

**TTSA01 : Managing Non-Functional Requirements and Business Goals****Description :**

ถ้าคุณกำลังเริ่ม Software Development และกำลังอยู่ในช่วง Get Functional Requirements แล้ว ต้องไม่ลืม Non-Functional Requirement และ Business Requirements ด้วย เพราะ Non-Functional Requirement เป็น Requirements ที่ "ต้อง" คำนึงถึงเสมอ เพราะถึง Software จะทำงานได้ทุกอย่างตาม Function แต่ไม่มีประสิทธิภาพ ไม่ปลอดภัย ไม่สิ้นเปลือง ไปคนละทางกับ Business Goal ก็จบกัน

**Instructor :**

Training Date : 4 ก.ย. 2567 - 6 ก.ย. 2567

fee : 12000 ฿ (ราคายังไม่รวม Vat 7%)

Days &amp; Duration : 3 Day(s) | 18 Hour(s)

Time : 10:00:00 - 17:00:00

Language : Thai

Venue : Training/Seminar Room, 3rd FL., Software Park Building Chaengwattana Road, Nonthaburi 11120

Type : Classroom

Category : Software Architecture and Design

[Mr.Narong Chansoi](#)**Objectives :**

หลักสูตรนี้ เน้นถึงการทำความเข้าใจกับความต้องการประเภท Non-Functional Requirements และ Business Requirements หรือ Business Goals โดยผู้เข้ารับการอบรมจะสามารถกำหนด, จัดกลุ่ม, วิเคราะห์ และจัดการได้อย่างมีประสิทธิภาพเพื่อให้ระบบ ที่จะพัฒนามีคุณภาพ คุ้มค่า และเกิดประโยชน์สูงสุด และนอกจากนี้ผู้เข้ารับการอบรมจะเข้าใจถึงการจัดการด้านคุณภาพโดยมีกรอบของ Non-Functional Requirements และ Business Requirements หรือ Business Goals เป็นตัวชี้นำ (KPI) และผู้เข้ารับการอบรมจะได้เรียนรู้และเข้าใจในทักษะที่เป็น Soft Skill อีกมากมาย

ปัญหาในการทำ Requirements Management ในปัจจุบันคือมีการใช้ของความรู้แบบเก่า ซึ่งไม่มีประสิทธิภาพและทันต่อการเปลี่ยนแปลงของเทคโนโลยีและกระบวนการพัฒนาซอฟต์แวร์สมัยใหม่ และการทำ Requirements Management แบบดั้งเดิมมักไม่ให้ความสำคัญกับสิ่งสำคัญอื่นมากมาย เช่น

- ความต้องการหรือเป้าหมายทางธุรกิจ (Business Goals)
- ความต้องการประเภท Non-Functional Requirements ซึ่งมักส่งผลกระทบต่อคุณภาพและความเสี่ยงมากกว่า Functional Requirements
- คุณภาพของสถาปัตยกรรมซอฟต์แวร์และระบบ
- ภาษาและการสื่อสาร
- การเจรจาต่อรอง
- การลดข้อขัดแย้งระหว่างการประชุม
- การจัดการการเปลี่ยนแปลง
- ความเข้าใจในภาคธุรกิจและโมเดลธุรกิจ
- ปัญหาการเชื่อมโยงภายในและระหว่างองค์กร และปัญหาองค์กรชั้น
- การตัดสินใจ
- การเงินและการประมาณการเบื้องต้น
- การวิจัยตลาด เช่น การทำแบบสำรวจ การรวบรวมความต้องการจากแหล่งข้อมูลต่างๆ
- กระบวนการพัฒนาซอฟต์แวร์สมัยใหม่ เช่น RUP, CMMI, Agile เช่น Extreme Programming, Scrum
- หลักการพัฒนาซอฟต์แวร์สมัยใหม่ เช่น Object-Oriented Programming
- การใช้ UML (Unified Modeling Language)
- การอธิบายความต้องการให้อยู่ในรูปแบบของ Software Definition
- ฯลฯ

นอกจากนี้ ปัญหาในการจัดการ Requirements ในแบบที่ผิดมักเน้นหนักที่การจัดการ Functional Requirements แต่ในความเป็นจริงปัจจัยที่จะส่งผลกระทบต่อสถาปัตยกรรมซอฟต์แวร์และระบบ คือ Non-Functional Requirements และ Business Requirements หรือ Business Goals ซึ่ง Requirements สองประเภทนี้หากจัดการผิดพลาด จะส่งผลกระทบต่อคุณภาพของสถาปัตยกรรมซอฟต์แวร์และระบบ ทำให้เกิดความเสียหายในด้านค่าใช้จ่าย กำลังคน เวลา ทรัพยากรมากมาย

การอบรมได้อ้างอิง องค์ความรู้จากผลงานการวิจัยและพัฒนาโดย Rational ในแนวทาง Unified Approach ซึ่งสามารถปรับใช้ได้กับหลายกระบวนการพัฒนาซอฟต์แวร์ (Software Development Process) เช่น RUP, CMMI, Agile (Extreme Programming, Scrum ฯลฯ) และองค์ความรู้ด้านสถาปัตยกรรมซอฟต์แวร์ของสถาบัน Software Engineering Institute (SEI) แห่งมหาวิทยาลัย Carnegie Mellon ประเทศสหรัฐอเมริกา และสถาบัน World Wide Institute of Software Architects และได้อ้างอิงความรู้อื่นอีกมากมาย เช่น การค้นหาและรับการปัญหาการเชื่อมโยง การวิจัยตลาด จิตวิทยาการสื่อสาร การเจรจาต่อรอง การเงิน เป็นต้น

การจัดการความต้องการ หรือ requirements Management ในปัจจุบันถือว่าเป็นปัญหาในขั้นวิกฤต เนื่องจากเป็นขั้นตอนที่สำคัญมาก เพราะเป็นขั้นตอนแรกๆ เพื่อทำความเข้าใจกับความต้องการก่อนที่จะนำผลการวิเคราะห์ความต้องการไป วิเคราะห์และออกแบบระบบ และเข้าสู่ขั้นตอนต่อไป ตามกระบวนการพัฒนาซอฟต์แวร์ ดังนั้นเห็นได้ว่าหากขั้นตอนแรกนี้ผิดพลาดจะมีโอกาสสูงที่จะส่งผลกระทบต่อ ขั้นตอนต่อไป ได้ โดยผลกระทบที่ตามมาอาจเต็มไปดวย ค่าใช้จ่าย กำลังคน เวลา และทรัพยากร

**Target Group :**

- การอบรมนี้ออกแบบมา สำหรับ System Analyst, Business Analyst, (Software) Sales Specialist, IT Manager / CIO / CTO, Software Architect, Programmer และผู้สนใจทั่วไปในการพัฒนาซอฟต์แวร์

## พื้นฐานของผู้เข้ารับการอบรม (Prerequisites)

- ผู้เข้ารับการอบรมควรมีความรู้ด้านกระบวนการพัฒนาซอฟต์แวร์มาก่อน

### Benefits :

ในการอบรมนี้ผู้เข้ารับการอบรมจะได้เข้าใจการจัดการ Non-Functional Requirements และ Business Requirements หรือ Business Goals โดยละเอียด เพื่อช่วยทำให้ส่วนการออกแบบสถาปัตยกรรมซอฟต์แวร์ (Software Architecture) มีคุณภาพและคุ้มค่ากับงบประมาณมากที่สุด และยังช่วยจัดการความต้องการด้าน Non-Functional Requirements และ Business Requirements หรือ Business Goals ได้อย่างมีประสิทธิภาพ

ผู้เข้ารับการอบรมจะได้เรียนรู้เทคนิคที่เป็น Non-Technical เพื่อนำมาประยุกต์มากมาย เช่น การวิจัยตลาด จิตวิทยาการสื่อสาร เทคนิคการตั้งคำถาม การลดความขัดแย้ง บุคลิกภาพ การทำความเข้าใจแผนธุรกิจ ทักษะทางภาษาที่จำเป็น ฯลฯ เพื่อประกอบการทำงานได้อย่างมีประสิทธิภาพ

และผู้เข้ารับการอบรมยังจะได้เรียนรู้หลักพื้นฐานและความสำคัญของการทำงาน สถาปัตยกรรมซอฟต์แวร์ (Software Architecture) อีก ด้วย

นอกจากนี้ผู้เข้ารับการอบรมจะได้ เรียนรู้พื้นฐานเทคนิคการประยุกต์โปรแกรมออฟฟิศ คือ โปรแกรมสเปรดชีต เช่น MS Excel เพื่อการทำ Requirements Attributes, Requirements Matrix, Requirements Measurement และ Requirements Traceability

### Course Outline :

- ปัญหาการทำ Requirements Management ในปัจจุบัน
- ความสำคัญและหลักการทำให้ Requirements Management เบื้องต้น
- สถาปัตยกรรมซอฟต์แวร์ (Software Architecture) เบื้องต้น
- วงจรการตอบสนองซึ่งกันและกันของธุรกิจและ สถาปัตยกรรม (Architectural Business Cycle)
- การกำหนดผู้ที่มีผลต่อความสำเร็จหรือล้มเหลวของโครงการ (Stakeholder), ผู้นำและผู้สนับสนุนทีม (Champion) และ ผู้เชี่ยวชาญเฉพาะด้าน (Domain Expert)
- กระบวนการพัฒนาซอฟต์แวร์แบบวนซ้ำ (Iterative and Incremental Development) สำหรับกระบวนการพัฒนาซอฟต์แวร์ สมัยใหม่
- ปรัชญา Agile เพื่อการพัฒนาซอฟต์แวร์ที่กระชับ รวดเร็ว เรียบง่าย ความเสี่ยงต่ำ และสร้างความพึงพอใจของลูกค้า (Customer Satisfaction) สูงสุด
- ช่วงการทำสถาปัตยกรรมซอฟต์แวร์ ในกระบวนการพัฒนาซอฟต์แวร์ (Architecture In The Life Cycle)
- คุณสมบัติ บทบาท หน้าที่รับผิดชอบ และทีมในการดำเนินการ
- ความสำคัญและ หลักการของ Business Modeling และ การวิเคราะห์ Business Model
- การแบ่งประเภทระหว่าง Functional Requirements, Non-Functional Requirements และ Business Goals หรือ Business Requirements
- ความสำคัญและผลกระทบของ Non-Functional Requirements และ Business Requirements หรือ Business Goals ที่มีต่อสถาปัตยกรรมซอฟต์แวร์และระบบฯ
- เทคนิควิเคราะห์ ปัญหา
- การระบุความต้องการ การเชิงธุรกิจ (Business Goals / Requirements / Stakeholder Needs) และการการวิเคราะห์จาก Vision Document หรือ Operational Concepts Document (OCD) หรือ แผนธุรกิจ (Business Plan)
- ทำความเข้าใจกับ คุณสมบัติเชิงคุณภาพ (Quality Attributes) และการกำหนด Non-Functional Requirements และ Business Goals เช่น Availability, Modifiability, Scalability, Security, Performance, Interoperability, Integrability, Testability, Usability, Reliability, Short Time to Market, Less Cost, Neat Feature, Competitiveness ฯลฯ
- การกำหนด คุณสมบัติด้านคุณภาพ (Quality Attributes) ของสถาปัตยกรรมซอฟต์แวร์ และระบบฯ
- การทำความเข้าใจกับความต้องการของลูกค้าและผู้ใช้ ที่อาจไม่ได้บอกตรง ๆ
- การลดข้อขัดแย้ง และจัดการการสื่อสารกับลูกค้า
- เทคนิคการสัมภาษณ์ การตั้งคำถาม จิตวิทยาการสื่อสาร การทำแบบสอบถาม เพื่อรวบรวมและค้นหา Non-Functional Requirements และ Business Requirements หรือ Business Goals แท้จริง หรือซ่อนอยู่ หรือถูกปิด
- การทำเวิร์คช็อป เพื่อให้วิสัยทัศน์และความต้องการเป็นไปในทิศทางเดียวกัน
- การค้นหาและรับมือกับปัญหาการเมือง ภายในและระหว่างองค์กร และการคอร์รัปชัน
- การเขียน Use Case Model และการวิเคราะห์ Use Case
- เทคนิคใหม่ในการ ระบุความต้องการด้วยการเขียน Storyboard และ Scenario โดยมีการประยุกต์ UML ด้วยเพื่อจำลองเหตุการณ์ที่สอดคล้องกับความต้องการ
- การกำหนดนิยาม (Definition)
- การกำหนดคุณสมบัติของความต้องการ (Requirement Attributes) และการทำ Requirement Matrix
- การวัดผลและประเมิน (Requirements Measurement), การกำหนดตัวขับเคลื่อนกระบวนการทาง สถาปัตยกรรม (Architectural Drivers) และทำความเข้าใจกับ Use Case ที่มีผลต่อสถาปัตยกรรมซอฟต์แวร์และระบบฯ (Architecturally Significant Use Cases)
- การกำหนดและ จัดการขอบเขตของระบบที่จะพัฒนา และ Requirements ที่มีผลต่อสถาปัตยกรรมซอฟต์แวร์และระบบฯ
- การจัดการเอกสาร Requirements ต่าง ๆ เช่น Requirements Specification, Use Case Specification / User Story เป็นต้น
- การตรวจสอบย้อน กลับ (Traceability) เพื่อให้ระบบฯ ถูกต้องตามความต้องการและตรวจสอบแก้ไขได้ง่ายและสะดวก
- การ Transform จาก Requirements สู่ Implementation และทำความเข้าใจกับ The Level of Abstraction และกระบวนการ Realization
- ทำความเข้าใจกับ Verification และ Validation
- การจัดการรวมกับการจัดการการเปลี่ยนแปลง (Change Management)

### Payment Condition :

Payment can be made by:

1. Cash or Credit Card or Bank Cheque payable to "สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ" (a post-dated cheque is not accepted) on the first day of the service or within the last day of the service.
2. **Account transfer** and send the proof of the payment (the deposit slip) to email [ttd@swpark.or.th](mailto:ttd@swpark.or.th)

- ธนาคารกรุงเทพ สาขาอุทยานวิทยาศาสตร์  
Saving Account Number: 080-0-00001-0  
Account Name: สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ
- ธนาคารกรุงไทย สาขาลาดไทย  
Saving Account Number: 152-1-32668-1  
Account Name: สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

**Notes:**

- Withholding tax (3%) is exempt.
- Should you need to withdraw, you must send the notice of the withdrawal in writing no later than 7 working days before the commencement date. The cancellation less than 7 days will be subject to a fine of 40% of the fee.
- Software Park Thailand reserves the rights to cancel courses due to unforeseen circumstances.

**Contact Person :**

For more information, contact our course coordinator on:

Namphon Pongyat

Tel: +66-2583-9992 Ext. 81427

Email: [namphon@swpark.or.th](mailto:namphon@swpark.or.th)



You are encouraged to use the course schedule as a guide to plan your training.

The schedule is accessible at [www.swpark.or.th](http://www.swpark.or.th) for more information.