

# Become the Software Engineer

“สู่ความเป็นวิศวกรซอฟต์แวร์”

By Kanjana Eiamsaard, Version 2.0 Created 2021-12-2

# Topics Covered

- ความเป็นมาและความสำคัญ
- กระบวนการทางซอฟต์แวร์ (Software processes)
- เครื่องมือสำหรับวิศวกรซอฟต์แวร์
  - Analysis and Design tool
  - Integrated Development Environment : IDE
  - Version Control System
  - Bug tracking system
  - Continuous Integration & Continuous Deployment: CI/CD
  - Project management tool
- จรรยาบรรณของวิศวกรซอฟต์แวร์

# ความเป็นมาและความสำคัญ

- พิจารณาซอฟต์แวร์โดยเทียบกับปัจจัยสี่
  - อาหาร ที่อยู่อาศัย เครื่องนุ่งห่ม ยารักษาโรค
  - มีอะไรอีกบ้างที่ไม่เกี่ยวข้องกับซอฟต์แวร์ ?
- ถ้ามีศาสตร์แขนงหนึ่งที่**นำไปสู่ความสำเร็จ**ของการพัฒนาโครงการด้านซอฟต์แวร์ จะเอาไหม ?
  - ไม่ใช่แค่ความรู้ทางเทคโนโลยีเท่านั้น
  - การได้มาซึ่งความต้องการของลูกค้า
  - การทำงานร่วมกันภายในทีมพัฒนา
  - การบริหารโครงการที่มีความซับซ้อน
  - การบริหารความเสี่ยง
  - การควบคุมคุณภาพ
  - ฯลฯ

# Digital era เกือบทุกสิ่งบนโลกใบนี้ ดำเนินไปด้วยระบบคอมพิวเตอร์



<https://www.za.in.th/content/67038/>



# Digital era cont.

หุ่นยนต์ผ่าตัด แผลเล็กแม่นยำ แพทย์ควบคุมจากทางไกล



<https://www.iurban.in.th/activity/robot-surgeons/>

Zomato ประกาศรับเงินเพิ่มทุนอีก 210 ล้านดอลลาร์ จาก Alipay สิงคโปร์



<https://www.blognone.com/node/105854>

# นิยาม

- ซอฟต์แวร์ (Software) หมายถึง ชุดคำสั่งที่ใช้สั่งการคอมพิวเตอร์ ซึ่งเป็นสิ่งสมมติที่ซับซ้อน เติบโตได้อย่างรวดเร็วและไม่มีข้อจำกัดทางกายภาพ แบ่งได้ 2 ชนิด
  - ซอฟต์แวร์ทั่วไป (Generic Software)
  - ซอฟต์แวร์เฉพาะกลุ่ม (Customized Software)
- ทั้งสองกลุ่มต่างกันว่า “ที่มาของความต้องการ”
- เหมือนกันใน 3 ด้าน
  - ความคุ้มค่าในการลงทุน (Cost)
  - คุณภาพของซอฟต์แวร์ (Quality)
  - ควบคุมเวลา (Time) ที่ใช้ในการผลิต •

## นิยาม (ต่อ)

- วิศวกรรมซอฟต์แวร์ (Software Engineering) คือ  
การพัฒนาซอฟต์แวร์ที่ประยุกต์ความรู้ ทฤษฎี เครื่องมือทางวิทยาการคอมพิวเตอร์มาใช้ในกระบวนการผลิต รวมถึงการบริหารจัดการกิจกรรมที่เกิดขึ้นตลอดช่วงชีวิตการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) ให้อยู่ภายใต้วัตถุประสงค์ของการนำไปใช้ คุณภาพ ความคุ้มค่า และเวลาที่กำหนด

## กระบวนการทางซอฟต์แวร์ (Software processes)

- Software specification
- Software development
- Software validation
- Software evolution





# กระบวนการทางซอฟต์แวร์

- หมายถึง ลำดับกิจกรรมทั้งหมดที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ แบ่งเป็น 4 กระบวนการหลัก
  - การระบุความต้องการของซอฟต์แวร์ (Software specification)
    - กิจกรรมที่ลูกค้าและวิศวกรร่วมกันรวบรวมข้อมูล วิเคราะห์ข้อกำหนด รายละเอียดของซอฟต์แวร์ที่จะสร้างขึ้น ข้อจำกัดในการนำไปใช้
  - การพัฒนาซอฟต์แวร์ (Software development)
    - กิจกรรมที่เกี่ยวข้องกับการออกแบบโครงสร้าง ส่วนประกอบ ตลอดจนการเขียนโปรแกรม

## กระบวนการทางซอฟต์แวร์ (ต่อ)

- การทวนสอบซอฟต์แวร์ (Software validation)
  - กิจกรรมที่ทำให้มั่นใจได้ว่าซอฟต์แวร์ที่สร้างขึ้น เป็นไปตามที่ลูกค้า/ผู้ใช้งานต้องการ และทำงานได้อย่างถูกต้อง
- การวิวัฒนาการของซอฟต์แวร์ (Software evolution)
  - กิจกรรมที่เกี่ยวข้องกับการปรับปรุงซอฟต์แวร์ เพื่อให้สอดคล้องกับสิ่งที่เปลี่ยนแปลงไป รวมไปถึงการจัดสิ่งที่อาจเป็นอุปสรรคสำหรับการปรับปรุงในอนาคต

## กระบวนการทางซอฟต์แวร์ (ต่อ)

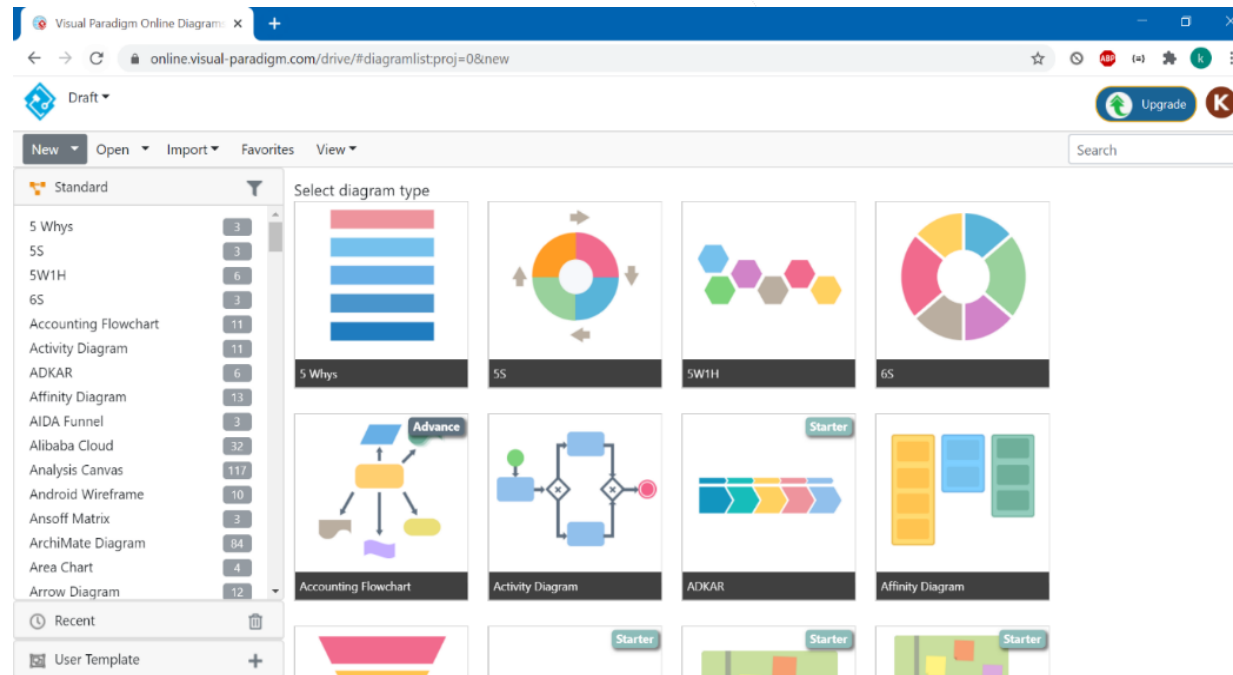
- เมื่อซอฟต์แวร์ถูกสร้างอย่างมีกระบวนการ ควรมีลักษณะ 7 อย่าง
  1. การนำซอฟต์แวร์มาปรับปรุงได้ในภายหลัง (Maintainability)
  2. ความเป็นที่พึ่งได้ของซอฟต์แวร์ต่อผู้ใช้ (Dependability)
  3. ความมั่นคงในสินทรัพย์ทางดิจิทัลและทางกายภาพของผู้ใช้ (Security)
  4. การทำงานอย่างมีประสิทธิภาพของซอฟต์แวร์ (Efficiency)
  5. ซอฟต์แวร์เป็นที่ยอมรับได้ในกลุ่มผู้ใช้ (Acceptability)



เครื่องมือสำหรับวิศวกรซอฟต์แวร์

# Analysis and Design tool

เครื่องมือที่ช่วยให้นักพัฒนาสามารถวิเคราะห์และออกแบบฯ เช่น Online Visual Paradigm , StartUML, Sparxsystems

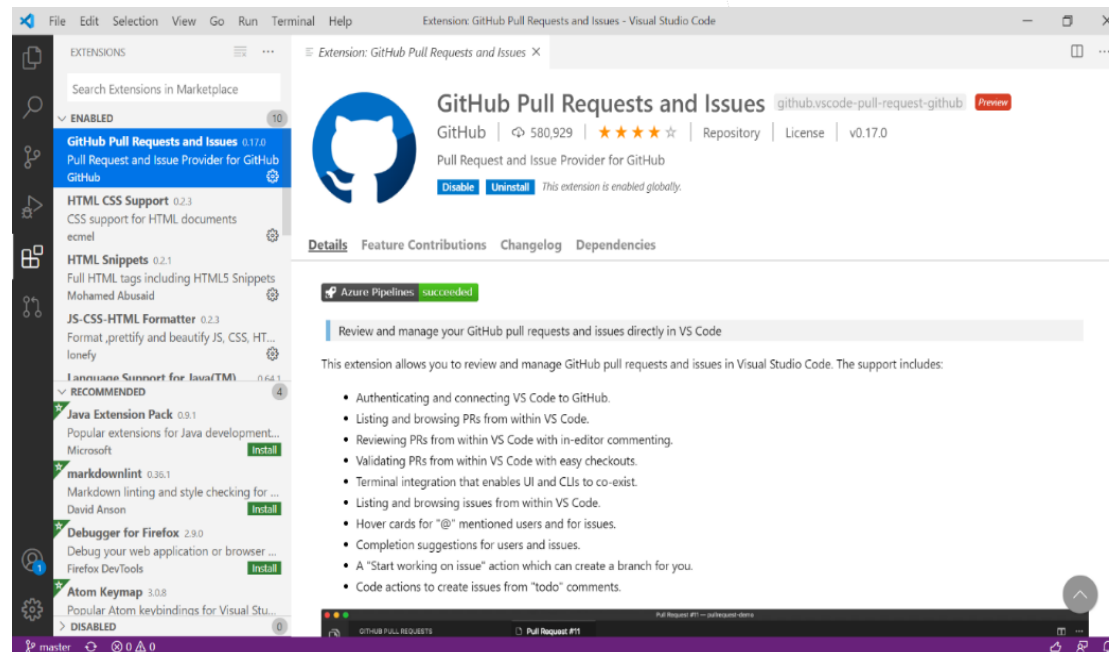


# Integrated Development Environment : IDE

สิ่งแวดล้อมสำหรับการพัฒนาแบบเบ็ดเสร็จ

รวบรวมสิ่งที่ต้องใช้ในการสร้าง การแปลภาษา และการประมวลผล ไว้เป็นหนึ่งเดียว

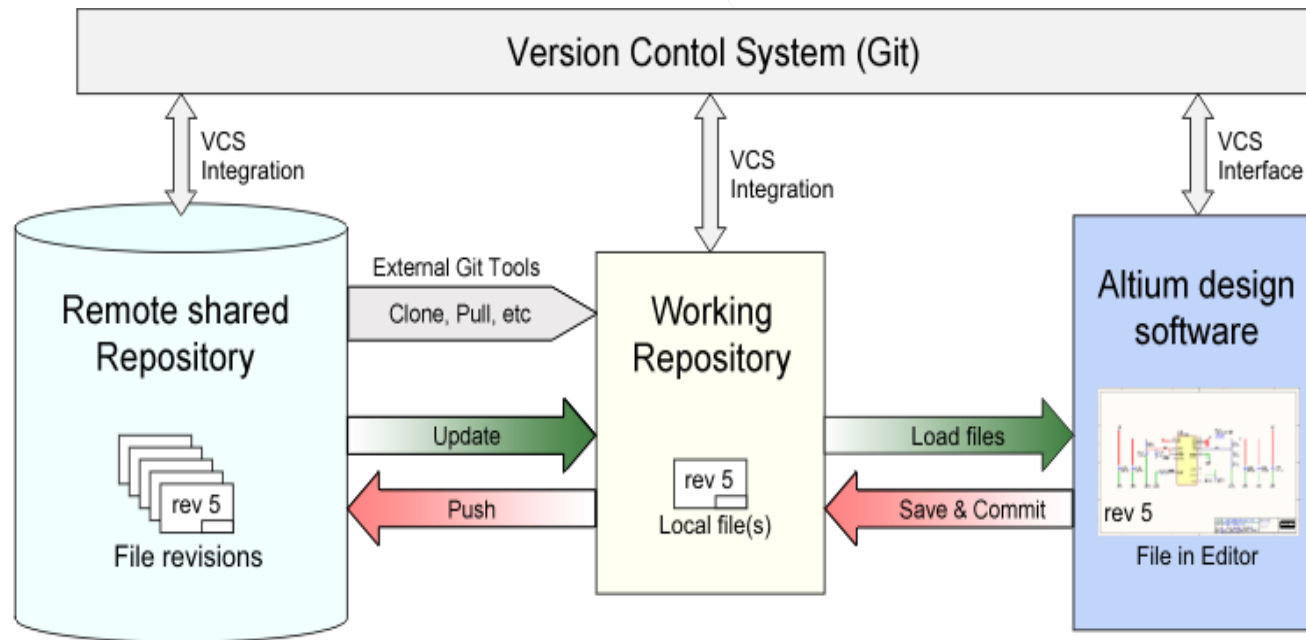
เช่น Sublime, Atom, Microsoft Visual Studio Code





# Version Control System

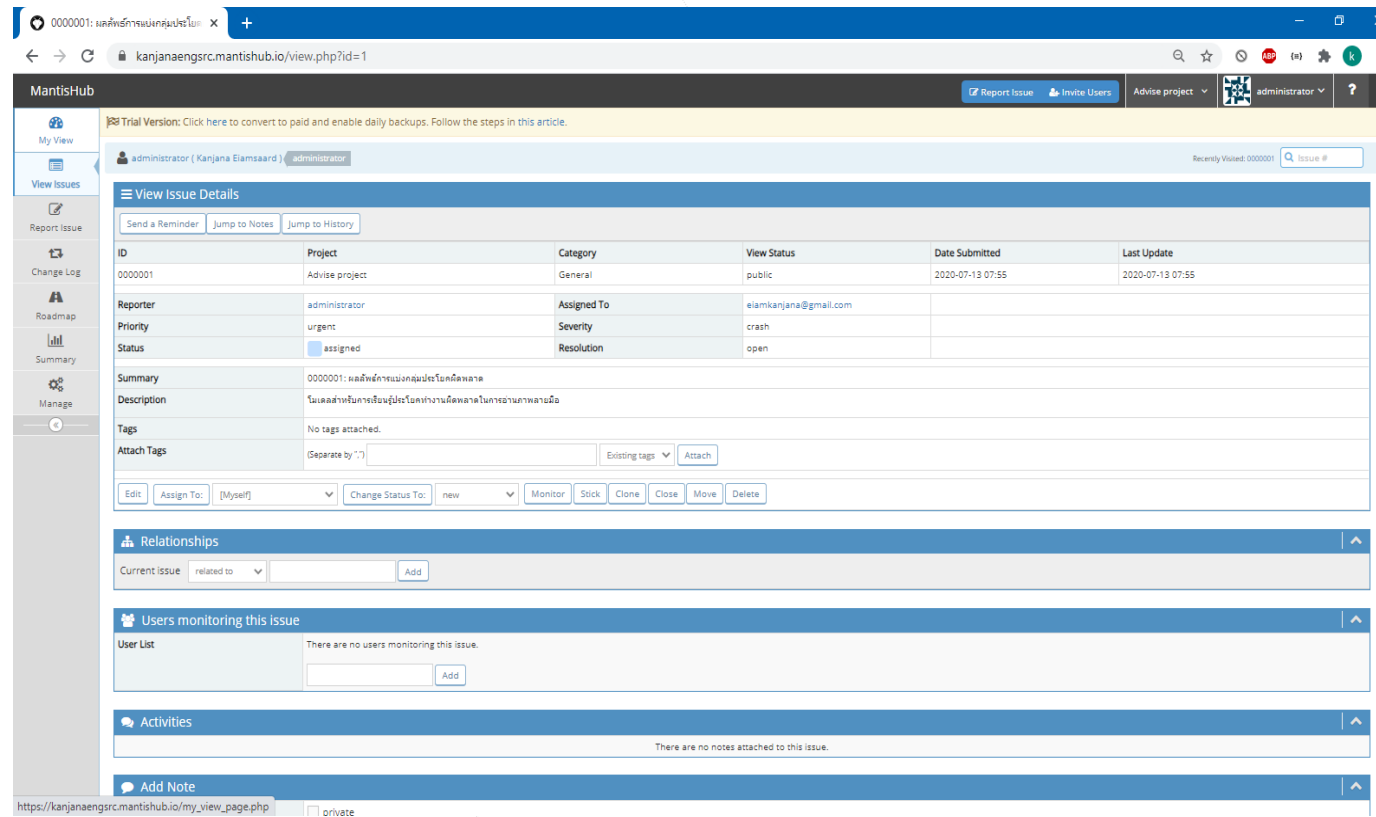
เครื่องมือสำหรับการควบคุมเวอร์ชันของซอฟต์แวร์ เช่น Git, CVS, SVN, Mercurial



ที่มา: <https://www.freecodecamp.org/news/the-beginners-guide-to-git-github/>

# Bug tracking system

เครื่องมือสำหรับควบคุมคุณภาพซอฟต์แวร์ เช่น JIRA, Trac, Bugzilla, Mantis Bug Tracker



The screenshot displays the Mantis Bug Tracker web interface. The browser address bar shows the URL `kanjanaengsrc.mantishub.io/view.php?id=1`. The interface includes a sidebar with navigation options like 'My View', 'View Issues', 'Report Issue', 'Change Log', 'Roadmap', 'Summary', and 'Manage'. The main content area is titled 'View Issue Details' and shows the following information for issue ID 0000001:

ID	Project	Category	View Status	Date Submitted	Last Update
0000001	Advise project	General	public	2020-07-13 07:55	2020-07-13 07:55

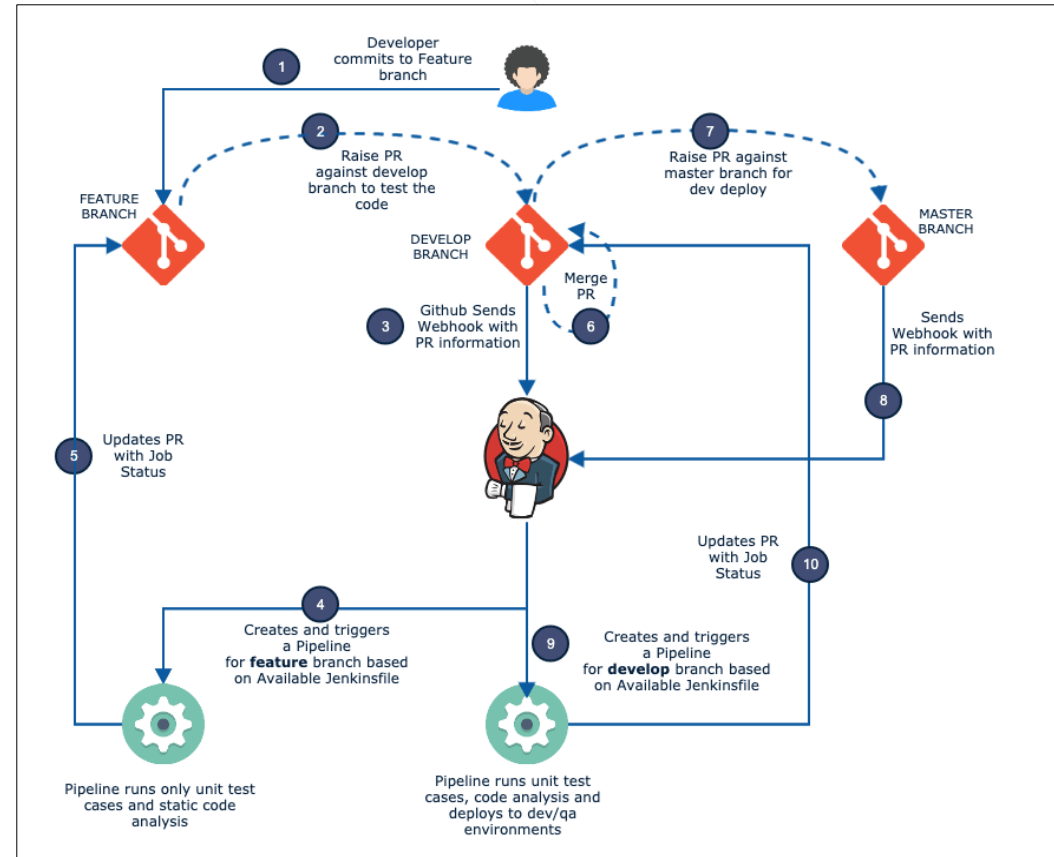
Additional details for the issue include:

- Reporter:** administrator
- Assigned To:** elamkajana@gmail.com
- Priority:** urgent
- Severity:** crash
- Status:** assigned
- Resolution:** open

The 'Summary' section contains the text: '0000001: แก้ไขการแจ้งเตือนระบบ'. The 'Description' section contains: 'โปรแกรมสำหรับแจ้งเตือนระบบทำงานผิดพลาดในทางความปลอดภัย'. The 'Tags' section shows 'No tags attached.' and the 'Attach Tags' section shows 'Existing tags' and an 'Attach' button. At the bottom, there are buttons for 'Edit', 'Assign To', 'Change Status To', 'Monitor', 'Stick', 'Clone', 'Close', 'Move', and 'Delete'. The 'Relationships' section shows 'Current issue related to' with an 'Add' button. The 'Users monitoring this issue' section shows 'There are no users monitoring this issue.' with a 'User List' and an 'Add' button. The 'Activities' section shows 'There are no notes attached to this issue.' with an 'Add Note' button. The footer shows the URL `https://kanjanaengsrc.mantishub.io/my_view_page.php` and a 'private' checkbox.

# Continuous Integration & Continuous Deployment: CI/CD

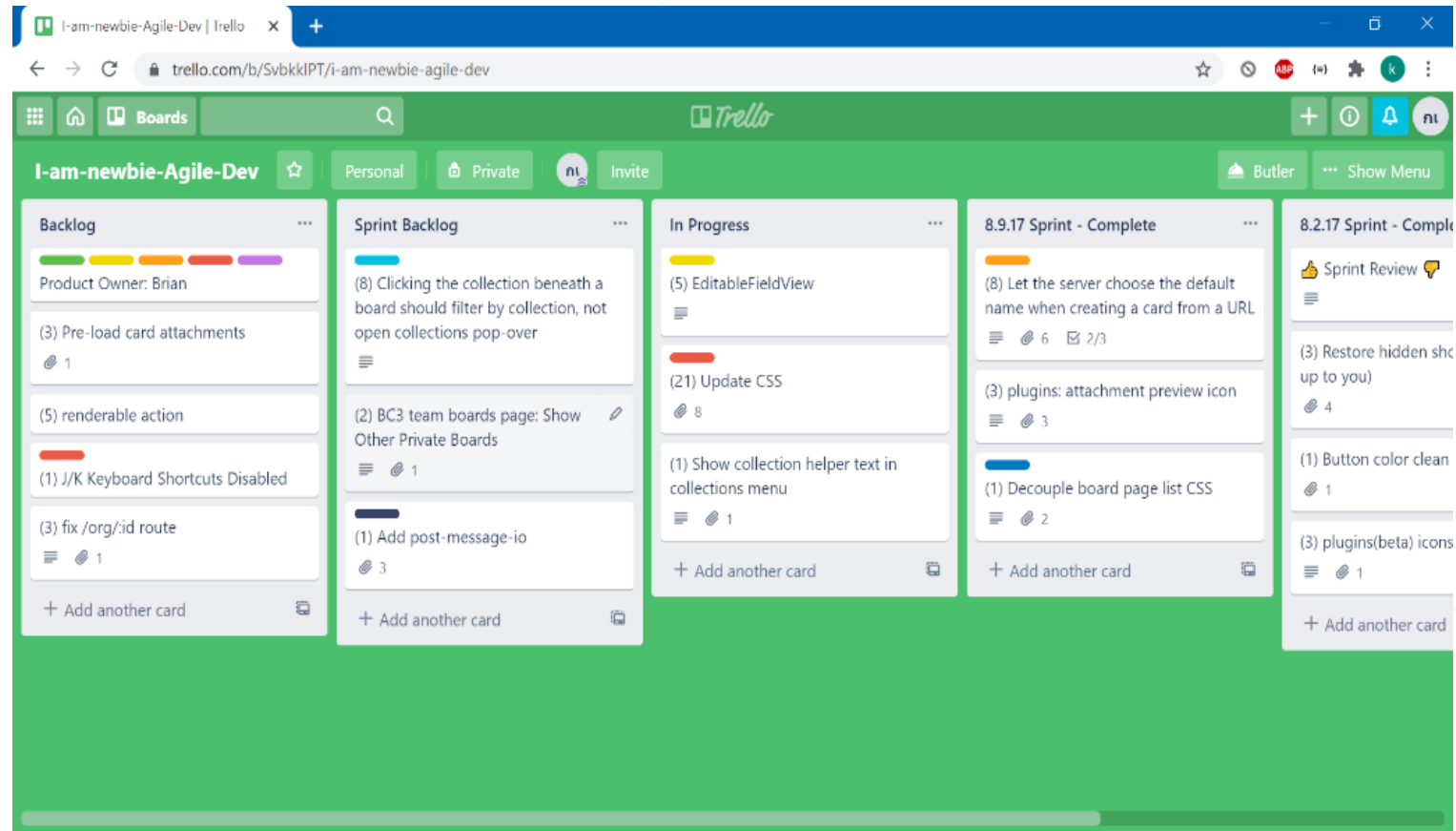
เครื่องมือสำหรับรวบรวมการพัฒนาและส่งมอบซอฟต์แวร์



ที่มา: <https://devopscube.com/jenkins-multibranch-pipeline-tutorial/>

# Project management tool

เครื่องมือสำหรับบริหาร คน เวลา งบประมาณ และงาน เช่น Kanbanboard, Lean Canvas และ Agile Sprint Board





# จรรยาบรรณของวิศวกรซอฟต์แวร์

# สมองดี...แล้วจิตใจล่ะ ดีไหม ?

- Confidentiality คือ การรักษาความลับของลูกค้าถึงแม้ว่าจะไม่ได้ทำข้อตกลงเป็นลายลักษณ์อักษร
- Competence คือ ไม่โกหกลูกค้าว่าสามารถทำงานที่ยากเกินความสามารถของตนได้
- Intellectual property rights คือ การปกป้องไม่ให้ผู้อื่นละเมิดลิขสิทธิ์ทางปัญญาของลูกค้า
- Computer misuse คือ ไม่นำความรู้ทางคอมพิวเตอร์ที่มีไปละเมิดความเป็นส่วนตัวของผู้อื่น



# อ้างอิง

