

## บทที่ 12

# การจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์ (Configuration Management)

วิชา วิศวกรรมซอฟต์แวร์ (04-06-306)



# วัตถุประสงค์การเรียนรู้

---

- เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์
- เพื่อให้ผู้เรียนสามารถบอกถึงการประยุกต์ใช้กระบวนการดังกล่าว กับ การพัฒนางานด้านซอฟต์แวร์ในเบื้องต้นได้



# หัวข้อ (Agenda)

---

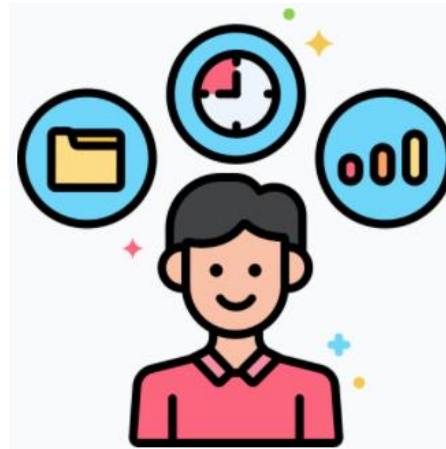
- บทนำ (Overview)
- การจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์
- การจัดทำเอกสารข้อเสนอให้เปลี่ยนระบบ (Change Request Form)
- การจัดการเวอร์ชันซอฟต์แวร์ (Software Version Management)
- เครื่องมือสำหรับการจัดการเปลี่ยนแปลง (Change Management Tool)
- สรุป (Summary)



# บทนำ (Overview)

Software systems always change during development and use

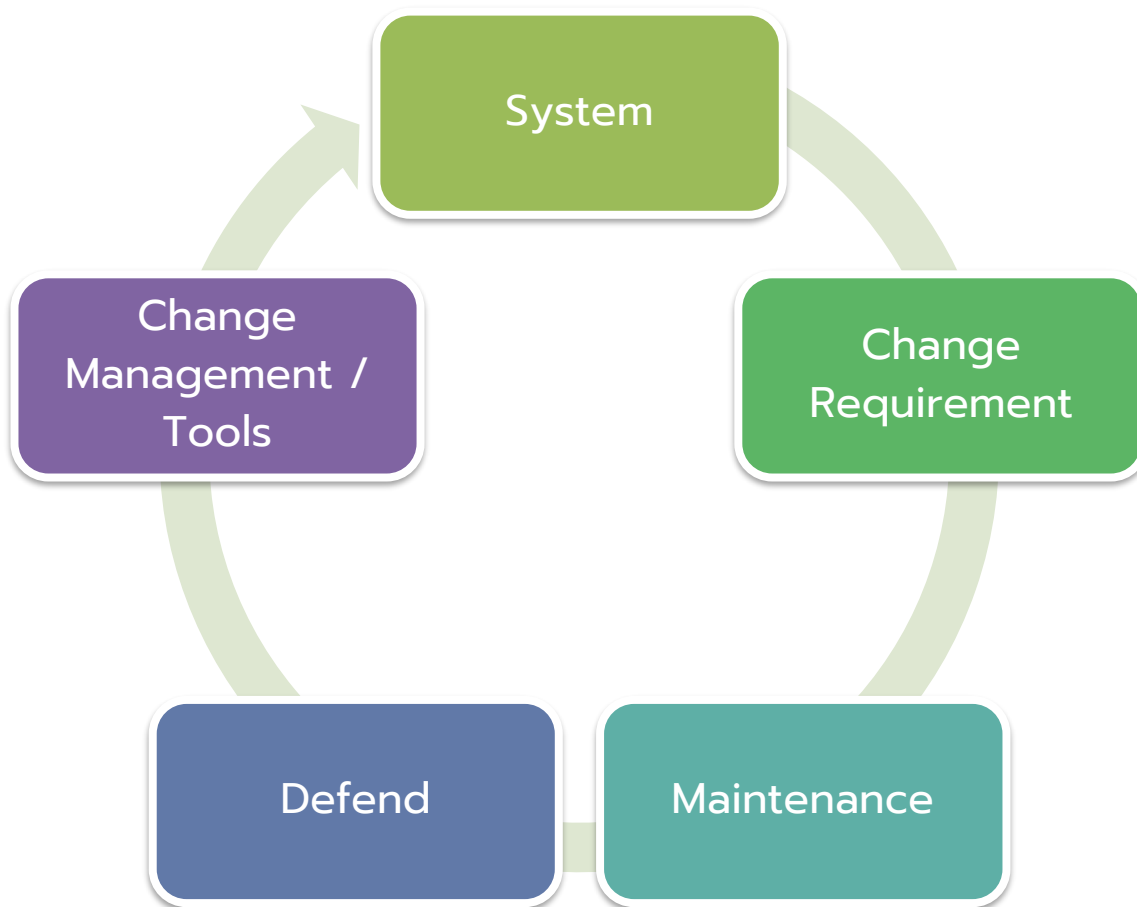
Configuration management (CM)



System requirements change

Configuration management is useful for individual projects

# การจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์ (Configuration Management)

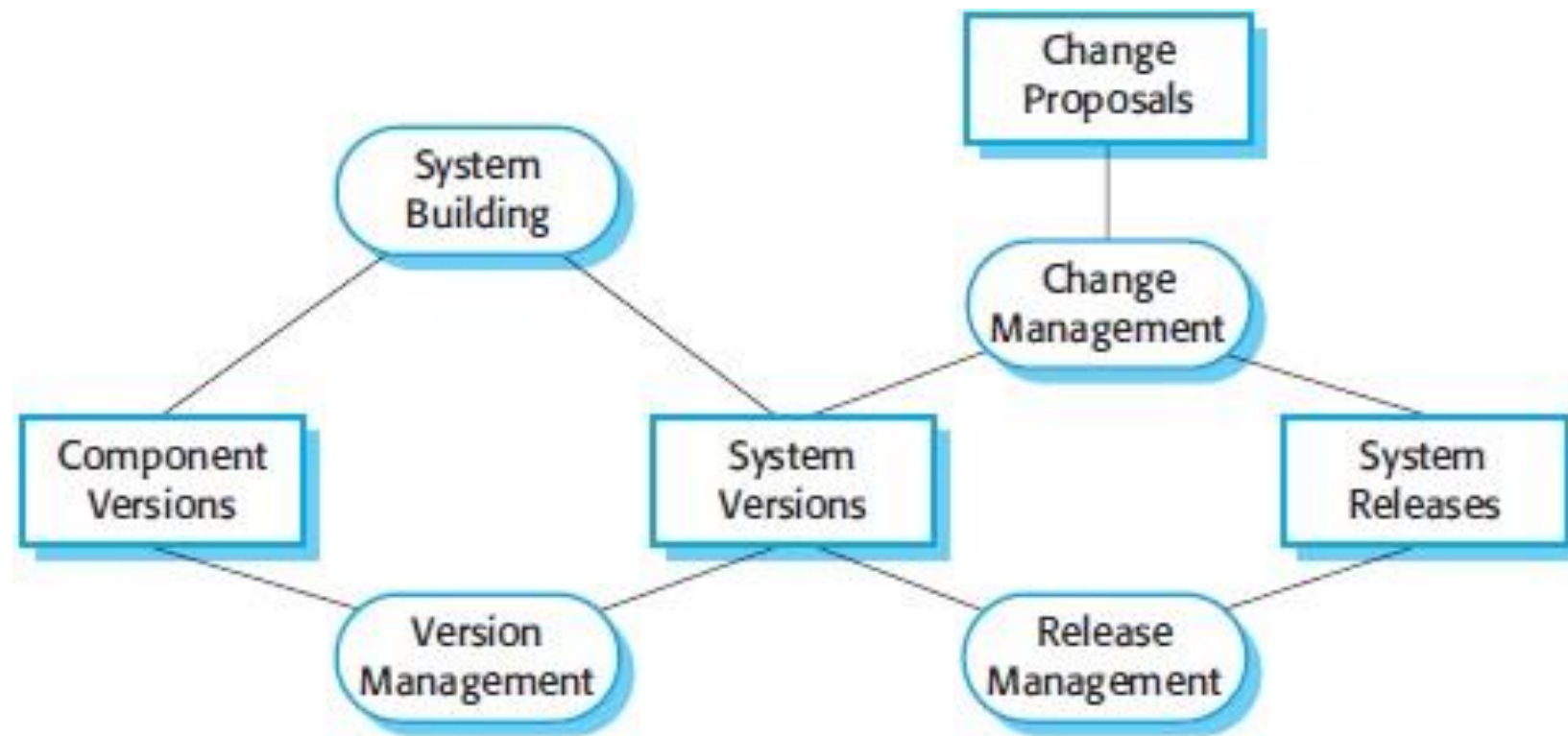


- เป็นการวิเคราะห์ต้นทุนและผลตอบแทนที่ได้รับจาก proposal ให้เปลี่ยนแปลงระบบ
- พิจารณาสິงที่ต้องเปลี่ยนแปลงคําค่ากับเงินลงทุนที่เสียไปหรือไม่ พร้อมกับตรวจสอบรายการที่ต้องเปลี่ยนแปลง



# Configuration management activities

- The configuration management of a software system product involves four closely related activities:

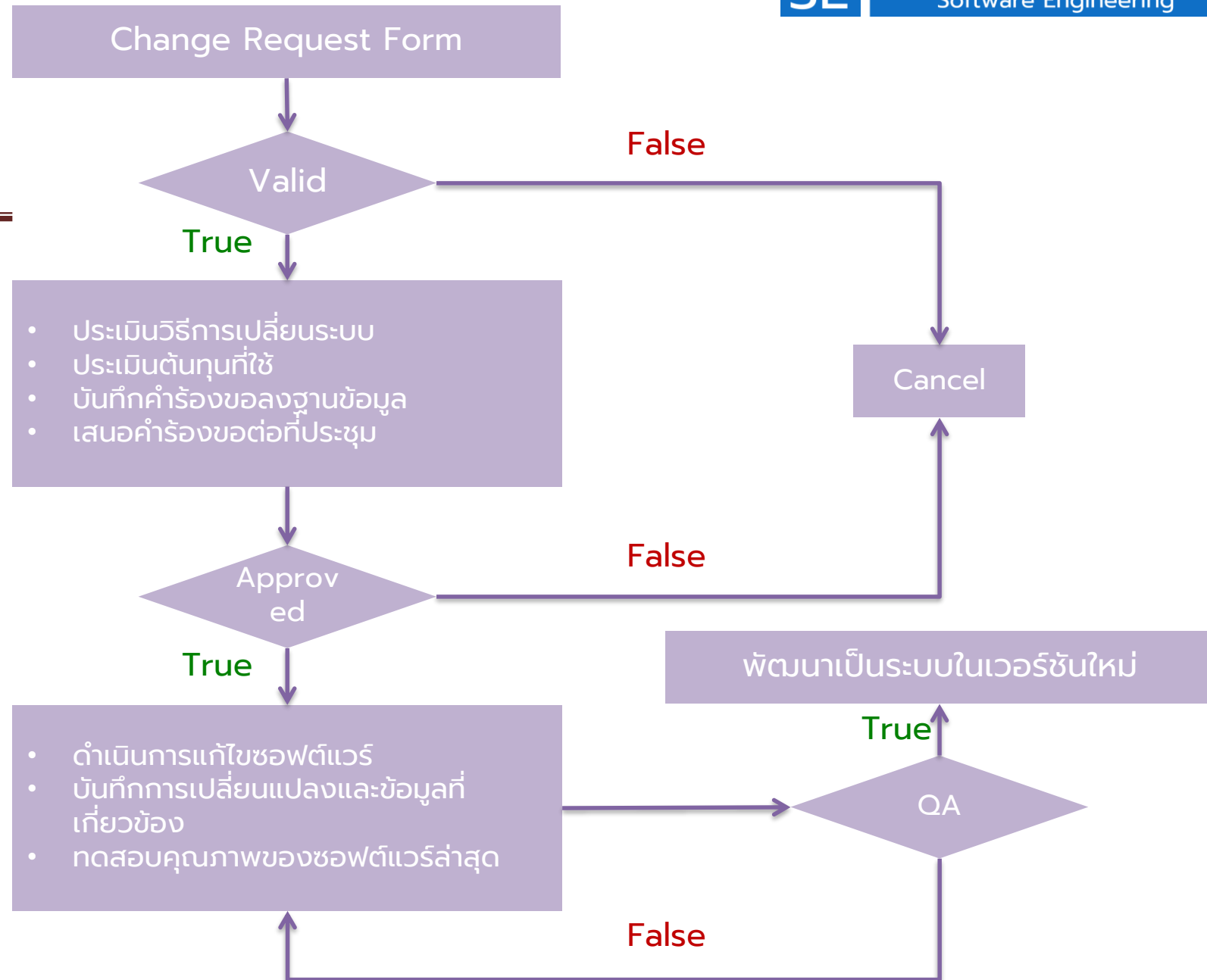


# CM terminology

Term	Explanation
Configuration item or software configuration item (SCI)	Anything associated with a software project (design, code, test data, document, etc.) that has been placed under configuration control. There are often different versions of a configuration item. Configuration items have a unique name.
Configuration control	The process of ensuring that versions of systems and components are recorded and maintained so that changes are managed and all versions of components are identified and stored for the lifetime of the system.
Version	An instance of a configuration item that differs, in some way, from other instances of that item. Versions always have a unique identifier, which is often composed of the configuration item name plus a version number.
Baseline	A baseline is a collection of component versions that make up a system. Baselines are controlled, which means that the versions of the components making up the system cannot be changed. This means that it should always be possible to re-create a baseline from its constituent components.
Codeline	A codeline is a set of versions of a software component and other configuration items on which that component depends.
Mainline	A sequence of baselines representing different versions of a system.
Release	A version of a system that has been released to customers (or other users in an organization) for use.
Workspace	A private work area where software can be modified without affecting other developers who may be using or modifying that software.
Branching	The creation of a new codeline from a version in an existing codeline. The new codeline and the existing codeline may then develop independently.
Merging	The creation of a new version of a software component by merging separate versions in different codelines. These codelines may have been created by a previous branch of one of the codelines involved.
System building	The creation of an executable system version by compiling and linking the appropriate versions of the components and libraries making up the system.



# Change management process





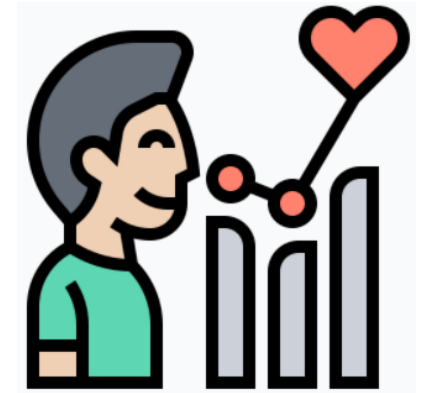
# Change management process (ต่อ)

- Significant factors that should be taken into account in deciding whether or not a change should be approved are:
  - *The consequences of not making the change*
    - When assessing a change request, you have to consider what will happen if the change is not implemented.
    - If the change is associated with a reported system failure, the seriousness of that failure has to be taken into account.
    - If the system failure causes the system to crash, this is very serious and failure to make the change may disrupt the operational use of the system.
    - On the other hand if the failure has a minor effect, such as incorrect colors on a display, then it is not important to fix the problem quickly, so the change should have a low priority.



# Change management process (ต่อ)

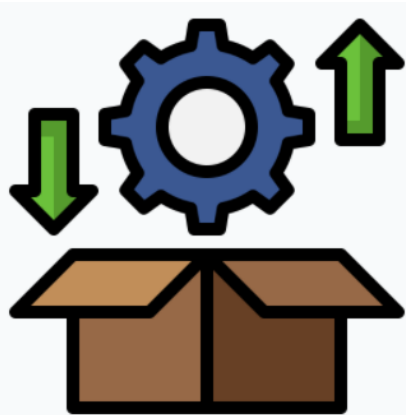
- *The benefits of the change*
  - Is the change something that will benefit many users of the system or is it simply a proposal that will primarily be of benefit to the change proposer?
- *The number of users affected by the change*
  - If only a few users are affected, then the change may be assigned a low priority.
  - In fact, making the change may be inadvisable if it could have adverse effects on the majority of system users.



# Change management process (ต่อ)



- *The costs of making the change*
  - If making the change affects many system components (hence increasing the chances of introducing new bugs) and/or takes a lot of time to implement, then the change may be rejected, given the elevated costs involved.



- *The product release cycle*
  - If a new version of the software has just been released to customers, it may make sense to delay the implementation of the change until the next planned release

# การจดทำเอกสารข้อเสนอการเปลี่ยนแปลง (Change Request Form)

- โครงการผลิตซอฟต์แวร์ขนาดใหญ่
  - ฉบับสมบูรณ์ ซึ่งรวมไว้ในกิจกรรมการจัดการเปลี่ยนแปลง
  - เขียน CRF
  - ใช้อย่างเป็นทางการ
- โครงการผลิตซอฟต์แวร์ขนาดเล็ก
  - มุ่งเน้นในการอธิบายความต้องการเปลี่ยนแปลงระบบ
  - ไม่เน้นวิธีปฏิบัติหรือขั้นตอนการเปลี่ยนแปลง
  - ทีมพัฒนาระบบเป็นผู้กำหนดขั้นตอนการปฏิบัติงาน
    - ตามสถานการณ์และสภาพแวดล้อม

ตัวอย่างแบบฟอร์มการเขียนข้อเสนอให้เปลี่ยนระบบ (CFR)  
<https://goo.gl/eiS2Xf>



# ตัวอย่าง เอกสารข้อเสนอการเปลี่ยนแปลง

Change  
Request  
Form

Change Request Form

**Project:** SICSA/AppProcessing

**Change requester:** I. Sommerville

**Requested change:** The status of applicants (rejected, accepted, etc.) should be shown visually in the displayed list of applicants.

**Number:** 23/02

**Date:** 20/01/09

**Change analyzer:** R. Looek

**Components affected:** ApplicantListDisplay, StatusUpdater

**Analysis date:** 25/01/09

**Associated components:** StudentDatabase

**Change assessment:** Relatively simple to implement by changing the display color according to status. A table must be added to relate status to colors. No changes to associated components are required.

**Change priority:** Medium

**Change implementation:**

**Estimated effort:** 2 hours

**Date to SGA app. team:** 28/01/09

**Decision:** Accept change. Change to be implemented in Release 1.2

**Change implementor:**

**Date submitted to QA:**

**Date submitted to CM:**

**Comments:**

**CCB decision date:** 30/01/09

**Date of change:**

**QA decision:**

# ตัวอย่าง ประวัติการเปลี่ยนแปลงคอมโพเนนท์

#SE Project

#SE-Tools/EDIT/FORM/DISPLAY/DISPLAY 1

#Object : User role

#Creator : Chumpol Mokrat

#Date : 2021-08-01

#Copy right : IT-CPC-BUSIT-RMUTTO

#Modification History

#Version	Developer	Date	Change	Note
# 1.0	John D.	2021-08-10	Additional search function.	Customer request
# 2.0	Devid C.	2021-09-20	Manual description	Process is wrong

▲ Commit or Logs

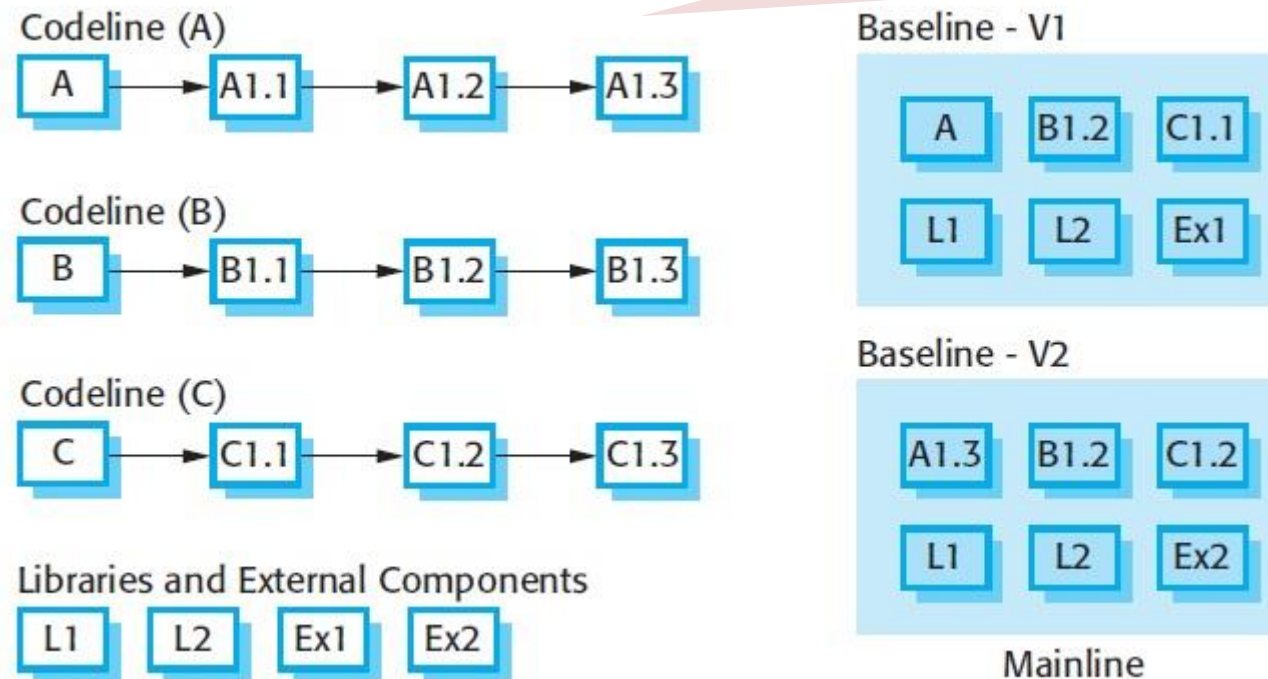
# การจัดการเวอร์ชันซอฟต์แวร์ (Software Version Management)

- เวอร์ชันซอฟต์แวร์ (Software Version)
  - รุ่นของระบบหรือซอฟต์แวร์
    - Microsoft Office 2015, ASP.NET 2.0
- วัตถุประสงค์ของการกำหนดรุ่นของซอฟต์แวร์
  - พังทช์ชิ้นการทำงานต่างกัน
  - ปรับปรุงคุณภาพของซอฟต์แวร์เดิม
  - สนับสนุนอุปกรณ์ที่แตกต่างกัน
- การจัดการเวอร์ชัน (Version Management)
  - กระบวนการจำแนกและติดตามการเปลี่ยนแปลงเวอร์ชันของระบบ หรือ ซอฟต์แวร์
  - ควบคุมไม่ให้เกิดการกำหนดเวอร์ชันปะปนกัน ช่วยวางแผนการพัฒนาระบบ เวอร์ชันใหม่มีความถูกต้อง
  - CASE Tool เป็นเครื่องมือช่วยจัดการเวอร์ชัน



# Codelines and baselines

a codeline is a sequence of versions of source code with later versions in the sequence derived from earlier versions

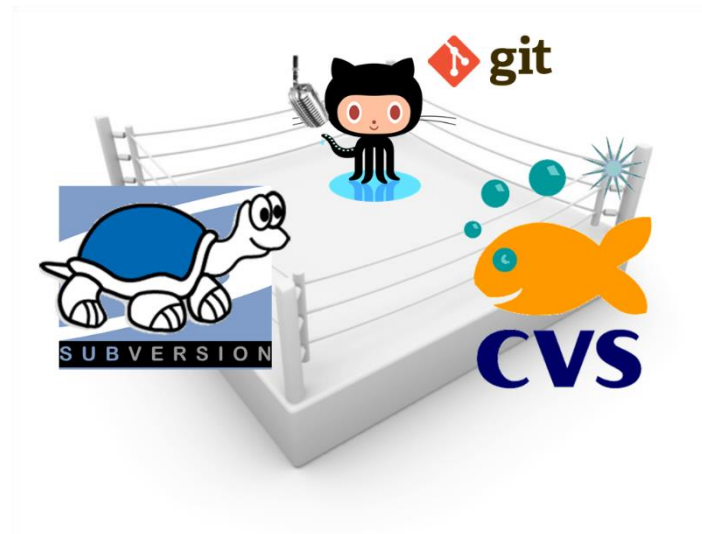


a baseline is a definition of a specific system.



# Version Management Tools

- Version control systems or source code control systems
- These tools identify, store, and control access to the different versions of components
  - CVS, Subversion and Git



# Version management

---

- Version management systems normally provide a range of features:

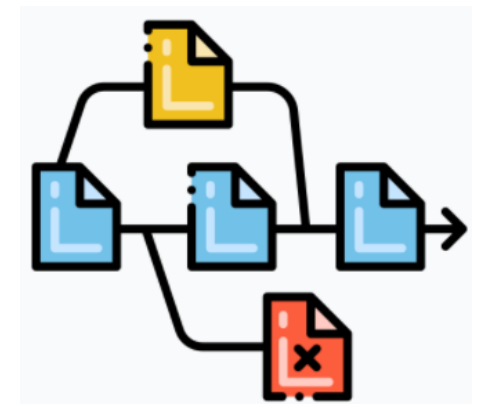
Version and release identification

Storage management

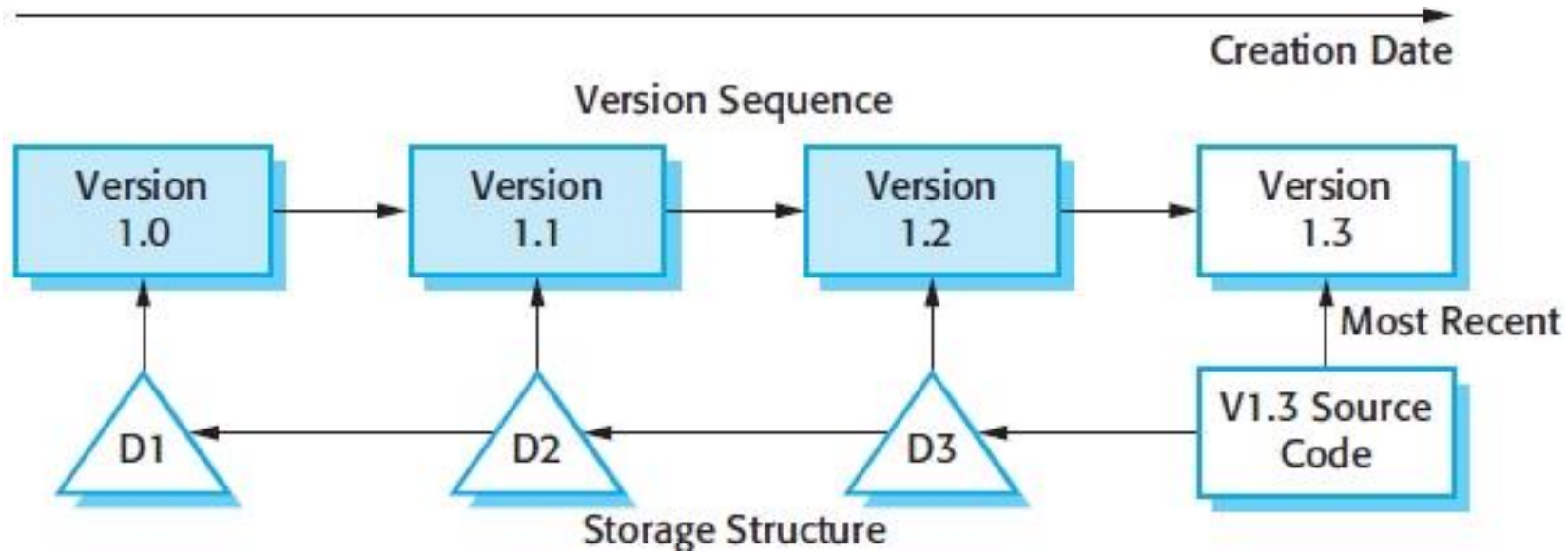
Change history recording

Independent development

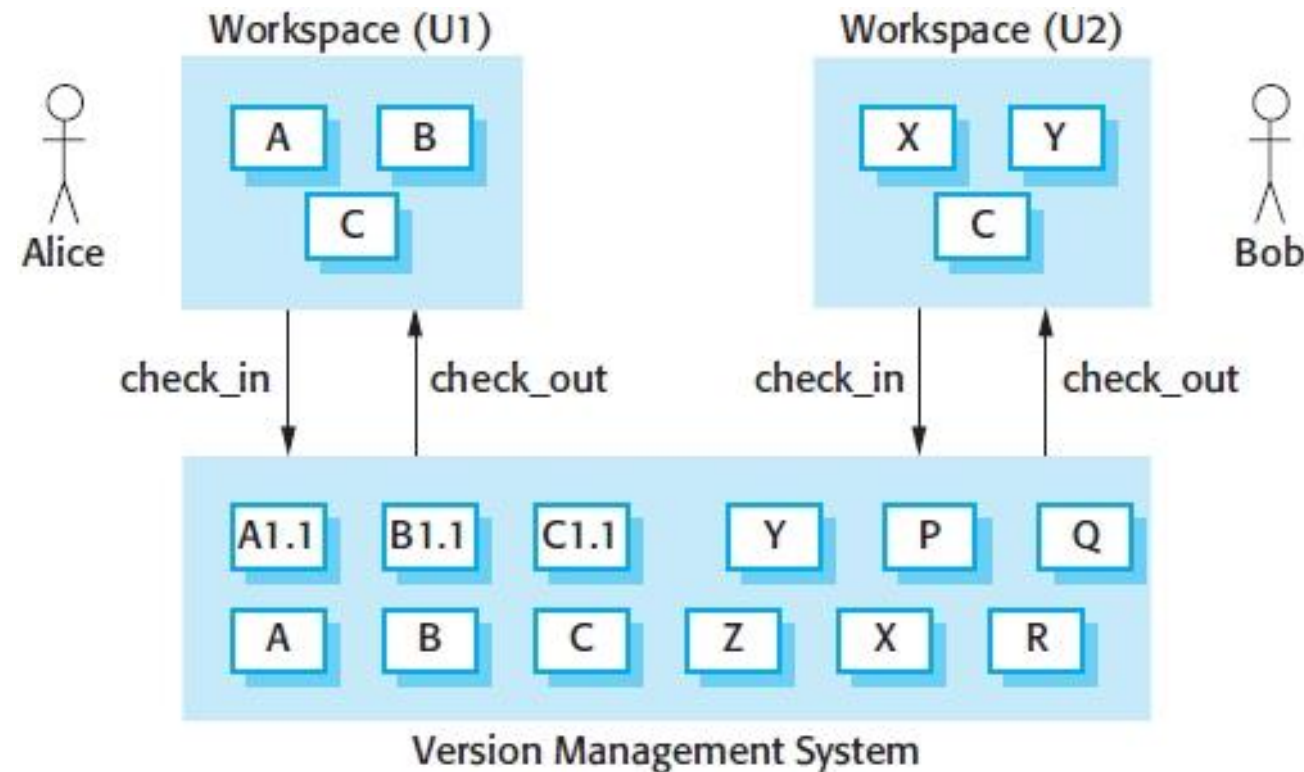
Project support



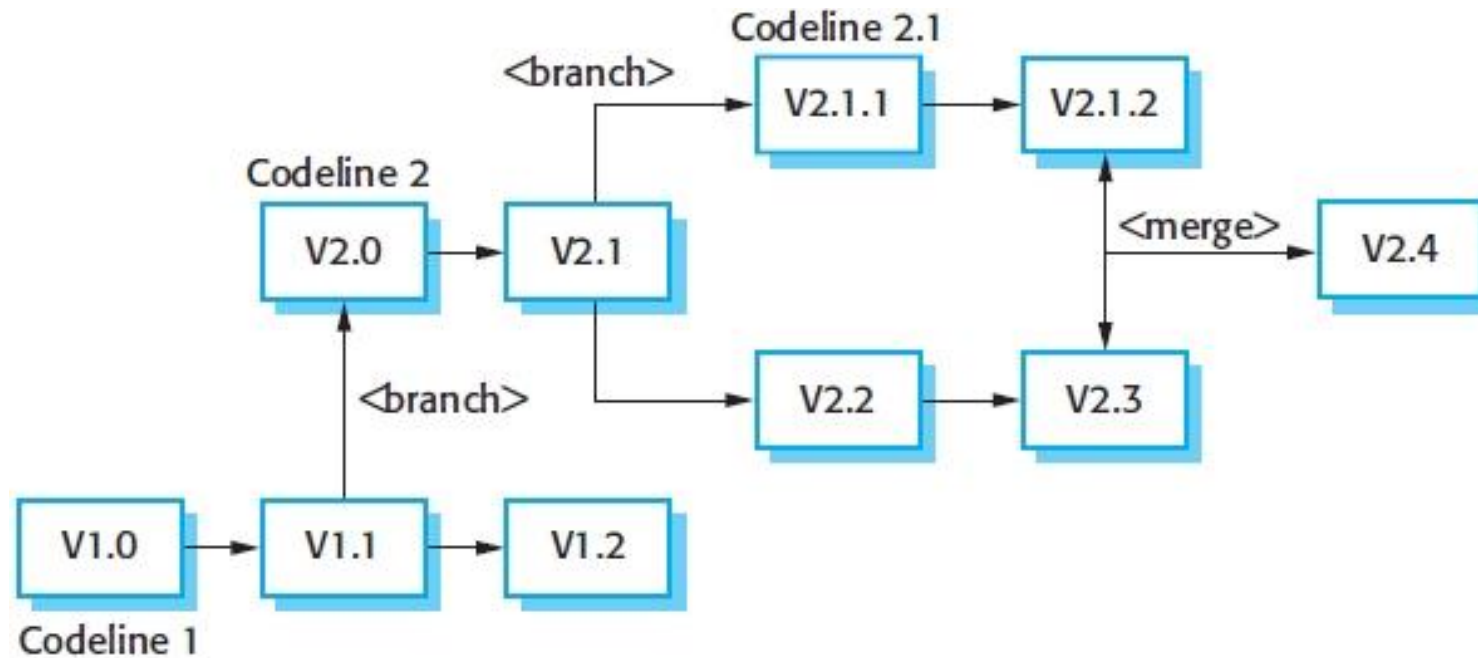
# Storage management using deltas



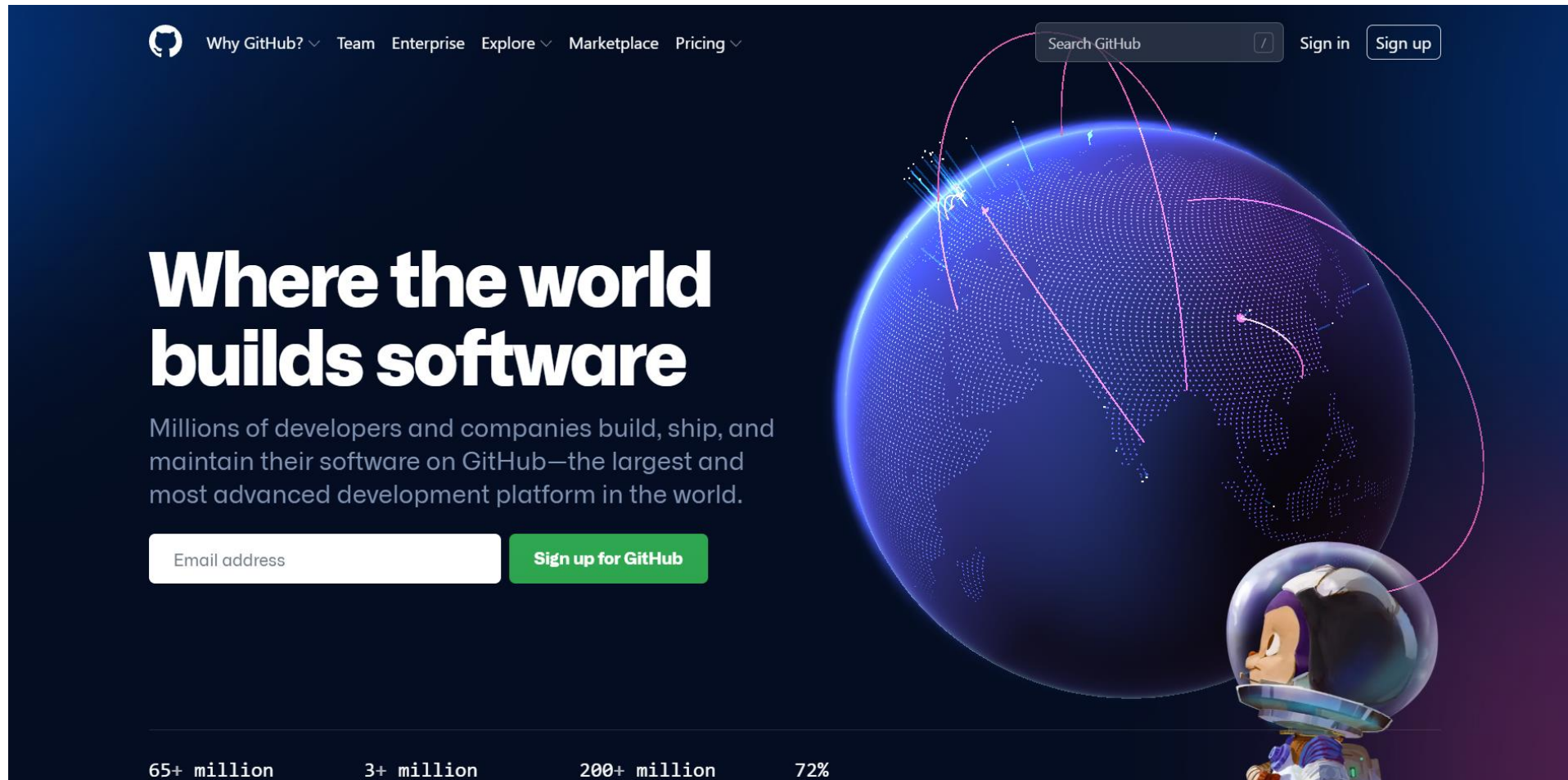
# Check-in and check-out from a version repository



# Branching and merging

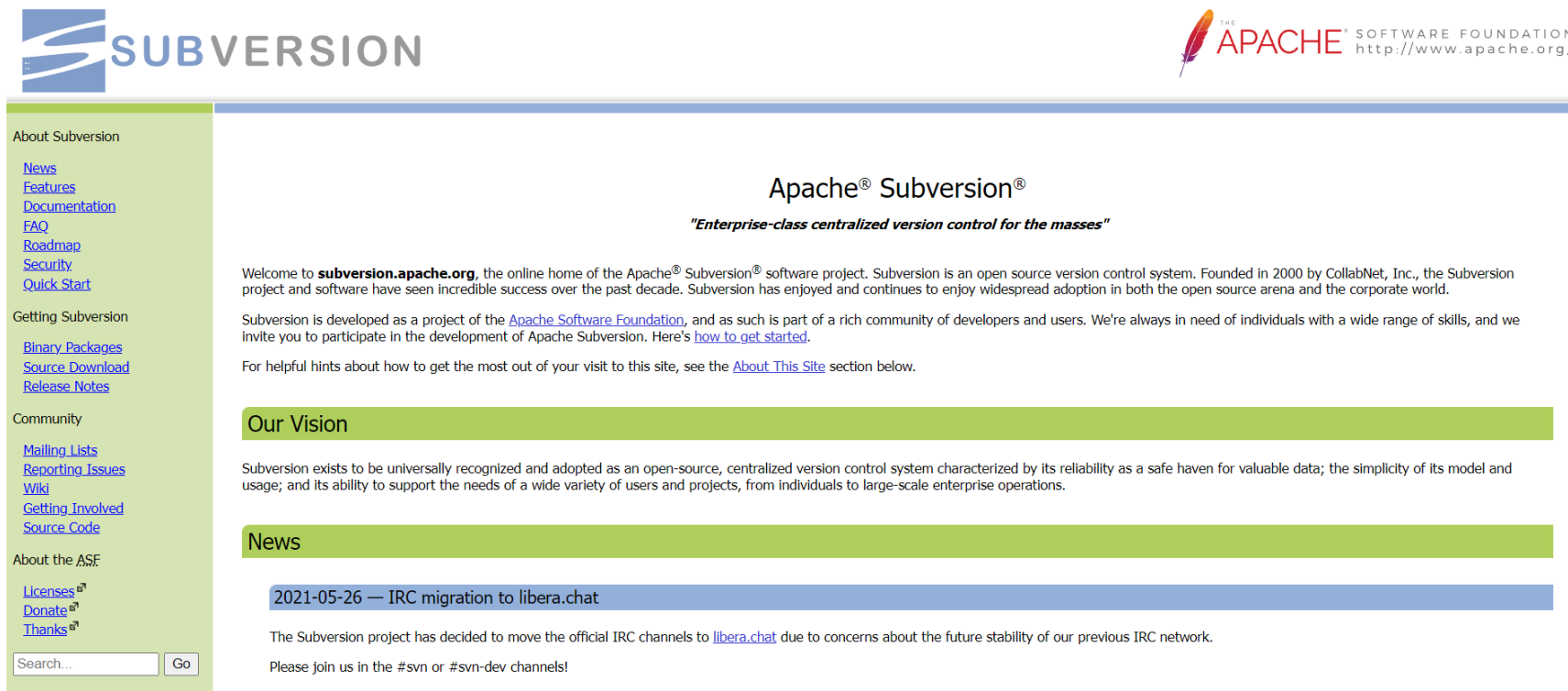


# GitHub



# Apache Subversion

**Apache Subversion** is fortunate to have a community that values documentation. This page contains pointers to various sources of documentation aimed at Subversion users and developers both of Subversion and of third-party tools with which Subversion integrates. Much of this documentation is itself freely licensed, so if you spot errors in them, please feel free to submit corrections to the relevant owners of the documentation.



The screenshot shows the Apache Subversion website homepage. At the top left is the Subversion logo, and at the top right is the Apache Software Foundation logo with the URL <http://www.apache.org/>. A left sidebar contains navigation links under categories: 'About Subversion' (News, Features, Documentation, FAQ, Roadmap, Security, Quick Start), 'Getting Subversion' (Binary Packages, Source Download, Release Notes), 'Community' (Mailing Lists, Reporting Issues, Wiki, Getting Involved, Source Code), and 'About the ASF' (Licenses, Donate, Thanks). Below these is a search bar. The main content area has the title 'Apache® Subversion®' with the tagline *"Enterprise-class centralized version control for the masses"*. It includes a welcome message, a description of the project as part of the Apache Software Foundation, and a link to 'About This Site'. Below this are sections for 'Our Vision' and 'News'. The 'News' section features a header for '2021-05-26 — IRC migration to libera.chat' and text stating the project's decision to move IRC channels to [libera.chat](#) due to concerns about the previous IRC network's stability, and an invitation to join the #svn or #svn-dev channels.

# สรุป (Summary)

---

- ศึกษาเรียนรู้เกี่ยวกับ
  - การจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์
  - การจัดทำเอกสารข้อเสนอให้เปลี่ยนระบบ (Change Request Form)
  - การจัดการเวอร์ชันซอฟต์แวร์ (Software Version Management)
  - เครื่องมือสำหรับการจัดการเปลี่ยนแปลง (Change Management Tool)



# กิจกรรมท้ายบท

---

- จงบอกความหมายหรือคำจำกัดความของการจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์ (Configuration Management) ตามความเข้าใจ
- จงบอกถึงคุณสมบัติที่ควรจะมีในเครื่องมือสำหรับสนับสนุนงานกระบวนการจัดการเปลี่ยนแปลง (Change Management Process)
- การกำหนดเวอร์ชันมีที่ลักษณะและมีอะไรบ้าง จงอธิบายพร้อมกับยกตัวอย่างในแต่ละลักษณะตามความเข้าใจ

# เอกสารอ้างอิง

---

- กิตติ ภัทต์วัฒนกุล, วิศวกรรมซอฟต์แวร์ (Software Engineering), กรุงเทพฯ: เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2552.
- Lan Sommerville, Software Engineering Ninth Edition, Pearson Education, Inc., publishing as Addison-Wesley, 2011.