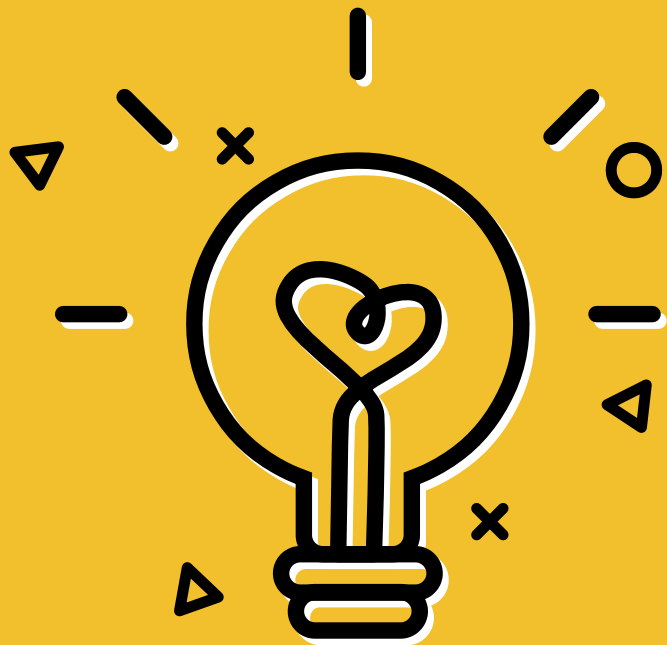




Rodzaje dokumentacji, sposoby jej prowadzenia

Dokumentacja w projekcie

Opracowanie: inż. Grzegorz Petri

RTFM!



*„Without requirements or design, programming is the art
of adding bugs to an empty text file“*

(Louis Srygley)



01 Cykl życia oprogramowania



◀ *W OpenSource nic nie ginie – jest forkowane ewentualnie zapominane*

Etapy Cyklu życia (wytwarzanego) oprogramowania

(Steps of Software Development Life Cycle)

01

Planowanie (*planning*)

- jakie problemy adresujemy?
- podejście zespołu deweloperskiego
- podstawowe funkcje oprogramowania
- etapy rozwijania projektu
- określenie wymagań
- *przypisanie ról i odpowiedzialności*

02

Analiza (*analysis*)

- studium wykonalności
- wysiłku jaki należy włożyć w projekt
- dobór zespołów deweloperskich
- szacowanie czasu realizacji
- szacowanie kosztów i ostatecznej ceny
- *licencje, zgodność prawna itp*

03

Projektowanie (*design*)

- opracowane procesy biznesowe i słownik funkcjonalności i usługi systemu
- szkice widoków i makiety układów (UI)
- struktury i schematy danych (ERD)
- diagramy i algorytmy (UML, BPMN)
- spis modułów i podsystemów

04

Implementacja

- modele danych, klas, schematy BD
- biblioteki, moduły
- interfejs użytkownika i interakcje
- system jako całość, instalatory

05

Testowanie (*testing*)

- Unit, Integration, Functional (TDD)
- zbiory danych (*DataSet*)
- Black-box / White-Box
- Automation, System (*build, end-user*)
- Performance, Security

06

Utrzymanie (*maintenance*)

- konserwacja w środowisku klienta
- zarządzanie zmianami (*środowiska, technologii, podstaw prawnych*)
- modyfikacje systemu (*zmiana potrzeb*)
- dostosowanie do potrzeb rynku
- mierzenie wydajności systemu



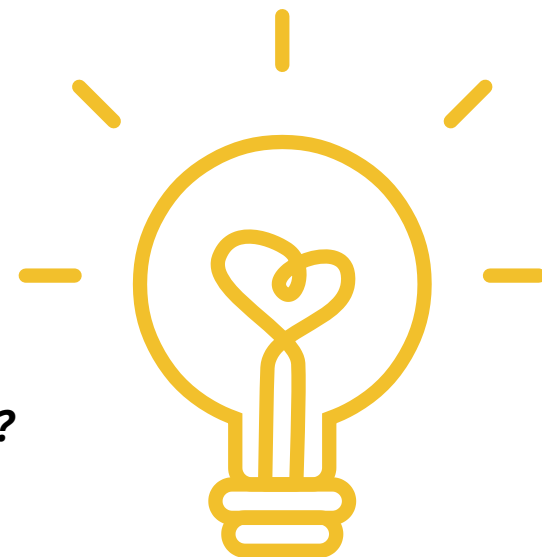
Etapy SDLC a metodologia

Nie ważne jak rozbudowany jest lub będzie projekt – SDLC będzie występować zawsze – mniej szczegółowy lub bardziej. SDLC będzie się też różnił w zależności od metodologii wytwarzania oprogramowania.



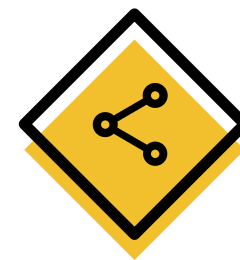
Każdy z etapów SDLC składa się z mniejszych działań, a te z kolej z konkretnych kroków.

Część działań wykonuje się dopiero po podjęciu decyzji. Jednak w każdym momencie może zajść też konieczność cofnięcia się, powtórzenia jakiegoś działania, czy wręcz rozpoczęcia od nowa.



Kto o tym decyduje? Kiedy i na jakiej podstawie?

- Metodologie wytwarzania oprogramowania



Waterfall
model

Incremental
model

Spiral model

Rapid
application
development
(RAD)

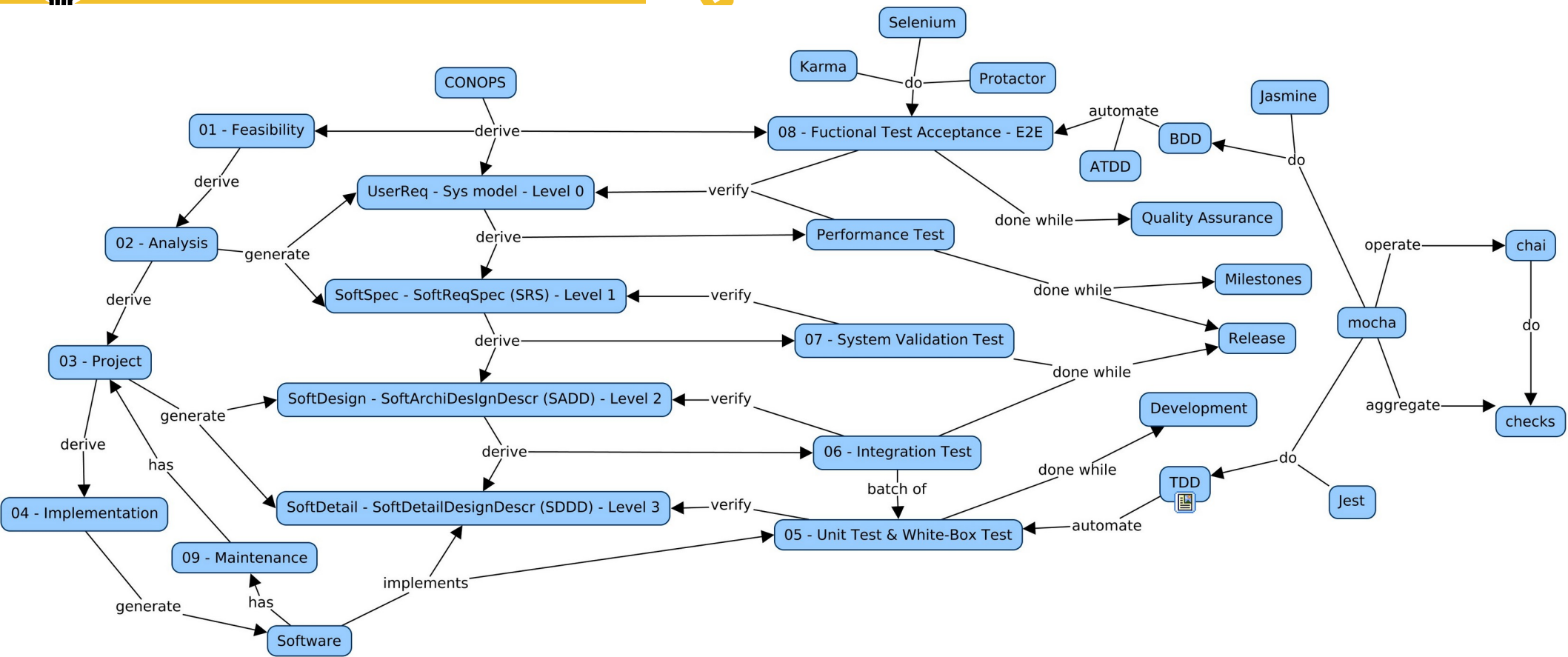
Continuous
integration Agile



i inne zaawansowane metodologie



- Wizualizacja SDLC





Uczestnicy projektu



Wspomniane osoby są **twórcami** oraz **odbiorcami** Wymagań.



„3 Biggest Software Lies: (...)“

#2 We're working on the documentation, (...)“

(Anonymous)



02 Typy dokumentacji



◀ *To, że TY nie potrzebujesz dokumentacji nie znaczy, że inni też*


01

Wymagania (*requirements*)

- cel wdrożenia
- odbiorcy/użytkownicy systemu
- przypadki użycia
- wymagania funkcjonalne
- wymagania niefunkcjonalne (*prawo, ciągłość pracy, czas ładowania systemu, jego ciągłej pracy, środowisko produkcyjne*)

02

Architektura / Projekt (*architecture / design*)

- architektura 3-wartkowa (WEB)
- MVC, MTV, Usług
- przepływ procesów
- architektura UI/UX



Ogólne typy 
dokumentacji

04

Techniczna (*technical*)

- deweloperska (*do bieżących prac oraz wsparcia i utrzymania systemu*)
- API, SDK, DevEnv
-  • wymagania systemowe
- administrator (*klient*)

03

Użytkownika (*end-user*)

- zdjęcia, zasoby do pobrania
- instrukcje, broszury, tutoriale
- FAQ, rozwiązania problemów, pomoc
- ! dokumenty użytkownika
- ! dokumenty administratora

04

Reklamy (*marketing*)

- zdjęcia z cechami oprogramowania
- broszury reklamowe
- pytania i odpowiedzi
- tutoriale
- specyfika systemu



- Typy dokumentacji

Szczegółowe typy

1. **Process documentation**

- dokumentacja procesu deweloperskiego

2. **Planning documentation** (SDP)

- plany deweloperskie (*Roadmap*)

3. **Estimate documentation**

- szacowanie kosztów i czasu

4. **Standards documentation**

- dokumentacja norm, zasad, standardów i formatów

5. **Metrics** documentation

6. **Scheduling documentation**

- dokumentacja harmonogramu prac i zadań (*np. Wykres Gantt'a*)

7. **Product documentation**

- system docs. & user docs.

8. **System** documentation

9. **Requirements** documentation

10. **Architecture and Design** documentation

11. **Technical** documentation

12. **User** documentation

- zasoby do pobrania, FAQ, rozwiązania częstych problemów, tutoriale, informacje pomocy

13. **End-user** documentation

- wymagania systemowe, instrukcje instalacji, uruchamianie i użytkowanie systemu, zdjęcia cech oprogramowania, rozwiązania częstych problemów, informacje kontaktowe dla użytkownika

14. **System administrator** documentation

- wymagania systemowe, instrukcje instalacji oraz aktualizacji, lista funkcjonalności, praca oprogramowania na różnych systemach, możliwe działania podczas awarii



Issues?

report@gplweb.pl
edu.gplweb.pl