

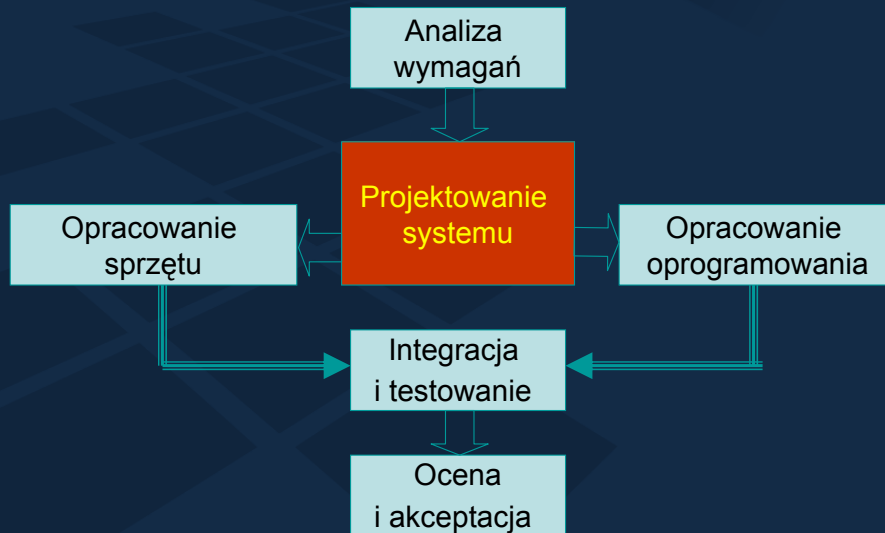


Metodyka projektowania komputerowych systemów sterowania

Andrzej URBANIAK



Projektowanie KSS



Metodyka projektowania KSS (2)

Projektowanie niezawodnych systemów sterujących opartych na mikroprocesorach jako urządzeniach zastępujących klasyczne regulatory wymaga systematycznego podejścia do procesu projektowania.

Etapy procesu powstawania projektu przedstawiono na rysunku. Charakterystyczną cechą procesu projektowania komputerowych systemów sterujących jest możliwość równoległego opracowania części sprzętowej i programowej.



Analiza wymagań

- Podstawowe funkcje
- Żądana dokładność
- Szybkość i niezawodność działania
- Ograniczenia dotyczące stosowanego sprzętu i oprogramowania
- Szacowane koszty projektu
- Spodziewane efekty wdrożenia

Dokument: *Specyfikacja wymagań*

Etap kończy opinia *recenzentów*

Metodyka projektowania KSS (3)

Pierwszym kluczowym etapem powstawania projektu jest analiza wymagań. Ważność tego etapu wynika z wagi podejmowanych decyzji odnośnie do:

- podstawowych realizowanych funkcji,
- żądaną dokładności,
- oczekiwanej szybkości i niezawodności działania,
- ograniczeń dotyczących stosowanego sprzętu i oprogramowania,
- szacowanych kosztów projektu,
- spodziewanych efektów wdrożenia.

Przejęcie do kolejnego etapu uwarunkowane jest pozytywną opinią recenzentów.



Projektowanie systemu

- Projektowanie systemu
 - Określenie architektury
 - Podział na podsystemy
 - Przypisanie funkcji poszczególnym podsystemom
 - Zdefiniowanie reguł współpracy podsystemów
 - Plan testów systemu

Etap kończy opinia *recenzentów*

Kolejny etap dotyczy ogólnego zaprojektowania systemu. Zestawienie zadań związanych z tym etapem obejmuje:

określenie architektury systemu, podział na podsystemy, przypisanie funkcji poszczególnym podsystemom, zdefiniowanie reguł współpracy podsystemów oraz plan testów systemu.



Projektowanie i implementacja podsystemów

- Współbieżna i niezależna realizacja sprzętu i oprogramowania podsystemów

- Wynikiem etapu jest zbiór programowych i sprzętowych komponentów systemu

Dokumentacja komponentów systemu

Etap kończy *REWIZJA*

- Funkcjonalna – czy wszystkie wymagania wymienione w specyfikacji są spełnione

- Fizyczna – czy komponenty są spójne i w pełni udokumentowane

Zgodnie z przyjętym i zaopiniowanym ogólnym projektem systemu w kolejnym etapie podjęta zostaje praca polegająca na realizacji sprzętu i oprogramowania. W zależności od charakteru projektu, część prac może być realizowana oddzielnie. Zwykle jednak występuje część zadań, które należy wykonać współbieżnie. W wyniku realizacji tego etapu powstają poszczególne komponenty sprzętowe i programowe systemu. Na zakończenie tego etapu dokonywana jest rozszerzona ocena polegająca na rewizji funkcjonalnej i fizycznej. Rewizja funkcjonalna dotyczy sprawdzenia, czy wszystkie wymagania zawarte w specyfikacji wymagań zostały spełnione. Rewizja fizyczna pozwala na sprawdzenie poszczególnych komponentów systemu: muszą one być spójne i powinny posiadać pełną dokumentację.



Integracja, testowanie, ocena

- Testowanie i integracja systemu
 - Połączenie wszystkich komponentów systemu i testowanie zgodnie z planem testów

- Przegląd kwalifikacyjny

Wynik: raport testowania

- Ocena i akceptacja

(etap realizowany przez użytkowników systemu)

Metodyka projektowania KSS (6)

Proces uruchomienia rozwiązania rozpoczyna się od testowania systemu w różnych konfiguracjach połączonych komponentów. Stosuje się w tym celu określone i ściśle zdefiniowane metodyki postępowania. Niezależnie od przyjętej metodyki postępowania ocenia się poprawność funkcjonalną działania całego systemu. Testowanie winno zakończyć się raportem wskazującym na elementy trudne oraz na możliwe słabe ogniwa systemu.

Ostateczna ocena i akceptacja przyjętego rozwiązania ma miejsce u użytkownika systemu.



Przykładowy szacunek kosztów

■Specyfikacja wymagań	18%
■Projekt wstępny	7%
■Projekt szczegółowy	15%
■Programowanie modułów	20%
■Integracja i testowanie komponentów	20%
■Instalacja i ocena	20%
■Koszty konserwacji	70% i więcej

Metodyka projektowania KSS (7)

W stosunku do klasycznego podejścia projektowego zmienia się rozkład kosztów na poszczególne etapy. W tradycyjnym projektowaniu przywiązywano największą wagę do projektu szczegółowego przy niewielkim udziale kosztowym etapów wstępnych (koncepcja i projekt wstępny). W przedstawionym powyżej rozkładzie kosztów widać wyraźnie równomierne ich rozłożenie na poszczególne etapy oraz wydatne zwiększenie nakładów na etap specyfikacji wymagań.

Zwróćmy również uwagę na fakt uwzględnienia w kosztach ogólnych istotnej części przeznaczanej na utrzymanie i konserwację systemu.



Cechy opisu wymagań wg IEEE Std 830, IEEE Std 1016

- **Jednoznaczność** – zapis każdego wymagania winien mieć *tylko jedna* interpretację
- **Kompletność** – specyfikacja wymienia *wszystkie* istotne wymagania, które powinny być spełnione
- **Weryfikowalność** – opisy wszystkich wymagań winny być tak sformułowane, aby było możliwe jednoznaczne *rozstrzygnięcie*, czy produkt finalny spełnia te wymagania czy nie
- **Spójność** – różne wymagania winny być opisane z użyciem *jednolitej terminologii* i nie mogą być wzajemnie sprzeczne
- **Modyfikowalność** – łatwość zmian (zależy od przejrzystości dokumentu: podział na rozdziały, spis treści, indeksy, odnośniki) i opis danego wymagania *tylko w jednym* miejscu
- **Powiązania** – uzasadnienie wymagania – przyczyna jego wprowadzenia (zalecenie użytkownika, normy jakości, przepisy)

Metodyka projektowania KSS (8)

Zaprojektowanie systemu sterującego wykorzystującego mikroprocesor jest niewątpliwie sztuką. Prawidłowo zaprojektowany i zrealizowany układ sterowania komputerowego jednak może nie znaleźć szerszego uznania, jeśli nie będzie spełniał określonych wymagań odnośnie do dokumentacji projektu.

Przykładowo podano poniżej zestaw istotnych cech opisu wymagań zgodny ze standardami IEEE Std 830 oraz Std 1016.

- **Jednoznaczność** – zapis każdego wymagania winien mieć *tylko jedna* interpretację
- **Kompletność** – specyfikacja wymienia *wszystkie* istotne wymagania, które powinny być spełnione
- **Weryfikowalność** – opisy wszystkich wymagań winny być tak sformułowane, aby było możliwe jednoznaczne *rozstrzygnięcie*, czy produkt finalny spełnia te wymagania czy nie
- **Spójność** – różne wymagania winny być opisane z użyciem *jednolitej terminologii* i nie mogą być wzajemnie sprzeczne
- **Modyfikowalność** – łatwość zmian (zależy od przejrzystości dokumentu: podział na rozdziały, spis treści, indeksy, odnośniki) i opis danego wymagania *tylko w jednym* miejscu
- **Powiązania** – uzasadnienie wymagania – przyczyna jego wprowadzenia (zalecenie użytkownika, normy jakości, przepisy)



Wzorcowy układ specyfikacji wymagań wg IEEE Std 830

1. Wstęp

- cel sporządzenia specyfikacji i spodziewany krąg czytelników
- zakres specyfikacji – lista nazw i funkcje jednostek oprogramowania
- definicje i skróty nazw
- dokumenty związane, do których występują odwołania i sposób dostępu do nich
- streszczenie poszczególnych części

Z tego samego powodu podano poniżej schemat wzorcowego układu dokumentu jakim jest specyfikacja wymagań (zgodnie ze standardem IEEE Std 830).

Powyższy schemat zawiera niezbędne elementy treściowe, które winny się znaleźć w dobrze przygotowanym opisie specyfikacji wymagań.

We wstępie specyfikacji winny się znaleźć następujące elementy:



Wzorcowy układ specyfikacji wymagań wg IEEE Std 830

2. Opis ogólny – intuicyjny opis wymagań i ograniczeń

- otoczenie oprogramowania (związki z innymi systemami)
- funkcje oprogramowania
- charakterystyka użytkowników
- ograniczenia narzucone na swobodę programowania
- przyjęte założenia

W drugim rozdziale pt. OPIS OGÓLNY należy opisać:

Rozdział 2 nie zawiera szczegółowych informacji o dużej złożoności. Powinien on być zrozumiały dla odbiorcy posiadającego ogólną wiedzę na temat projektowania systemów informatycznych.



Wzorcowy układ specyfikacji wymagań wg IEEE Std 830

3. Wymagania szczegółowe – najważniejszy rozdział

- wymagania na wydajność przetwarzania (np..liczba końcówek, dokładność obliczeń, czas reakcji)
- wymagania dotyczące sposobu komunikacji z użytkownikiem, komunikacji oprogramowania ze sprzętem, itp..
- ograniczenia projektowe w swobodzie programowania (prawne, zgodność ze standardami, konieczność współpracy z innymi systemami)
- cechy szczególne (organizacja bazy danych, zabezpieczenie danych, praca w warunkach awarii, śledzenie komunikacji z użytkownikiem, spodziewane kierunki modyfikacji)

4. Indeks – lista ważniejszych nazw i terminów

Metodyka projektowania KSS (11)

Najważniejszy rozdział 3 obejmuje szczegółowe wymagania, w pełni czytelne dla specjalisty i niezbędne do jednoznacznego ustalenia zakresu prac projektowych. Obejmuje on:

Opis specyfikacji wymagań kończy lista najważniejszych nazw i terminów. Obserwuje się tendencje do przyjmowania wielu skrótów nazw i terminów co może prowadzić do nieporozumień. Staranne sporządzenie tej listy gwarantuje pełną komunikatywność treści.



Projektowanie oprogramowania

- **Jednostki konfiguracyjne** – stanowią pewne samodzielne elementy, np.. oprogramowanie wbudowane przyrządu pomiarowego lub robota
- **Komponenty** – realizują dobrze zdefiniowane funkcje, możliwe do samodzielnego testowania
- **Moduły** – najmniejsze jednostki dekompozycji programu

Metodyka projektowania KSS (12)

Omówienie zasad projektowania uzupełnimy jeszcze o szersze uwagi odnośnie do projektowania oprogramowania. Problematyka projektowania oprogramowania podejmowana jest w obszarze inżynierii oprogramowania, poczynione w tym miejscu uwagi mają potwierdzić konieczność metodycznego podejścia do tego fragmentu projektowania oraz uświadomić, że nawet przy pełnej dostępności sprzętowej funkcjonalność systemu buduje oprogramowanie.

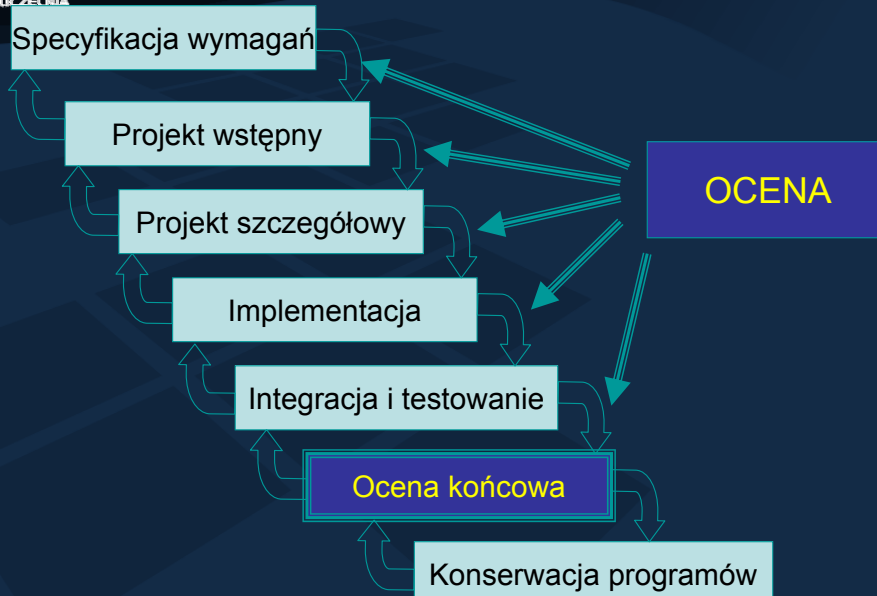
Sprawna realizacja oprogramowania wymaga jego podziału na określone ściśle elementy. Są to:

- **Jednostki konfiguracyjne**, które stanowią pewne samodzielne elementy, np.. oprogramowanie wbudowane przyrządu pomiarowego lub robota;
- **Komponenty**, które realizują dobrze zdefiniowane funkcje, możliwe do samodzielnego testowania;
- **Moduły** stanowiące najmniejsze możliwe jednostki dekompozycji programu.

Projektowanie oprogramowania ma swoją specyfikę, czego wyrazem jest m.in.. rozwijana intensywnie dziedzina pod nazwą inżynierii oprogramowania.



Projektowanie oprogramowania



Metodyka projektowania KSS (13)

Kaskadowy schemat procesu projektowania oprogramowania zakłada konieczność oceny wyników każdego etapu przed podjęciem następnego. Negatywna ocena powoduje powrót do poprzedniego etapu – jego weryfikację lub powtórzenie.

Zakres prac realizowanych w poszczególnych etapach zestawiono poniżej:

- **Analiza i specyfikacja systemu**
 - Wynik etapu: dokument określający funkcje realizowane przez oprogramowanie, powiązania z otoczeniem, dokładność, szybkość i wydajność obliczeń
- **Projekt wstępny**
 - Cel: dekompozycja na komponenty i przypisanie funkcji do poszczególnych komponentów i określenie sposobu ich testowania
 - Wynik etapu:
 - wstępna specyfikacja projektu oprogramowania
 - plan testów kwalifikacyjnych
- **Projekt szczegółowy**
 - Wynik etapu:
 - Specyfikacja projektu opisująca podział na moduły, funkcje i algorytmy działania modułów
 - Podstawowe struktury danych
 - Opisy testów kwalifikacyjnych
 - Opisy testów komponentów i modułów
- **Implementacja – programowanie i testowanie modułów**
 - Wynik etapu:
 - Kod źródłowy i listing programu
 - Instrukcje i raporty testowania modułów
 - Instrukcje testowania komponentów
- **Integracja i testowanie komponentów**
 - Wynik etapu:
 - uaktualniona specyfikacja projektu i kod źródłowy programu
 - raporty testowania komponentów
 - instrukcja testowania jednostki konfiguracyjnej
- **Instalacja i ocena końcowa**
 - Wynik etapu:



Projektowanie KSS – główne błędy

- Niedocenianie roli początkowych faz projektu
- Brak systematycznej oceny etapów prac
- Znikome korzystanie ze sprawdzonych metod inżynierii oprogramowania oraz oprogramowania typu CAD
- Nieadekwatny i mało wiarygodny proces testowania
- Niepełna dokumentacja

Metodyka projektowania KSS (14)

Proces projektowania komputerowych systemów sterujących jest bardzo złożony i nie zawsze kończy się sukcesem. Można wskazać kilka najbardziej elementarnych zasad, których przestrzeganie pozwoli uniknąć najczęściej spotykanych błędów. Należą do nich:

- Niedocenianie roli początkowych faz projektu
- Brak systematycznej oceny etapów prac
- Znikome korzystanie ze sprawdzonych metod inżynierii oprogramowania oraz oprogramowania typu CAD
- Nieadekwatny i mało wiarygodny proces testowania
- Niepełna dokumentacja

Komentarz do wyżej wymienionych błędów wydaje się być zbędny, gdyż nawet mniej doświadczony projektant potrafi wskazać przykłady potwierdzające nierzadko fatalne skutki takich błędów.