

CS 361 – Software Engineering

Team Formation and Roles

Kamonphop Srisopha
Churee Techawut



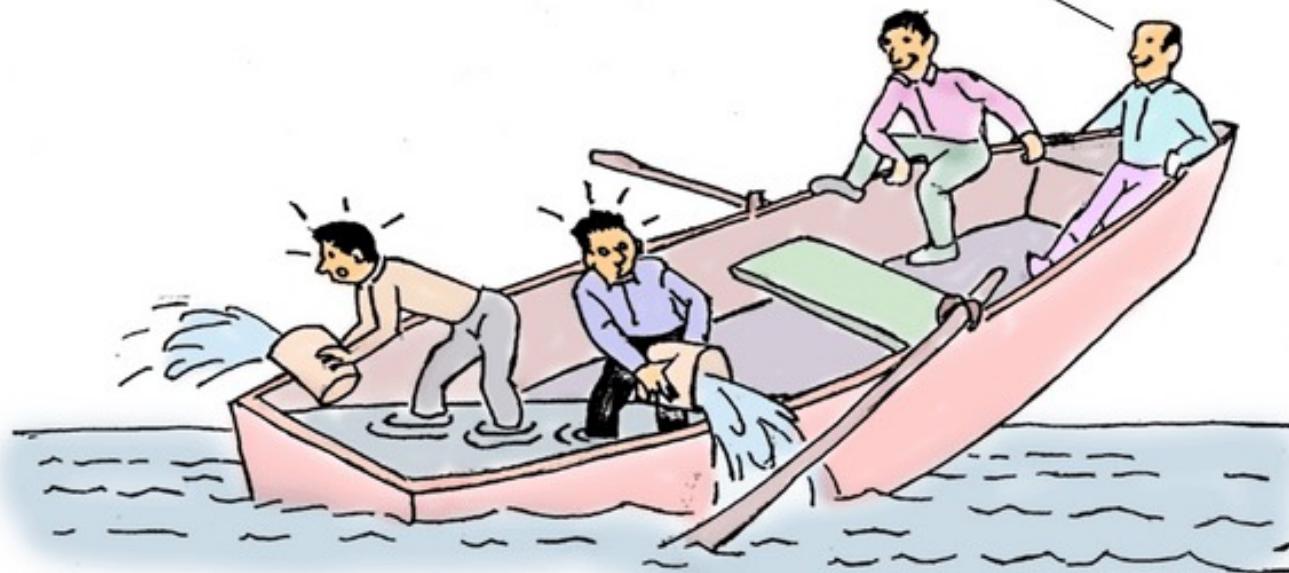
Faculty of Science, Chiang Mai University
คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

Agenda

- Team Formation
- Team Roles and Responsibilities
- Client Interaction
 - Kickoff Meeting

Team Formation

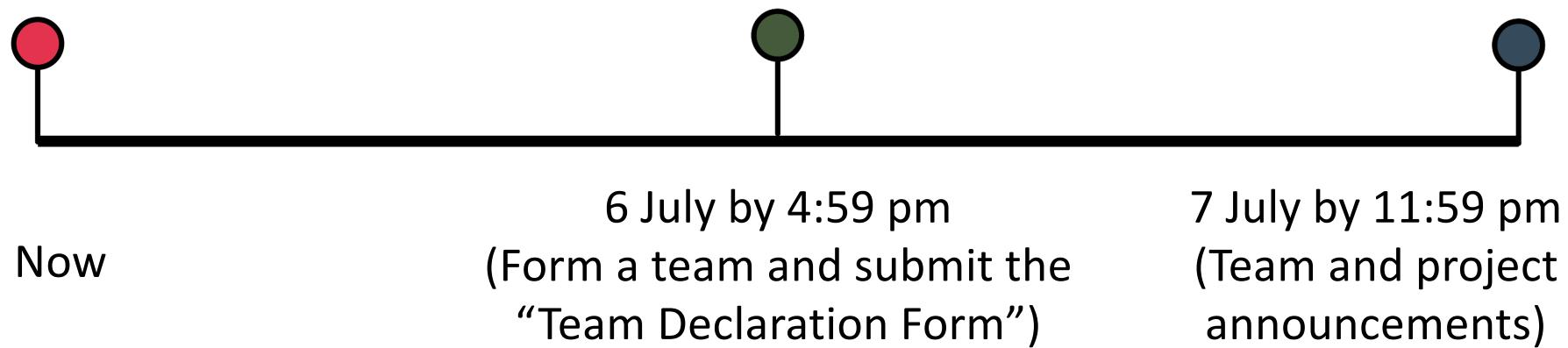
Sure glad that hole isn't at our end.



Team Formation

- 8 ทีม, 7 คนต่อทีม
- มี **12** project proposals
 - สามารถดูรายละเอียดแต่ละ proposal ได้ที่: ดู link ได้บน Course Syllabus ใน **Mango**
 - เลือก Project ที่อยากรับมา **5 projects**
- เริ่มต้นหาทีม หรือหาสมาชิกในทีม
 - ใช้ MS Teams ในการช่วยหาทีม (จะมีเวลาห้ายคืนให้นักศึกษาหาทีมอีกทางหนึ่ง)
 - เมื่อได้ทีมแล้ว ให้ สมาชิกคนที่รับหน้าที่เป็น Scrum Master/ Project Manager ทำ “Team Declaration Form”(<https://cmu.to/3HwLp>) แล้ว ส่งภายในวันที่ **6 กรกฎาคม** เวลา **4:30 pm**
- อาจารย์จะประกาศว่าทีมไหนมีโครงและได้ Project อะไรภายในวันที่ **7-9 กรกฎาคม**
- **Note:** นักศึกษาที่หาทีมไม่ได้ภายในเวลาดังกล่าวจะถูกนำไปรวมกับทีมที่ขาดสมาชิก

Team Formation Timeline



Choosing Team Members

- Skillset
- Role Preference
- Project Preference



Microsoft Teams



A mixer event



Peer Evaluation (Quest)

Overall Evaluation

คำอธิบายคะแนนมีดังต่อไปนี้

- **5 คะแนน:**

สมาชิกคนนี้ทำหน้าที่ที่ได้รับมอบหมายได้ถึงดีมาก
บางทีทำเกินกว่าที่ได้รับมอบหมาย ใส่ใจรายละเอียดต่างๆ
คุณภาพผลงานที่ได้รับมอบหมายออกแบบ ช่วยเหลือและ support ทีมดีมาก

- **4 คะแนน:** สมาชิกคนนี้ทำหน้าที่ที่ได้รับมอบหมายได้ดี

แต่อาจจำเป็นใช้ในบางกรณี
แต่รวมแล้วสมาชิกคนนี้เป็นสมาชิกที่ช่วยเหลือและ support ทีมได้ดี

- **3 คะแนน:** สมาชิกคนนี้ช่วยเหลือทีมเป็นที่น่าพอใจ แต่มืออย่างน้อยหนึ่งครั้งที่เพื่อนคนอื่นในทีม
ต้องทำงานที่เข้าได้รับมอบหมายแทน

- **2 คะแนน:**

สมาชิกคนนี้ช่วยเหลือทีมบางครั้ง
ส่วนมากเพื่อนในทีมจะเป็นต้องตื่นเช้าเป็นประจำและอาจมาสาย (เช่น
เมื่อมีประชุม).
เพื่อนคนอื่นในทีมอาจจะต้องเข้ามาทำงานที่สมาชิกคนนี้ได้รับมอบหมายแทนหรือทำซ้ำอีกรอบ
เนื่องจากงานที่ได้มาไม่ค่อยดี

- **1 คะแนน:** สมาชิกคนนี้ทำตัวเป็นอุปสรรคของทีม ทำให้ทีมไม่คีบหน้า ไม่ช่วยเหลือเพื่อนในทีม
ขาดการติดต่อบ่อยครั้ง

Overall Evaluation Score (ให้คะแนนสมาชิกคนนี้ตาม list ด้านบน) *

1 2 3 4 5

เป็นอุปสรรคของทีม

ดีมาก

- At the end of the semester, we will ask you to evaluate your teammates.
- Your peer evaluation XP score (100 XP) will be based on the feedback given by your teammates.

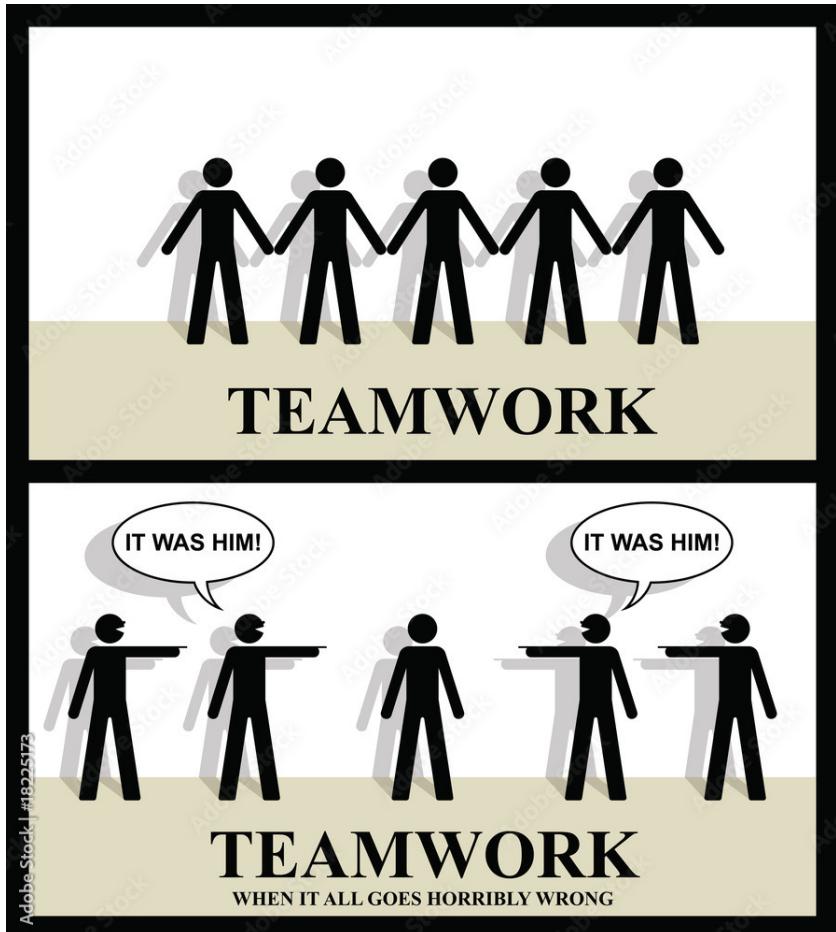
	ไม่เลย	นานๆ ครั้ง	บางครั้ง	มักจะ	เป็นประจำ
เพื่อนสมาชิกคนนี้ได้ทำหน้าที่ตามบทบาทที่เขาได้รับมอบหมายหรือไม่ (เช่น Scrum Master, DevOps, RQ Manager, ...)?	<input type="radio"/>				
เพื่อนสมาชิกคนนี้ได้ช่วยเหลือเพื่อนในทีมคนอื่นๆ เมื่อเห็นสมควร	<input type="radio"/>				
เพื่อนสมาชิกคนนี้สื่อสารกับคนในทีมอย่างมีประสิทธิภาพ	<input type="radio"/>				
เพื่อนสมาชิกคนนี้แสดงให้เห็นว่าเขาได้ลงแรงและใช้เวลาที่เหมาะสมกับ project	<input type="radio"/>				
Quality ของงานที่เพื่อนสมาชิกคนนี้ได้รับมอบหมายออกแบบต่างๆ	<input type="radio"/>				

10 Qualities For an Excellent Team Member

1. Show genuine commitment
2. Be flexible
3. Don't stay in the shadows
4. Be reliable and responsible
5. Actively listen
6. Keep your team informed
7. Always be ready to help
8. Support and respect others
9. Be a problem-solver
10. Recognize when you are wrong



Working In A Team Can Be Great



ประโยชน์:

- แก้ปัญหาใหญ่ๆได้ในเวลาอันสั้น
- ใช้ประสบการณ์ร่วมของทุกคน
- สร้าง network/ friends

ความเสี่ยงหรือปัญหาที่อาจเกิดขึ้น:

- ปัญหาในการสื่อสารและประสานงาน
- การตัดสินใจ
- (โครงสร้าง) ความขัดแย้งหรือความไม่ไว้วางใจระหว่างสมาชิกในทีม

3 Big Questions Your Team Needs To Talk About



การสื่อสาร

How will the team communicate?

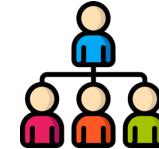
Key: Not communicating will guarantee miscommunication



การตัดสินใจ

How will the team make decisions?

Key: Strive for consensus (Majority vote), evaluate options, compromise



โครงสร้างการจัดการ

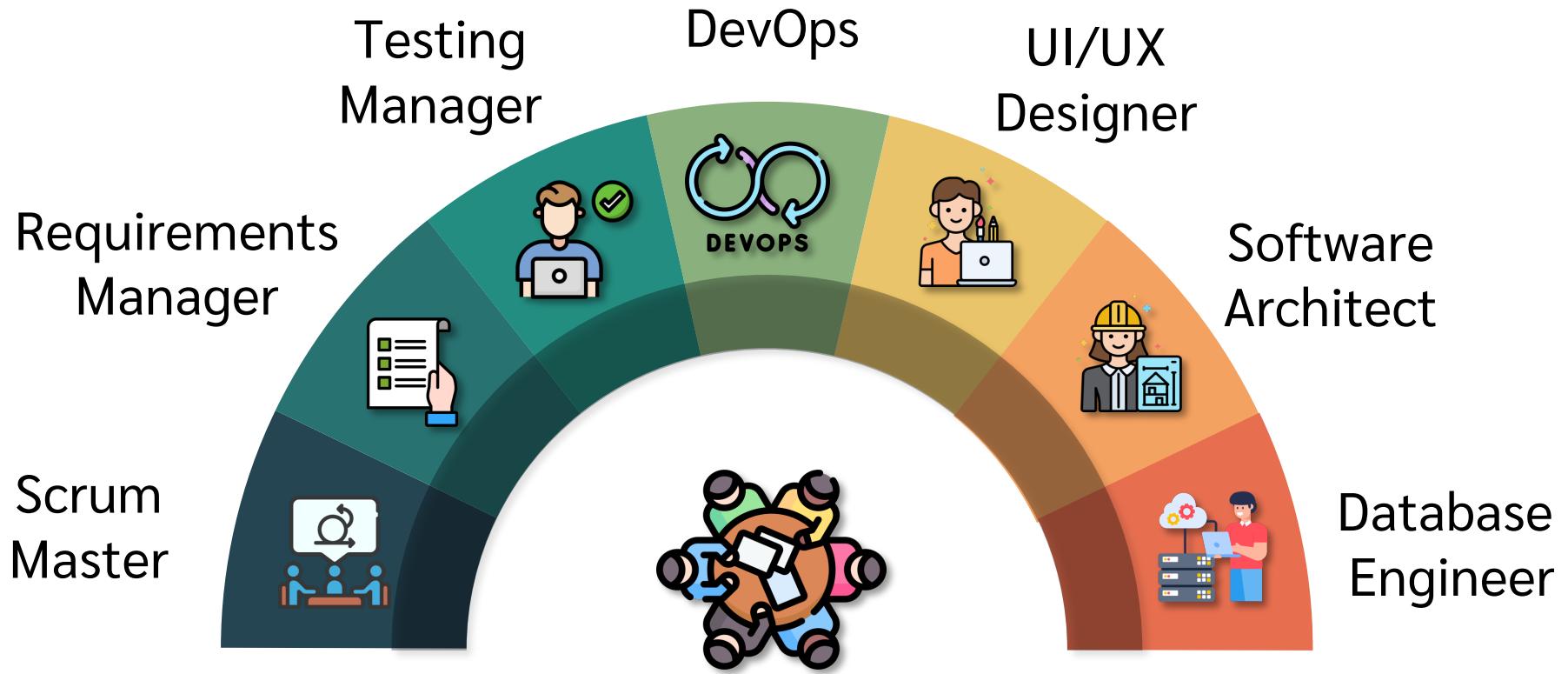
How will the team divide tasks?

Key: Have a clear vision and purpose, define clear roles and responsibilities

Roles

บทบาทในทีม

Team Roles



ทั้งนี้ทุก Role ต้องส่งเอกสาร 1 ฉบับเกี่ยวกับ Role นั้นๆ

(ขึ้นอยู่กับ sprint, จะมี template ให้ และ มีค่า XP = 100 XP)

Scrum Master/ Project Manager

- เรียกได้ว่าเป็น “โค้ชของทีม”
- เป็นบุคคลที่มีบทบาทสำคัญในการบริหารจัดการทีมให้ทำงานในกรอบ Scrum, ติดตามว่า ทุกคนได้ทำงานตามที่รับมอบหมายหรือไม่, และช่วยแก้ปัญหาต่างๆที่อาจเกิดขึ้นในทีม
- ทำหน้าที่จัดการการประชุมต่างๆ เช่น เพื่อรีวิวและวางแผนงานในแต่ละวัน นอกจากนี้ Scrum Master ยังต้องจัดการ Sprint Planning, Sprint Review, และ Retrospective ซึ่ง เป็นการประชุมที่เกิดขึ้นตอนเริ่มและจบของแต่ละ Sprint และ Daily Scrum.
- ทำหน้าที่เป็นผู้ประสานงานหลักของทีมกับผู้ว่าจ้าง
- ทางคณานารย์จะติดต่อกับ Scrum Master ก่อนหากมีปัญหาหรือคำถามเกี่ยวกับทีม ตัว Scrum Master เองก็ต้องเป็นผู้ประสานงานกับคณานารย์เพื่อ report สิ่งต่างๆ เช่น หากมี สมาชิกมีปัญหา เป็นต้น

Scrum Master/ Project Manager



เหตุผลที่ควรเลือกเป็น Scrum Master:

- ชอบการจัดการวางแผนหรือการกระทำสิ่งใด ๆ อย่างเป็นระบบระเบียบ
- อยากศึกษาและเข้าใจรู้วิธี Agile/Scrum อย่างลึกซึ้งถ่องแท้
- รู้สึกโอบอุ่นที่จะต้องเป็นผู้ประสานหรือติดต่อหลักกับคณาจารย์และผู้ว่าจ้าง
- คิดว่าตนเองสามารถติดตามสิ่งต่างๆ ในทีมได้อย่างใกล้ชิด และ มีภาวะเป็นผู้นำสูง ไม่เชินอายที่จะต้องคุยกับเพื่อนในทีม 1-1



เหตุผลที่ไม่ควรเลือกเป็น Scrum Master:

- นักศึกษาไม่ชอบการจัดการ
- นักศึกษาคิดว่าไม่สามารถตามสิ่งต่างๆ ที่เกิดขึ้นได้ทีมได้อย่างสม่ำเสมอ
- นักศึกษาไม่ชอบที่จะเป็นผู้ติดต่อสื่อสารหลักของทีม

Requirements Manager (System Analyst)

- เรียกได้ว่าเป็น “ผู้จัดการความต้องการ”
- เป็นบุคคลที่มีบทบาทในการวิเคราะห์ จัดการ และแปลความต้องการของผู้ว่าจ้าง หรือลูกค้า เพื่อสร้างระบบหรือซอฟต์แวร์ที่ตอบสนองความต้องการของคนเหล่านั้น
- เข้าใจว่าผู้ว่าจ้างต้องการให้ทีมพัฒนา “อะไร” “เพื่อประโยชน์อะไร” และ “เพื่อ ใด” แปลงข้อมูลเหล่านั้นมาเป็นข้อจำกัดทางเทคนิคของระบบที่จะพัฒนา เช่น user stories
- ไม่ใช่ว่า ผู้จัดการความต้องการ ต้องทำ process การหาความต้องการของระบบ เพียงคนเดียว แต่ทำหน้าที่เป็นผู้นำทีมในการเริ่ม process (เช่น เป็นผู้นำเปิดประดีนซักถามผู้ว่าจ้าง หรือเป็น facilitator ใน process การจัดลำดับความสำคัญ ของความต้องการ)
- คล้ายๆ Product Owner ใน Scrum

Requirements Manager (System Analyst)



เหตุผลที่ควรเลือกเป็น RQs Manager:

- สนใจในกระบวนการหาความต้องการและการจัดเรียงความสำคัญของความต้องการ
- มีทักษะการสื่อสารและมนุษย์สัมพันธ์ดี ชอบการซักถามและค้นข้อมูลจากผู้ถูกถาม
- สนใจที่จะรู้เชิงลึกว่าผู้ว่าจ้างต้องการอะไร และจะตอบโจทย์ผู้ว่าจ้างได้อย่างไร



เหตุผลที่ไม่ควรเลือกเป็น RQs Manager:

- ไม่มั่นใจในทักษะการสื่อสารของตนเอง
- สนใจทำหน้าที่อื่น
- ไม่มั่นใจในทักษะการวิเคราะห์ข้อมูลของตนเอง

Testing Manager

- เรียกได้ว่าเป็น “ผู้คุมคุณภาพ”
- หน้าที่สำคัญคือการสร้างความเชื่อมั่นว่าระบบทำงานได้อย่างถูกต้อง โดยการเลือกกลยุทธ์การทดสอบ (Testing Strategies) และสร้างแผนการทดสอบ (Testing Plan) ที่เหมาะสมกับทีม
- หน้าที่สำคัญคือเตือนหรือกำชับให้ทีมมีการเขียน unit test
- หน้าที่สำคัญคือกระจายงานให้สมาชิกในทีมในการทำ Testing
- หน้าที่สำคัญคือการจัดการ Meeting กับ UI/UX Designer และผู้ว่าจ้างเพื่อทำ Capability Drive-through/ Beta Testing ในตอนท้ายหรือตอนทำ Sprint Review
- หน้าที่สำคัญคือการสร้างเอกสาร และรายงานสถานะของการทดสอบ ในตอนท้ายของภาคการศึกษา
- หมายเหตุ: ผู้คุมคุณภาพไม่ได้เป็นคนเดียวที่ทดสอบระบบ แต่เป็นกิจกรรมที่ต้องทำทุกคน

Testing Manager



เหตุผลที่ควรเลือกเป็น Testing Manager:

- สนใจในกระบวนการและเทคนิคของการทดสอบระบบ
- ชอบเห็นสถานะการทดสอบต่างๆ เป็นสีเขียว
- เป็นคนมีความคิดสร้างสรรค์ มีความนิดนองรอบ ชอบคิดว่ามีวิธีไหนบ้างที่ระบบจะ Fail ได้ ชอบเล่นเป็นนักสืบหรือ hacker
- สนใจการจัดการกระบวนการทดสอบกับผู้ว่าจ้าง



เหตุผลที่ไม่ควรเลือกเป็น Testing Manager:

- ไม่เห็นความสำคัญของการทดสอบระบบ
- อยากรับหน้าที่อื่นมากกว่า
- คิดว่าตนเองไม่มีความคิดสร้างสรรค์ ชอบอยู่ในกรอบ
- คิดว่าตนเองจะไม่เข้าใจระบบที่จะพัฒนาได้ดีพอ

DevOps Manager

- เรียกได้ว่าเป็น “ฝ่าย IT ของกลุ่ม”
- รับผิดชอบในการจัดการ, สนับสนุน, และเลือกเทคโนโลยีต่างๆที่ทีมใช้ เช่น จัดการและให้คำแนะนำเกี่ยวกับการใช้งาน Github, CI/CD (Jenkins), การ Deploy ระบบลงบน Cloud server, Monitoring tools, เป็นต้น
- กล้าที่จะเปลี่ยนการตั้งค่าต่างๆของคอมพิวเตอร์หรือเทคโนโลยีที่นำมาใช้
- วิเคราะห์ประสิทธิภาพการทำงานของทีมและนำเทคโนโลยีต่างๆมาใช้เพื่อเพิ่มประสิทธิภาพในการทำงานของทีม
- ช่วยหาคำตอบและแก้ไขปัญหาทางด้านเทคโนโลยีที่เกิดขึ้น หรือเป็น contact point ที่คนในทีมจะเข้าหาหากเกิดปัญหาทางด้านเทคโนโลยีที่ทีมใช้อยู่
- ช่วยให้แน่ใจว่าทีมมีเครื่องมือที่ต้องการใช้ในเวลาที่ต้องการใช้

DevOps Manager



เหตุผลที่ควรเลือกเป็น DevOps Manager:

- คุณเคยอยากรู้เพิ่มเติมเกี่ยวกับเครื่องมือต่างๆ เช่น Github, Jenkins, Kubernetes, Docker, Ansible เป็นต้น
- ชอบปรับเปลี่ยนการตั้งค่า กล้าลองถูกลองผิด เพื่อให้สิ่งต่างๆทำงานได้ดีและมีประสิทธิภาพ
- มีความอดทนในการช่วยเหลือผู้อื่นและชอบค้นคว้า



เหตุผลที่ไม่ควรเลือกเป็น DevOps Manager:

- คิดว่าตนเองมันใจว่าจะช่วยเพื่อนในทีมแก้ปัญหาทางเทคนิคได้
- ไม่มีประสบการณ์หรือไม่อยากรู้เพิ่มเติมเกี่ยวกับเครื่องมือต่างๆข้างต้น
- อยากรับหน้าที่อื่นมากกว่า

*วิชานี้มีทุนให้ทุกกลุ่ม deploy งานบน cloud server (ที่เสียเงิน)

UI/UX Designer

- เรียกได้ว่าเป็น “ดีไซเนอร์ของทีม”
- หน้าที่รับผิดชอบหลักคือการออกแบบรูปแบบรู้สึกษณ์โดยรวม (look-and-feel) ของตัวโครงการ และจัดการสไตล์ในทุก ๆ หน้า
- ทำงานร่วมกับ Requirements Manager เพื่อศึกษาให้เข้าใจว่าผู้ใช้เป็นใคร มีข้อจำกัดอะไรบ้าง และต้องเข้าใจเป้าหมายและวัตถุประสงค์ของโครงการอย่างถ่องแท้
- เป็นบุคคลที่ความมีประสบการณ์ด้าน Front-end ที่สุดในทีม (รู้วิธีการใช้เทคโนโลยีสำคัญๆ เช่น HTML, CSS, JS, Bootstrap หรือเครื่องมือ เช่น Figma เป็นต้น)
- ตรวจสอบ Personas ที่สมาชิกในทีมสร้าง
- ทำงานร่วมกับ Testing Manager ในการจัดการทำ Beta Testing หรือ Capability Drive-through กับผู้ว่าจ้างหรือผู้ใช้ที่เป็นกลุ่มเป้าหมาย
- จัดทำเอกสารรายงานผลลัพธ์ของ Beta Testing ในตอนท้ายของการศึกษา

UI/UX Designer



เหตุผลที่ควรเลือกเป็น UI/UX Designer:

- มีประสบการณ์หรืออยากเรียนรู้เกี่ยวกับเทคโนโลยีทาง front-end ต่างๆ เช่น JS/Bootstrap/CSS/React/Vue.js/Angular หรือเครื่องมือเช่น Figma
- มีความสนใจในการ design หรือชอบนิ่งวิธีต่างๆ ที่ผู้ใช้สามารถ interact กับระบบได้ ชอบนำวิทยาศาสตร์กับศิลปะมาร่วมกัน เป็นคนมีสไตล์
- อยากให้โครงงานออกมา “ดูดี” และ “ใช้งานได้ดี”



เหตุผลที่ไม่ควรเลือกเป็น UI/UX Designer:

- ไม่มีประสบการณ์และไม่อยากศึกษาการออกแบบหน้า interface ของเว็บหรือ mobile app เลย
- คิดว่าตนเองไม่มีความคิดสร้างสรรค์หรือไม่สนใจว่าผู้ใช้จะ interact กับระบบอย่างไร

Tech Lead/ Software Architect

- เรียกได้ว่าเป็น “สภานิกของทีม”
- รับผิดชอบในการเลือกและตัดสินใจทางเทคนิค ดูว่าทีมควรเลือกใช้ Framework อะไรหรือ Library อะไร, Standard แบบไหน, หรือเทคโนโลยี Stack ไหน
- รับผิดชอบการนำทีมวิเคราะห์ความเป็นไปได้ของระบบ
- รับผิดชอบในการออกแบบระดับสูง (high-level) หรือแม่แบบ (blue-print) ของตัวระบบ เช่นว่า component นี้เชื่อมต่อกับ component อื่นๆในระบบอย่างไร การติดต่อสื่อสารหากันระหว่าง services ที่ใช้เป็นอย่างไร
- รับผิดชอบในตรวจสอบว่าทีมได้พัฒนาระบบทามที่ได้ออกแบบไว้
- รับผิดชอบในการบริหาร (ระบุ, วิเคราะห์, ประเมิน, และเสนอวิธีการในการจัดการ) ความเสี่ยงทางเทคนิคต่างๆที่สามารถเกิดขึ้นได้

Tech Lead / Software Architect



เหตุผลที่ควรเลือกเป็น Software Architect:

- มีประสบการณ์ในการออกแบบซอฟต์แวร์ รู้จักภาษา UML และเคยใช้ Architecting Tools พวก Draw.io, Visual Paradigm, StarUML
- มีความสนใจใน design patterns และ technology stack ต่างๆ
- คิดว่าจะเข้าใจภาพรวมของโปรเจคได้เป็นอย่างดี



เหตุผลที่ไม่ควรเลือกเป็น Software Architect:

- ไม่มีประสบการณ์และไม่สนใจการออกแบบระบบ
- อยากรับหน้าที่อื่นมากกว่า
- ไม่มีความสนใจเกี่ยวกับ technology stack หรือ design patterns
- ไม่อยากเป็นผู้เลือกและตัดสินใจ

Database Engineer

- เรียกได้ว่าเป็น “**คนจัดการฐานข้อมูลของทีม**”
- ทำงานร่วมกับ Software Architect แต่จะรับผิดชอบใส่ส่วนของการออกแบบและสร้างโมเดลของฐานข้อมูล ว่าฐานข้อมูลควรมีโครงสร้าง (structure) และการจัดระเบียบอย่างไร ให้มีประสิทธิภาพมากที่สุด
- ศึกษา, ประเมิน, และเลือกใช้ชนิดของฐานข้อมูล (Database Management System) ให้เหมาะสมกับทีม
- ค้นหาและแก้ไขปัญหาที่เกี่ยวข้องกับฐานข้อมูลต่างๆ ที่อาจเกิดขึ้น
- รับผิดชอบในการตรวจสอบว่าข้อมูลที่เข้าและออกของฐานข้อมูลนั้นถูกต้องและมีเหตุมีผล เช่น ฐานข้อมูลไม่ควรจะรับเข้าราคาสินค้าที่เป็นเลขติดลบ
- ช่วยทีมปรับจูนประสิทธิภาพของการเข้าถึงฐานข้อมูล เช่น SQL Query ต่างๆ ที่สามารถใช้ในทีมเขียนขึ้น

Database Engineer



เหตุผลที่ควรเลือกเป็น Database Engineer:

- มีประสบการณ์หรือสนใจในการออกแบบและจัดการฐานข้อมูล
- เคยใช้หรืออยากรู้ศึกษาเพิ่มเติมเกี่ยวกับ Database Management System ชนิดต่างๆ เช่น MySQL, Postgres, MongoDB, Redis, Firebase เป็นต้น



เหตุผลที่ไม่ควรเลือกเป็น Database Engineer:

- ไม่มีประสบการณ์และไม่สนใจฐานข้อมูล
- อยากรับหน้าที่อื่นมากกว่า
- ไม่เห็นความสำคัญของการจัดการฐานข้อมูล

Client Interaction

First Meeting, Project Kickoff

Before the meeting:

- นัดเวลาประชุมที่ทุกคนต้องสามารถเข้าประชุมได้
- เตรียมคำานวณ
- อ่านข้อมูลที่เกี่ยวข้อง

Structuring the meeting (Agenda):

- แนะนำตัวของคนในทีมและ background
- ให้ Client แนะนำตัวและ Project
- ช่วงถามคำถามที่เตรียมไว้
- ทีมพูดถึง timeline
- ทีมพูดถึง ways to communicate/ work together/ be in touch (e.g., regular meeting)
- ทีมพูดถึง next activites



Tips: คำาณส่วนใหญ่ควร focus เรื่อง system usage (การใช้งานของระบบ) แทนที่จะเป็นการพัฒนาระบบ และหลีกเลี่ยงการใช้ technical terms

Some Questions to Ask Your Client

- ระบบที่มีอยู่ในปัจจุบัน (อาจจะเป็นแบบ manual) อุปสรรค หรือ ปัญหาที่อยากให้ระบบที่ต้องการเข้ามาแก้ไข
- ใครมีส่วนเกี่ยวข้องโดยตรงกับตัวโครงการ (success-critical stakeholders)
- เป้าหมายของตัวโครงการ (Goals and Value Propositions)
- ประโยชน์เชิงปริมาณและเชิงคุณภาพที่คาดว่าจะได้รับ (Quantitative and Qualitative benefits)
- ผู้ที่จะได้รับผลประโยชน์หรือผู้ใช้งานโครงการนี้ (Beneficiaries)
- Features/Capabilities ที่สำคัญต่อความสำเร็จของตัวโครงการ
- เทคโนโลยีและข้อจำกัดต่างๆ
- คิดว่าความสำเร็จของตัวโครงการหมายถึงอะไร หน้าตาเป็นอย่างไร
- มีหน้าตาของ project คร่าวๆที่ได้คิดไว้แล้วหรือไม่ ระบบที่มีความใกล้เคียง
- และคำรามอื่นๆที่เป็นประโยชน์ต่อความเข้าใจในตัวโครงการของกลุ่ม

Make Your Client Part of the Team

- Schedule a recurring meeting (at least for an hour every two weeks
- **A MUST**)
- Involve your client in any major decision-making.
- Update them on the progress and ask for their feedback regularly.
- Add them to your group chat (if they are okay with it).
- Include them in **some or all** of the **Scrum activities** (for example, sprint reviews, sprint planning, requirements prioritization).
 - Daily Stand-ups (if possible)





Client Evaluation (Quest) – 100 XP

At the end of the semester, your client will evaluate the team on multiple criteria

- **Overall professionalism/performance**

- เช่น ความใส่ใจในการทำงาน, ทัศนคติต่อตัวโครงการ, การให้ความสำคัญกับ feedback ที่ได้รับ, ความกระตือรือร้น เป็นต้น

- **Team collaboration/communication**

- เช่น ความสามารถทำในการทำงานร่วมกัน, การติดต่อสื่อสารที่คุ้มกัน, การใช้เครื่องมือสื่อสาร, การนัดประชุมและเข้าประชุม เป็นต้น

- **The quality of the project**

- เช่น ผลงานตรงกับความต้องการและความคาดหวังของผู้ใช้ (needs and expectations), มีการตรวจสอบคุณภาพด้วยการ testing, ความสามารถในการใช้งานของโครงการและหน้าตา interface (Usability), ประสิทธิภาพการทำงานอยู่ในระดับที่น่าพอใจ เป็นต้น

Questions ?

