รายงาน

โครงการ SideSpark

เรื่อง ระบบสำรวจไอเดีย side hustle สำหรับนักศึกษา

จัดทำโดย

นางสาววรรณภา ดงงาม	รหัสนักศึกษา 6409700082
นางสาวณัฐกฤตา ธนัทนันท์	รหัสนักศึกษา 6709618034
นายเปรมช์ เผ่าเสถียรฐานนท์	รหัสนักศึกษา 6709618075
นายพรเทวา จินดาธรรม	รหัสนักศึกษา 6709618109
นางสาวปาณภัทร จันทร์คีรี	รหัสนักศึกษา 6709700014
นายทัตพงศ์ ทรายคำ	รหัสนักศึกษา 6709700048

เสนอ

ผศ. ดร.วรวรรณ ดีอัซ การบาโย

รายงานนี้เป็นส่วนหนึ่งของรายวิชา ทนด. 241 ระบบนิเวศซอฟต์แวร์สำหรับนักพัฒนา

ภาคการศึกษาที่ 2 ปีการศึกษา 2568

สาขาวิทยาศาสตรบัณฑิตสาขาวิชาเทคโนโลยีและนวัตกรรมดิจิทัล (หลักสูตรใหม่ พ.ศ. 2566)

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ศูนย์ลำปาง

1. บทนำ

SideSpark เป็นระบบที่ช่วยนักศึกษาค้นหาไอเดียหารายได้เสริม (side hustle) ที่เหมาะสมกับทักษะของตนเอง พร้อมทั้งมีเครื่องมือสำหรับการวางแผนธุรกิจขนาดเล็ก และการติดตามรายได้อย่างเป็นระบบ

ระบบประกอบด้วยฟีเจอร์หลัก ได้แก่ Idea Explorer, Mini Business Plan, Income Tracker และ Progress Dashboard

เอกสารฉบับนี้จัดทำขึ้นเพื่อสรุปบทเรียนที่ได้รับจากการพัฒนาโครงการผ่านกระบวนการ Agile จำนวน 3 Sprint โดยมุ่งเน้นการสะท้อนปัญหา การทำงานของทีม และการปรับปรุงกระบวนการพัฒนาอย่างต่อเนื่องตลอดโครงการ

2. การสะท้อนผล Sprint ที่ 1

2.1 สิ่งที่ทำได้ดี

2.1.1 สามารถเริ่มต้นออกแบบระบบและกำหนดแนวคิดของโครงการได้อย่างชัดเจน

2.1.2 มีการวางโครงสร้างของฟีเจอร์หลักของระบบ ได้แก่ Idea Explorer, Business Plan และ Income Tracker

2.1.3 เริ่มต้นการพัฒนาและออกแบบ UI เบื้องต้นได้ตามแผนที่กำหนด

2.2 ปัญหาที่พบ

2.2.1 การออกแบบ UI ไม่สอดคล้องกับทิศทางที่เหมาะสม โดยมีการออกแบบในลักษณะ mobile-first แทนที่จะเป็น desktop-first

2.2.2 Requirement ยังขาดความชัดเจนในช่วงเริ่มต้นของโครงการ

2.2.3 ไม่สามารถดำเนินการแก้ไข feedback จาก QA ได้ทันเวลา ส่งผลให้บางประเด็นถูกส่งต่อไปยัง Sprint 2

2.3 สิ่งที่ปรับปรุง

2.3.1 ปรับแนวทางการออกแบบ UI ให้สอดคล้องกับ desktop-first มากขึ้น

2.3.2 เพิ่มกระบวนการทบทวน requirement ก่อนเริ่มการพัฒนาในแต่ละรอบ

2.3.3 วางแผนการรับและจัดการ QA feedback ให้มีประสิทธิภาพและรวดเร็วมากขึ้นในรอบถัดไป

3. การสะท้อนผล Sprint ที่ 2

3.1 สิ่งที่ทำได้ดี

3.1.1 ระบบมีความสมบูรณ์มากขึ้นในส่วนของฟีเจอร์หลัก

3.1.2 การพัฒนาในส่วน backend และ frontend สามารถเชื่อมต่อกันได้ดีขึ้น

3.1.3 ทีมมีความเข้าใจ flow ของระบบมากขึ้นเมื่อเทียบกับ Sprint แรก

3.2 ปัญหาที่พบ

3.2.1 การกระจายงานภายในทีมยังไม่สมดุล ส่งผลให้เกิด bottleneck กับสมาชิกบางคน ซึ่งส่งผลกระทบต่อความเร็วในการพัฒนาและการส่งมอบงาน

3.2.2 การกระจายงานภายในทีมยังไม่ทั่วถึง

3.2.3 ปัญหาบางส่วนที่ตกค้างจาก Sprint 1 ยังไม่ได้รับการแก้ไขอย่างครบถ้วน

3.3 สิ่งที่ปรับปรุง

3.3.1 พยายามกระจายงานให้สมาชิกทุกคนมีส่วนร่วมในการพัฒนาอย่างเหมาะสมมากขึ้น

3.3.2 ปรับปรุงการสื่อสารภายในทีมให้มีความถี่และชัดเจนมากขึ้น เพื่อลดปัญหางานค้าง

3.3.3 เริ่มจัดลำดับความสำคัญของงานให้มีความชัดเจนมากขึ้น

4. การสะท้อนผล Sprint ที่ 3

4.1 สิ่งที่ทำได้ดี

4.1.1 ระบบโดยรวมมีความสมบูรณ์มากขึ้น และสามารถใช้งานได้ในระดับหนึ่ง

4.1.2 Dashboard และฟีเจอร์การติดตาม สามารถทำงานได้ตามเป้าหมาย

4.1.3 มีการปรับปรุง UI ให้มีความเหมาะสมและสอดคล้องกับการใช้งานมากขึ้น

4.2 ปัญหาที่พบ

4.2.1 การ deploy ระบบขึ้น AWS EC2 ยังไม่สามารถดำเนินการได้สำเร็จภายในระยะเวลาที่กำหนด

4.2.2 ระยะเวลาที่เหลือมีอย่างจำกัด ส่งผลให้บางฟีเจอร์ยังไม่ได้รับการปรับปรุงอย่างเต็มที่

4.2.3 ระบบยังคงมีบางส่วนที่ยังไม่เสถียร

4.3 สิ่งที่ปรับปรุง

4.3.1 วางแผนการ deploy ตั้งแต่ช่วงเริ่มต้นของโครงการ

4.3.2 มีการปรับลด scope ของฟีเจอร์บางส่วน เพื่อให้สามารถส่งมอบระบบที่มีความเสถียรและครบถ้วนภายในระยะเวลาที่กำหนด

4.3.3 เพิ่มกระบวนการทดสอบระบบให้มากขึ้นในช่วงท้ายของแต่ละ Sprint

5. บทเรียนที่ได้รับจากโครงการ

5.1 การออกแบบระบบควรเริ่มต้นจาก requirement ที่มีความชัดเจนก่อนเริ่มการพัฒนา

5.2 การทำงานแบบ Agile จำเป็นต้องอาศัยการสื่อสารภายในทีมอย่างสม่ำเสมอ

5.3 การแบ่งงานที่ไม่สมดุลสามารถส่งผลให้เกิด bottleneck ในการพัฒนา

5.4 การรับ feedback จาก QA และอาจารย์ควรดำเนินการแก้ไขโดยทันที ไม่ควรปล่อยค้างไว้

5.5 การวางแผนการ deploy ควรถูกกำหนดตั้งแต่ช่วงเริ่มต้นของโครงการ ไม่ควรทำในช่วงท้าย

6. สิ่งที่ควรปรับปรุงหากทำโครงการอีกครั้ง

6.1 ควรเริ่มจากการออกแบบ system architecture และ requirement ให้มีความชัดเจนตั้งแต่ Sprint แรก

6.2 ควรมีการกระจายงานให้สมาชิกในทีมอย่างเท่าเทียมมากขึ้น

6.3 ควรเพิ่มการสื่อสารและการประชุมติดตามผล (daily/weekly check-in) อย่างสม่ำเสมอ

6.4 ควรวางแผนด้าน deployment และ testing ตั้งแต่ช่วงกลางของโครงการ

6.5 ควรลด scope ของฟีเจอร์เพื่อให้สามารถส่งมอบงานที่มีความสมบูรณ์และเสถียรมากขึ้น

7. สรุป

โครงการ SideSpark ทำให้ทีมได้รับประสบการณ์ในการพัฒนาซอฟต์แวร์แบบ Agile อย่างแท้จริง ทั้งในด้านการออกแบบระบบ การพัฒนา การทำงานเป็นทีม และการจัดการปัญหาที่เกิดขึ้นระหว่างการดำเนินงาน แม้จะมีข้อจำกัดด้านเวลาและการจัดการทรัพยากรภายในทีม แต่ทีมสามารถพัฒนาระบบให้มีฟีเจอร์หลักครบถ้วน และได้รับบทเรียนสำคัญซึ่งสามารถนำไปประยุกต์ใช้ในการพัฒนาโครงการในอนาคตได้