20.04.2020r

## SDLC

Cykl wytwarzania oprogramowania nazywany również SDLC (Software Development Life Cycle).



1. **Planowanie (Planning)** – Zebranie i analiza wymagań przedstawionych przez klientów, interesariuszy, ekspertów z branży, na podstawie badań rynkowych. W oparciu o wymagania planowane są dalsze działania i opracowywana jest strategia.
2. **Analiza (Analysis)** – Definiowanie problemów jakie może napotkać zespół podczas wytwarzania oprogramowania, analiza ograniczeń, analiza celów jako funkcji, rekomendacje.
3. **Projektowanie (Design)** – Zaprojektowanie produktu na podstawie zebranych wymagań i przeprowadzonych analiz, konsultacji
4. **Implementacja (Implementation)** – Budowanie produktu, tworzenie kodu
5. **Testowanie i integracja (Testing & Integration)** – Testowania oprogramowania oraz intergacja z bibliotekami, bazami danych lub innymi programami.
6. **Utrzymanie (Maintenance)** – Przeszkolenie użytkowników oprogramowania, przekazanie dokumentacji, zapewnienie pomocy w korzystaniu z oprogramowania, bieżące rozwiązywanie powstałych błędów, awarii.

Zalety:

-Jest wysoka jakość, efektywność i ekonomiczność procesu wytwarzania.  
-Można zastosować dwa podejścia przy wytwarzaniu oprogramowania z zastosowaniem SDLC i są to podejście tradycyjne (Waterfall) oraz zwinne (Agile).

## STLC

Cykl życia testowania. Musimy spełnić wszystkie kryteria wejścia i wyjścia dla wszystkich poziomów w cyklu testowania oprogramowania.

**Kryteria wejścia:** Kryteria wejścia wskazują przedmioty, które musimy spotkać przed rozpoczęciem testów.  
**Kryteria wyjścia:** Kryteria wyjścia określają elementy, które należy wypełnić, zanim będziemy mogli ukończyć testowanie.

1. **Analiza wymagań** - Podczas tej fazy jest badanie oprogramowania w celu zidentyfikowania sprawdzalnych wymagań. Wymagania mogą być funkcjonalne (co ma robić) i niefunkcjonalne (bezpieczeństwo)
2. **Faza planowania** - Przygotowanie planu testu/dokumentu strategicznego dla różnych rodzajów testów, wybór narzędzi testowych i strategii. Planowanie zasobów oraz określanie ról i obowiązków.
3. **Integracja z testami** – tworzenie przypadków testowych, tworzenie danych testowych. Przejrzyj i sprawdź przypadki testowe oraz skrypty.  
   W tej fazie przypadki testowe i skrypty testowe są tworzone, weryfikowane i przerabiane. Identyfikujemy, tworzymy i oceniamy dane testowe do późniejszej edycji.
4. **Ustaw środowisko testowe** - Zrozumienie wymaganej architektury, środowiska i tworzenie listy wymagań sprzętowych i programowych dla środowiska testowego
5. **Faza implementacji/wykonanie testu** – przygotowanie przypadków testowych, dokumentacja wyników i raport błędów.
6. **Zamknięcie cyklu testowego** - Ocena kryteriów ukończenia cyklu na podstawie czasu, pokrycia testowego, kosztów, oprogramowania, kluczowych celów biznesowych i jakości, oraz przygotowanie statystyk. Tworzenie raportu

## V model

Sekwencyjny model wytwarzania oprogramowania. Na każdym etapie powinniśmy testowa nasze oprogramowanie, począwszy od planowania. Im później wykryjemy błędy tym wyższe będą koszty naprawy błędu.  
Służy temu V model, obrazuje on etapy tworzenia oprogramowania oraz odpowiadające każdemu etapowi testy które powinny być wykonywane.   
Najczęściej posiadający cztery poziomy testowania odpowiadające czterem poziomom rozwoju oprogramowania.

* testy modułowe,
* testy integracyjne,
* testy systemowe,
* testy akceptacyjne.

