



Wykłady z inżynierii oprogramowania.  
Zarządzanie konfiguracją, zarządzanie zmianami.

**Zawartość**

1. Zarządzanie konfiguracją, Configuration Management.....	2
1.1 Standardy zarządzania konfiguracją .....	2
1.2 Podstawowe pojęcia zarządzania konfiguracją (CM).....	2
2. Zarządzanie konfiguracją i zmianami, Configuration & Change Management (C&CM).....	4
2.1 Definicja procesu zarządzania konfiguracją i zmianami .....	4
2.2 Proces zarządzania konfiguracją i zarządzania zmianami .....	4
2.4 System zarządzania konfiguracją.....	6
2.5 Wybór elementów konfiguracji (CI).....	7
3. System zarządzania wersjami .....	9
3.1 Struktura katalogowa systemu zarządzania wersjami.....	9
4. Zarządzanie zmianami .....	11
4.1 Żądanie zmiany .....	11
4.2 Organizacja spotkań CCB.....	12
4.3 Analiza ryzyka zarządzania zmianami.....	13
4.4 Zasady skutecznego zarządzania zmianami.....	13
4.5 Plan zarządzania konfiguracją (CM Plan) .....	15
4.6 Najlepsze praktyki CM .....	16
4.7 Zasady CM.....	17
5. Narzędzia zarządzania konfiguracją .....	17

## 1. Zarządzanie konfiguracją, Configuration Management

### 1.1 Standardy zarządzania konfiguracją

ANSI/IEEE Std 828, Standard for Software Configuration Management Plans.

ANSI/IEEE Std 1042, Guide to Software Configuration Management.

EIA Standard IS-649 National Consensus Std for Configuration Management, 1995.

MIL STD-480, Engineering Changes, Deviations and Waivers.

MIL-STD-483A, Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs.

MIL-STD-973, Configuration Management.

MIL-HDBK-61, Guidelines for Configuration Management,

[http://everyspec.com/MIL-HDBK/MIL-HDBK-0001-0099/MIL-HDBK-61A\\_15177/](http://everyspec.com/MIL-HDBK/MIL-HDBK-0001-0099/MIL-HDBK-61A_15177/)

### 1.2 Podstawowe pojęcia zarządzania konfiguracją (CM)

**Identyfikacja konfiguracji (configuration identification)** - wybór elementów, które będą podlegały procedurom konfiguracji.

**Elementy podlegające konfiguracji (Configuration Items), można podzielić na elementy**

- sprzętowe (hardware),
- programowe (software).

**Dla wybranych elementów podlegających konfiguracji CI należy określić**

- zasady dokumentowania zmian w konfiguracji,
- zbiór informacji 'baseline' i jego konfiguracji.

**Computer Software Configuration Item (CSCI)** - programowe (softwarowe) elementy wybrane w procesie *identyfikacji konfiguracji* (configuration identification) podlegające zarządzaniu konfiguracją (CM).

**Baseline** - jest to zbiór informacji projektu o określonym numerze wersji.

Zbiór informacji 'baseline' stanowi pełną informację o projekcie, tzn. informacja o wszystkich CSCI w określonej chwili czasu.

Wszelkie zmiany w treści informacji są rejestrowane jako zmiany w stosunku do zbioru informacji 'baseline'.

Co pewien czas zbiór informacji 'baseline' z zarejestrowanymi zmianami jest definiowany jako nowa wersja zbioru informacji 'baseline'.

**Elementy zbioru informacji 'baseline'**

- zbiór wymagań,
- projekt edycji systemu,
- konfiguracja systemu (produktu).

**Kontrola konfiguracji (configuration control)**

planowa i systematyczna kontrola, ocena, koordynacja i akceptacja zmian w konfiguracji.

Kontrola dokonywanych zmian w konfiguracji odbywa się zgodnie z przyjętymi procedurami w oparciu o zdefiniowany zbiór informacji 'baseline'.

### **Audyt konfiguracji (configuration audit)**

formalna kontrola elementów konfiguracji oprogramowania i dokumentacji.

#### **Rozróżniane są dwa typy audytu**

- fizyczny audyt konfiguracji (Physical Configuration Audit - PCA)
- funkcjonalny audyt konfiguracji (Functional Configuration Audit - FCA).

***Audyt fizyczny*** - ma na celu sprawdzenie czy stan elementów konfiguracji (CI) jest zgodny z danymi zawartymi w dokumentacji.

***Audyt funkcjonalny*** - ma na celu sprawdzenie czy elementy konfiguracji (CI) spełniają funkcje określone w specyfikacji.

***Rejestracja stanu konfiguracji (Configuration status accounting)*** - rejestracja i generowanie raportów o stanie konfiguracji.

#### **Rejestrowane i raportowane są**

- proponowane zmiany w konfiguracji,
- zaakceptowane zmiany w konfiguracji,
- stan realizacji zaakceptowanych zmian w konfiguracji,
- bieżący stan konfiguracji 'baseline',
- historia zmian w konfiguracji,
- wyniki kontroli konfiguracji (configuration audit).

***Żądanie zmian (Change request)*** - formalne wystosowanie żądanie zmiany w funkcjonalności systemu.

Żądanie zmian mogą wystosowywać klienci, użytkownicy, analitycy, programiści, testerzy proponując

- zmiany w istniejącej funkcjonalności systemu,
- dodanie/usunięcie funkcjonalności,
- poprawieniu działania systemu,
- usunięciu znalezionych błędów.

### ***Configuration Control Board (CCB)***

grupa osób upoważnionych do definiowania i kontroli zmian w funkcjonalności systemu.

W skład CCB mogą wchodzić przedstawiciele klientów i użytkowników, analitycy, testerzy.

### ***Engineering Review Board (ERB)***

grupa osób upoważnionych do definiowania i kontroli żądań związanych z procesem analizy i projektowania, implementacji i testów systemu.

#### **Rola ERB**

- definiowanie zadań i ich ocena w procesie analizy i projektowania, implementacji i testów systemu,
- ocena proponowanych zmian,
- definiowanie nowych zmian,
- określanie kryteriów akceptowalności zmian,
- określanie kryteriów zarządzania zmianami.

## 2. Zarządzanie konfiguracją i zmianami, Configuration & Change Management (C&CM)

### Element zarządzania projektem informatycznym polegający na

- identyfikowaniu i dokumentowaniu elementów konfiguracji tworzonego oprogramowania,
- kontroli zmian w elementach konfiguracji tworzonego oprogramowania i związanym z tymi elementami dokumentów,
- zbieraniu i udostępnianiu informacji niezbędnych do zarządzania elementami konfiguracji, włączając określanie statusu proponowanych i implementowanych zmian w konfiguracji,
- kontroli konfiguracji, tzn. kontroli zgodności konfiguracji z odpowiednimi specyfikacjami.

### 2.1 Definicja procesu zarządzania konfiguracją i zmianami

#### Definicja (MIL-STD-973) Configuration Mangement

(\*) Identyfikowanie i dokumentowanie funkcjonalnych i fizycznych charakterystyk elementów konfiguracji (Configuration Items).

(\*) Kontrola zmian w konfiguracji i dokumentach konfiguracji.

(\*) Rejestracja informacji niezbędnych do zarządzania konfiguracją.

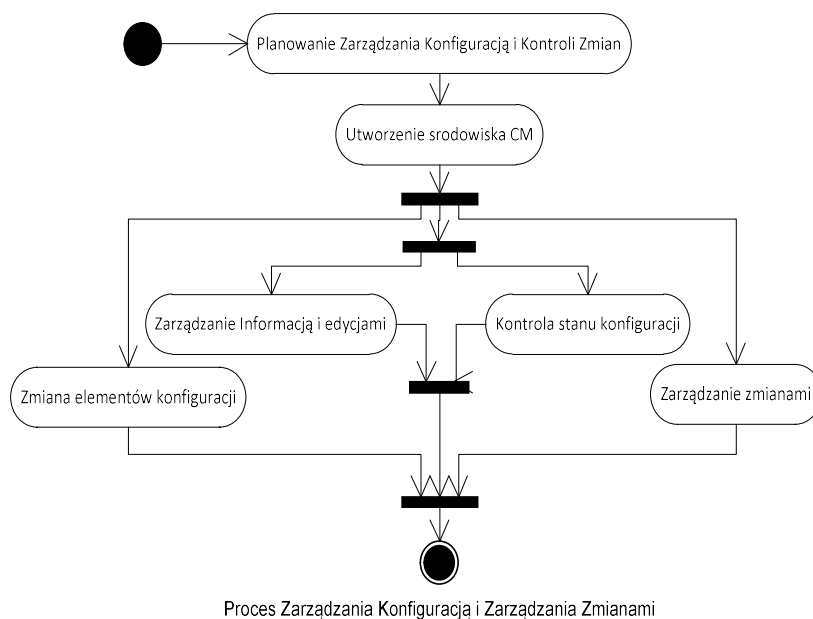
(\*) Kontrola zgodności stanu konfiguracji z dokumentacją konfiguracji.

#### Definicja (ANSI/IEA 649-1998) Configuration Mangement

Proces zarządzania polegający na definiowaniu i kontroli spójności fizycznych i funkcjonalnych charakterystyk systemu z charakterystykami określonymi w fazie projektowania.

Kontrola spójności między definicją produktu, konfiguracją produktu (fizyczna konfiguracja) i danymi zawartymi w dokumentacji konfiguracji produktu.

### 2.2 Proces zarządzania konfiguracją i zarządzania zmianami



## 2.3 Cele zarządzania konfiguracją i zarządzania zmianami

### Kontrola konfiguracji dotyczy

- kodów źródłowych tworzonego oprogramowania,
- plików binarnych tworzonego oprogramowania,
- danych na których operują tworzone programy,
- dokumentacji tworzonego oprogramowania (specyfikacji, instrukcji instalacyjnych i konfiguracyjnych, dokumentacji użytkownika),
- dokumentacji związanej z prowadzeniem projektu (planów, raportów, notatek ze spotkań).

### Zarządzanie konfiguracją określa

- **zasady identyfikacji konfiguracji**  
Określenie kto ma dostęp do jakich informacji w projekcie (określenie współdostępu do informacji).  
Określenie kto jest właścicielem poszczególnych informacji.  
Określenie jakie informacje podlegają procesowi zarządzania konfiguracją.  
Określenie zasad identyfikacji aktualnych informacji.
- **zasady kontroli konfiguracji**  
Określenie poziomów kontroli przez jakie muszą przejść dane podlegające procesowi kontroli konfiguracji.  
Określenie osób mających prawo do dokonywania zmian na każdym poziomie kontroli konfiguracji (określenie praw dostępu).  
Określenie osób mających prawo do akceptowania zmian (określenie praw do autoryzacji zmian).  
Określenie procedur dokonywania zmian w danych (procedur żądania zmian, procedur autoryzacji zmian, procedur kontroli zmian, procedur kontroli wersji).  
Określenie procedur informowania o dokonanych zmianach.
- **zasady rejestracji zmian w konfiguracji i generowania raportów o dokonanych zmianach**  
Określenie procedur rejestracji dokonywanych zmian w konfiguracji w dokumentach CM.  
Określenie zasad przypisywania statusu dokonywanym zmianom (żądanie zmiany, zmiana zaakceptowana, zmiana zaimplementowana).  
Określenie procedur generowania raportów z dokumentów CM.  
Określenie zasad dystrybucji raportów.
- **zasady ochrony danych o konfiguracji**  
Określenie zasad archiwizowania i odtwarzania z informacji podlegającej kontroli konfiguracji.  
Określenie zasad archiwizowania i odtwarzania działającego systemu w środowisku programistycznym, testowym.
- **zasady oceny jakości kontroli konfiguracji**  
Określenie zasad oceny zgodności konfiguracji z danymi zawartymi w dokumentach CM.  
Określenie procedur ulepszania procesu zarządzania konfiguracją.  
Określenie metryk konfiguracji (CM metrics) służących do kontroli jakości tworzonego oprogramowania.

### Elementami środowiska zarządzania konfiguracją i zmianami są

- System zarządzania konfiguracją (np. ClearCase).
- System zarządzania zmianami (rejestracja zmian i ich statusów, śledzenie historii zmian).
- System zarządzania wersjami (np. CVS, Source Save).
- Repozytorium projektu (Software Development Library).
- Środowisko do kompilowania kodów źródłowych projektu.

## 2.4 System zarządzania konfiguracją

### System zarządzania konfiguracją służy do

- przechowywania informacji powstających w procesie tworzenia oprogramowania,
- wspierania procesów instalacji i utrzymania systemu.

### System zarządzania konfiguracją pozwala rozwiązać problemy związane z

- jednoczesnym uaktualnianiem informacji tworzonych w procesie tworzenia oprogramowania,
- udostępnianiem i rozpowszechnianiem informacji o zmianach w konfiguracji,
- tworzeniem wielu wersji (drzewa wersji) elementów tworzonego oprogramowania.

Zbiór informacji 'baseline' jest to zbiór informacji projektu określonym numerze wersji.

### Przyczyny tworzenia zbioru informacji 'baseline'

- możliwość powrotu do poprzednich wersji tworzonego systemu, dokumentacji i innych informacji o projekcie (reproducibility),
- łatwa możliwość określenia historii zmian, sekwencji zmian wersji (traceability),
- możliwość śledzenia i porównywania zmian w poszczególnych zbiorach informacji 'baseline'.

### Zalety tworzenia zbioru informacji 'baseline'

- zbiór informacji 'baseline' stanowi punkt odniesienia do oficjalnych informacji w projekcie,
- 'baseline' może być punktem wyjścia do tworzenia nowego projektu lub pod-projektu,
- istnienie historii zmian o systemie pozwala na powrót się do poprzednich wersji systemu z pełnym zbiorem informacji 'baseline' aktualnych w danej wersji,
- okresowe definiowanie nowego zbioru 'baseline' pozwala na dokładniejszą kontrolę treści informacji o projekcie.

### Tworzenie nowego zbioru informacji 'baseline':

- po osiągnięciu określonego celu w projekcie (milestone),
- po zamknięciu określonej fazy w procesie wytwarzania oprogramowania,
- po dokonaniu przeglądu i oceny (kontroli) informacji 'baseline'.

Tworzenie nowego zbioru informacji 'baseline' może zależeć od przyjętego modelu wytwarzania oprogramowania.

### Przykład.

W modelu 'wodospadu' tworzenie nowego zbioru 'baseline' może następować po każdym etapie wytwarzania oprogramowania

- zbiór 'baseline' fazy wymagań (requirements baseline),
- zbiór 'baseline' fazy projektowania (design baseline),
- zbiór 'baseline' fazy implementacji (code baseline),
- zbiór 'baseline' produktu (product baseline),
- zbiór 'baseline' edycji (software accepted baseline).

### **Przykład zbioru 'baseline'.**

Początkowa wersja zbioru 'baseline' (initial 'baseline'):

procedury projektu, plan zarządzania projektem, specyfikacja wymagań, kontrakt, plan wytwarzania oprogramowania, plan CM, plan QA, szablony dokumentów.

Kolejne wersje zbioru 'baseline':

initial 'baseline', dokumentacja fazy analizy i projektowania, kody źródłowe, pliki binarne systemu, pliki instalacyjne edycji systemu, plan testów, skrypty testowe, plan instalacji, dokumentacja użytkownika, release notes.

## **2.5 Wybór elementów konfiguracji (CI)**

Elementy konfiguracji (CI) dzielone są na zbiory charakteryzujące części systemu niezależnie projektowane, implementowane, testowane, dokumentowane.

### **Kryteria tworzenia zbiorów elementów konfiguracji (CI)**

- podział na podstawie wagi CI (krytyczne elementy, ... ),
- podział według złożoności CI,
- podział według obszarów funkcjonalnych,
- podział według częstości zmian dokonywanych w CI,
- podział CI według platformy systemowej.

### **Każdy element konfiguracji (CI) powinien**

- posiadać unikalny identyfikator (opisowa nazwa CI),
- posiadać dokładny opis,
- być łatwo identyfikowany w systemie i dokumentacji,
- posiadać procedurę kontroli zmian.

### **Zasady skutecznego zarządzania konfiguracją**

- Planowanie i dokumentowanie
  - zadań stawianych przed zarządzaniem konfiguracją,
  - działań związanych z zarządzaniem konfiguracją,
  - zakresu odpowiedzialności osób realizujących zadania zarządzania konfiguracji.
- Zdefiniowanie ogólnego procesu wytwarzania oprogramowania i identyfikacja miejsc w których będzie kontrolowana jakość i konfiguracja produktu.  
Określenie jednolitego formatu i zasad prezentowania danych i informacji w dokumentacji, kodzie.
- Dokonywanie przeglądu i wyboru narzędzi które mogą wspierać procesy wytwarzania oprogramowania i zarządzania konfiguracją.
- Określenie procedur zarządzania konfiguracją  
Określenie zasad administrowania biblioteką projektu (Software Development Library) służącej do kontroli zmian konfiguracji.  
Zarządzanie informacjami 'baseline'.  
Określenie zasad generowania raportów o konfiguracji (komu, kiedy, jaki zakres informacji).

## Inżynieria oprogramowania

- Określenie procedur kontroli konfiguracji.  
Procedury kontroli powinny obejmować zasady wyznaczania struktury produktu, wyboru elementów konfiguracji, dokumentowania elementów konfiguracji.
- Określenie procedur przygotowania edycji produktu (systemu).  
Zasady generowania plików binarnych (kompilacji kodu źródłowego).  
Zasady generowania dokumentacji użytkownika.
- Określenie procedur archiwizacji pakietów i dokumentacji.  
Określenie zasad archiwizacji (backup) i odtwarzania z archiwum (restore/recovery) pakietów instalacyjnych i dokumentacji w środowisku developerskim, testowym, produkcyjnym (u klienta).



### 3. System zarządzania wersjami

**Zbiory informacji projektu mogą być pogrupowane w systemie zarządzania wersjami według kategorii**

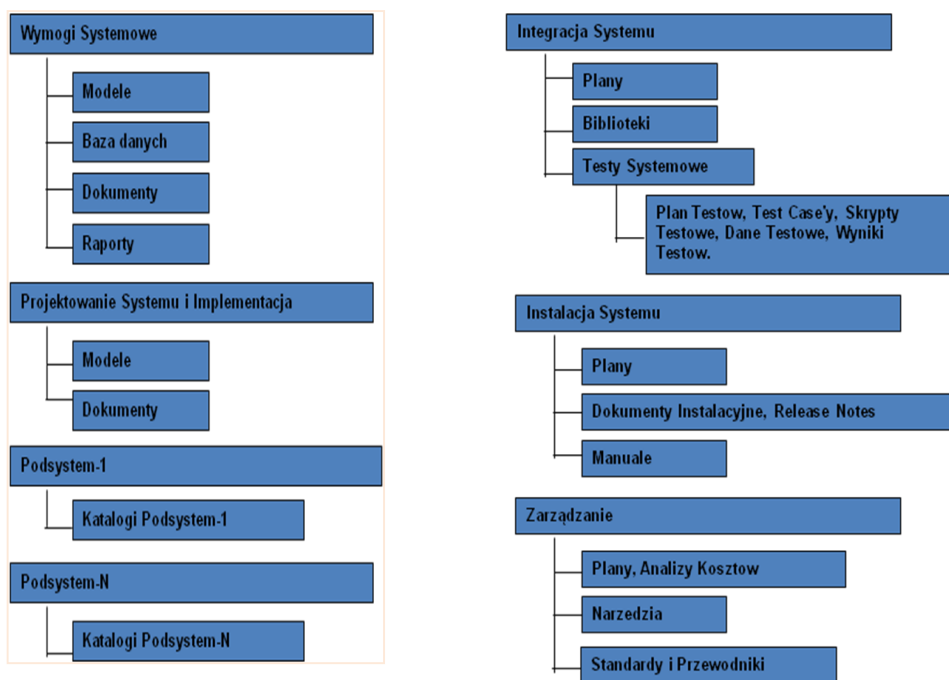
- wymagania (requirements),
- analiza funkcjonalna (business modeling),
- analiza Systemowa i Projektowanie (analysis & design),
- implementacje (implementation),
- testy (tests),
- instalacje (deployment),
- konfiguracja (Configuration & Change Management),
- zarządzanie projektem (Project Management).

#### 3.1 Struktura katalogowa systemu zarządzania wersjami

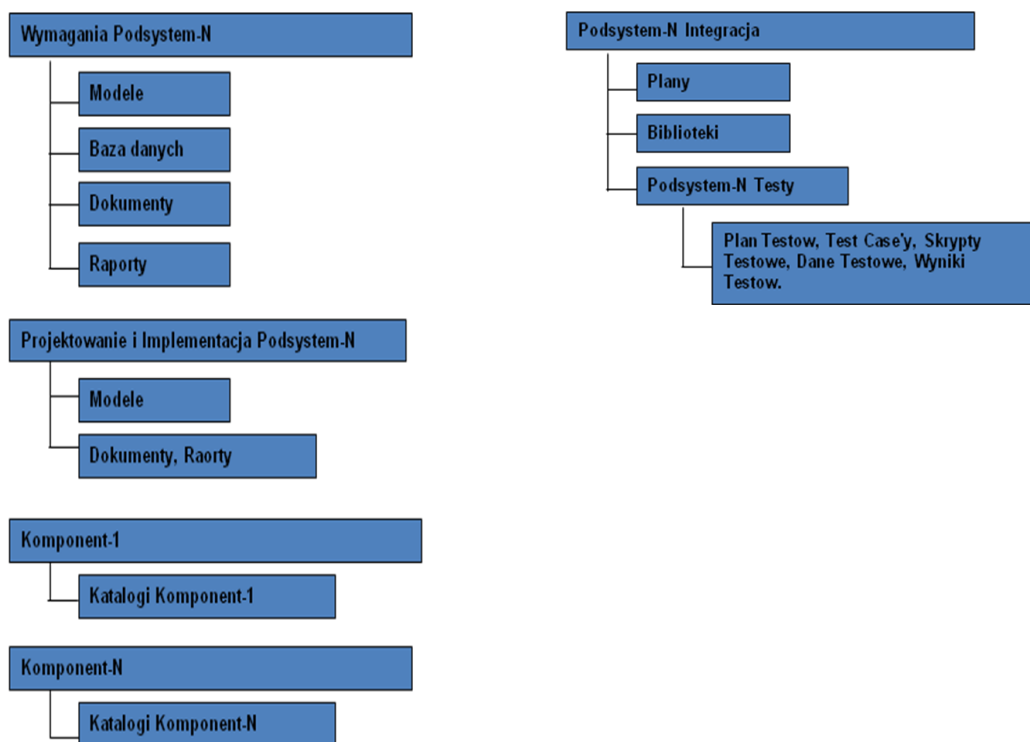
**Struktura katalogowa Systemu Zarządzania Wersjami** dla tworzonego oprogramowania (Product Directory Structure) może być podzielona na poziomy:

- struktura katalogowa systemu (System Product Directory Structure),
- struktura katalogowa podsystemu (Subsystem Directory Structure),
- struktura katalogowa komponentów (Component Directory Structure).

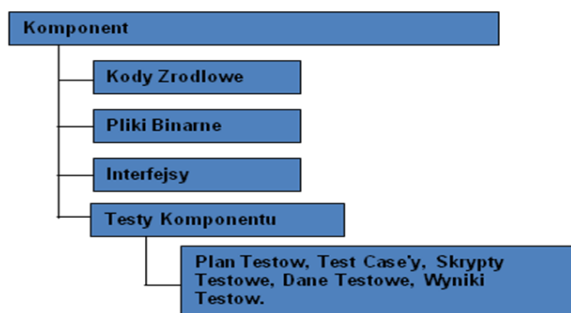
Struktura katalogowa - poziom systemu.



Struktura katalogowa - poziom podsystemu.



Struktura katalogowa - poziom komponentów.



#### 4. Zarządzanie zmianami

**Zarządzanie zmianami, *Change Request Management*** - proces akceptowania i kontroli zmian w funkcjonalności tworzonego systemu.

***Configuration Control Board (CCB)*** - grupa osób (klienci, użytkownicy, programiści, analitycy, testerzy) upoważnionych do definiowania i kontroli zmian w funkcjonalności systemu.

**Żądanie zmiany (*Change Request*)** - formalnie wystosowane żądanie zmiany w funkcjonalności systemu.

##### Cele procesu zarządzania zmianami

- określenie procedur akceptowania/odrzućania proponowanych zmian w tworzonym systemie,
- określenie procedur dokumentowania proponowanych zmian,
- określenie procedur kontroli statusów zmian,
- określenie procedur identyfikacji historii zmian w tworzonym systemie.

##### Proces zarządzania zmianami ma zapewnić, że proponowane zmiany

- podlegają procedurze kwalifikacji do realizacji,
- podlegają procedurze oceny. Określenie: wagi, priorytetu, kosztu zamiany,
- są dokumentowane,
- są kontrolowane w procesie implementacji,
- są testowane,
- podlegają procedurze akceptacji,
- są uwzględniane z nowej wersji zbioru 'baseline'.

#### 4.1 Żądanie zmiany

**Formularz żądania zmiany, *Change Request Submit Form*** - formularz na którym zgłaszane są żądania zmian.

##### Elementy formularza 'Żądania zmiany'

- kto żąda zmiany (submitting organization)?
- numer żądania (tracking no.),
- osoba do kontaktu (contact person), Nr. telefonu, adres e-mail (telephone, mailing address),
- data żądania zmiany (date),
- nazwa zmiany (title),
- wersja systemu której zmiana dotyczy (change location tag),
- powód zmiany (reason for change),
- opis proponowanej zmiany (proposed change).

##### Cechy 'żądania zmiany'

- priorytet (priority),
- waga (severity),
- status,
- poziom ryzyka.

##### Priorytety (priority) nadawane 'żądaniu zmiany'

- niski (low)
- średni (medium)
- wysoki (high)
- pilny (urgent).

**Wagi (severity) nadawane 'żądaniu zmiany'**

- mała zmiana (enhancement),
- średnia zmiana,
- poważna zmiana.

**Wagi określone są na podstawie ilości zasobów potrzebnych na implementację zmiany**

- czas implementacji,
- liczbę programistów,
- ilość i czas trwania testów.

**Statusy nadawane 'żądaniu zmiany'**

- dostarczony (submitted),
- otwarty (open),
- przydzielony (assigned),
- przetestowany (tested).
- zamknięty (closed)
- do wyjaśnienia (to clarify)
- odrzucony (rejected).

## **4.2 Organizacja spotkań CCB**

**CCB review meeting** - spotkania CCB są organizowane w celu przeglądu żądań zmian w funkcjonalności systemu.

**Rola CCB jest:**

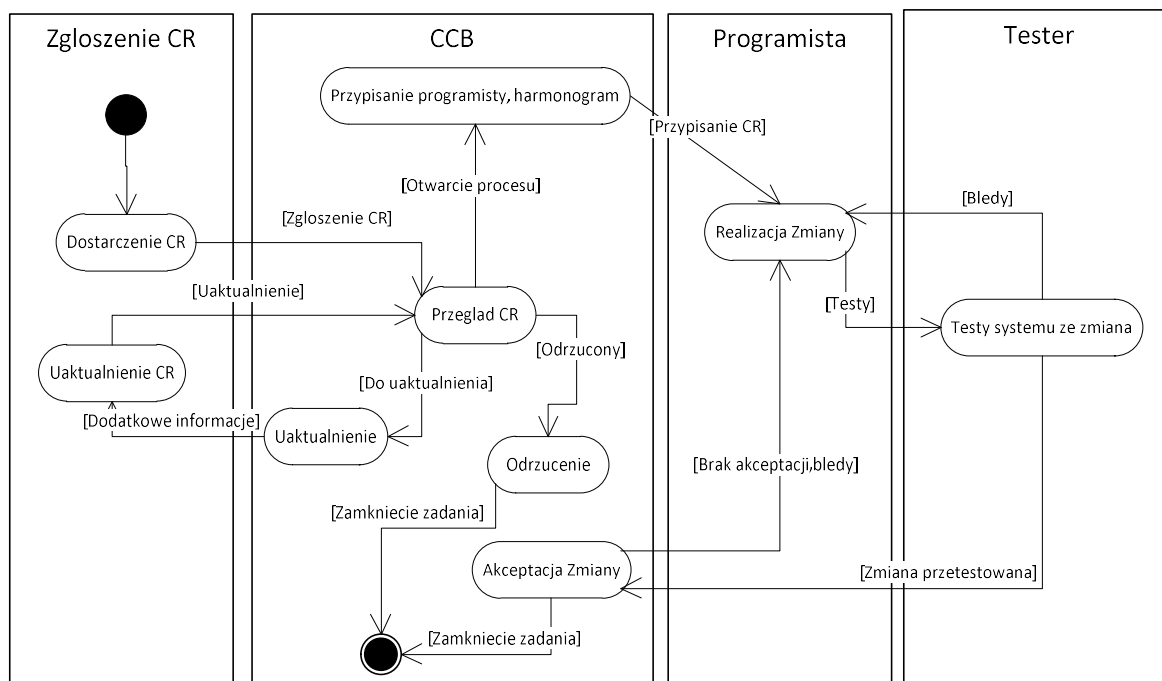
- ocena czy zadanie zmiany może być zrealizowane.
- określenie wersji lub edycji (release) w której zadanie zostanie zrealizowane.
- nadanie żądaniu zmiany: priorytetu, wagi, statusu.
- określić: zasoby potrzebne do realizacji zadania, harmonogram realizacji, poziom ryzyka.

### 4.3 Analiza ryzyka zarządzania zmianami

#### Poziomy ryzyka nadawane 'Żądaniu zmiany'

- niskie ryzyko (ryzyko akceptowalne),
- wysokie ryzyko,
- ryzyko nieakceptowalne.

Przykład procesu Zarządzania Zmianami.



Przykład Procesu Zarządzania Zmianami

### 4.4 Zasady skutecznego zarządzania zmianami

#### Zasady skutecznego zarządzania zmianami:

- Określić procesy zarządzania zmianami i ich role w CM.
- Określić procedury zarządzania zmianami.
- Zbudować środowisko zarządzania zmianami.
- Powołać i określić role CCB (Configuration Control Board).
- Powołać i określić role ERB (Engineering Review Board).

#### Kierownik Zarządzania Konfiguracją (Configuration Manager).

##### Do zadań kierownika Zarządzania Konfiguracją należy

- określenie procedur zarządzania konfiguracją,
- identyfikacja zadań CM w projekcie,
- utworzenie środowiska służącego do zarządzania konfiguracją,
- dokonywanie kontroli konfiguracji zgodnie z przyjętymi procedurami,
- tworzenia raportów z przeprowadzanych kontroli,
- definiowanie i zarządzanie zbiorem informacji 'baseline',
- tworzenie planów zarządzania konfiguracją,
- udoskonalanie procedur zarządzania konfiguracją,
- zarządzanie biblioteką projektu,
- tworzenie edycji (określanie konfiguracji do danej edycji, kompilowanie kodu edycji).

**Lista czynności kierownika Zarządzania Konfiguracją**

- przygotowanie pakietów instalacyjnych do testów, dla klienta (kompilowanie kodów źródłowych),
- Określania struktury tagów, nadawanie nowych tagów (np. dla nowej wersji 'baseline', nowej edycji, nowego projektu),
- zarządzanie systemem kontroli wersji (nadawanie uprawnień użytkownikom),
- administrowanie system zarządzania zmianami.

**Kierownik Zarządzania Zmianami (Change Control Manager).**

**Do zadań osoby odpowiedzialnej za Zarządzanie Zmianami należy**

- określenie procedur zadania zmian (Change Request),
- określenie procedur zarządzania zmianami,
- zarządzanie zmianami w konfiguracji,
- kontrola procesu zmian,
- współtworzenie planów Zarządzania Konfiguracją w zakresie zarządzania zmianami.

#### 4.5 Plan zarządzania konfiguracją (CM Plan)

**Plan zarządzania konfiguracją** - określa procedury i metody jakie będą stosowane w danym projekcie do zarządzania konfiguracją (CI).

**Elementy planu zarządzania konfiguracją (Standard ISO 10007).**

1. Zakres dokumentu
2. Referencje
3. Terminologia dokumentu
4. System zarządzania konfiguracją
5. Proces zarządzania konfiguracją
  - 5.1 Identyfikacja konfiguracji (CI)
    - 5.1.1 Struktura produktu.
    - 5.1.2 Wybór elementów konfiguracji (CSCI)
    - 5.1.3 Dokumentowanie elementów konfiguracji
    - 5.1.4 Wersje
    - 5.1.5 Definicja zbioru baseline
  - 5.2 Kontrola konfiguracji
  - 5.3 Rejestracja stanu konfiguracji
  - 5.4 Audyt konfiguracji
    - 5.4.1 Audyt funkcjonalny
    - 5.4.2 Audyt fizyczny
6. Organizacja procesu zarządzania konfiguracją
7. Procedury zarządzania konfiguracją
  - 7.1 Procedura identyfikacji konfiguracji
  - 7.2 CCB
  - 7.3 Procedura kontroli konfiguracji
  - 7.4 Procedura rejestracji stanu konfiguracji
  - 7.5 Procedura audytu konfiguracji
  - 7.6 Harmonogram zarządzania konfiguracją
8. Kontrola procesu zarządzania konfiguracją

## Elementy planu CM

### 1. Wstęp

*[Zawiera cele, zakres stosowania CM w danym projekcie.]*

### 2. Organizacja dokumentu

*[Zawiera opis roli CM w projekcie, opis relacji między procesem CM a innymi procesami (Zarządzania projektem, QA, procesów wytwarzania oprogramowania), relacji między planem CM a innymi planami projektu.]*

### 3. Harmonogram projektu

*[Zawiera opis faz projektu, celów pośrednich (milestones) projektu w odniesieniu do celów CM. Zawiera zasady określania i zarządzania zbiorem informacji 'baseline'.]*

### 4. Identyfikacja konfiguracji

*[Zawiera opis elementów tworzonego oprogramowania podlegającego zarządzaniu konfiguracją, określa strukturę biblioteki projektu, określa strukturę identyfikatorów konfiguracji (tagów), zasady nazewnictwa i numerowania wersji.]*

### 5. Zarządzanie interfejsami

*[Interface Management-Describes the procedures for identification of interface requirements, establishment of both internal and external interface agreement processes and procedures, and how the project will participate in the interface control working groups.]*

### 6. Zasady rejestracji statusów konfiguracji

*[Określa zasady jak i kiedy będą rejestrowane informacje o stanie konfiguracji. W jaki sposób informacje o stanie konfiguracji będą udostępniane. Informacje o stanie konfiguracji zawierają: informacje o stanie edycji.]*

### 7. Zasady kontroli konfiguracji

*[Zawiera plan i procedury przeprowadzania kontroli (audytu) procesów zarządzania i tworzonego oprogramowania.]*

### 8. Zarządzanie podwykonawcami

*[Zawiera zasady zarządzania podwykonawcami i kontroli zgodności ich procedur z procedurami projektu.]*

## 4.6 Najlepsze praktyki CM

### Best Practice #1:

Make Configuration Management Everyone's Job.

### Best Practice #2:

Create an Environment and Engineering Process That Enables Configuration Management.

### Best Practice #3:

Define and Document the CM/Engineering Process, Then Select the Tool Set (Automation) to Support the Process.

### Best Practice #4:

The CM Staffing Should Consist of Individuals with Technical Expertise to Support the Development and Maintenance of the Product.

### Best Practice #5:

The CM Plan and Procedures Need to Be Developed and Documented in the Same Way a Software Development Plan Is Created at the Initial Stages of a Project.



#### **4.7 Zasady CM**

##### CM Rules

###### RULE 1:

CM Must Manage the Ownership of Information and Bring Work Products under Control in a Timely Manner.

###### RULE 2:

Early Identification and Change Control of Artifacts and Work Products Is Integral to a Project.

###### RULE 3:

The Change Management Process Must Be Simple, Consistent with the Culture and Requirements of the Project, and Supported by the Methods and Tools of the Project.

###### RULE 4:

The Actions, Roles, and Responsibilities of the Engineering Review Board (ERB) and the Change Control Board (CCB) Should Be Documented.

###### RULE 5:

Change Control Must Be a Primary Focus of the Program and Must Be Integrated into the Culture of the Project.

###### RULE 6:

All Information That Is Placed under CM Control or Promoted in the CM Hierarchy Must Be Promoted Based on Successful Completion of a Quality Gate.

###### RULE 7:

All Proposed Changes to CM-Controlled Products Should Be Properly Classified.

###### RULE 8:

All CM Library Structures or Releases from the CM Library Must Be Recorded in a Software Version Description (SVD) or Comparable Document.

###### RULE 9:

It Is Essential That CM Be Continually Aware of the Exact Status of Products under Control, the Relationships between these Products, Problems, and Open Issues, and Who Is Using the Products.

###### RULE 10:

The Worst Way to Establish a CM System Is to Purchase a Tool Set and then Devise a Process to Fit the Tools. Establish the CM Process That Fits Your Project and Culture, then Find a Tool Set That Supports That Process.

#### **5. Narzędzia zarządzania konfiguracją**

ClearCase,

<http://www-03.ibm.com/software/products/pl/clearcase>

SoftBench,

HP, <http://h20565.www2.hp.com/portal/site/hpsc/>

Visual Source Safe,

Microsoft Corp., [http://msdn.microsoft.com/en-us/library/3h0544kx\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/3h0544kx(v=vs.80).aspx)