłać hasło użytkownika do serwera szyfrując je algorytmem MD5.
Kategoria	Poufność
Priorytet	H

Identyfikator	NFR30
Opis	W dostarczonej dokumentacji technicznej musi zawierać się instrukcja przeprowadzania modyfikacji (w tym aktualizacji systemu do nowszej wersji).
Kategoria	Łatwość zamiany
Priorytet	M

Dodatkowo, często podaje się uzupełniające informacje jak trudność realizacji czy sposób weryfikacji, czy dane wymaganie zostało spełnione.

Prototyp interfejsu użytkownika

Przy tworzeniu przypadków użycia, mimo unikania szczegółów interfejsu, powstaje już pewna wizja tego, w jaki sposób pozycjonowane będą konkretne elementy na ekranie i jak użytkownik będzie się poruszał po systemie. Jest to pewne wyobrażenie systemu, którego źródłem jest tworzący go zespół, klient czy docelowy użytkownik. Na prototyp, który powinien powstać, wpływ mają również wymagania pozafunkcjonalne oraz pewne decyzje architektoniczne.

Na tym etapie (i ogólnie, w kontekście całego projektu) zwykle poprzestaje się na prototypie o niskiej wierności (ang. *low fidelity*) z uwagi na czas tworzenia oraz to, że najważniejsze jest pokazanie idei graficznej i funkcjonalnej systemu, a nie jego wszystkie szczegóły graficzne.

Prototyp ulega również wielu zmianom pod wpływem jego oceny przez wiele osób, w związku czym należy przygotowywać makietę w taki sposób, aby była łatwa w modyfikacji.