

Metody oraz strategie testowania oprogramowania.

Testowanie oprogramowania jest istotnym etapem w procesie jego wytwarzania, mającym na celu zapewnienie jakości i poprawności działania. Istnieje wiele metod oraz strategii testowania, z których korzysta się w zależności od rodzaju projektu, dostępnych zasobów i wymagań.

Metody Testowania:

Jednostkowe: Testowanie pojedynczych elementów kodu.

Integracyjne: Testowanie interakcji między modułami.

Systemowe: Całościowe testowanie systemu.

Akceptacyjne: Potwierdzenie spełnienia wymagań użytkownika.

Wydajnościowe: Sprawdzenie szybkości i skalowalności systemu.

Bezpieczeństwa: Identyfikacja zagrożeń i zabezpieczeń.

Strategie Testowania:

Czarna Skrzynka: Testowanie bez wiedzy na temat wewnętrznej implementacji.

Biała Skrzynka: Testowanie z uwzględnieniem struktury kodu.

Szara Skrzynka: Połączenie podejść czarno- i białoskrzynkowego.

Regresywne: Testowanie po każdej zmianie w celu zabezpieczenia przed regresją.

Progresywne: Testowanie nowych funkcji przed pełnym wdrożeniem.

Migawkowe: Testowanie stabilnej wersji systemu.

Istotności (Smoke): Szybkie sprawdzenie głównych funkcji systemu.

Obciążeniowe: Testowanie pod dużym obciążeniem.

Niestandardowe (Ad-hoc): Nieterminowe testowanie bez wcześniejszego planu.

Metody identyfikacji wymagań systemu informatycznego.

Identyfikacja wymagań systemu informatycznego jest procesem zbierania, analizy i dokumentowania potrzeb oraz oczekiwań użytkowników i interesariuszy.

Wywiady: Rozmowy z interesariuszami w celu zrozumienia ich potrzeb.

Analiza dokumentów: Przegląd istniejących dokumentów biznesowych.

Ankiety: Rozpowszechnianie pisemnych kwestionariuszy wśród użytkowników.

Warsztaty: Zorganizowane spotkania z interesariuszami w celu zbierania wymagań.

Prototypowanie: Tworzenie modeli systemu do uzyskania feedbacku od użytkowników.

Scenariusze użycia: Opisywanie, jak system będzie używany w różnych sytuacjach.

Analiza procesów biznesowych: Badanie i zrozumienie istniejących procesów.

Badania rynkowe: Analiza trendów i standardów branżowych.

Analiza problemu: Identyfikacja problemów, z którymi borykają się użytkownicy.

Analiza ryzyka: Identyfikacja potencjalnych zagrożeń i problemów.

Metodologie wytwarzania systemów informatycznych.

Istnieje wiele różnych metodologii wytwarzania systemów informatycznych, z których każda ma swoje zalety, wady i obszary zastosowania.

Model kaskadowy (Waterfall):

- Proces ten jest liniowy i składa się z etapów takich jak analiza, projektowanie, implementacja, testowanie, wdrożenie i konserwacja.
- Każdy etap musi zostać ukończony przed rozpoczęciem kolejnego.
- Jest to dobra metoda w przypadku prostych projektów, gdzie wymagania są stabilne.

Model spiralny:

- Opiera się na modelu kaskadowym, ale dodaje elementy iteracyjne.
- Proces jest podzielony na wiele cykli, zwanych spiralami, z których każdy obejmuje planowanie, ryzyko, inżynierię i ewaluację.
- Jest bardziej elastyczny niż model kaskadowy, co umożliwia dostosowanie do zmieniających się wymagań.

Metodyka zwinna (Agile):

- Skupia się na elastyczności i dostosowywaniu się do zmieniających się potrzeb klienta.
- Popularne ramy pracy Agile to Scrum, Kanban, XP (eXtreme Programming) itp.
- Rozwój jest prowadzony w krótkich iteracjach, zwanych sprintami, z częstym dostarczaniem działającego oprogramowania.

Model V-Model:

- Jest to rozwinięcie modelu kaskadowego, w którym etapy rozwoju są powiązane z odpowiadającymi im etapami testowania.
- Każdy etap rozwoju ma swój równoważny etap testowania.

Prototypowanie:

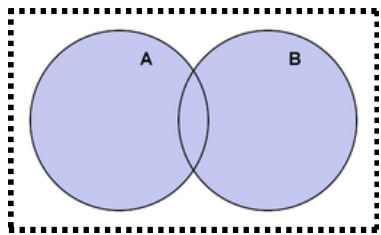
- Zamiast opracowywać pełny system od razu, tworzony jest prototyp, który jest następnie testowany i ulepszany.
- Wartość prototypowania polega na uzyskaniu szybkiej informacji zwrotnej od użytkowników.

Wybór odpowiedniej metodologii zależy od rodzaju projektu, dostępnych zasobów, wymagań klienta i innych czynników. W praktyce wiele organizacji łączy elementy różnych metodologii, dostosowując się do specyfiki danego projektu.

Działania na zbiorach.

☐ Suma zbiorów:

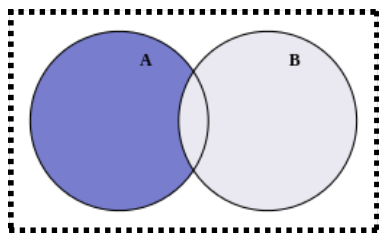
Sumą zbiorów A i B nazywamy zbiór tych elementów, które należą do zbioru A lub do zbioru B



Jeżeli $A=\{1,2,5\}$ i $B=\{1,3,4\}$, to $A \cup B=\{1,2,3,4,5\}$

☐ Różnica zbiorów:

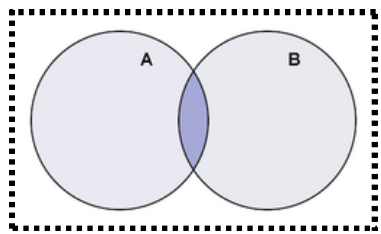
Różnicą zbiorów A i B nazywamy zbiór tych elementów, które należą do zbioru A , a które nie należą do zbioru B



Jeśli $A=\{1,2,5\}$ i $B=\{1,3,4\}$, to $A \setminus B=\{2,5\}$

☐ Iloczyn zbiorów:

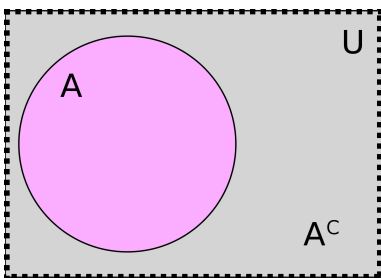
Iloczynem/Częścią wspólną zbioru A i B nazywamy zbiór tych elementów, które należą jednocześnie do zbioru A i do zbioru B



Jeśli $A=\{1,2,5\}$ i $B=\{1,3,4\}$, to $A \cap B=\{1\}$

☐ Dopełnienie zbioru:

Dopełnieniem zbioru A z przestrzeni U nazywamy zbiór tych elementów przestrzeni U , które **nie** należą do zbioru A



Jeśli $A=\{1,2,3\}$, a przestrzenią U jest zbiór wszystkich liczb całkowitych dodatnich, to dopełnieniem zbioru A będzie zbiór $A'=\{4,5,6,7,8,\dots\}$

Własności działań na zbiorach i prawa De Morgana

- $(A \cup B)' = A' \cap B'$ -- I prawo De Morgana
- $(A \cap B)' = A' \cup B'$ -- II prawo De Morgana
- $A \cup B = B \cup A$ -- przemienność dodawania zbiorów
- $A \cap B = B \cap A$ -- przemienność mnożenia zbiorów
- $(A \cup B) \cup C = A \cup (B \cup C)$ -- łączność dodawania zbiorów
- $(A \cap B) \cap C = A \cap (B \cap C)$ -- łączność mnożenia zbiorów
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ -- rozdzielność dodawania zbiorów względem mnożenia
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ -- rozdzielność mnożenia zbiorów względem dodawania