

การเรียนรู้-การสอน 1-2567

รหัสวิชา ENGCE301 การออกแบบและพัฒนาซอฟต์แวร์
Software Design and Development

โดย

นายธนิต เกตุแก้ว (thanit@rmutl.ac.th)

หลักสูตรวิศวกรรมคอมพิวเตอร์, สาขาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่

วิชาชีพเลือกแนะนำ (12 หน่วยกิต)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์ (ปรับปรุง พ.ศ.2565)

นักศึกษาควรรับคำแนะนำจากอาจารย์ที่ปรึกษาประจำห้อง และเมื่อขึ้นปีที่ 2 ควรพิจารณาติดต่ออาจารย์ที่ปรึกษาโครงการ เพื่อวางแผนในการเลือกรายวิชาลงทะเลเบียน ซึ่งจะช่วยให้บรรลุเป้าหมายของการเรียน และสำเร็จการศึกษาในระยะเวลาที่เหมาะสม

วิชาชีพเลือกแนะนำ แบบที่ 1 เลือกภายในแขนงวิชา 3 กลุ่ม ตามความสนใจของนักศึกษา

- (1) กลุ่มสถาปัตยกรรมคอมพิวเตอร์และเครือข่าย
- (2) กลุ่มวิศวกรรมซอฟต์แวร์
- (3) กลุ่มวิศวกรรมปัญญาประดิษฐ์

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์ (ปรับปรุง พ.ศ.2565)

ปี 1/1 (19 หน่วยกิต)

ปี 1/2 (21 หน่วยกิต)

ปี 2/1 (21 หน่วยกิต)

ปี 2/2 (21 หน่วยกิต)

ปี 3/1 (21 หน่วยกิต)

ปี 3/2 (14 หน่วยกิต)

ปี 4/1 (6 หน่วยกิต)

ปี 4/2 (12 หน่วยกิต)

ศึกษาทั่วไป 3 หน่วยกิต

ศึกษาทั่วไป 3 หน่วยกิต

ศึกษาทั่วไป 3 หน่วยกิต

ศึกษาทั่วไป 3 หน่วยกิต

ศึกษาทั่วไป 3 หน่วยกิต

ศึกษาทั่วไป 3 หน่วยกิต

Co-operative Education in Computer Engineering

ศึกษาทั่วไป 3 หน่วยกิต

Fund. Cal. for Engineers

Applied Cal. for Engineers

Engineering Drawing

Electrical Instruments and Measurements

Fundamental of Engineering Training

Computer Programming

Engineering Electronics

Fund. Physics for Engineers

Digital Circuits

Basic Computer Engineering Skills

Discrete Mathematics

Programming for Computer Engineers

Electric Circuits

Electrical Engineering Mathematics

Data Communication and Networks

Computer Architecture and Organization

Statistics for Computer Engineering

Data Structures and Algorithms

ศึกษาทั่วไป 3 หน่วยกิต

Microprocessor and Interface

Digital Systems Design

Artificial Intelligence and Machine Learning

Operating Systems

Database System

ศึกษาทั่วไป 3 หน่วยกิต

Embedded Systems and Internet of Everything

Computer and Data Security

System Analysis and Design

Pre-Co-Operative Education

ซีพเลือก 3 หน่วยกิต

ซีพเลือก 3 หน่วยกิต

Computer Engineering Pre-Project

เลือกเสรี 3 หน่วยกิต

หรือ

Computer Engineering Practice

ซีพเลือก 3 หน่วยกิต

ศึกษาทั่วไป 3 หน่วยกิต

Computer Engineering Project

เลือกเสรี 3 หน่วยกิต

พื้นฐานวิชาชีพ

ซีพบังคับ (กลุ่มฮาร์ดแวร์และสถาปัตยกรรม)

ซีพบังคับ (กลุ่มโครงสร้างพื้นฐานของระบบ)

ซีพบังคับ (กลุ่มเทคโนโลยีและวิธีการทางซอฟต์แวร์)

ซีพบังคับ (กลุ่มเทคโนโลยีเพื่องานประยุกต์)

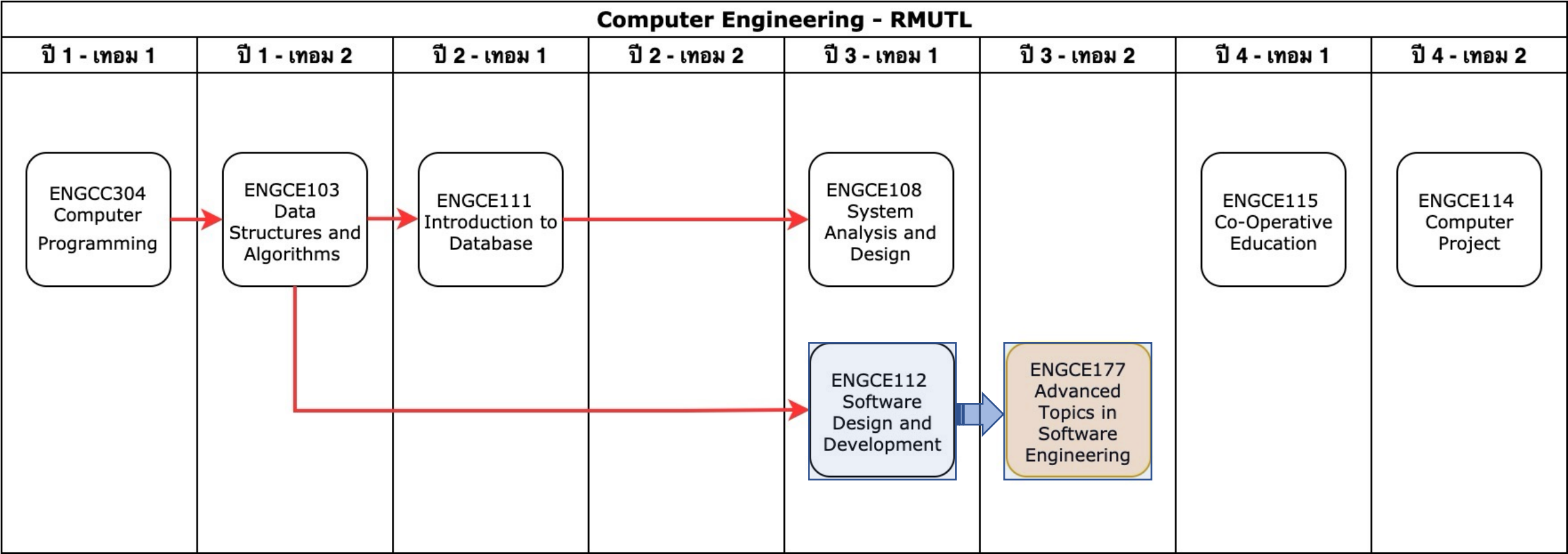
ซีพบังคับ (กลุ่มฝึกวิชาชีพและโครงงาน)

วิชาชีพเลือกแนะนำ แบบที่ 2 เพื่อให้เกิดความเชี่ยวชาญในแต่ละสายงานอาชีพ ตามเป้าหมายของนักศึกษา

สายงานอาชีพ	วิชาชีพเลือกแนะนำ
(1) วิศวกรคอมพิวเตอร์	ENGCE202 การสั่งงานด้วยอุปกรณ์เคลื่อนที่แบบพกพา ENGCE204 วิทยาการหุ่นยนต์ขนาดเล็ก ENGCE173 การวิเคราะห์และออกแบบเชิงวัตถุ ENGCE405 การเรียนรู้เชิงลึกเบื้องต้น
(2) นักวิชาการคอมพิวเตอร์	ENGCE131 การสื่อสารเชิงดิจิทัล ENGCE306 การเขียนโปรแกรมบนเว็บ ENGCE311 วิทยาการข้อมูลและการวิเคราะห์ ENGCE402 คอมพิวเตอร์กราฟิกส์
(3) นักวิเคราะห์ระบบคอมพิวเตอร์	ENGCE161 การประมวลผลสัญญาณดิจิทัล ENGCE162 สัญญาณและระบบ ENGCE205 ระบบอัตโนมัติเบื้องต้น ENGCE309 การประมวลผลแบบคลาวด์
(4) นักพัฒนาซอฟต์แวร์	ENGCE302 การกำหนดความต้องการและการออกแบบทางซอฟต์แวร์ ENGCE303 การตรวจสอบความสมเหตุสมผลและการทวนสอบ ENGCE304 กระบวนการซอฟต์แวร์และการประกันคุณภาพ ENGCE186 สถาปัตยกรรมซอฟต์แวร์
(5) โปรแกรมเมอร์	ENGCE301 การออกแบบและพัฒนาซอฟต์แวร์ ENGCE174 การเขียนโปรแกรมเชิงวัตถุ ENGCE306 การเขียนโปรแกรมบนเว็บ ENGCE307 การพัฒนาโปรแกรมประยุกต์สำหรับอุปกรณ์เคลื่อนที่

Subjects Study Roadmap

From Zero to Fullstack



รหัสวิชา ENGCE301 และ ENGCE112 การออกแบบและพัฒนาซอฟต์แวร์

การออกแบบและพัฒนาซอฟต์แวร์

ทฤษฎี + ปฏิบัติ 3(2-3-5)

Software Design and Development

รหัสรายวิชาเดิม : ไม่มี

วิชาบังคับก่อน : ENGCE103 โครงสร้างข้อมูลและขั้นตอนวิธี

ศึกษาและฝึกปฏิบัติการเกี่ยวกับ แบบจำลองข้อมูล และระบบฐานข้อมูล การเลือกใช้เครื่องมือในการพัฒนาซอฟต์แวร์ มาตรฐานของซอฟต์แวร์ และการจัดการข้อจำกัดเชิงวิศวกรรม กระบวนการทัศน์การเขียนโปรแกรมและเทคนิคการเลือกคำสั่งในการแก้ไขปัญหาในเชิงโปรแกรมรูปแบบการเขียนโปรแกรมเชิงโครงสร้าง แบบจำลองการไม่ประสานกันสำหรับการเขียนโปรแกรมที่มีการประมวลผลร่วมกัน การออกแบบและเขียนโปรแกรมเชิงวัตถุ กระบวนการผลิตซอฟต์แวร์ กระบวนการทดสอบและคุณภาพของซอฟต์แวร์ การสร้างหน่วยทดสอบ รวมถึงการประยุกต์ใช้งานสำหรับอินเทอร์เน็ตในทุกสรรพสิ่ง

เนื้อหารายวิชา

เรียนรู้เกี่ยวกับวิศวกรรมซอฟต์แวร์ ปัญหาของการออกแบบและผลิตซอฟต์แวร์ ขั้นตอนการดำเนินการสำหรับผลิตซอฟต์แวร์ เทคนิคและเครื่องมือของการออกแบบซอฟต์แวร์ การวิเคราะห์หน่วยความจำ การวิเคราะห์เวลาการประมวลผล การทดสอบโปรแกรม การจัดทำคู่มือรายละเอียดซอฟต์แวร์ การจัดการและบริหารโครงการซอฟต์แวร์ การออกแบบซอฟต์แวร์การบริหารเพื่อตัดสินใจอย่างชาญฉลาด

วัตถุประสงค์ เพื่อให้ นักศึกษาสามารถ

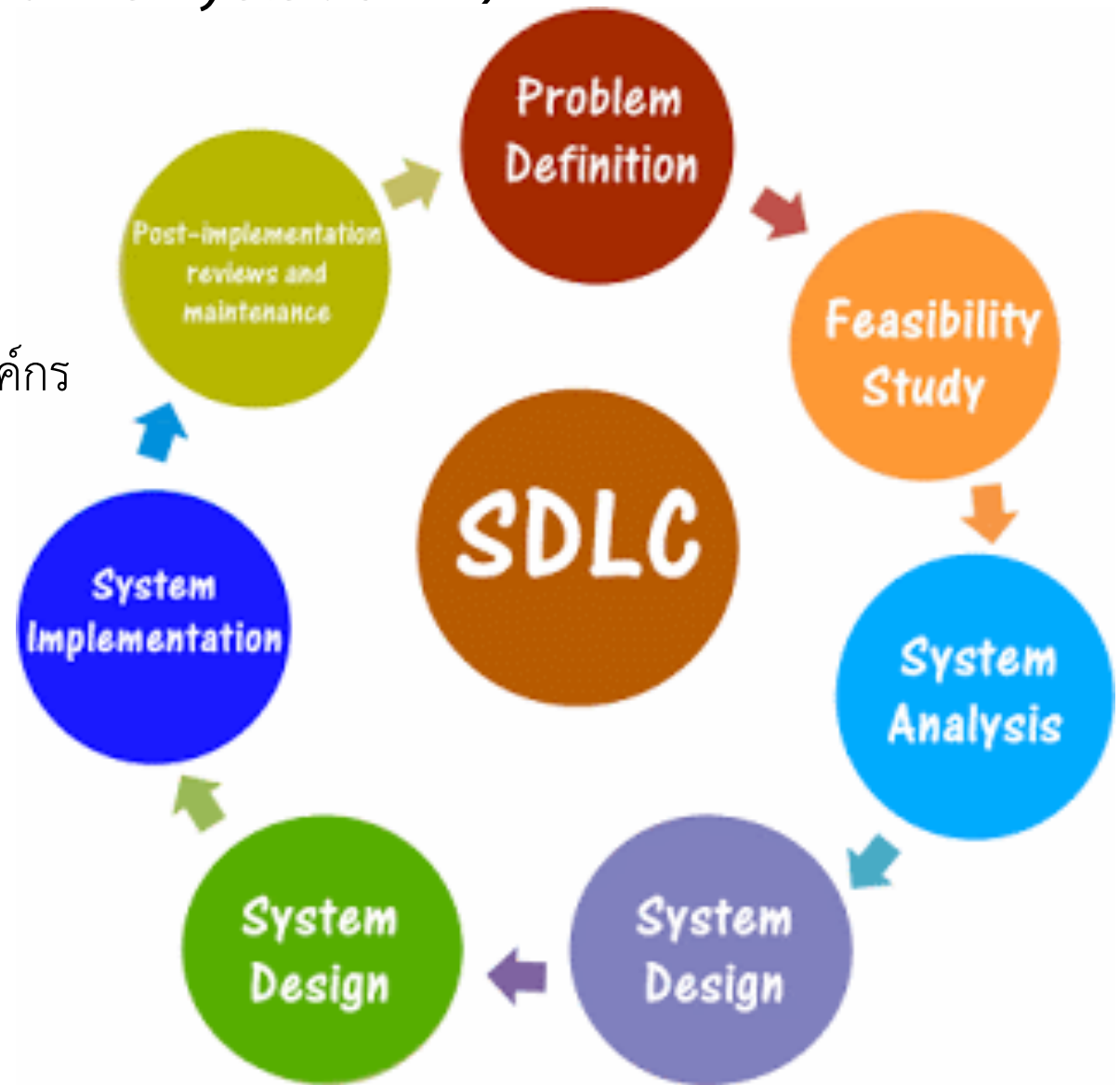
1. เข้าใจหลักการและกระบวนการวิศวกรรมซอฟต์แวร์
2. เข้าใจหลักการของการรวบรวมความต้องการซอฟต์แวร์รวมถึงข้อกำหนดซอฟต์แวร์
3. เข้าใจหลักการและขั้นตอนในการออกแบบซอฟต์แวร์
4. วางแผนและบริหารจัดการโครงการพัฒนาซอฟต์แวร์
5. จัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์แต่ละขั้นตอนได้อย่างมีประสิทธิภาพ
6. รู้จักเครื่องมือที่สามารถนำมาใช้กับงานวิศวกรรมซอฟต์แวร์
7. เข้าใจหลักการและขั้นตอนในการทดสอบและบำรุงรักษาซอฟต์แวร์
8. ทำงานร่วมกันเป็นกลุ่มได้อย่างมีประสิทธิภาพ

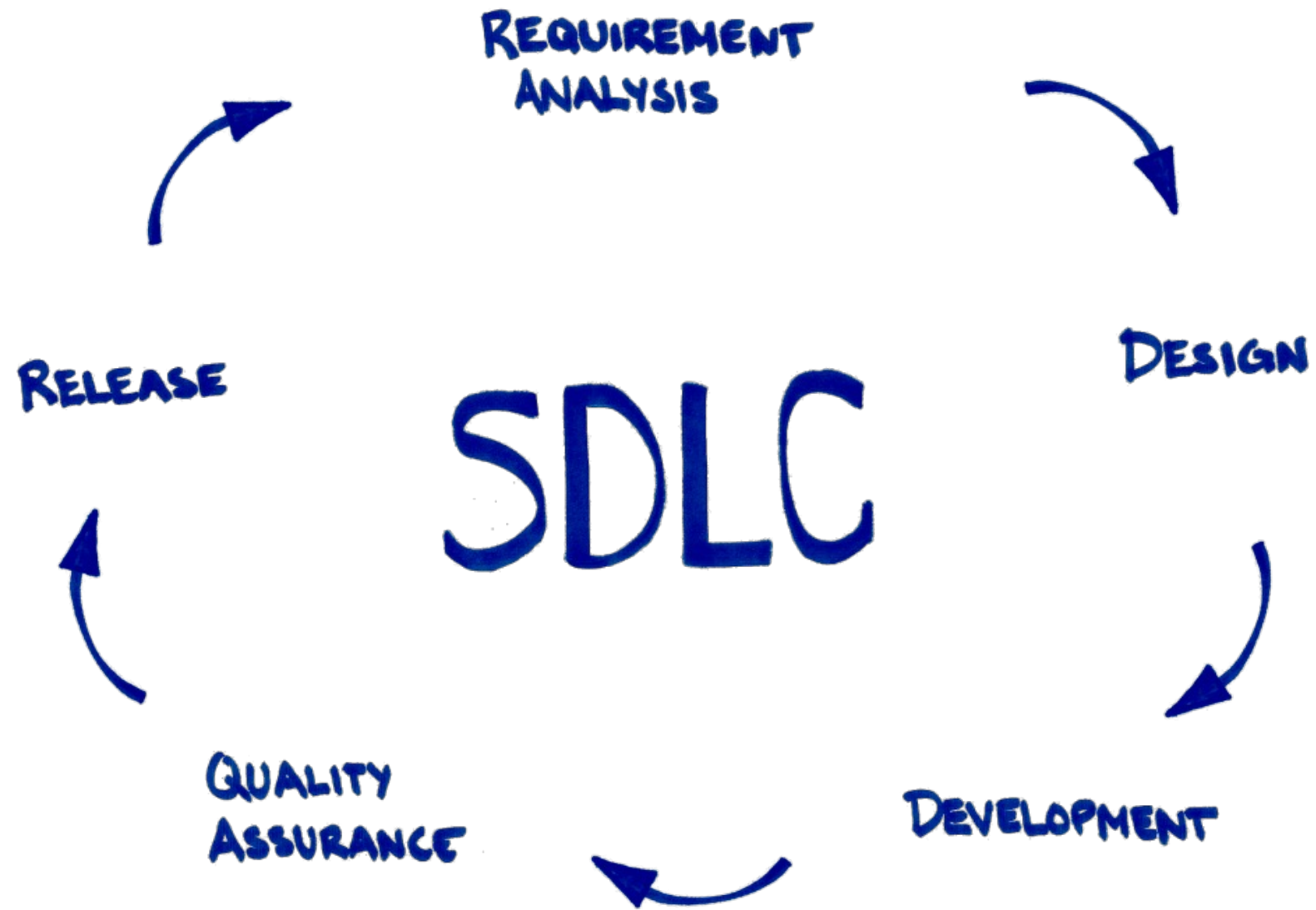
References

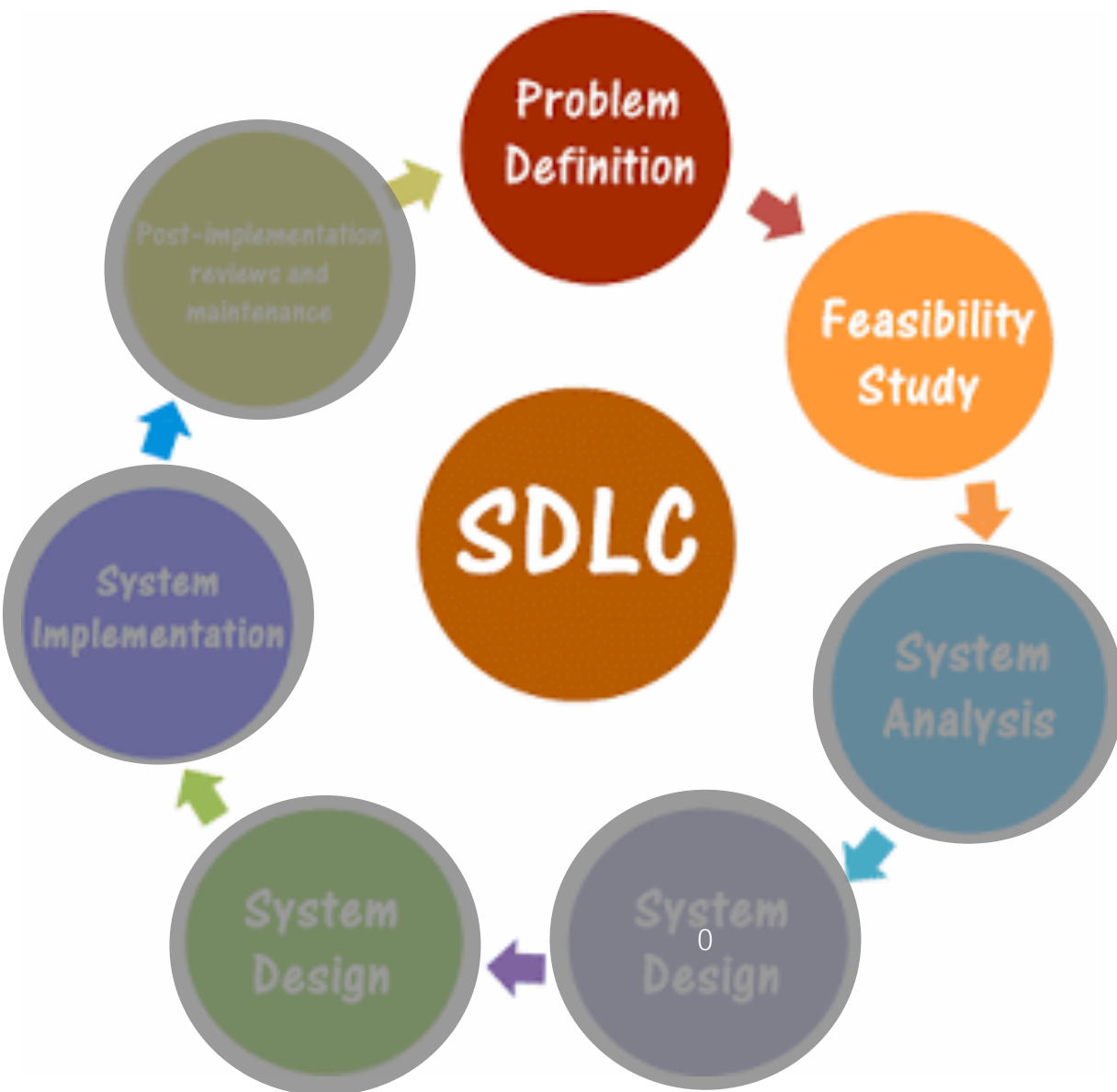
- เอกสารประกอบการสอนวิชาวิศวกรรมซอฟต์แวร์ by Aj. NesT The Series
- Roger S. Pressman and Bruce R. Maxim. Software Engineering a Practitioner's Approach. Eighth Edition. McGraw-Hill, 2014.
- Ian Sommerville, Software Engineering, 10th Edition, Addison-Wesley, 2015.
- Ivan Marsic. Software Engineering. 2012.

วงจรการพัฒนาระบบ (System Development Life Cycle : SDLC)

คือ การแบ่งขั้นตอนกระบวนการพัฒนาระบบงาน
หรือระบบเทคโนโลยีสารสนเทศด้วย
เพื่อช่วยแก้ปัญหาทางธุรกิจหรือตอบสนองความต้องการขององค์กร
โดยระบบที่จะพัฒนานั้นอาจเป็นการพัฒนาระบบใหม่หรือ
การปรับปรุงระบบเดิมให้ดีขึ้นก็ได้
การพัฒนาระบบแบ่งออกเป็น 7 ขั้นตอน





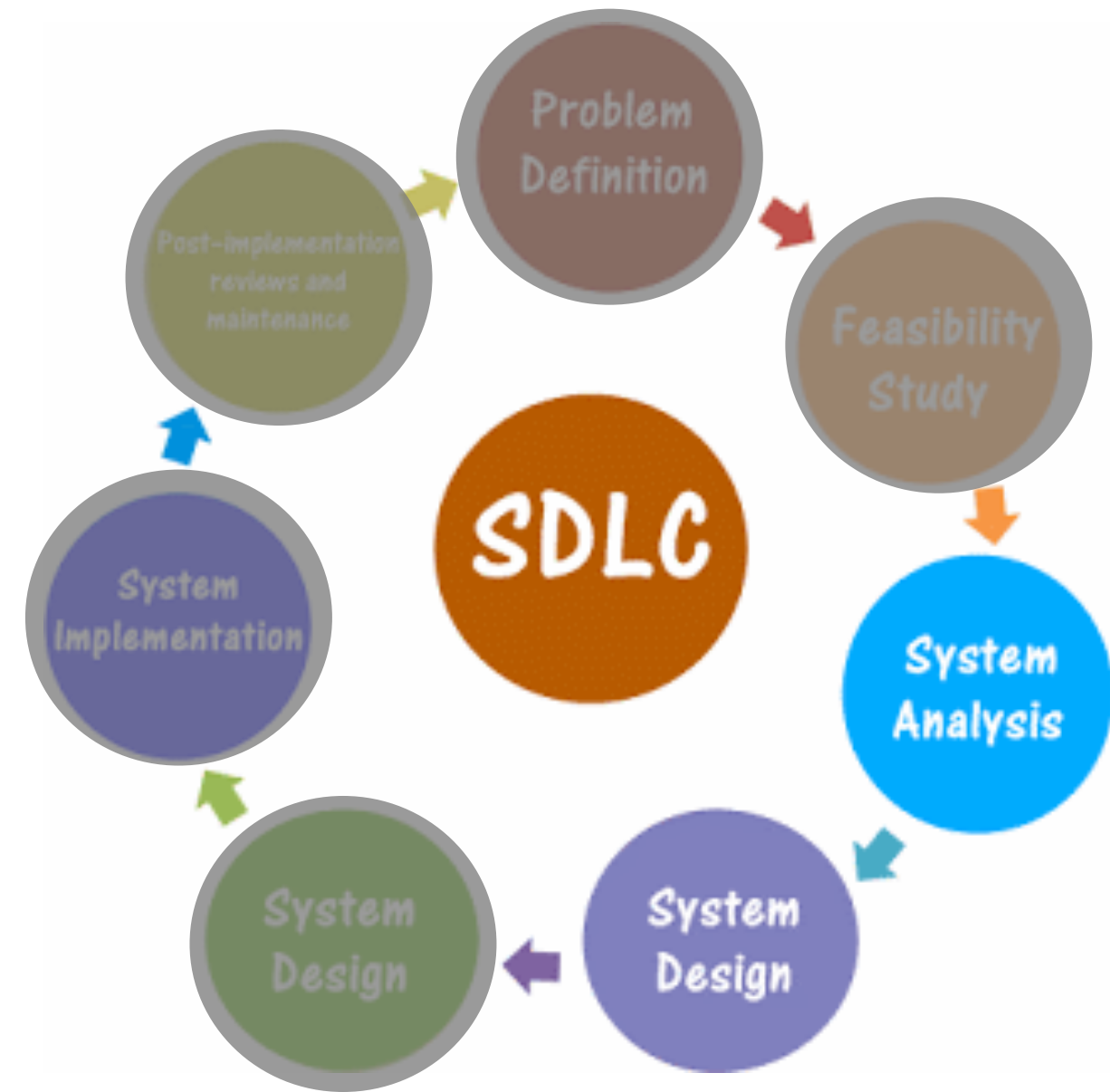


เข้าใจปัญหาและที่มา

ปัญหาของระบบ เป็นโปรแกรมที่ใช้งานระบบเดิมเหล่านี้นั้นในระยะเวลาเนิ่นนานอาจจะเป็นโปรแกรมที่เขียนขึ้นมาแล้ว ไม่ได้มีการเชื่อมต่อถึงระบบ

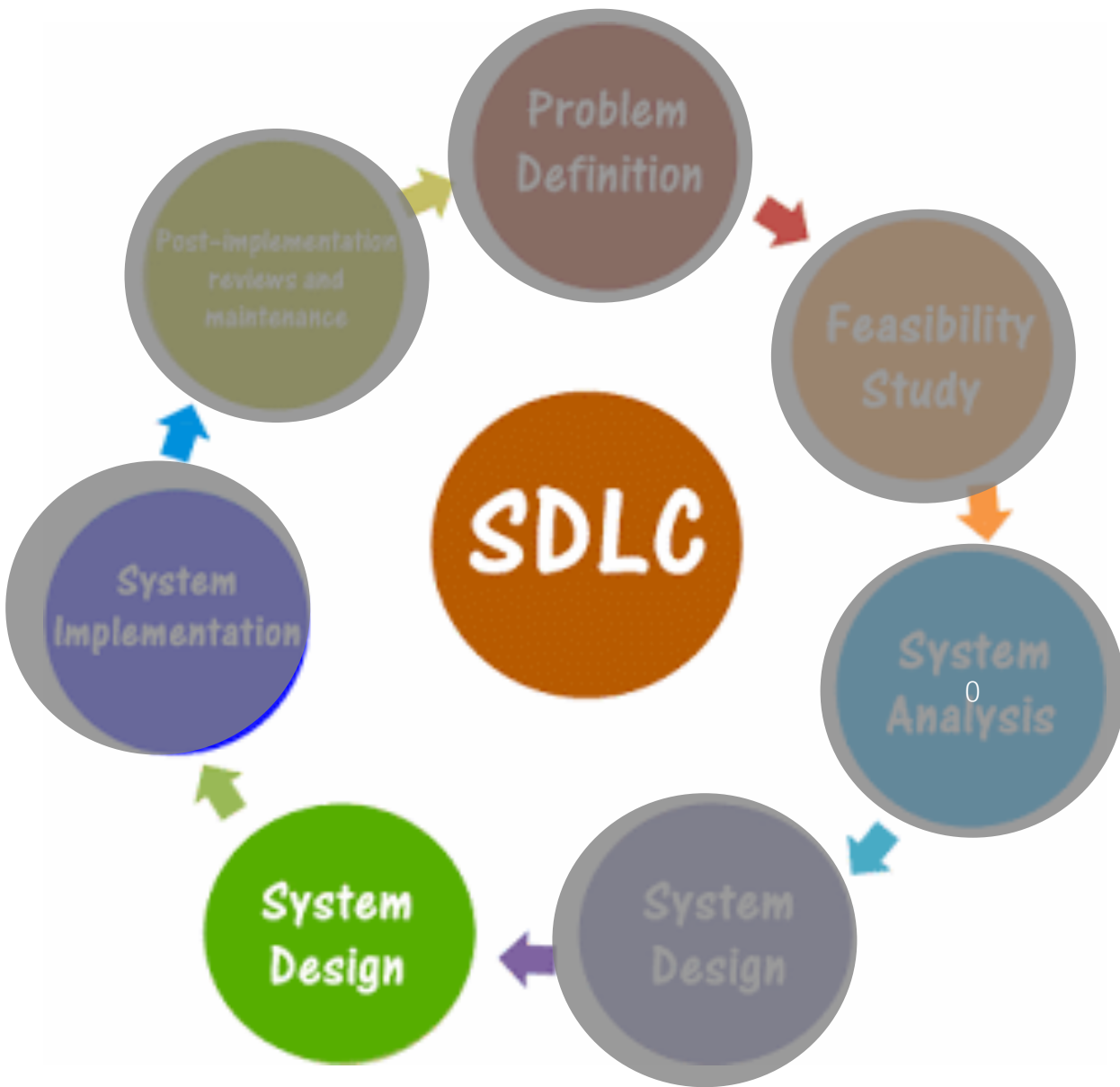
ดังนั้น นักวิเคราะห์ระบบจึงมองเห็นปัญหาที่เกิดขึ้นที่เกี่ยวข้องกับระบบที่จะพัฒนา แล้วดำเนินการแก้ไข ต้องมีการรับรู้ถึงปัญหา ค้นหาต้นเหตุของปัญหา

รวบรวมปัญหาเดิมของระบบ ศึกษาความเป็นไปได้ของโครงการพัฒนาระบบ จัดเตรียมทีมงาน และกำหนดเวลาในการทำงาน ลงมือดำเนินการ



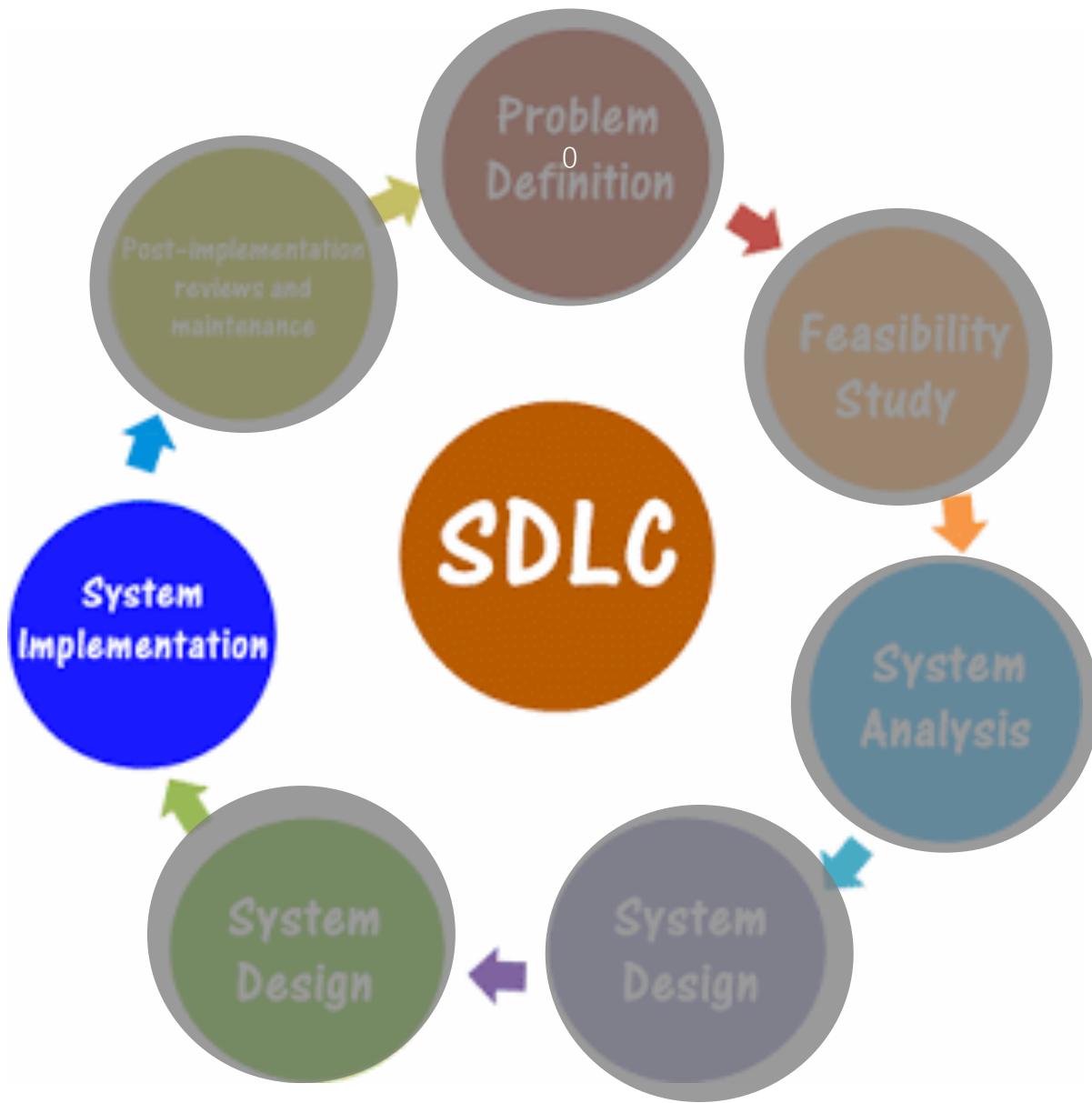
วิเคราะห์งาน

จะต้องรวบรวมความต้องการ มากที่สุด การสืบค้นความต้องการ
ของผู้ใช้ สามารถดำเนินการได้ มีการรวบรวมเอกสารสัมภาษณ์
การออกแบบ การสอบถาม และการสังเกตการณ์บน
สภาพแวดล้อมการทำงาน นักวิเคราะห์ต้องไปพัฒนาความ
ต้องการของระบบใหม่ด้วยการพัฒนาแบบจำลองขึ้นมา ได้แก่
จำลองกระบวนการ DFD: Data Flow Diagram
และแผนภาพ ER



ออกแบบงาน

เป็นระยะที่นำผลลัพธ์ที่ได้จากการวิเคราะห์ ว่าระบบดำเนินการอย่างไรเพื่อให้เกิดตามที่ต้องการ การออกแบบรายงาน ออกแบบหน้าจออินพุตข้อมูล การออกแบบผังงานระบบ การออกแบบฐานข้อมูล และออกแบบโปรแกรม



พัฒนาระบบและทดสอบ

การพัฒนาระบบ โดยนักโปรแกรมเมอร์หรือนักวิเคราะห์ระบบได้ทำการออกแบบไว้ มาเขียนชุดคำสั่งเพื่อสร้างระบบขึ้นมา เลือกภาษาที่จะเขียน สามารถนำเครื่องมือมาช่วยในพัฒนา และสร้างเอกสารประกอบ



การติดตั้ง

เป็นขั้นตอนการนำระบบที่พัฒนาจนสมบูรณ์มาติดตั้ง และเริ่มใช้งานจริง ในส่วนนี้นอกจากติดตั้งระบบใช้งานแล้ว ยังต้องมีการจัดเตรียมขั้นตอนการสนับสนุนส่งเสริม การใช้งานให้สามารถใช้งานได้อย่างสมบูรณ์ โดยจัดทำหลักสูตรฝึกอบรมผู้ใช้งาน

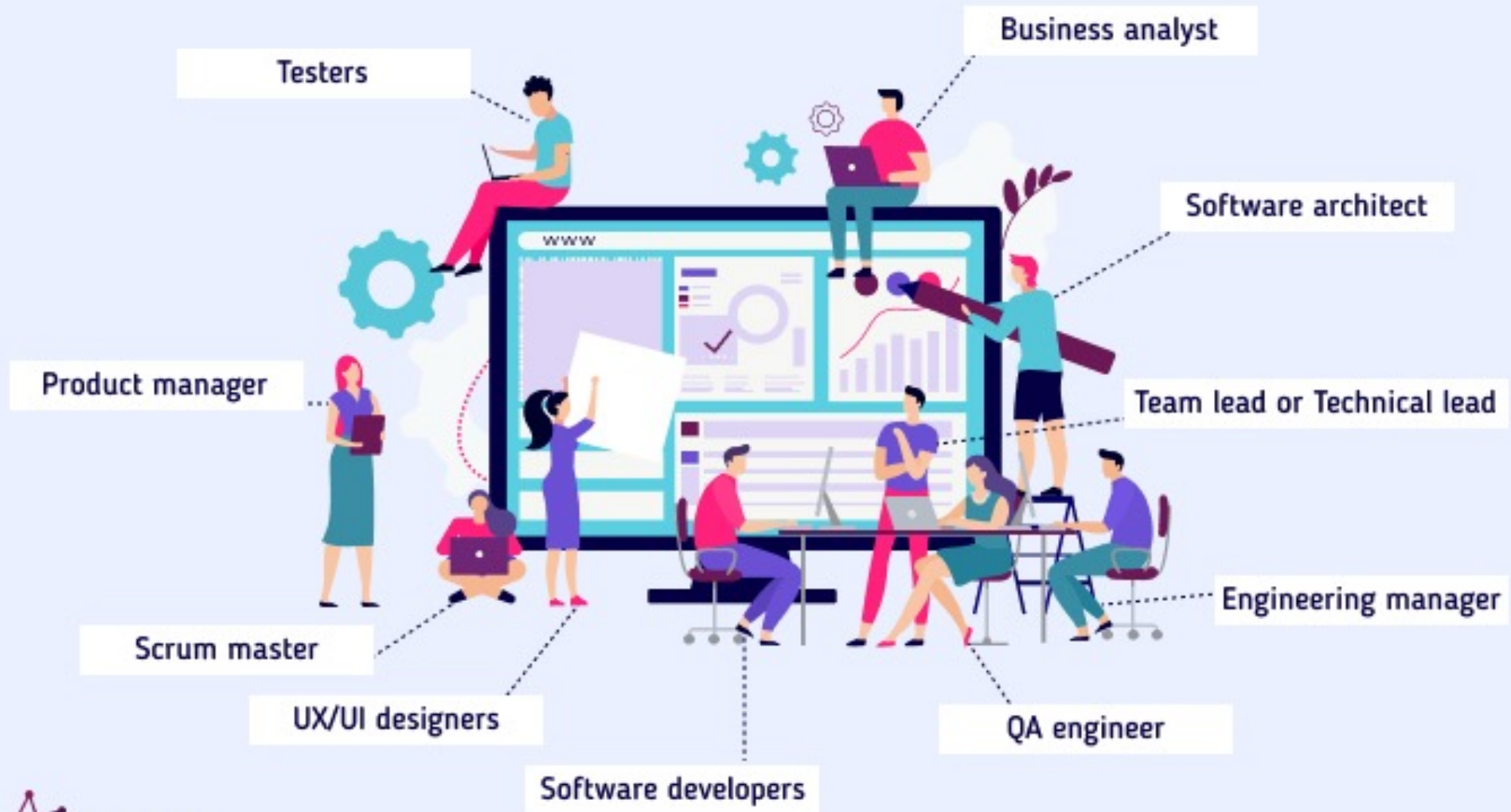
บำรุงรักษา

เป็นขั้นตอนการบำรุงรักษาระบบต่อเนื่องหลังจากเริ่มดำเนินการ ผู้ใช้ระบบอาจจะพบกับปัญหาที่เกิดขึ้นภายหลัง เช่น ปัญหาเนื่องจากความไม่คุ้นเคยกับระบบใหม่ ติดตามประเมินผล เก็บรวบรวมคำร้องขอให้ปรับปรุงระบบ วิเคราะห์ข้อมูลร้องขอให้ปรับปรุงระบบ

Software development team structure



Your perfect development team





Project Manager

Project Manager ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

ผู้จัดการโครงการหรือ Project Manager มีหน้าที่ที่ต้องรับผิดชอบในการเป็นผู้นำและประสานงานโครงการพัฒนาซอฟต์แวร์ พวกเขา มีหน้าที่รับผิดชอบในการตรวจสอบให้แน่ใจว่าโครงการเสร็จสิ้นตามเวลา ภายในงบประมาณ และเพื่อความพึงพอใจของผู้มีส่วนได้ส่วนเสีย(stakeholders)



Product Owner

Product Owner ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

Product Owner เป็นมืออาชีพที่รับผิดชอบในการเป็นตัวแทนผลประโยชน์ของผู้มีส่วนได้ส่วนเสียและลูกค้าในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ พวกเขาทำหน้าที่เป็นผู้ประสานงานระหว่างทีมพัฒนาและผู้มีส่วนได้ส่วนเสียและมีหน้าที่รับผิดชอบในการกำหนดและจัดลำดับความสำคัญของคุณสมบัติของผลิตภัณฑ์

<https://www.trueplookpanya.com/knowledge/content/57712>



Software Team Lead

Software Team Lead ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

หัวหน้าทีมพัฒนาซอฟต์แวร์(Software Team Lead) คือบุคคลที่รับผิดชอบในการเป็นผู้นำทีมนักพัฒนาซอฟต์แวร์และดูแลให้โครงการพัฒนาซอฟต์แวร์เสร็จสิ้นตามกำหนดเวลา ภายในงบประมาณ และเป็นไปตามมาตรฐานคุณภาพที่กำหนด

<https://www.trueplookpanya.com/knowledge/content/57712>



Product Owner

Product Owner ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

Product Owner เป็นมืออาชีพที่รับผิดชอบในการเป็นตัวแทนผลประโยชน์ของผู้มีส่วนได้ส่วนเสียและลูกค้าในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ พวกเขาทำหน้าที่เป็นผู้ประสานงานระหว่างทีมพัฒนาและผู้มีส่วนได้ส่วนเสียและมีหน้าที่รับผิดชอบในการกำหนดและจัดลำดับความสำคัญของคุณสมบัติของผลิตภัณฑ์

<https://hocco.co/th/blog/software-developer-job-description/>

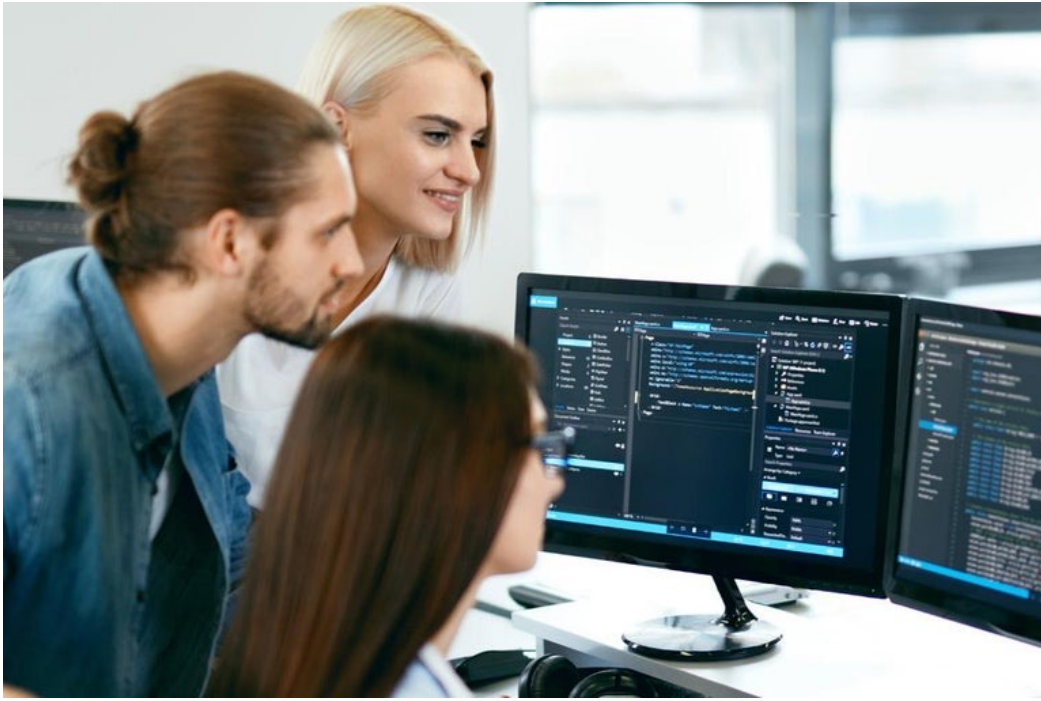


System Analyst

Systems analyst ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

นักวิเคราะห์ระบบ(Systems Analyst) มีบทบาทสำคัญในการพัฒนาระบบซอฟต์แวร์ พวกเขาทำหน้าที่รับผิดชอบในการวิเคราะห์ความต้องการของผู้ใช้และผู้มีส่วนได้ส่วนเสีย และใช้ข้อมูลนั้นในการออกแบบและพัฒนาซอฟต์แวร์ที่ตอบสนองความต้องการเหล่านั้น

<https://www.computerscience.org/careers/systems-analyst/>



UI/UX designer ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

นักออกแบบ UI/UX มีหน้าที่รับผิดชอบการออกแบบ user interfaces และ user experiences สำหรับแอปพลิเคชันซอฟต์แวร์ พวกเขามีหน้าที่รับผิดชอบในการสร้างการออกแบบที่ดึงดูดสายตาและเป็นมิตรกับผู้ใช้ซึ่งทำให้ผู้ใช้สามารถโต้ตอบกับซอฟต์แวร์ได้ง่าย

Quality Assurance(QA) / Tester ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

มีหน้าที่รับผิดชอบในการตรวจสอบว่าผลิตภัณฑ์ซอฟต์แวร์ตรงตามมาตรฐานคุณภาพที่กำหนดโดยบริษัทและลูกค้า พวกเขามีส่วนร่วมในกระบวนการพัฒนาตั้งแต่ต้นจนจบ และทำงานอย่างใกล้ชิดกับทีมพัฒนาเพื่อระบุและแก้ไขปัญหาที่อาจส่งผลกระทบต่อคุณภาพของซอฟต์แวร์

DevOps

Result?

Make a thing !
With features !

It compiles !
Ship it !

It took all night to
launch

Don't know
If want ..



Biz



Dev



Ops



User

DevOps engineer ต้องทำอะไรบ้างในการพัฒนาซอฟต์แวร์?

DevOps engineer มุ่งเน้นแนวทางปฏิบัติในการรวมทีมพัฒนาและทีมปฏิบัติการเข้าด้วยกันเพื่อปรับปรุงการทำงานร่วมกันและประสิทธิภาพการทำงาน พวกเขามีหน้าที่รับผิดชอบในการทำให้กระบวนการพัฒนาซอฟต์แวร์เป็นไปอย่างอัตโนมัติและคล่องตัว และดูแลให้มั่นใจว่าซอฟต์แวร์ได้รับการปรับใช้และบำรุงรักษาในลักษณะที่เชื่อถือได้และมีประสิทธิภาพ

การประเมินผล

สอบ Midterm	25%
สอบ Final	25%
Term Project	30%
งานกลุ่ม/Quiz	10%
เข้าเรียน	10%
<u>รวม 100%</u>	

แบ่งกลุ่มตำแหน่งของสมาชิกในทีม

1. Team Leader
2. System Analyst
3. Developer
4. QA/Tester

Let 's join your Team