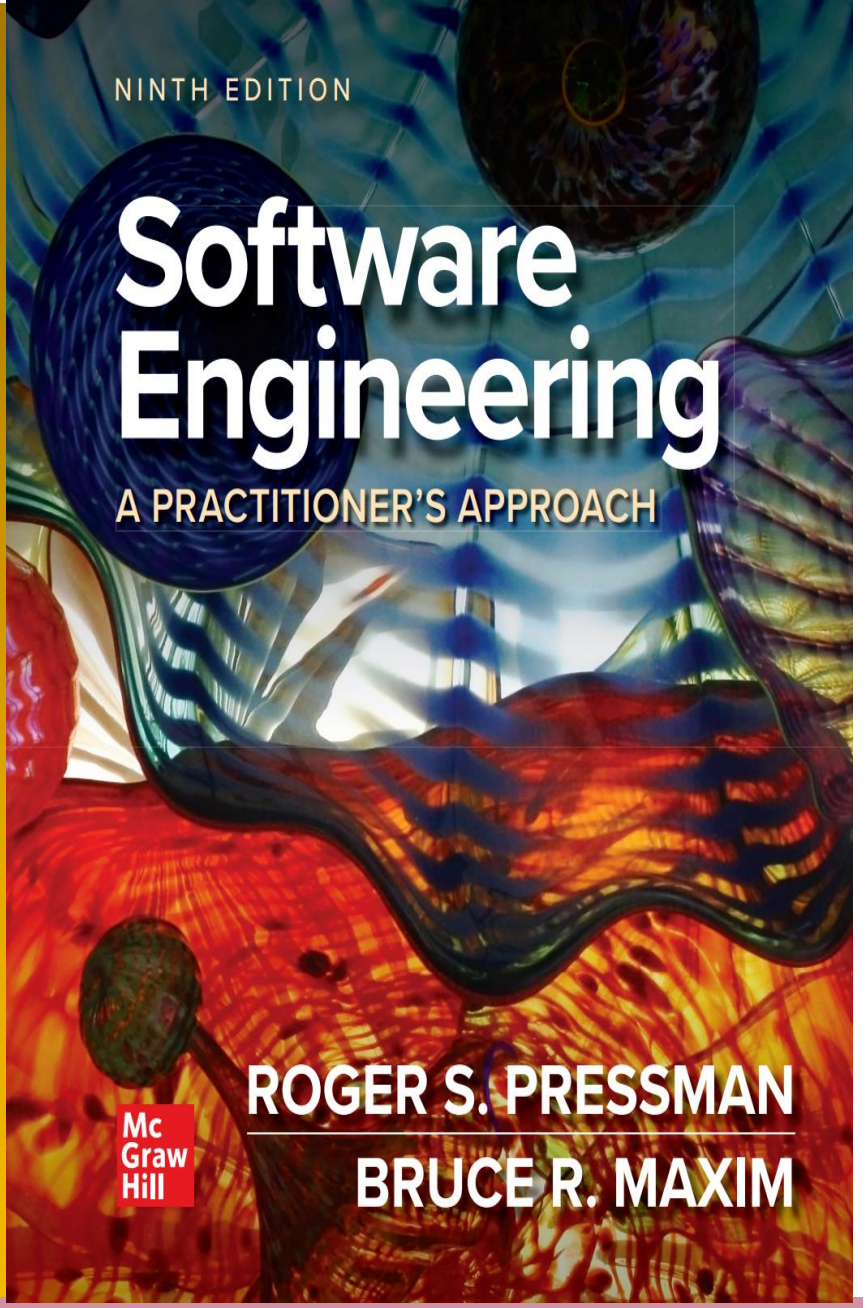
مدیریت پیکربندی نرم‌افزار

Scan for More Information:



Website: ataghinezhad.github.io, Email:a0taghinezhad@gmail.com

هدف فصل (مدیریت پیکربندی نرم‌افزار)

این فصل به معرفی مفهوم **مدیریت پیکربندی نرم‌افزار**

Software Configuration Management - SCM می‌پردازد.

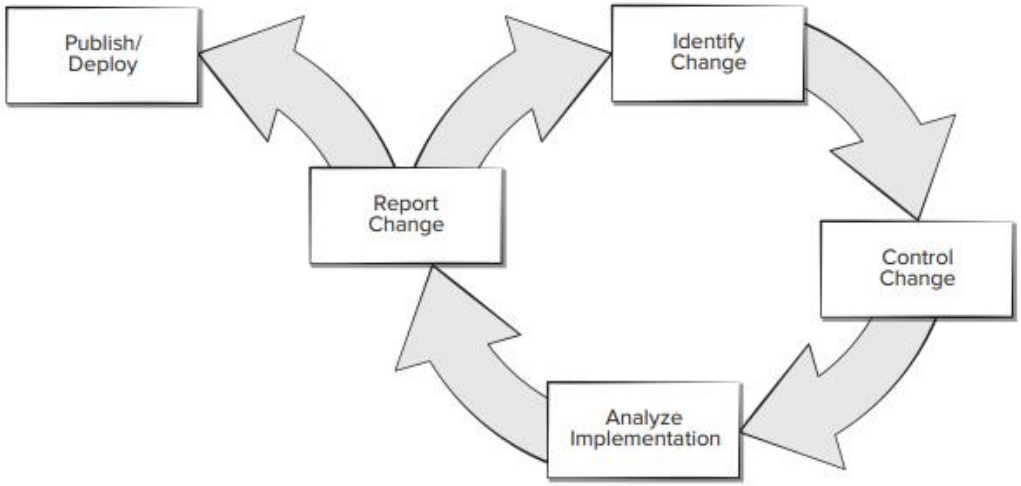
شامل مجموعه‌ای از فعالیت‌ها و ابزارها برای شناسایی، کنترل، پیگیری و گزارش‌دهی تغییرات در کلیه محصولات تولیدشده در چرخه حیات نرم‌افزار است.

اهداف اصلی این فصل عبارتند از:

- اطمینان از اینکه هر آیتم پیکربندی (کد منبع، مستندات، مدل‌ها، داده‌ها و...) دارای شناسه یکتا و نسخه‌بندی‌شده است.
- فراهم‌سازی سیاست‌ها، ابزارها و ممیزی‌هایی برای مدیریت تغییرات بدون آسیب به انسجام محصول.
- پشتیبانی از همکاری بین مدیران پروژه، مدیر پیکربندی، مهندسان و مشتریان با تعریف فرایندهای مشخص کنترل تغییر.
- تسهیل در ادغام سریع، تست قابل اعتماد و گزارش‌گیری دقیق به منظور تحویل نرم‌افزار با کیفیت و به‌موقع .

مقدمه بر مدیریت پیکربندی نرم‌افزار(SCM)

- **مدیریت پیکربندی نرم‌افزار** به مجموعه فعالیت‌هایی گفته می‌شود که تغییرات در محصولات نرم‌افزاری را «شناسایی»، «کنترل»، «پیگیری» و «گزارش» می‌کند.
- تمرکز بر حفظ یکپارچگی و قابلیت بازگشت به نسخه‌های قبلی.
- نقش کلیدی در کاهش ریسک و بهبود کیفیت.



مقدمه بر مدیریت پیکربندی نرم‌افزار(SCM)

مرحله	فعالیت	توضیح
Report Change	تیم UX پیشنهاد می‌دهد رنگ دکمه از آبی به سبز تغییر کند برای بهبود نرخ کلیک.	RFCثبت می‌شود.
Identify Change	بررسی می‌شود که این دکمه در سه صفحه مشترک استفاده شده.	۳ فایل UI تحت تأثیرند.
Control Change	تیم طراحی و توسعه بررسی می‌کنند که این تغییر باعث تداخل با تم برند نمی‌شود.	تصویب اولیه انجام می‌شود.
Analyze Implementation	نیاز به تغییر CSS در یک فایل اصلی و تست UI در ۲ حالت تیره/روشن.	تأثیر کم و کم‌ریسک است.
Publish/Deploy	نسخه جدید منتشر و در توضیحات انتشار ذکر می‌شود: "بهبود UI برای تجربه کاربری بهتر."	تغییر به Production می‌رود.

نیازمندی‌ها و مزایا

- **ردیابی تغییرات :**مشاهده تاریخچه کامل هر فایل و قابلیت مقایسه نسخه‌ها.
- **همکاری تیمی :**تسهیل کار موازی با شاخه‌بندی و ادغام کنترل‌شده.
- **بازگشت به عقب :**در صورت بروز خطا، بازگشت به آخرین نسخه پایدار.
- **گزارش‌گیری :**تهیه گزارش‌های خودکار از وضعیت پروژه.

نقش‌ها در فرایندSCM

- **مدیر پیکربندی :(Configuration Manager)** تدوین سیاست‌ها، تأیید درخواست‌های تغییر.
- **توسعه‌دهنده :**چک‌اوت/چک‌این، گزارش خطا، اجرای تغییرات.
- **مدیر پروژه :**نظارت بر گزارش‌های وضعیت، مدیریت زمانبندی.
- **ذینفعان :(Stakeholders)** ارائه بازخورد و تصویب نسخه اولیه نهایی.

نقش‌ها در فرایندSCM

۱ .مدیر پیکربندی(Configuration Manager)

نقش اصلی :تضمین رعایت سیاست‌ها، کنترل تغییرات، حفظ انسجام و قابلیت ردیابی کد.

وظایف:

- **تدوین سیاست‌ها و فرآیندها:**
  ایجاد دستورالعمل‌هایی مانند "همه تغییرات باید از طریق سیستم درخواست تغییر (Change Request System)انجام شوند".
- **تأیید درخواست‌های تغییر:(Change Requests)**
  بررسی و تأیید اینکه آیا یک تغییر پیشنهادی توجیه‌پذیر است، بر روی سایر اجزا تأثیر منفی ندارد، و باید در نسخه بعدی اعمال شود یا نه.

نقش‌ها در فرایندSCM

۲ .توسعه‌دهنده(Software Developer)

نقش اصلی :پیاده‌سازی و تست تغییرات کد، همکاری در ادغام (merge) و مدیریت نسخه‌های محلی.

وظایف:

- **چک‌اوت/چک‌این:(Checkout/Check-in)**
دریافت نسخه‌ای از کد منبع (checkout) برای کار روی آن و ارسال نسخه تغییر یافته (check-in) به مخزن اصلی.
- **گزارش خطا:(Bug Report)**
شناسایی باگ‌ها و ثبت آن‌ها در سیستم ردیابی مثلاً Jira یاBugzilla
- **اجرای تغییرات:**
اعمال تغییرات تأییدشده و تست آن‌ها در محیط محلی یا آزمایشی.

نقش‌ها در فرایندSCM

۳ .مدیر پروژه(Project Manager)

نقش اصلی :نظارت بر روند توسعه و هماهنگی منابع.

وظایف:

- **بررسی گزارش‌های وضعیت:**
  مطالعه گزارش‌های مدیر پیکربندی CMبرای ارزیابی پیشرفت، تأخیرها، و مشکلات احتمالی.
- **مدیریت زمانبندی:**
  اطمینان از تحویل نسخه‌ها مطابق با برنامه زمانی پروژه.

نقش‌ها در فرایندSCM

۴ .ذینفعان(Stakeholders)

نقش اصلی :ارائه بازخورد، تأیید نسخه نهایی و اطمینان از برآورده شدن نیازها.

وظایف:

- **بازخورد دادن:**
  بررسی نسخه‌های آزمایشی و ارائه پیشنهاد یا اصلاح.
- **تصویب نسخه اولیه نهایی:**
  تأیید اینکه نسخه منتشر شده معیارهای مورد نظر را دارد و قابل استفاده است.

اجزای کلیدی SCM

1. **Component Elements عناصر مولفه ای**
   - مخزن فایل، پایگاه‌داده SCIs
2. **Process Elements عناصر فرآیندی**
   - رویه‌های تعریف شده برای کنترل نسخه و تغییر
3. **Construction Elements عناصر ساخت**
   - ابزارهای Build اتوماتیک و CI/CD
4. **Human Elements عناصر انسانی**
   - آموزش، فرهنگ تیمی، مستندسازی

اجزای کلیدی SCM

1. در سیستم‌های **مدیریت پیکربندی نرم‌افزار SCM**، عناصر مختلفی برای پشتیبانی از کنترل مؤثر تغییرات و نگهداری سیستم وجود دارد. این عناصر در چهار دسته اصلی قابل طبقه‌بندی هستند: **عناصر مؤلفه‌ای، فرآیندی، ساخت، و انسانی**

۱- عناصر مؤلفه‌ای (Component Elements)

تعریف :منابع اصلی پروژه که تحت مدیریت پیکربندی قرار دارند.

شامل:

- **مخزن فایل (File Repository): محلی که فایل‌های کد منبع، مستندات و اسکریپت‌ها نگهداری می‌شوند. مانند Git.**
- **پایگاه داده (Software Configuration Items) SCIs: مجموعه‌ای از مؤلفه‌ها (کد، مستندات، فایل‌های پیکربندی) که به‌عنوان واحدهای قابل ردیابی شناخته می‌شوند.**

اجزای کلیدی SCM

۲ .عناصر فرآیندی(Process Elements)

تعریف :سیاست‌ها و رویه‌هایی که برای مدیریت تغییرات و کنترل نسخه اعمال می‌شوند.

شامل:

- رویه‌های کنترل نسخه :مانند الزام به نوشتن توضیح برای هر commit یا استفاده از pull request برای بازبینی تغییرات.
- مدیریت تغییر :(Change Management) فرآیند ثبت، بررسی، تأیید یا رد درخواست‌های تغییر.

اجزای کلیدی SCM

۳ .عناصر ساخت(Construction Elements)

تعریف :ابزارها و روش‌هایی که برای ساخت (build) و انتشار خودکار محصول استفاده می‌شوند.

شامل:

- ابزارهای ساخت اتوماتیک :مانندMaven ، Gradle، یا Make برای کامپایل و بسته‌بندی کد.
- ابزارهای :CI/CD مانندJenkins ، GitLab CIیا GitHub Actions برای تست، ادغام، و استقرار خودکار.

اجزای کلیدی SCM

۴ .عناصر انسانی(Human Elements)

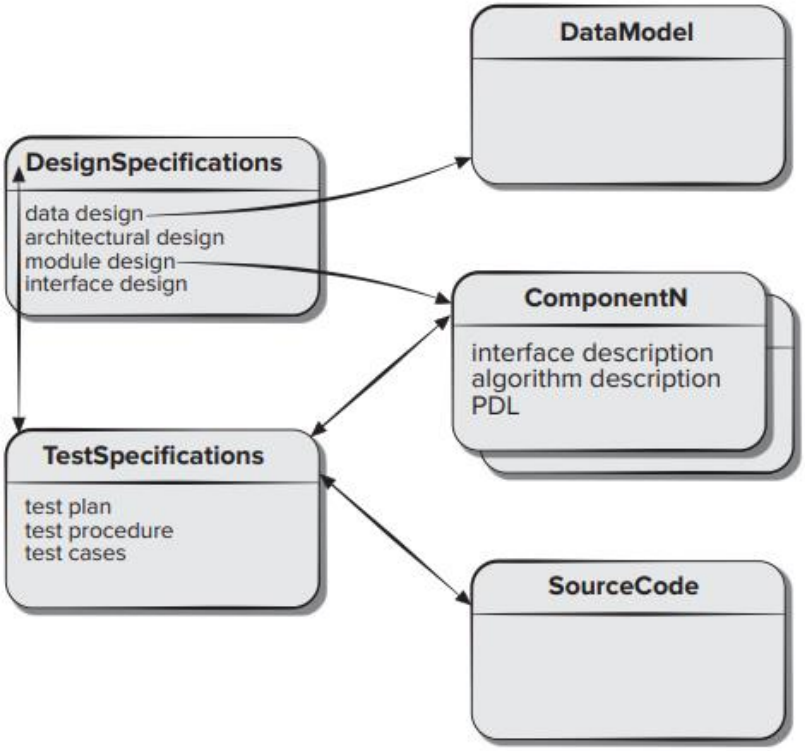
تعریف :افراد و فرهنگ سازمانی که به درستی و اثربخشی سیستم CM کمک می‌کنند.

شامل:

- آموزش :برگزاری جلسات آموزشی برای آشنایی تیم با فرآیندهای.CM
- فرهنگ تیمی :ایجاد ذهنیت همکاری، مسئولیت‌پذیری، و مستندسازی در تیم توسعه.
- مستندسازی :نوشتن دقیق تغییرات، دلیل تغییر، مراحل تست و تأییدات.

Software Configuration Item (SCI)

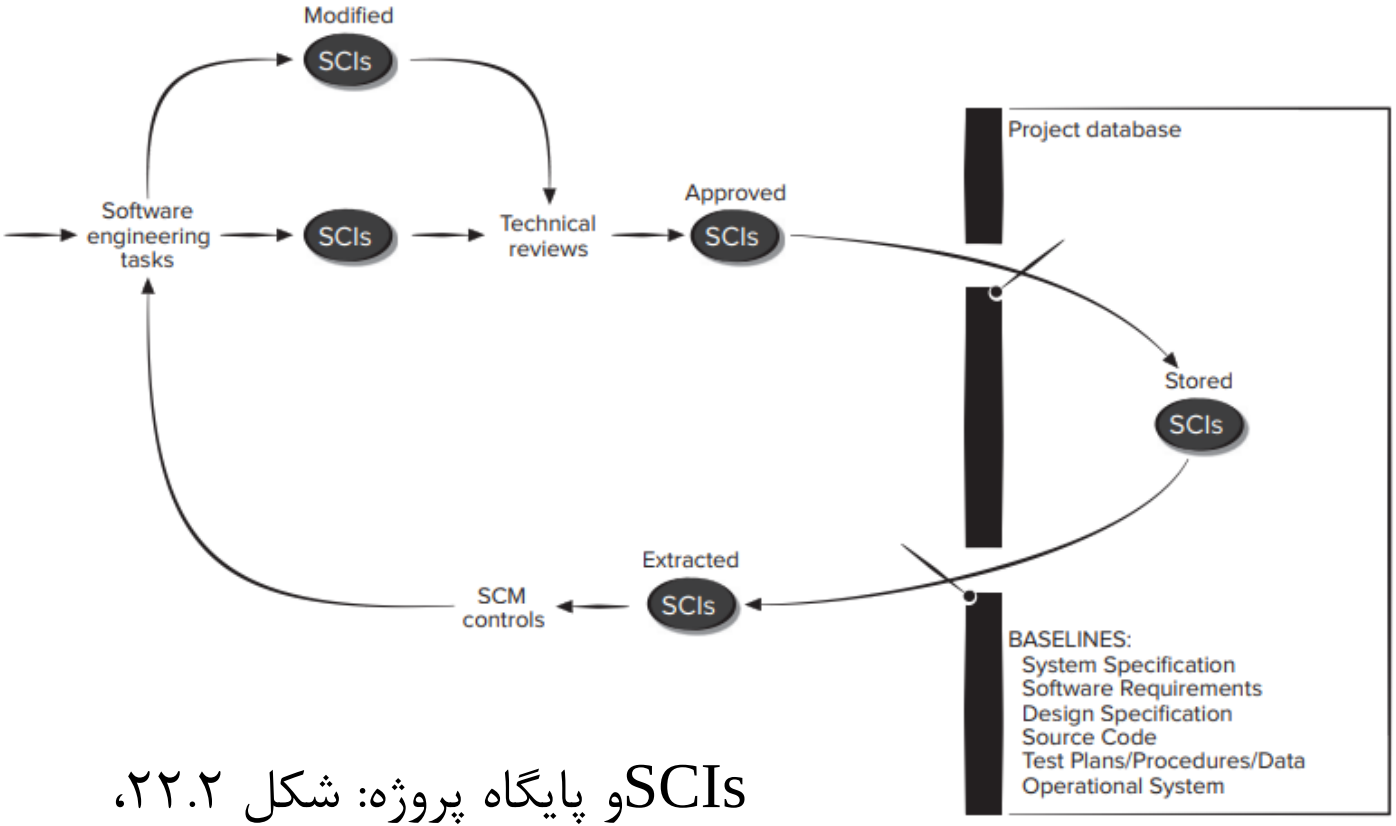
- کوچک‌ترین واحد قابل کنترل: فایل سورس، سند طراحی، دیاگرام UML.
- هر SCI باید شناسه یکتا، توضیحات متادیتا و وابستگی‌ها داشته باشد.



- اشیاء پیکربندی: شکل ۲۲.۳، صفحه ۴۴۲

Baseline: Snapshotو

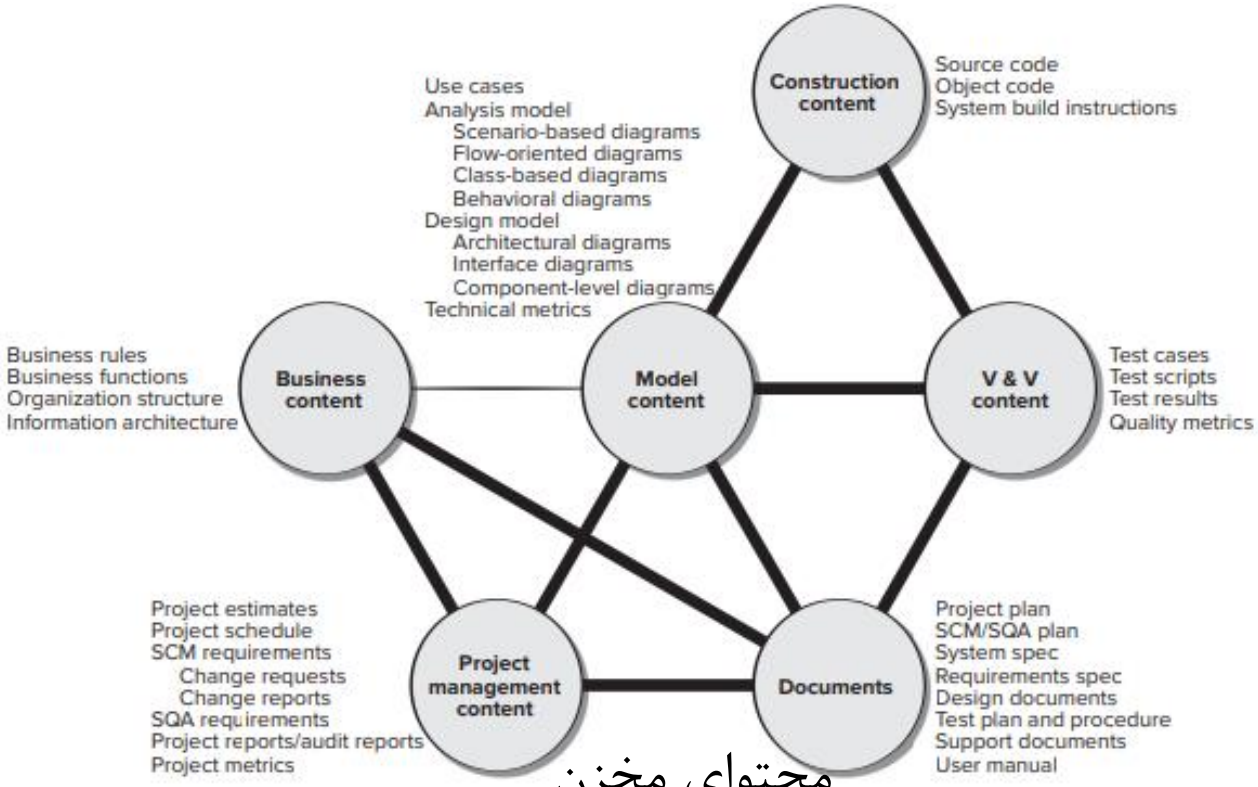
- **Baseline** نسخه رسمی مصوب یک یا چند SCI که پس از آن هر تغییر تنها از طریق روند رسمی اعمال می‌شود.
- **Snapshot** کپی لحظه‌ای از وضعیت مخزن در یک زمان مشخص، بدون الزام رسمی.



SCIsو پایگاه پروژه: شکل ۲۲.۲،

کنترل نسخه(Version Control)

- مرکزی SVN **(Centralized):** ، Perforce
- توزیعی Git **(Distributed):** ، Mercurial
- عملیات اصلی Commit: ، Update/Fetch، Merge، Branch، Tag.



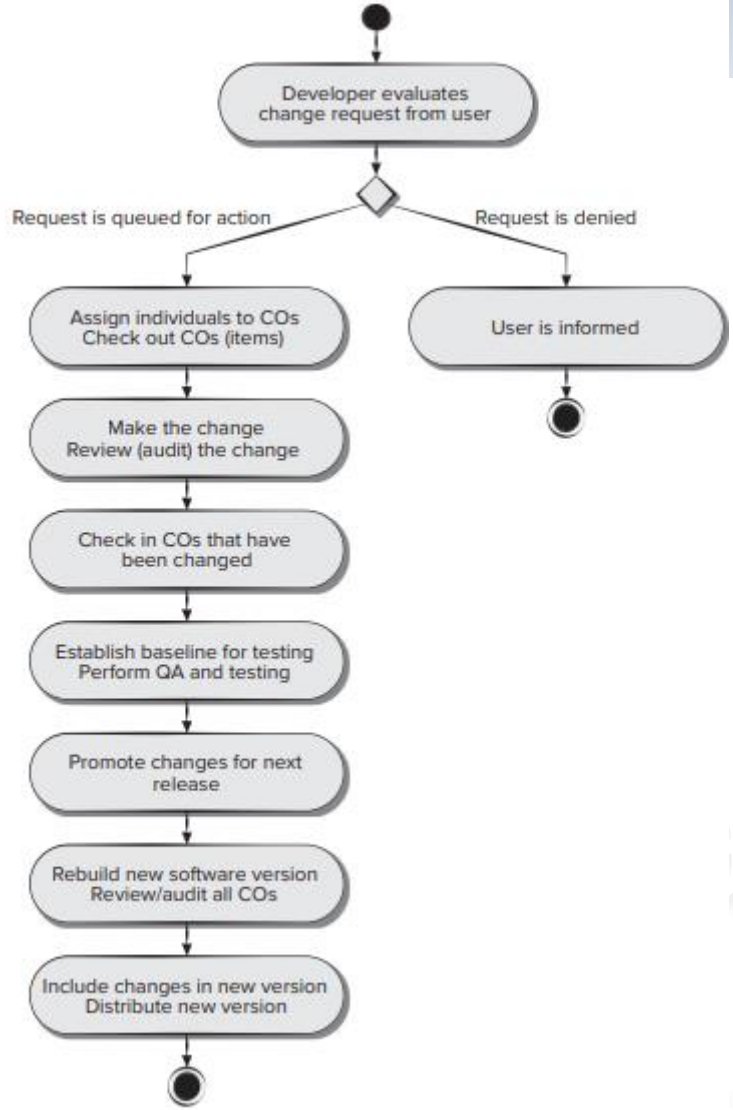
محتوای مخزن

شاخه‌بندی و ادغام(Branching & Merging)

- **Branching:** ایجاد محیط مجزا برای توسعه ویژگی‌ها یا رفع باگ.
- **Merging:** ادغام تغییرات از شاخه‌های مختلف به شاخه اصلی.
- تکنیک‌های رایج: Git Flow ،. GitHub Flow.

لایه کنترل نسخه در فرایند :SCM شکل ۲۲.۵، صفحه ۴۴۸

کنترل تغییر (Change Control)



کنترل تغییر: شکل ۲۲.۶، صفحه ۴۵۰

1. **RFC (Request for Change):** ثبت رسمی درخواست و مستندات لازم.
2. **ارزیابی تأثیر:** تحلیل هزینه، زمان، ریسک.
3. **ECO (Engineering Change Order):** دستور رسمی برای اعمال تغییر.
4. **توسعه و تست:** پیاده‌سازی تغییر و اجرای تست‌های مرتبط.

ممیزی پیکربندی(Configuration Audit)

- ممیزی عملکردی **Functional Audit**تأیید اینکه تغییرات به اهداف عملکردی رسیده‌اند.
- ممیزی فیزیکی **Physical Audit**تأیید وجود همه SCIs و انطباق با خط پایه Baseline.
- گزارش ممیزی شامل یافته‌ها، موارد انحراف و پیشنهادات اصلاحی است.

یکپارچگی مداوم(Continuous Integration)

- ادغام خودکار تغییرات در مخزن مرکزی پس از هرCommit
- اجرای خودکار Build و تست‌ها.
- کشف سریع خطاها و کاهش ریسک ادغام.

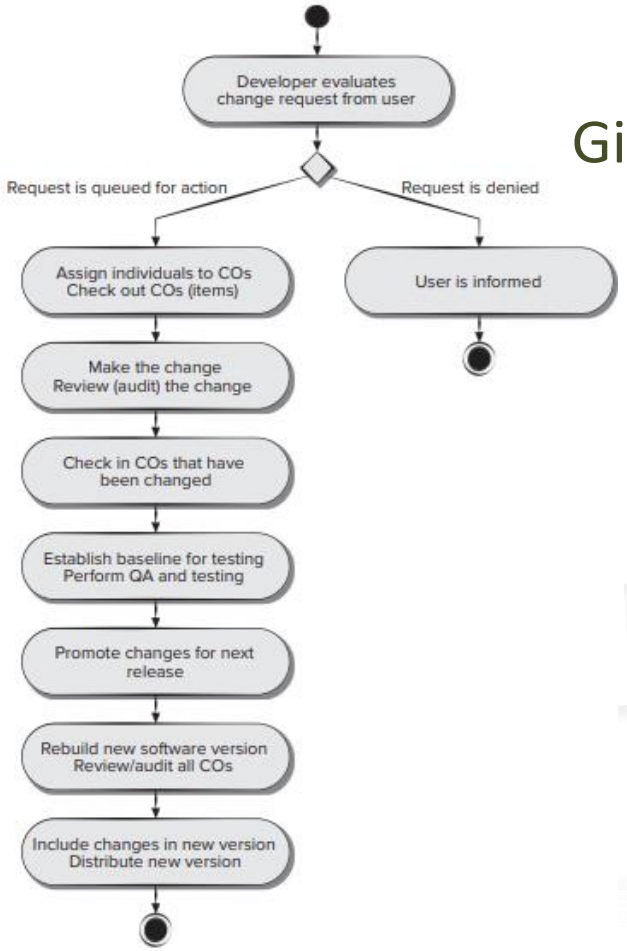
استقرار مداوم(Continuous Deployment)

گسترش خودکار پس از موفقیت CI به محیط‌های تست.Production

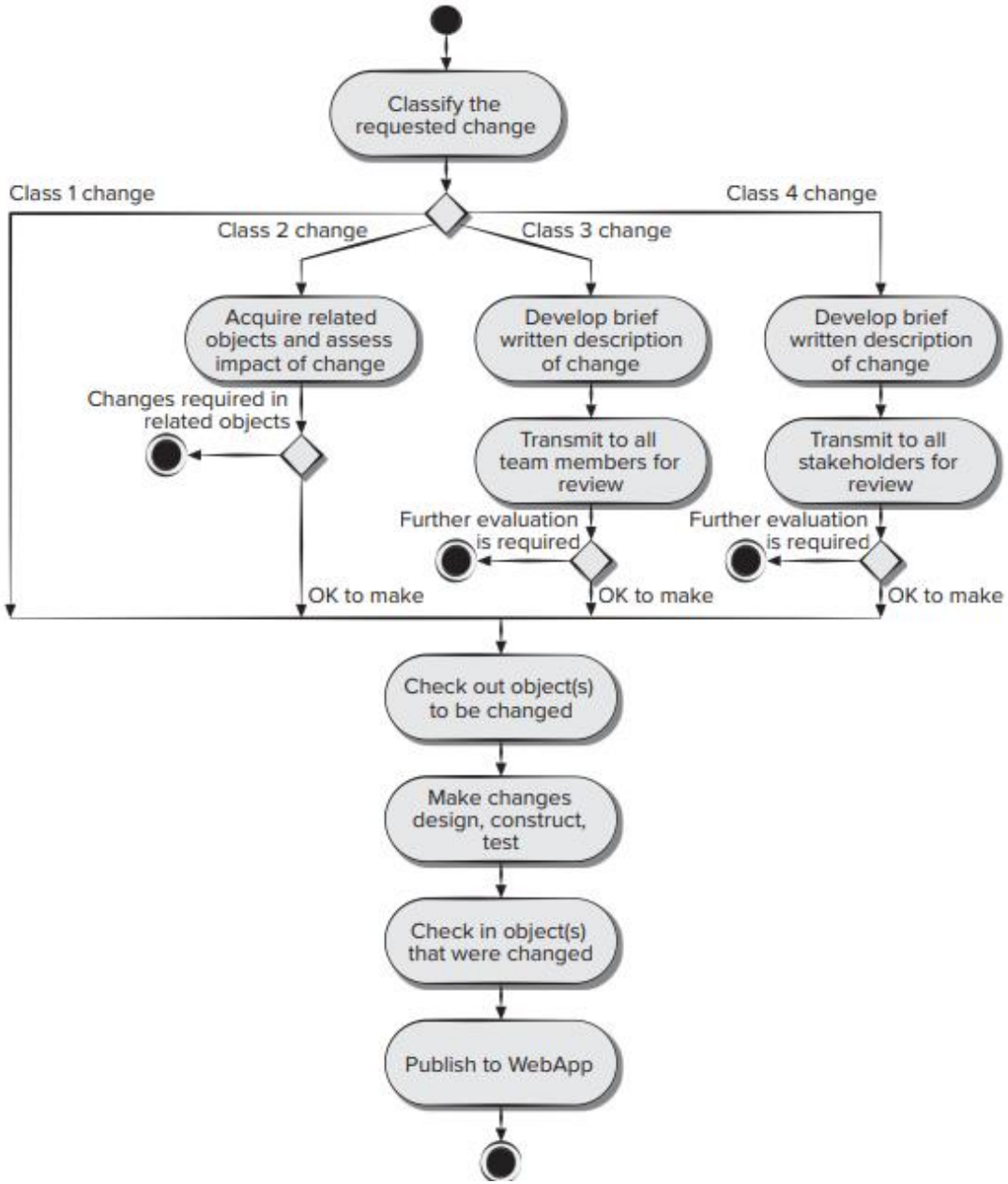
- کاهش زمان چرخه انتشار از هفته‌ها به دقیقه‌ها.
- الزامات: تست خودکار کامل، مانیتورینگ قوی.

گزارش وضعیت(Status Reporting)

- **محتوای گزارش :**نسخه مخزن، شاخه‌های فعال، وضعیتMerge Request ها، تست‌های ناموفق.
- **ابزارها :**داشبوردCI/CD ، Pluginهای.GitLab/GitHub
- ارائه گزارش زمان‌بندی منظم (روزانه/هفتگی).



تغییرات در WebApps و موبایل



- طبقه‌بندی تغییرات کلاس ۱ تا ۴
- فرایند سریع و چابک برای WebApps

- پایان