

# บทที่ 12 การจัดการและควบคุมการเปลี่ยนแปลง ในการพัฒนางานด้านซอฟต์แวร์ (Configuration Management)

วิชา วิศวกรรมซอฟต์แวร์ (04-06-306)



# วัตถุประสงค์การเรียนรู้

---

- เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์
- เพื่อให้ผู้เรียนสามารถบอกถึงการประยุกต์ใช้กระบวนการดังกล่าว กับ การพัฒนางานด้านซอฟต์แวร์ในเบื้องต้นได้



# หัวข้อ (Agenda)

---

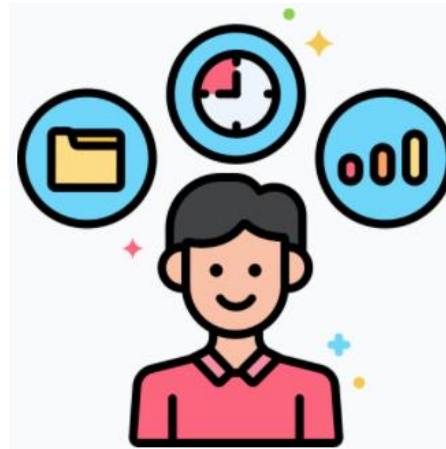
- บทนำ (Overview)
- การจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์
- การจัดทำเอกสารข้อเสนอให้เปลี่ยนระบบ (Change Request Form)
- การจัดการเวอร์ชันซอฟต์แวร์ (Software Version Management)
- เครื่องมือสำหรับการจัดการเปลี่ยนแปลง (Change Management Tool)
- สรุป (Summary)



# บทนำ (Overview)

การเปลี่ยนแปลงของระบบซอฟต์แวร์ใน  
ระหว่างการพัฒนาและใช้งาน

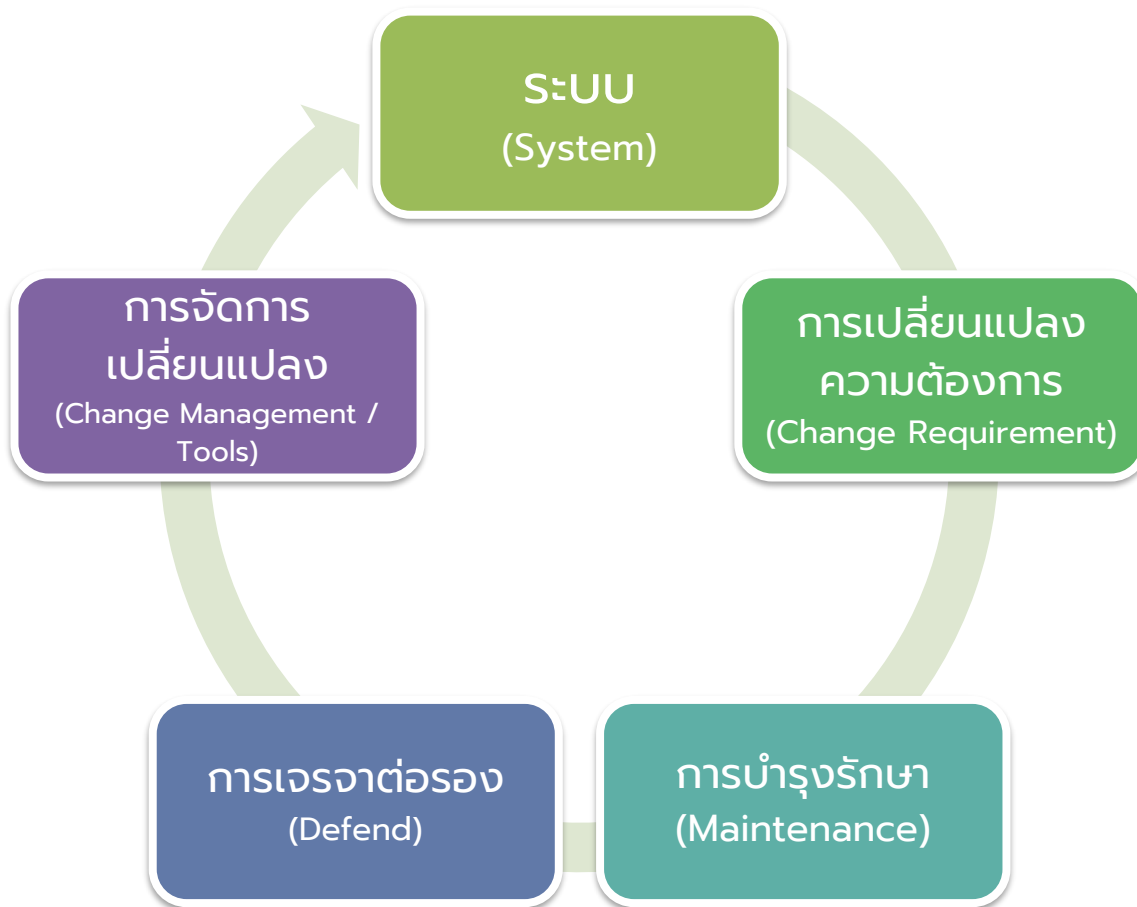
การจัดการและควบคุมการเปลี่ยนแปลง  
(Configuration management: CM)



การเปลี่ยนแปลงความต้องการระบบ

CM เป็นประโยชน์ต่อการดำเนินการของ  
โครงการ

# การจัดการและควบคุมการเปลี่ยนแปลงในการพัฒนางานด้านซอฟต์แวร์ (Configuration Management)

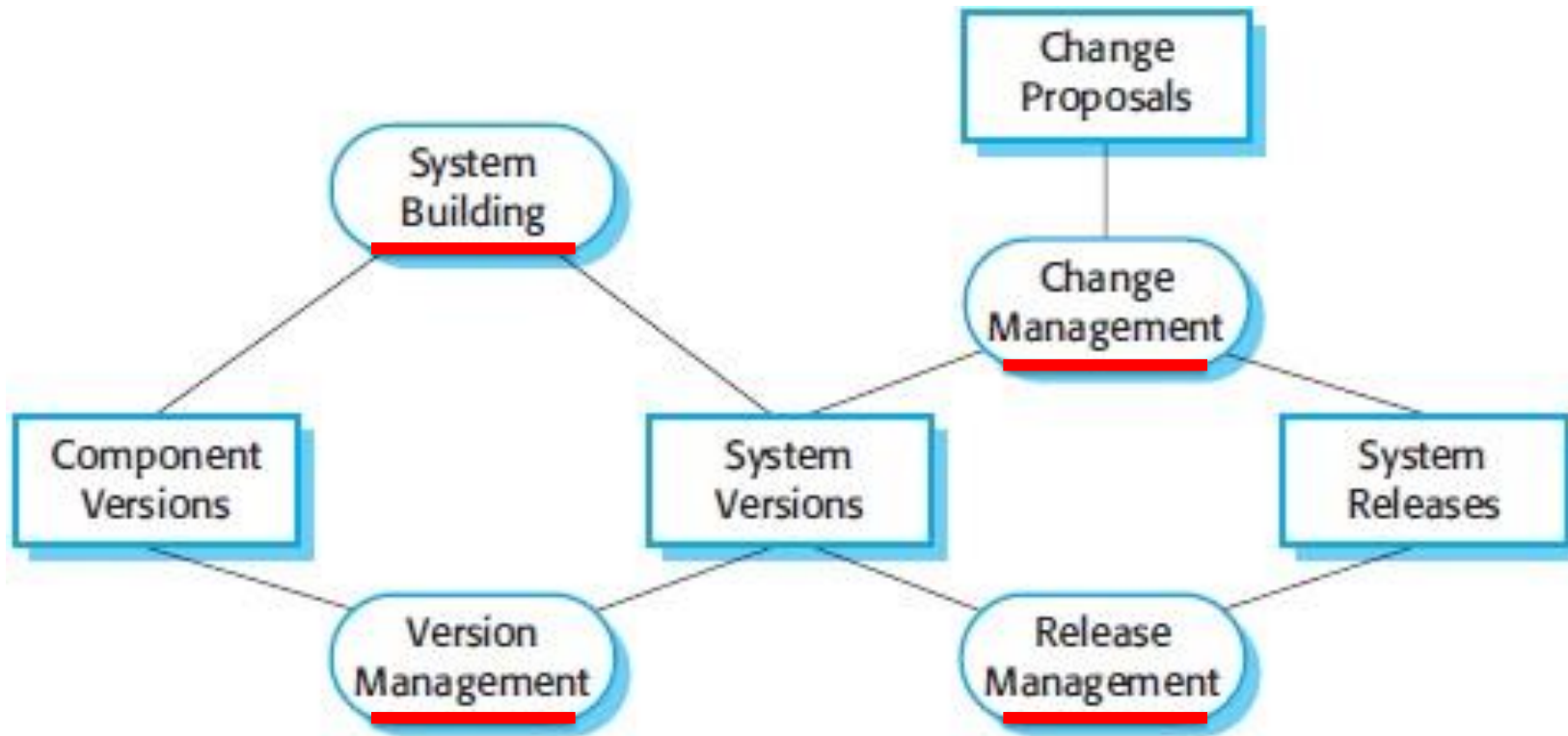


- เป็นการวิเคราะห์ต้นทุนและผลตอบแทนที่ได้รับจาก proposal ให้เปลี่ยนแปลงระบบ
- พิจารณาสິงที่ต้องเปลี่ยนแปลงคําค่ากับเงินลงทุนที่เสียไปหรือไม่ พร้อมกับตรวจสอบรายการที่ต้องเปลี่ยนแปลง



# กิจกรรมในการจัดการและควบคุมการเปลี่ยนแปลง (Configuration management activities)

- การจัดการและควบคุมการเปลี่ยนแปลงในผลิตภัณฑ์ที่เป็นระบบซอฟต์แวร์ ประกอบด้วย 4 กิจกรรมที่เกี่ยวข้องกัน ดังนี้





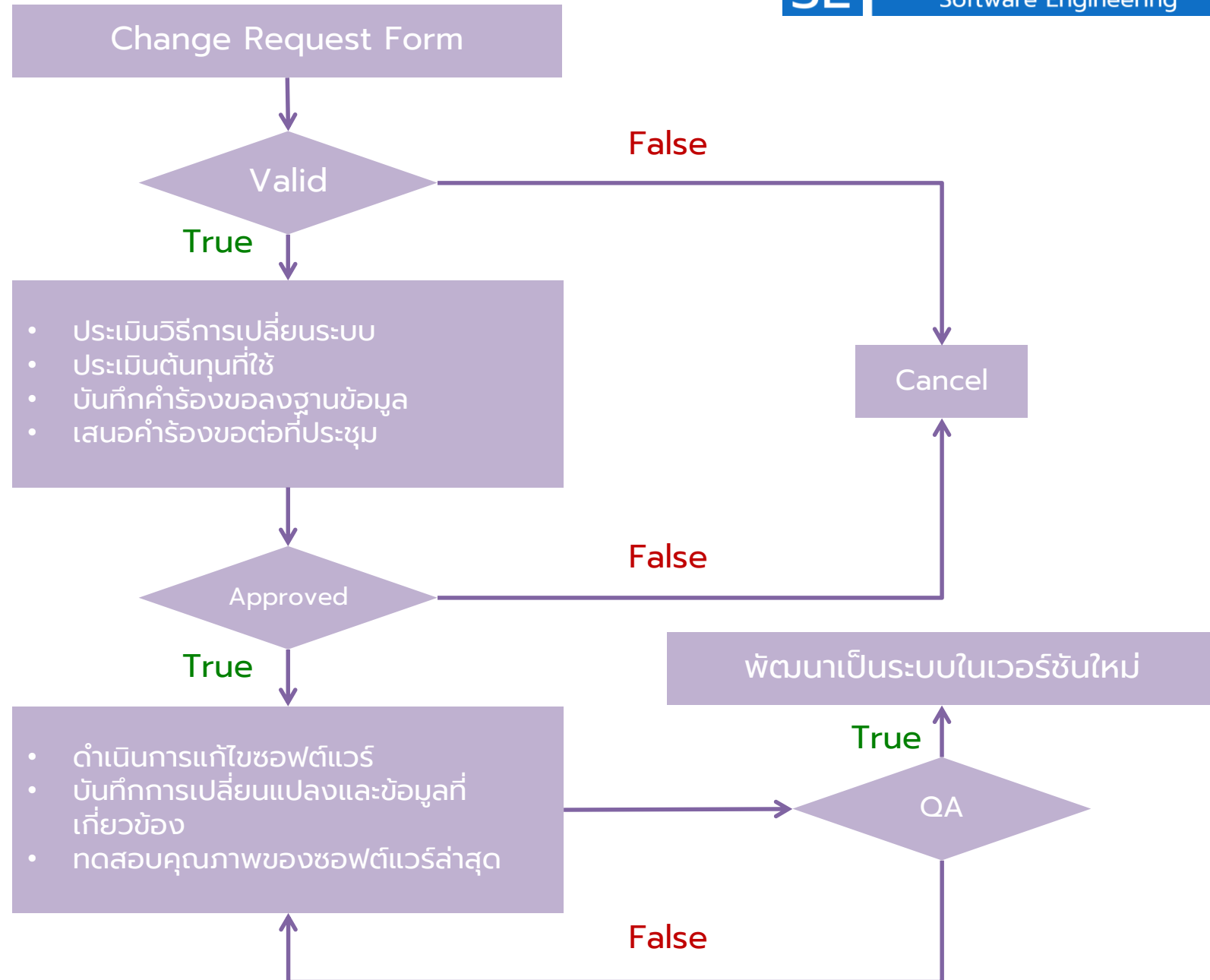
# คำศัพท์ CM (CM terminology)

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.



# กระบวนการ การจัดการ การเปลี่ยนแปลง

## Change Management Process





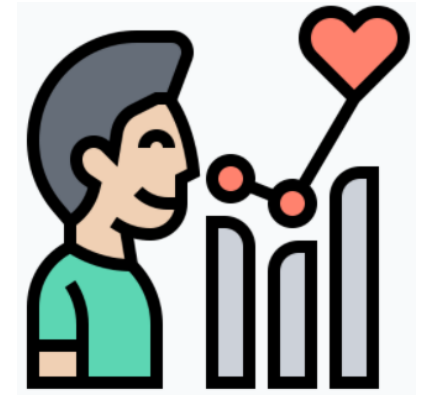
# กระบวนการจัดการการเปลี่ยนแปลง (ต่อ)

- **ปัจจัย**สำคัญที่ควรนำมาพิจารณาร่วมในการ**ตัดสินใจ**ในการ**เปลี่ยนแปลง**ที่เกิดขึ้นควรได้รับการ**อนุมัติ**หรือไม่:
  - *ผลจากการไม่ดำเนินการเปลี่ยนแปลง (The consequences of not making the change)*
    - ถ้ามีการ**ประเมินคำขอเปลี่ยนแปลง** ต้องมีการพิจารณาว่า**เกิดสิ่งใด**ขึ้นในกรณีที่ไม่มีการ**เปลี่ยน**
    - ถ้าการเปลี่ยนแปลงเกี่ยวข้องกับ**ความล้มเหลว**ของ**ระบบ** ต้องคำนึงถึง**ความร้ายแรง**ของความล้มเหลวด้วย
    - ถ้า**ความล้มเหลว**ของระบบทำให้**ระบบหยุดทำงาน** ถือว่า**ร้ายแรง**มากและอาจ**ขัดขวาง**การดำเนินงานในการใช้งานระบบ
    - ถ้า**ความล้มเหลว**มีผลกระทบเพียง**เล็กน้อย** เช่น สีแสดงผลไม่ถูกต้อง อาจ**ไม่สำคัญ**ที่ต้องแก้ปัญหาโดยเร็ว สามารถระบุระดับความสำคัญ**ต่ำ**



# กระบวนการจัดการการเปลี่ยนแปลง (ต่อ)

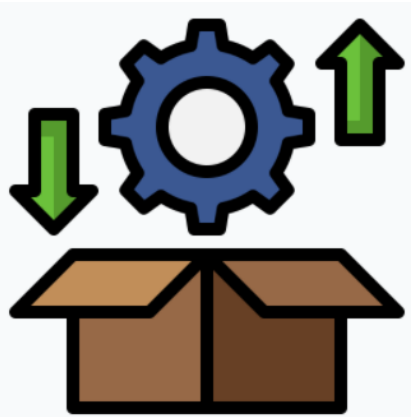
- ประโยชน์ที่ได้รับจากการเปลี่ยนแปลง  
(The benefits of the change)
  - บางครั้งการเปลี่ยนแปลงอาจเป็นประโยชน์ต่อผู้เสนอขอฯ เป็นหลัก หรือ ต่อผู้ใช้งานในระบบเป็นหลัก
- จำนวนผู้ใช้ที่ได้รับผลกระทบจากการเปลี่ยนแปลง  
(The number of users affected by the change)
  - หากมีผู้ใช้งานจำนวนน้อยที่ได้รับผลกระทบ การเปลี่ยนแปลงนั้นอาจกำหนดลำดับความสำคัญต่ำ
  - ไม่แนะนำให้มีการเปลี่ยนแปลง หากส่งผลกระทบต่อผู้ใช้งานระบบส่วนใหญ่



# กระบวนการจัดการการเปลี่ยนแปลง (ต่อ)



- *งบประมาณที่ใช้ในการเปลี่ยนแปลง*  
(The costs of making the change)
  - หากการเปลี่ยนแปลง**ส่งผลกระทบต่อส่วนประกอบ**ต่าง ๆ ของระบบ (เพิ่มโอกาส**เจอ BUG** ใหม่ ๆ)
  - หากการเปลี่ยนแปลง**ใช้เวลา**ในการดำเนินการ**นาน**
  - กรณีดังกล่าวอาจ**ปฏิเสธ** เนื่องจากใช้**งบประมาณสูง**



- *รอบการเผยแพร่ผลิตภัณฑ์*  
(The product release cycle)
  - หาก**ซอฟต์แวร์เวอร์ชันล่าสุด**มีการ**เปิดตัว**กับลูกค้าไปในเร็ว ๆ นี้ อาจ**ชะลอ**การนำ**การเปลี่ยนแปลง**ไปดำเนินการจนกว่าจะมีการ**เปิดตัวตามแผน**ครั้งถัดไป

# การจดทำเอกสารข้อเสนอการเปลี่ยนแปลง (Change Request Form)

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

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



# ตัวอย่าง เอกสารข้อเสนอการเปลี่ยนแปลง

## 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:**

เอกสารข้อเสนอ  
การเปลี่ยนแปลง  
(Change Request Form)

# 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 หรือ Logs

# การจัดการเวอร์ชันซอฟต์แวร์ (Software Version Management)

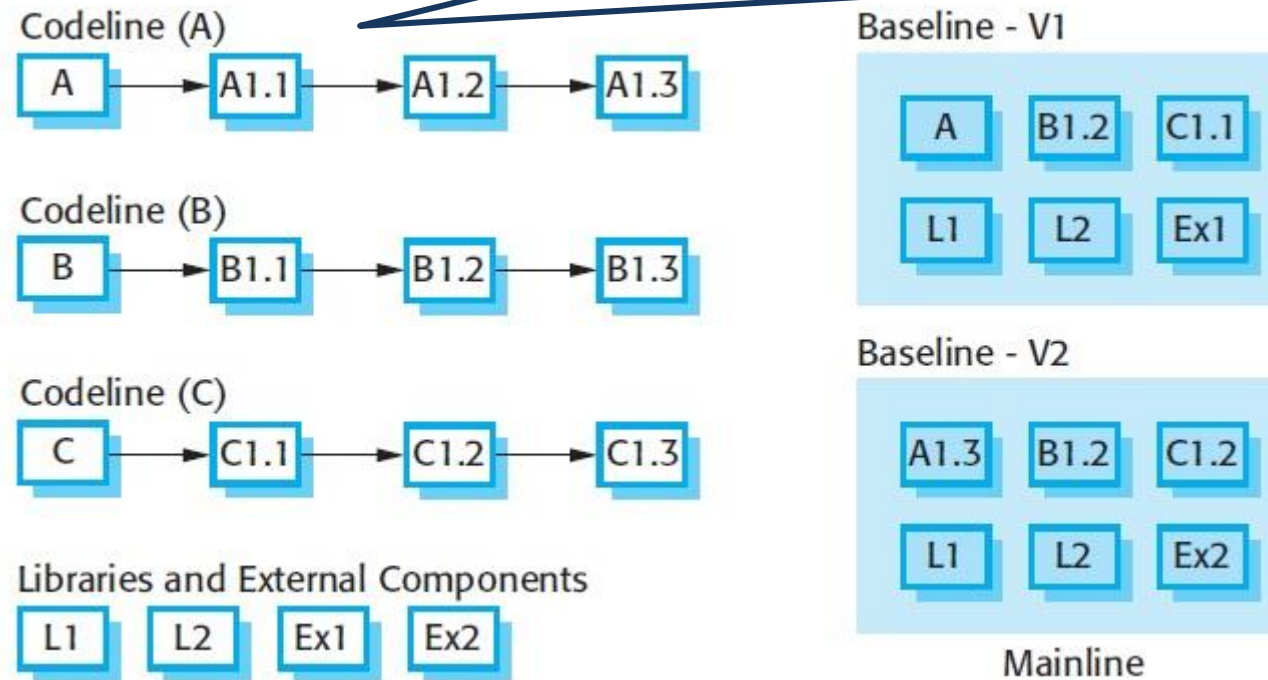
- เวอร์ชันซอฟต์แวร์ (Software Version)
  - รุ่นของระบบหรือซอฟต์แวร์
    - Microsoft Office 2015, ASP.NET 2.0
- วัตถุประสงค์ของการกำหนดรุ่นของซอฟต์แวร์
  - พังค์ชันการทำงานต่างกัน
  - ปรับปรุงคุณภาพของซอฟต์แวร์เดิม
  - สนับสนุนอุปกรณ์ที่แตกต่างกัน
- การจัดการเวอร์ชัน (Version Management)
  - กระบวนการจำแนกและติดตามการเปลี่ยนแปลงเวอร์ชันของระบบ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





# โค้ดไลน์และเบสไลน์ (Codelines and baselines)

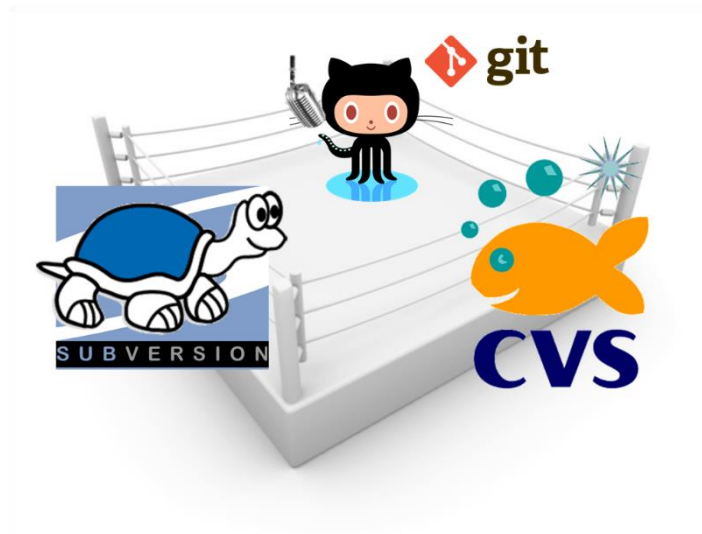
**โค้ดไลน์ (Codeline)** เป็นลำดับเวอร์ชันของรหัสคำสั่งกับเวอร์ชันล่าสุด โดยลำดับมาจากเวอร์ชันก่อนหน้า ใช้กับการจัดการเวอร์ชันกับคอมโพเนนต์ของระบบ เพื่อให้มีเวอร์ชันที่แตกต่างกันในแต่ละคอมโพเนนต์



**เบสไลน์ (Baseline)** เป็นการนิยามของระบบให้มีความเฉพาะมากขึ้น

# เครื่องมือการบริหารจัดการเวอร์ชัน (Version Management Tools)

- ระบบการควบคุมเวอร์ชัน หรือ ระบบการควบคุมซอร์สโค้ด
- สามารถดำเนินการด้วยเครื่องมือต่าง ๆ เพื่อระบุ จัดเก็บและควบคุมการเข้าถึงคอมโพเนนต์ต่าง ๆ จากต่างแหล่งจัดเก็บได้ด้วยเครื่องมือ ดังนี้
  - CVS, Subversion และ Git



# การจัดการเวอร์ชัน (Version management)

- คุณสมบัติของระบบการจัดการเวอร์ชัน

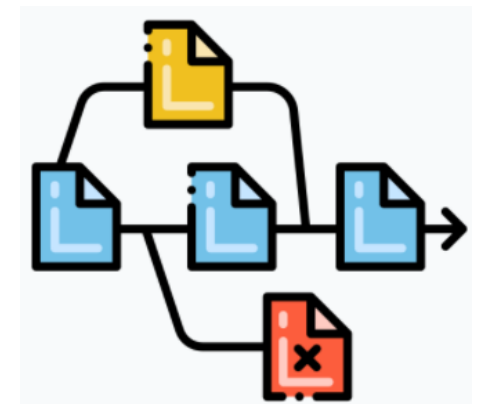
การระบุเวอร์ชันและเผยแพร่  
(Version and release identification)

การจัดการแหล่งจัดเก็บ  
(Storage management)

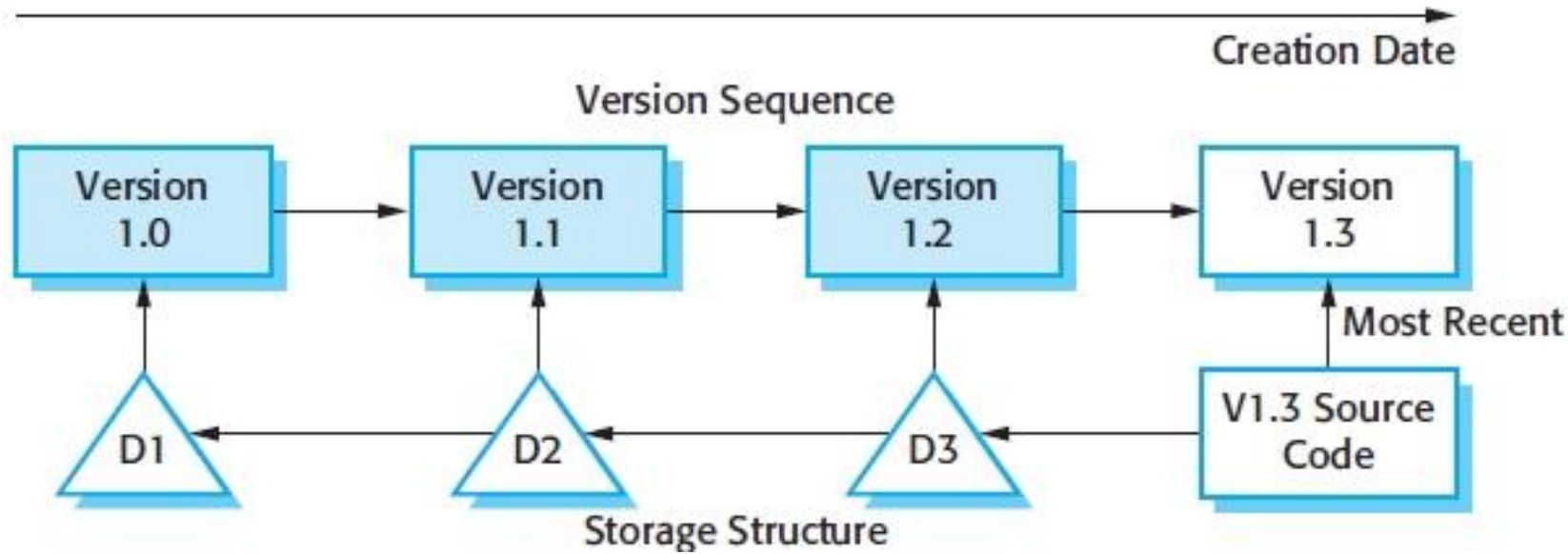
การบันทึกประวัติการเปลี่ยนแปลง  
(Change history recording)

การพัฒนาแบบอิสระ  
(Independent development)

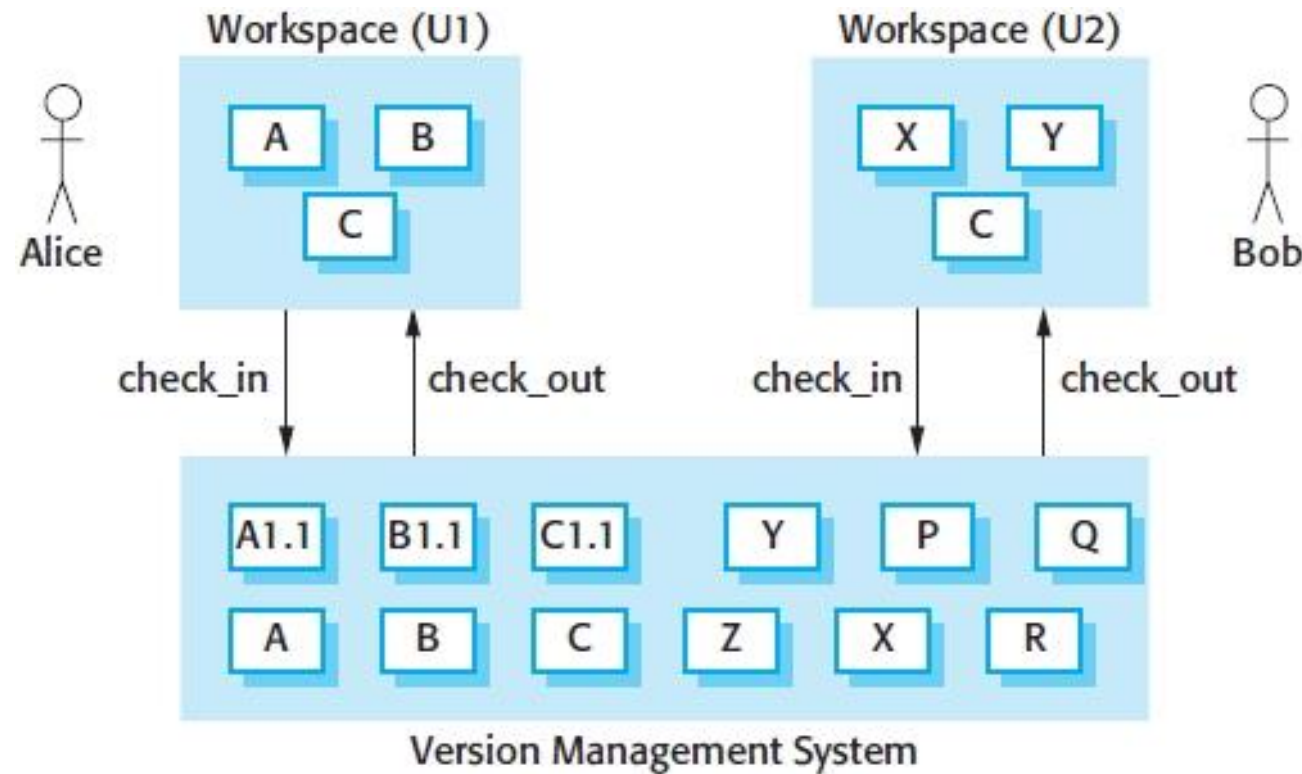
การสนับสนุนโครงการ  
(Project support)



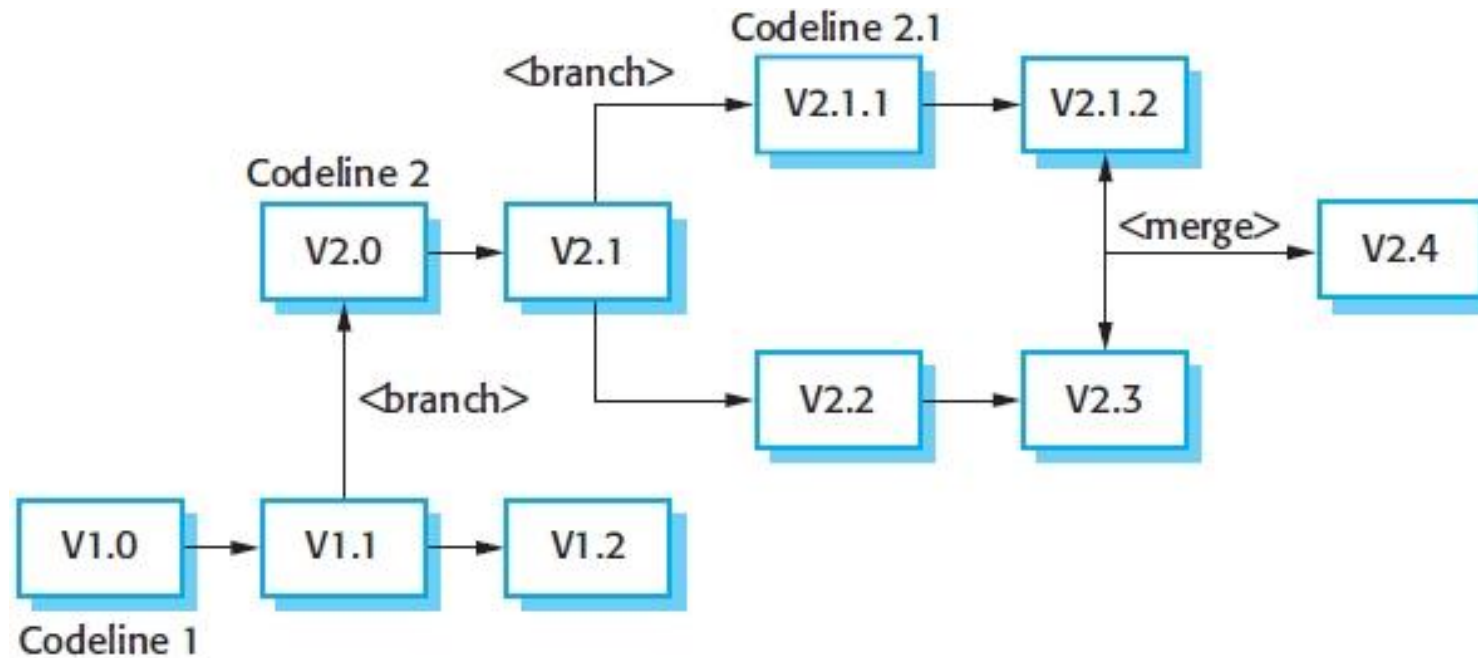
# การจัดการแหล่งจัดเก็บด้วยเดลต้า (Storage management using deltas)



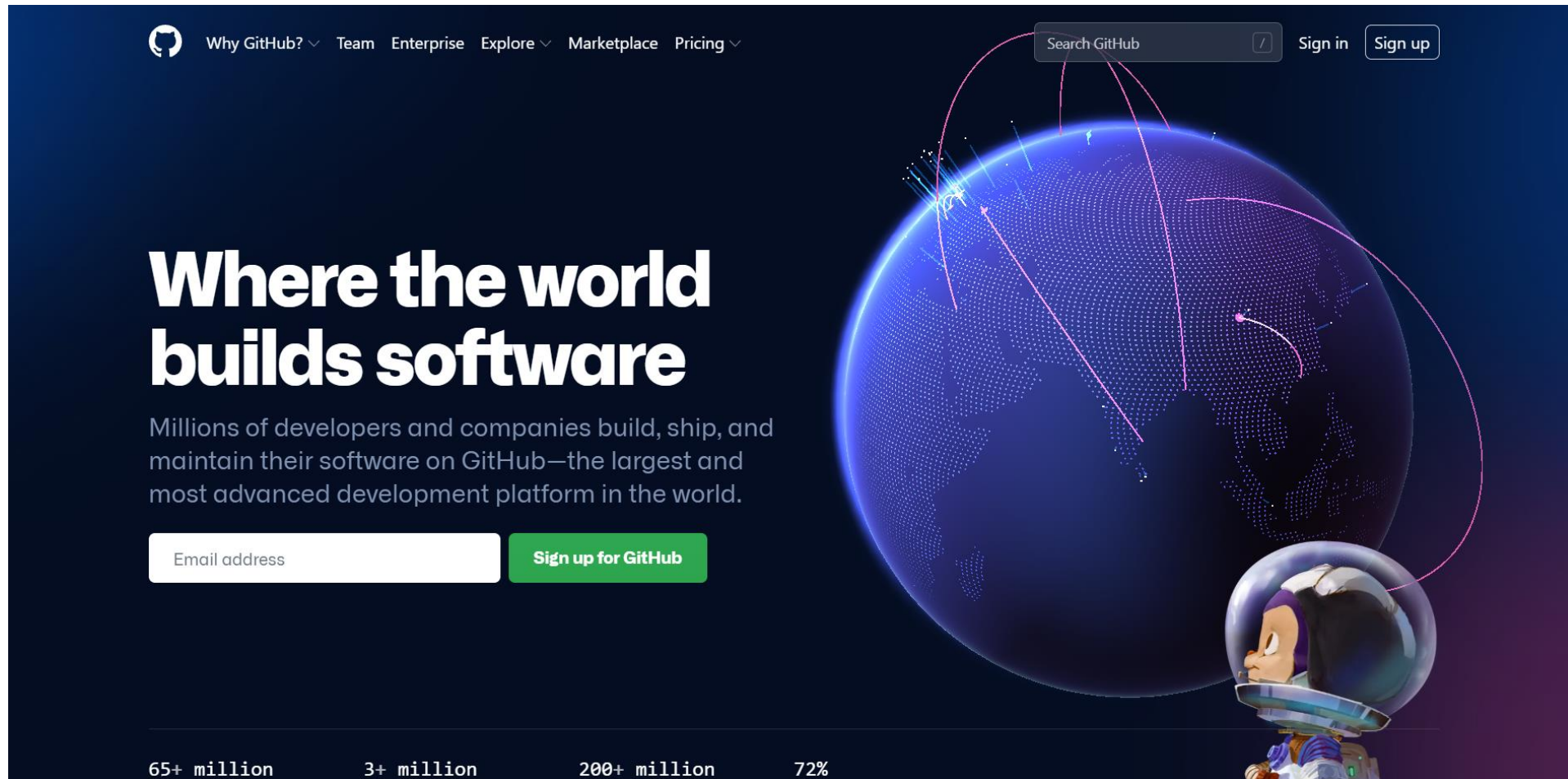
# การเช็คอินและเช็คเอาท์จากแหล่งจัดเก็บเวอร์ชัน (Check-in and check-out from a version repository)



# การแตกแขนงและการรวมเวอร์ชัน (Branching and merging)



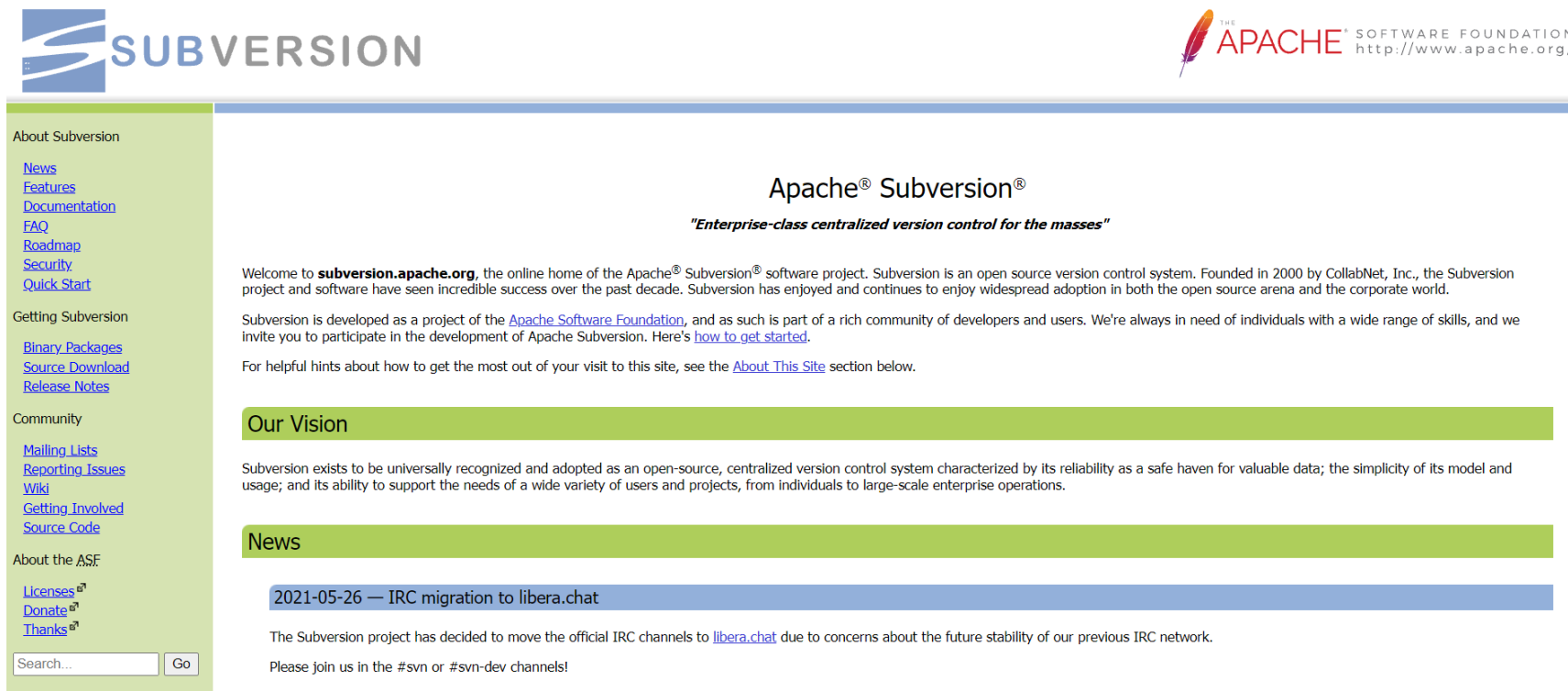
# GitHub





# 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 three categories: 'About Subversion' (News, Features, Documentation, FAQ, Roadmap, Security, Quick Start), 'Getting Subversion' (Binary Packages, Source Download, Release Notes), and 'Community' (Mailing Lists, Reporting Issues, Wiki, Getting Involved, Source Code). Below these is 'About the ASF' (Licenses, Donate, Thanks) and 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 two green section headers: 'Our Vision' and 'News'. The 'News' section features a blue header for '2021-05-26 — IRC migration to libera.chat' and text stating the project's decision to move IRC channels to [libera.chat](#) and a request 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.