หลักสูตรฝึกอบรมฐานสมรรถนะ สาขาซอฟต์แวร์และการประยุกต์ อาชีพนักพัฒนาระบบ ระดับ 5 (Programmer)

วัตถุประสงค์ของหลักสูตร

1. สามารถแก้ไขข้อผิดพลาดของโปรแกรม โดยหาจุดผิดพลาดตามบันทึกข้อผิดพลาดและทดสอบโปรแกรมหลังจากการแก้ไขข้อผิดพลาดดังกล่าวได้
2. สามารถดำเนินการพัฒนาโปรแกรมแบบซับซ้อนในระดับองค์กรขนาดใหญ่ (Enterprise / Big Scale)โดยอ่าน Functional / Program Specification / UML และเขียนโปรแกรมตาม Functional / Program Specification / UML ได้
3. สามารถออกแบบกระบวนการพัฒนาโปรแกรมแบบ Continuous Delivery และ Continuous Integration โดยวิเคราะห์ Continuous Delivery และ Continuous Integration และออกแบบ Continuous Delivery และ Continuous Integration ได้
4. สามารถพัฒนาโปรแกรมตามมาตรฐานและตรวจสอบมาตรฐานการเขียนโปรแกรม (Code Review) โดยวางแผนในการพัฒนา Code Review และพัฒนา Code Review ได้

คุณสมบัติผู้เข้ารับการประเมิน

1. ผู้ที่สำเร็จการศึกษาระดับ ปริญญาตรี ในด้านการพัฒนาโปรแกรม หรือที่เกี่ยวข้อง หรือ
2. มีประสบการณ์ทำงานด้านการพัฒนาโปรแกรม หรือที่เกี่ยวข้องไม่น้อยกว่า 5 ปี หรือ
3. ได้รับรองคุณวุฒิวิชาชีพ สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาซอฟต์แวร์และการประยุกต์ อาชีพนักพัฒนาระบบ ระดับ 4 แล้วเป็นระยะเวลาไม่น้อยกว่า 1 ปี

หน่วยสมรรถนะที่ใช้ในอบรมและการประเมิน

1. แก้ไขข้อผิดพลาด
2. ดำเนินการพัฒนาโปรแกรมแบบซับซ้อน (Enterprise / Big Scale)
3. ออกแบบ Continuous Delivery และ Continuous Integration
4. พัฒนาโปรแกรมตามมาตรฐานและตรวจสอบมาตรฐานการเขียนโปรแกรม (Code Review)

จำนวนผู้เข้ารับการประเมิน

50 คน / หลักสูตร

ระยะเวลาการอบรม พร้อมสอบประเมิน

ฝึกอบรม จำนวน 2 วัน (12 ชั่วโมง)

- ภาคทฤษฎี 4 ชั่วโมง
- ภาคปฏิบัติ 8 ชั่วโมง

สอบประเมิน จำนวน 1 วัน

- ความรู้ทฤษฎี สอบประเมินในวันอบรม
- ความรู้ทฤษฎีเชิงปฏิบัติ โดยการสอบสัมภาษณ์ (ภายหลัง)

เกณฑ์การผ่านการฝึกอบรม

1. ผู้เข้าอบรมต้องเข้ารับการฝึกอบรม ไม่น้อยกว่าร้อยละ 80 ของระยะเวลาฝึกอบรมตลอดหลักสูตร
2. ผู้เข้าอบรมต้องทดสอบประเมินความรู้ภาคทฤษฎีด้วยแบบประเมินผลก่อนการฝึกอบรม (Pre-Test) เกณฑ์ผ่านไม่น้อยกว่าร้อยละ 70
   ผู้เข้าอบรมต้องทดสอบประเมินความรู้ภาคทฤษฎีด้วยแบบประเมินผลหลังการฝึกอบรม (Post-Test) เกณฑ์ผ่านไม่น้อยกว่าร้อยละ 70

เกณฑ์การผ่านการประเมิน

- "ผ่านการประเมิน" หมายถึง ผู้เข้ารับการประเมินมีผลคะแนนความรู้ทฤษฎี และความรู้ทฤษฎีเชิงปฏิบัติ โดยการสอบสัมภาษณ์ ไม่น้อยกว่าร้อยละ 70
- “ไม่ผ่านการประเมิน” หมายถึง ผู้เข้ารับการประเมินมีผลคะแนนความรู้ทฤษฎี และความรู้ทฤษฎีเชิงปฏิบัติ โดยการสอบสัมภาษณ์ น้อยกว่าร้อยละ 70

กำหนดการจัดกิจกรรมอบรม

วันที่ 1

09:00-12:00 น. บรรยายความรู้หัวข้อ “การแก้ไขข้อผิดพลาด”

- การหาจุดผิดพลาด ตามบันทึกข้อผิดพลาด
- การแก้ไขข้อผิดพลาดของโปรแกรม
- การทดสอบการแก้ไขข้อผิดพลาดของโปรแกรม
- ฝึกปฏิบัติ (Workshop)

บรรยายความรู้หัวข้อ “การพัฒนาโปรแกรมแบบซับซ้อน (Enterprise/Big Scale)”

- การอ่าน Functional/Program Specification/UML (Enterprise/ Big Scale)
- ฝึกปฏิบัติ (Workshop)

13:00-16:00 น. บรรยายความรู้หัวข้อ “การพัฒนาโปรแกรมแบบซับซ้อน (Enterprise/Big Scale)” (ต่อ)

- การเขียนโปรแกรมตาม Functional/Program Specification/UML (Enterprise/Big Scale)
- ฝึกปฏิบัติ (Workshop)

บรรยายความรู้หัวข้อ “การออกแบบ Continuous Delivery และ Continuous Integration”

- การวิเคราะห์ Continuous Delivery และ Continuous Integration
- ฝึกปฏิบัติ (Workshop)

วันที่ 2

09:00-12:00 น. บรรยายความรู้หัวข้อ “การออกแบบ Continuous Delivery และ Continuous Integration” (ต่อ)

- การออกแบบ Continuous Delivery และ Continuous Integration
- ฝึกปฏิบัติ (Workshop)

บรรยายความรู้หัวข้อ “การพัฒนาโปรแกรมตามมาตรฐานและตรวจสอบมาตรฐานการเขียนโปรแกรม (Code Review)”

- การวางแผนในการพัฒนา Code Review
- การจัดทำรายงาน Unit test

13:00-16:00 น. บรรยายความรู้หัวข้อ “การพัฒนาโปรแกรมตามมาตรฐานและตรวจสอบมาตรฐานการเขียนโปรแกรม (Code Review)” (ต่อ)

- การพัฒนา Code Review
- ฝึกปฏิบัติ (Workshop)

แนะนำ และอธิบายกระบวนการประเมิน

- แนวทางการสอบประเมิน
- สอบความรู้ทฤษฎี 60 นาที

การสอบประเมินความรู้ทฤษฎีเชิงปฏิบัติ โดยการสอบสัมภาษณ์

การสอบสัมภาษณ์ทางเจ้าหน้าที่สอบจะแจ้งกำหนดการให้ทราบในภายหลัง เพื่อนัดหมายวันและเวลาสอบ

อุปกรณ์ที่ต้องใช้สำหรับอบรมและประเมิน อาชีพนักพัฒนาระบบ ระดับ 5 (Programmer)

ลำดับ	อุปกรณ์	จำนวน	หมายเหตุ
1	คอมพิวเตอร์	1 เครื่อง	เครื่องคอมพิวเตอร์ 1 เครื่องต่อผู้อบรม และเข้ารับการประเมิน 1 คน • มีหน่วยความจำไม่น้อยกว่า 4 GB • ติดตั้งระบบปฏิบัติการไม่ต่ำกว่า Windows 7 64-bit หรือ Windows เวอร์ชัน ใหม่กว่า
2	ซอฟต์แวร์ที่ใช้ในการอบรม เพื่อฝึกปฏิบัติ (Workshop) และภาษาที่ใช้เขียนโค้ด (Coding)	อย่างใดอย่างหนึ่ง	1. โปรแกรม Text Editor เช่น EditPlus, Notepad ++, Sublime Text 3 และ NetBeans เป็นต้น 2. โปรแกรม Web Browser เช่น Google Chromeหรือ Mozilla Firefox เป็นต้น 3. ภาษาที่ใช้เขียนโค้ด (Coding) เช่น PHP, Visual Studio Code, Java, C# และ Python เป็นต้น

** ผู้อบรมและเข้ารับการประเมิน สามารถเตรียมซอฟต์แวร์อย่างใด อย่างหนึ่ง
สำหรับการฝึกปฏิบัติในระหว่างการอบรม และการสอบประเมิน **

แผนโครงสร้างหลักสูตร

หน่วยสมรรถนะ Unit of Competence: การแก้ไขข้อผิดพลาด

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การหาจุดผิดพลาด ตามบันทึกข้อผิดพลาด	1. สามารถระบุตำแหน่งจุดที่มีข้อผิดพลาดได้ 2. สามารถอธิบายสาเหตุของข้อผิดพลาดที่เกิดขึ้นได้ 3. สามารถอธิบายผลกระทบจากข้อผิดพลาดที่เกิดขึ้นกับโปรแกรมย่อยได้ 4. สามารถเสนอหรือแนะนำวิธีการแก้ไขข้อผิดพลาดในแต่ละจุดได้ 5. สามารถตรวจสอบจุดอ่อนหรือช่องโหว่ของโปรแกรมได้	การค้นหาจุดผิดพลาด และการกำหนดแผนการทดสอบโปรแกรมย่อยหลังจากที่ได้รับการแก้ไขจุดผิดพลาดเป็นสิ่งสำคัญ ซึ่งแผนการทดสอบจะต้องเป็นไปตามลำดับการทำงานของโปรแกรม และควรคำนึงถึงความมั่นคงปลอดภัยของโปรแกรมอย่างเหมาะสม • ความสำคัญและประโยชน์ของบันทึกข้อผิดพลาด (Error Log) • องค์ประกอบและการแปลความหมายของข้อความแสดงข้อผิดพลาด • การวิเคราะห์ไฟล์และบรรทัดที่ระบุในข้อผิดพลาด • การประเมินผลกระทบและระบุจุดอ่อนจากข้อผิดพลาดที่เกิดขึ้น • ขั้นตอนการแก้ไขข้อผิดพลาด • การทดสอบการแก้ไขข้อผิดพลาดของโปรแกรม
2. การแก้ไขข้อผิดพลาดของโปรแกรม	1. สามารถแก้ไขจุดผิดพลาดตามที่มีการระบุตำแหน่งไว้แล้วได้ 2. สามารถอธิบายวิธีการแก้ไขข้อผิดพลาดที่เกิดขึ้นได้ 3. สามารถแก้ไขจุดอ่อนหรือช่องโหว่ของโปรแกรมได้	
3. การทดสอบการแก้ไขข้อผิดพลาดของโปรแกรม	1. สามารถกำหนดแผนการทดสอบโปรแกรมย่อยหลังรับการแก้ไขจุดผิดพลาดแล้วได้ 2. สามารถทดสอบโปรแกรมย่อยให้เป็นไปตามลำดับการทำงานของโปรแกรมได้ 3. สามารถเลือกวิธีป้องกันการทดสอบโปรแกรมย่อยโดยคำนึงถึงวิธีการป้องกันความมั่นคงปลอดภัยได้	

หน่วยสมรรถนะ Unit of Competence : การพัฒนาโปรแกรมแบบซับซ้อน (Enterprise/Big Scale)

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การอ่าน Functional / Program Specification / UML (Enterprise/Big Scale)	1. สามารถอ่านและเข้าใจลำดับของ Pseudo code / Flowchart / UML ในระดับ Enterprise / Big Scale ได้ 2. สามารถบอกผลลัพธ์ของ Functional / Program Specification ในระดับ Enterprise / Big Scale ได้	การเข้าใจ และการอ่านเอกสารเชิงเทคนิค เป็นทักษะสำคัญที่ต้องการสำหรับนักพัฒนาซอฟต์แวร์ในการพัฒนาโครงการขนาดใหญ่ หรือโครงการระดับองค์กร การทำความเข้าใจอย่างถูกต้อง และการที่นักพัฒนาเข้าใจเอกสารเหล่านี้ จะช่วยให้สามารถปรับปรุง และสร้างซอฟต์แวร์ที่มีคุณภาพสูง และได้ตรงตามความต้องการ สร้างความเชื่อมั่นให้กับลูกค้า หรือองค์กร และสร้างผลิตภัณฑ์ที่เหมาะสมแก่การใช้งานได้อย่างมีประสิทธิภาพ มีคุณภาพสูงสุด นักพัฒนาซอฟต์แวร์จึงควรมีทักษะในสามเรื่อง ดังนี้ • การอ่าน Functional Specifications (FS) • การอ่าน Program Specifications (PS) • การอ่าน Unified Modeling Language (UML)
2. การเขียนโปรแกรมตาม Functional / Program Specification / UML (Enterprise / Big Scale)	1. สามารถเขียนโปรแกรมตาม Functional / Program Specification / UML ในระดับ Enterprise / Big Scale ได้ 2. สามารถเขียนโปรแกรม ในระดับ Enterprise / Big Scale โดยคำนึงถึงความมั่นคงปลอดภัยได้ 3. สามารถตรวจสอบผลลัพธ์ของโปรแกรมตามที่กำหนดใน Functional / Program Specification / UML ในระดับ Enterprise / Big Scale ได้ 4. สามารถจัดทำเอกสาร API Document Specification ได้	

หน่วยสมรรถนะ Unit of Competence : การออกแบบ Continuous Delivery และ Continuous Integration

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การวิเคราะห์ Continuous Delivery และ Continuous Integration	1. สามารถรวบรวมฟังก์ชันการทำงานของระบบ 2. สามารถวิเคราะห์ความสัมพันธ์ของฟังก์ชันการทำงานของระบบ	การออกแบบ CI / CD ที่มีประสิทธิภาพช่วยให้ทีมพัฒนาซอฟต์แวร์สามารถตอบสนองต่อความต้องการของตลาดได้เร็วขึ้นและลดความเสี่ยงในการปล่อยซอฟต์แวร์ที่มีปัญหา นอกจากนี้ยังช่วยให้การปรับปรุงซอฟต์แวร์เป็นไปอย่างราบรื่นและมีความต่อเนื่อง สร้างความไว้วางใจในสินค้าซอฟต์แวร์และกระบวนการพัฒนา การใช้งาน CI / CD จึงเป็นสิ่งสำคัญที่นักพัฒนาซอฟต์แวร์ควรให้ความสำคัญและพัฒนาความเข้าใจในการออกแบบ และการใช้งานให้ดีที่สุด
2. การออกแบบ Continuous Delivery และ Continuous Integration	1. สามารถออกแบบ Continuous Delivery และ Continuous Integration ของระบบได้ 2. สามารถเลือกเครื่องมือในดำเนินการ Continuous Delivery และ Continuous Integration ได้	

หน่วยสมรรถนะ Unit of Competence : การพัฒนาโปรแกรมตามมาตรฐานและตรวจสอบมาตรฐานการเขียนโปรแกรม (Code Review)

ชื่อหัวข้อวิชา (Content Title)	เกณฑ์ในการปฏิบัติงาน (Performance Criteria)	รายละเอียดเนื้อหาวิชา (Content)
1. การวางแผนในการพัฒนา Code Review	1. สามารถกำหนดปัจจัยต่าง ๆ ที่เกี่ยวข้องกับการพัฒนา Code Review ได้ 2. สามารถวางแผนการพัฒนา Code Review ได้	การวางแผน และการดำเนินการที่มีประสิทธิภาพในการทบทวนโค้ด ไม่เพียงช่วยเพิ่มคุณภาพผลิตภัณฑ์ซอฟต์แวร์เท่านั้น แต่ยังช่วยให้ทีมงานเรียนรู้และเติบโตร่วมกัน ทำให้การพัฒนาซอฟต์แวร์เป็นกระบวนการที่ต่อเนื่องและได้มาตรฐาน ขั้นตอนการวางแผนในการพัฒนา Code Review สำหรับนักพัฒนาซอฟต์แวร์มีดังนี้ • ทำหนดวัตถุประสงค์ของการทบทวนโค้ด • ทำหนดข้อทำหนดและมาตรฐานในการเขียนโค้ด • เลือกเครื่องมือสำหรับการทบทวนโค้ด • ทำหนดโครงสร้างการทบทวนโค้ด • การฝึกอบรมและการเรียนรู้ต่อเนื่อง
2. การพัฒนา Code Review	1. สามารถพัฒนา Code Review ได้ 2. สามารถตรวจสอบคุณภาพ Code Review จากการพัฒนา Code Review ได้	