

- Walidacja produktu pod kątem dostosowania do określonych potrzeb i oczekiwań odbiorców.

Uszczegółowieniem polityki testów jest strategia testów. Strategia testów opisuje ogólne podejście do testów w organizacji i zwykle zawiera opis realizowanych zwyczajowo poziomów testów wraz z kryteriami przejścia między poziomami, typowe techniki i narzędzia oraz procedury wykorzystywane w procesie testowym, ze szczególnym uwzględnieniem wskazania sposobu, w jaki środki te mają łagodzić typowe ryzyka produktowe.

Polityka i strategia testów wyznaczają ogólne cele i ramy procesów testowych, są więc pomocne przy opracowywaniu koncepcji ogólnego procesu testowego, realizowanego na poziomie całej organizacji.

Przyjrzyjmy się zatem poszczególnym elementom strategii testowej.

### 3.3. Poziomy testów

#### Pojęcie weryfikacji i walidacji

Tematu poziomów testów nie sposób wyjaśnić bez przypomnienia koncepcji jednych z podstawowych elementów zarządzania jakością oprogramowania: weryfikacji i walidacji (V&V, *verification and validation*). Zgodnie z CMMI® walidacja sprawdza, czy produkt lub jego elementy (w tym specyfikacja) spełnia zamierzone zastosowanie produktu, jeśli umieści się go w docelowym środowisku, weryfikacja zaś „dostarcza punktów kontrolnych, w których wybrane produkty końcowe lub pośrednie (np. specyfikacje) są sprawdzane pod kątem spełnienia wymagań odnośnie do owych produktów” [CMMI].

Na pierwszy rzut oka pojęcia walidacji i weryfikacji wydają się identyczne, różnicę stanowi kilka istotnych słów:

- Walidacja sprawdza, czy produkt spełnia wymagania konieczne do określonego użycia. Odnosi się do oceny spełnienia oczekiwań interesariuszy, w szczególności docelowych użytkowników w realnych warunkach użycia produktu.
- Weryfikacja natomiast polega na sprawdzeniu, czy w produkcie zaimplementowano określone, zdefiniowane uprzednio wymagania.

Cele weryfikacji i walidacji są w różny sposób realizowane przez określone poziomy testów.

#### Poziomy testów

Testowaniu można poddawać nie tylko produkt końcowy, lecz także produkty cząstkowe na różnym etapie rozwoju. Przykładem „półproduktu” może być kod źródłowy

czy poszczególne interfejsy, weryfikowane jednostkowo przed pełną integracją systemu. To swoiste zróżnicowanie zakresu testów odzwierciedla koncepcja poziomów testów. Przykładem zastosowania koncepcji poziomów testów jest wykonane czynności weryfikacji dla poszczególnych komponentów rozwiązania (testy jednostkowe) czy też ocena działania produktu jako całości (testy systemowe). Organizacja testów w formie poziomów umożliwi zróżnicowanie celów testowania i systematyczną kontrolę jakości na każdym etapie wytwarzania produktu.

Liczba i definicja poziomów testów różnią się w zależności od źródła, np. SWEBOK© wyróżnia trzy główne poziomy testów – jednostkowe, integracyjne i systemowe – podczas gdy inne modele (np. model V) wyróżniają jeszcze jeden niezbędny poziom – testy akceptacyjne (akceptacji) [SWEBOK].

Przyjrzyjmy się zatem typowym poziomom testów.

#### **Testy jednostkowe (modułowe)**

Testy jednostkowe (*unit test*), inaczej testy programistyczne, skupiają się na weryfikacji działania indywidualnych elementów oprogramowania. Takimi elementami mogą być metody czy klasy, poddawane testom w izolacji od pozostałych elementów produktu. Celem testów modułowych jest sprawdzenie działania poszczególnych składników systemu bez „szumów komunikacyjnych”, czyli niezależnie od innych składników produktu. Ten poziom testów zwykle wymaga dostępu do kodu źródłowego, co powoduje, iż najczęściej wykonywany jest przez osoby znające języki programowania. Te testy są zwykle w pełni zautomatyzowane oraz ich realizacja jest wspierana przez narzędzia.

#### **Testy integracji**

Testy integracji/integracyjne (*integration test*) weryfikują komunikację między komponentami oprogramowania, np. poszczególnymi funkcjami, modułami czy nawet systemami. Zazwyczaj testy te są wykonywane po testach jednostkowych indywidualnych modułów lub po testach systemowych – jeśli integracja dotyczy systemów.

Sama integracja poszczególnych elementów może przebiegać według kilku podejść. Klasyczne podejścia do testowania integracyjnego zawierają m.in. podejście zstępujące (*top-down*) oraz wstępujące (*bottom-up*), podczas gdy nowoczesne strategie integracji opierają się na architekturze (technicznej lub biznesowej) – co przekłada się na integrację komponentów lub podsystemów na podstawie uprzednio zidentyfikowanych wątków (czy procesów) funkcjonalnych. Niemal w każdym przypadku rekomendowaną strategią integracji jest podejście przyrostowe umożliwiające minimalizację ryzyka i łatwiejszą identyfikację źródeł potencjalnych usterek.

#### **Testy systemowe**

Testowanie systemowe (*system test*) można określić jako weryfikację zachowania całego systemu. Testy te realizowane są zwykle wtedy, gdy dostępny jest już cały produkt