

# Pakeitimų valdymas

# Konfigūracijos valdymas

SCM - Software configuration management.

SCM - rinkinys veiklų, skirtų valdyti pokyčius projekto gyvavimo ciklo metu. Į SCM galima žvelgti kaip į programinės įrangos kokybės užtikrinimo veiklą.

Programinės įrangos konfigūracijos valdymo  
uždaviniai:

# Programinės įrangos konfigūracijos valdymo uždaviniai:

1. Nustatyti pokyčius.
2. Kontroliuoti pokyčius.
3. Užtikrinti pokyčių tinkamą įgyvendinimą.
4. Informuoti apie pokyčius suinteresuotus asmenis.

# Programų sistemų kūrimo produktai

1. Programinė įranga (išeities tekstai ir vykdomieji failai).
2. Programinę įrangą aprašantys dokumentai.
3. Duomenys naudojami programinės įrangos.

# Programinēs ierangos konfigurācijas elementai

SCI - software configuration item.

SCI - programinēs ierangos kūrimo metu sūkurtoš informācijas elementas (pvz. UML diagrama arba pilnas architektūros dokumentas), kuriam yra priskirtas pavadinimas.

Hierarchinė struktūra - vienas SCI elementas gali būtī kito dalis.

# Pagrindiniai pokyčių šaltiniai

“Nesvarbu kuriame sistemos kūrimo etape esame, sistema keisis, ir noras ją keisti išliks visuose sistemos kūrimo ciklo etapuose” [Ber80]

[Ber80] Bersoff, E., V. Henderson, and S. Siegel, Software Configuration Management ,Prentice Hall, 1980.

# Pagrindiniai pokyčių šaltiniai

- Naujos verslo ar rinkos sąlygos.
- Nauji suinteresuotos šalies poreikiai reikalauja pakeisti produktų funkcionalumą.
- Reorganizavimas ar verslo augimas/mažėjimas keičia projekto prioritetus ar komandos struktūrą.
- Biudžeto ar laiko apribojimai reikalauja iš naujo apibrėžti kuriamą sistemą.



# SCM dalyviai

- Projekto vadovas.  
Kuria ataskaitas apie sistemos būklę.
- SCM koordinatorius.  
Siekia užtikrinti, kad laikomasi procedūrų skirtų kurti, keisti ir testuoti produktą.
- Programinės įrangos inžinieriai.  
Siekia dirbti efektyviai, komunikuoja apie atliktus pakeitimus.
- Klientas.  
Naudojasi produktu ir laikosi formalių procedūrų pranešti apie rastas klaidas ir pakeitimų prašymus.

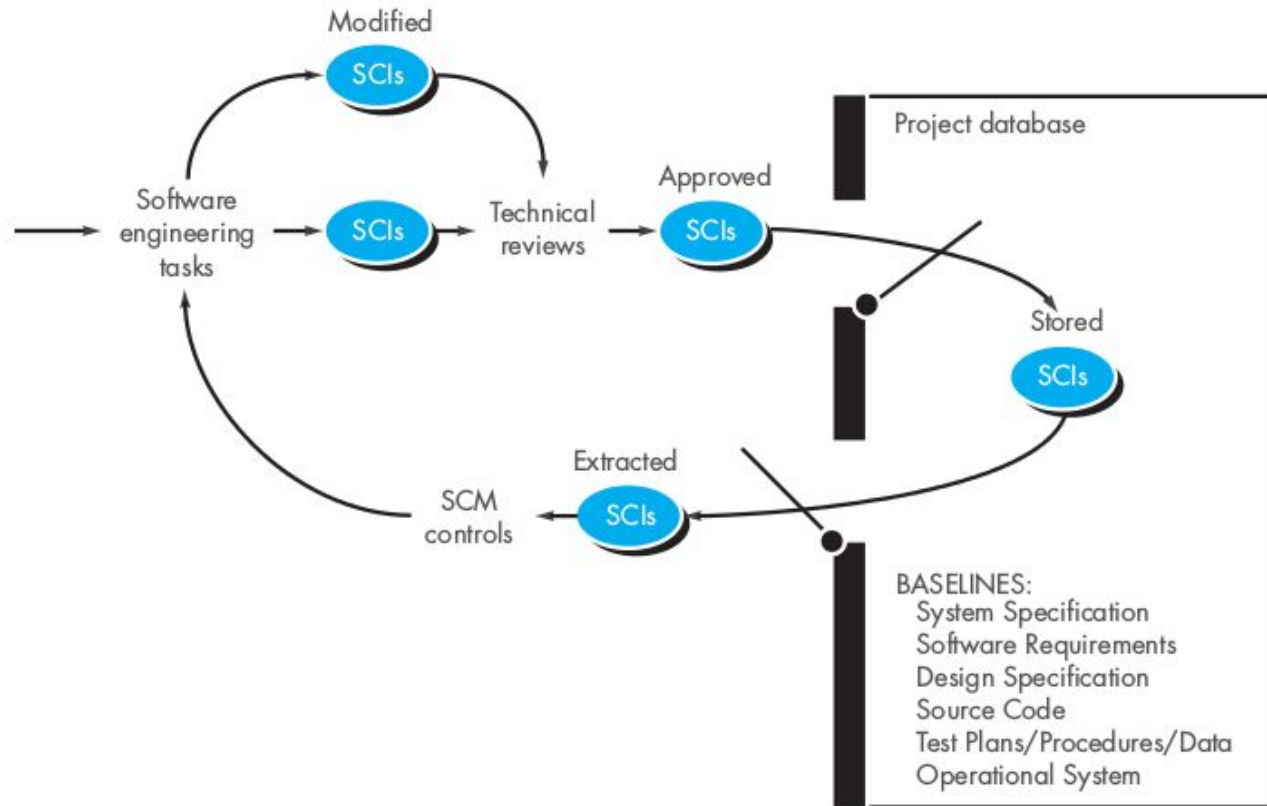
# Bazinis elementas

Specifikacija ar produktas, kuriam buvo atlikta formali peržiūra ir dėl kurio susitarta, kuris vėliau tarnauja kaip bazė tolimesniam vystymui ir kuris gali būti keičiamas tik naudojant formalias procedūras.

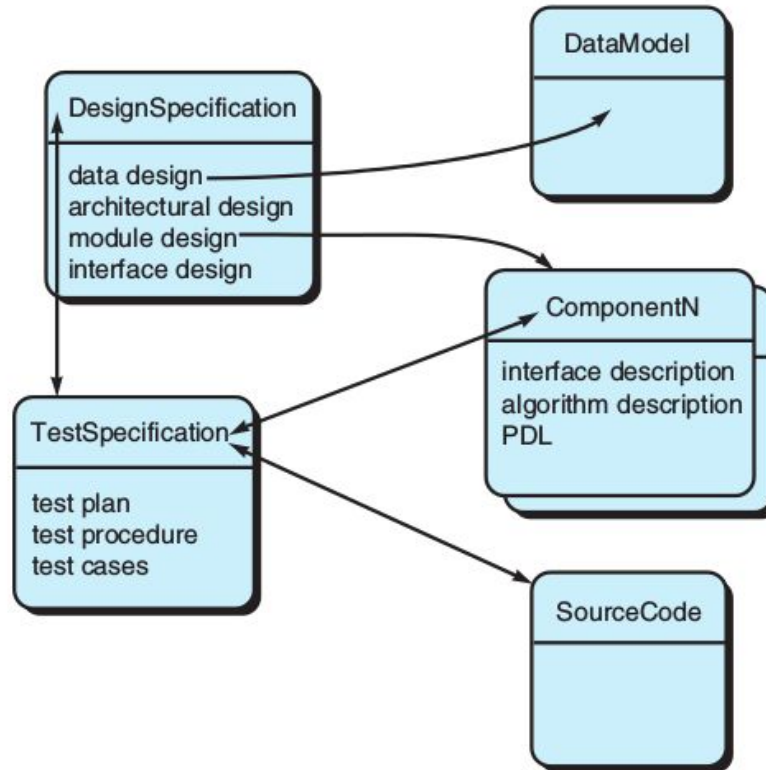
Prieš programinės įrangos elementui tampant baziniu elementu jis gali būti keičiamas greitai ir neformaliai. Po to pakeitimai šiam elementui gali būti vykdomi tik per formalias procedūras, skirtas įvertinti pakeitimams.

**FIGURE 29.1**

**Baselined  
SCIs and  
the project  
database**

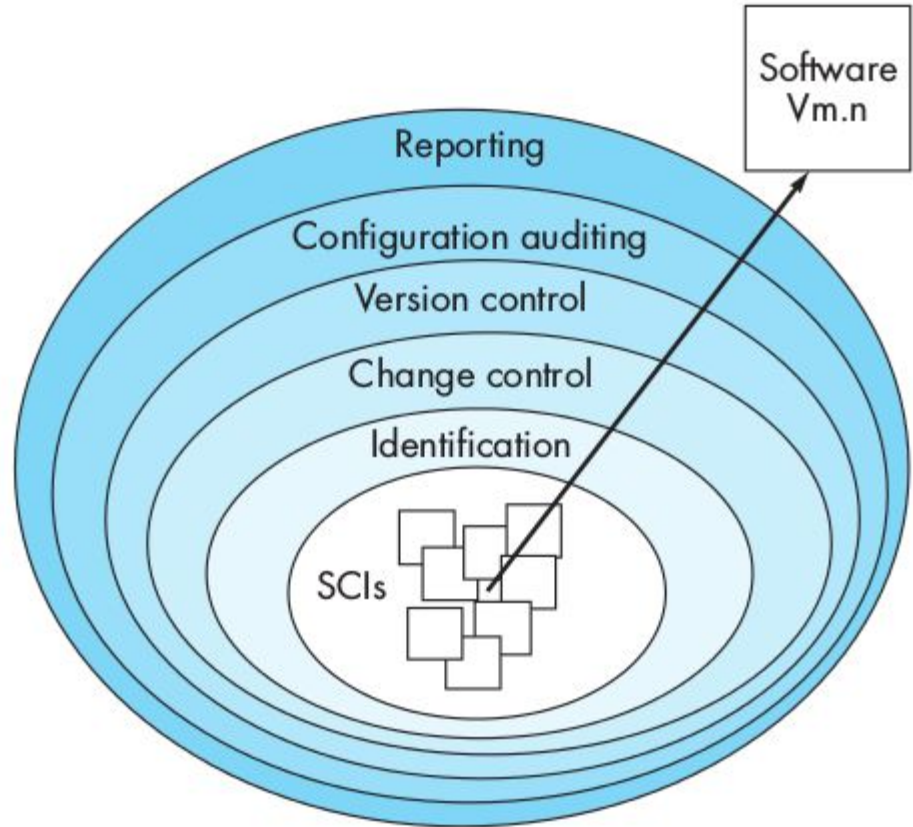


# SCI sudaro konfigurācijas objektus



# SCM užduotys

- Identifikacija.
- Versijų kontrolė.
- Pakeitimų kontrolė.
- Konfigūracijos auditas.
- Pranešimų teikimas.



# Identifikacija

Galima identifikuoti dviejų tipų objektus [Cho89]:

1. Baziniai objektai - informacijos vienetas sukurtas analizės, programavimo ar testavimo metu, pvz. reikalavimų specifikacijos skyrius, funkcijos programinis kodas.
2. Agreguoti objektai - bazinių objektų rinkinys, pvz. duomenų modelis.

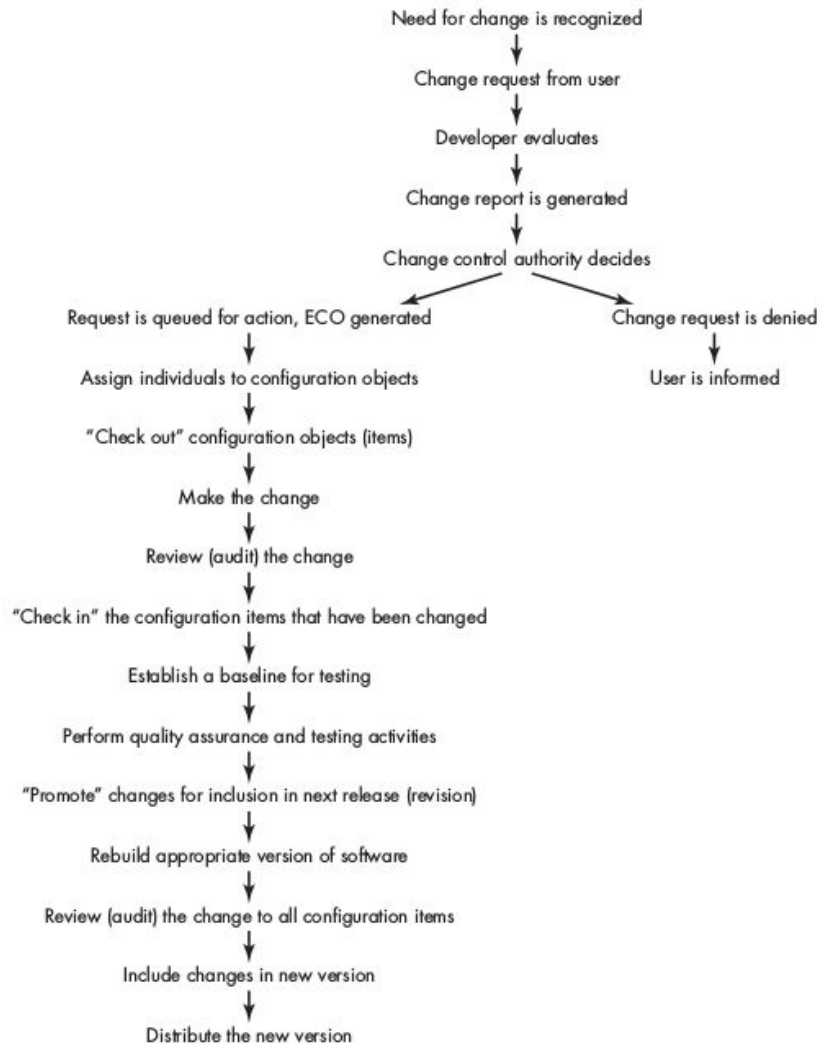
Kiekvienas SCI identifikuojamas - jam priskiriamas unikalus pavadinimas, aprašymas, nusakomos sąsajos su kitais SCI.

# Pakeitimų kontrolė

Per daug varžanti pakeitimų kontrolė sukuria problemas, per laisva pakeitimų kontrolė kuria kitas problemas. Dideliems programinės įrangos projektams nekontroliuojami pakeitimai veda prie chaoso.

Prieš SCI tampa baziniu elementu jam taikoma *informali pakeitimų kontrolė*. SCI tapus baziniu elementu jam keisti taikoma *projekto lygio pakeitimo kontrolė*. Programuotojas turi gauti asmesns atsakingo už pakeitimų kontrolę leidimą norint daryti pakeitimą. Projektą atidavus vartotojui pradedama taikyti *formali pakeitimų kontrolė*.

# Formali pakeitimų kontrolė





# Versijų kontrolė

Versijų kontrolė - kombinacija procedūrų ir įrankių, skirtų valdyti konfigūracijos elementų versijas. Versijų kontrolės sistemai reikalingos keturios pagrindinės savybės:

# Versijų kontrolė

Versijų kontrolė - kombinacija procedūrų ir įrankių skirtų valdyti konfigūracijos elementų versijas. Versijų kontrolės sistemai reikalingos keturios pagrindinės savybės:

1. Projekto repozitorija su visais konfigūracijos objektais.
2. Versijų valdymo funkcionalumas saugantis visas konfigūracijos objektų versijas, arba saugantis tik pakitusias objektų vietas ir gebantis iš to atgaminti pilnas versijas.
3. Galimybė sukonstruoti specifinę visos sistemos versiją iš atskirų konfigūracijos objektų.
4. Klaidų registravimo ir sekimo sistema.

# Konfigūracijos auditas

Konfigūracijos auditas papildo techninę apžvalgą ir atsako į šiuos klausimus:

# Konfigūracijos auditas

Konfigūracijos auditas papildo techninę apžvalgą ir atsako į šiuos klausimus:

- Ar techninė apžiūra buvo atlikta siekiant nustatyti korektiškumą techniniame lygmenyje?
- Ar buvo laikomasi programų sistemų kūrimo proceso standartų?
- Ar buvo nurodytas pakeitimo autorius, data?
- Ar buvo laikomasi SCM procedūrų skirtų pakeitimo registravimui ir informavimui apie pakeitimą?
- Ar susiję SCl buvo atnaujinti?

Būsenos ataskaitos

# Būsenos ataskaitos

- Kas pakeista?
- Kas atliko pakeitimą?
- Kada pakeitimas įvykdytas?
- Kas bus paveikta pakeitimo?

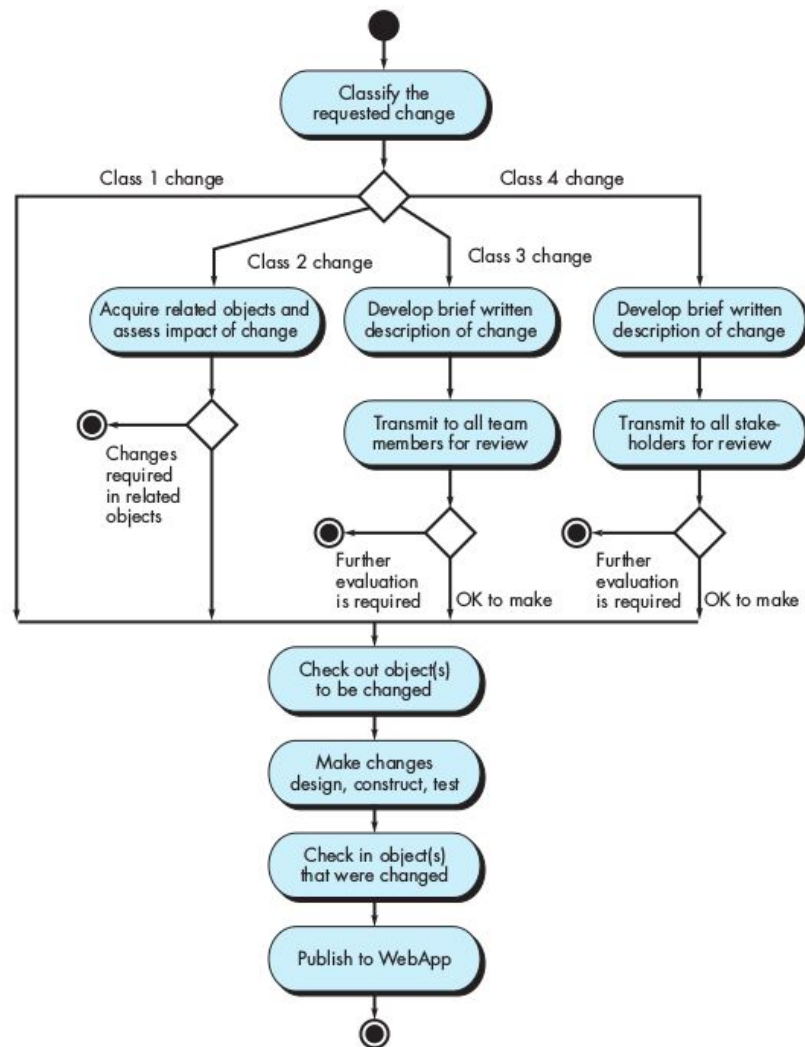
# SCM Web ir mobilioms aplikacijoms

Web ir mobilių aplikacijų kūrimo metu dažnai naudojamas iteratyvus ir inkrementinis proceso modelis kuris taiko Agile metodologijas. Naudojant šią metodologiją komandos kas trumpą laiko tarpą paleidžia vis naują patobulintą produkto versiją. Todėl Agile metodologijoje į pakeitimus žiūrima ne taip griežtai kaip SCM.

# Pakeitimų kategorijos

1. Pirma kategorija - pakeitimas, kuris ištaiso klaidą ar sukuria naują lokalių funkcionalumą.
2. Antra kategorija - funkcijos pakeitimas, kuris turį įtaką kitiems turinio ar funkciniam komponentams.
3. Trečia kategorija - turinio ar funkcinis pakeitimas, kuris turi stiprų poveikį visoje aplikacijoje.
4. Ketvirta kategorija - didelis dizaino pakeitimas.





Klausimai?