

บทที่ 2

สรุปสาระสำคัญจากเอกสารที่เกี่ยวข้อง

ในการพัฒนาระบบฐานข้อมูลข่าวสารภายในองค์กร ได้ทำการศึกษาทฤษฎีและหลักการต่างๆ ที่สามารถนำมาประยุกต์เข้ากับงานได้ โดยแบ่งออกเป็นหัวข้อต่างๆ ได้ดังต่อไปนี้

2.1 Multimedia Database

2.2 Intranet/PHP Library

2.3 การพัฒนาซอฟต์แวร์แบบ AGILE

2.4 Extreme Programming (XP)

2.1 Multimedia Database

คือฐานข้อมูลที่บันทึกข้อมูลทั้งตัวเลข อักษรระ เสียง ภาพเคลื่อนไหว ที่สามารถแสดงได้พร้อมกันในเวลาเดียวกันและสามารถโต้ตอบกับผู้ใช้ได้ด้วย

Media data	Special nature	Typical digital file formats
Text data		ASCII, RTF, HTML, SGML, XML, TXT
Audio data	Time dependence with digitized data size can affect quality	WAV, AU, MPEG-MP3
Image data	Results from pictures, drawings and photographs	JPEG, GIF, TIFF, BMP, PNG
Video data	Time-dependent sequence of video frames	AVI, FLV, MPEG

2.2 Intranet/PHP Library

อินทราเน็ต (Intranet) คือเครือข่ายคอมพิวเตอร์ที่เชื่อมโยงการสื่อสารด้วยระบบ โพรโทคอลทีซีพี/ไอพี(TCP/IP) ซึ่งเป็นระบบโพรโทคอลในการสื่อสารของเครือข่ายอินเทอร์เน็ต (Internet) ดังนั้น โปรแกรมเพื่อการสื่อสารบนเครือข่ายอินทราเน็ตจึงเป็นซอฟต์แวร์ชนิดเดียวกับที่ใช้ในการสื่อสารบนเครือข่ายอินเทอร์เน็ต ส่วนความแตกต่างที่ชัดเจนระหว่างเครือข่าย อินทราเน็ตกับเครือข่ายอินเทอร์เน็ต คือ อินเทอร์เน็ตเป็นเครือข่ายคอมพิวเตอร์ที่ครอบคลุมทั้งโลก อินเทอร์เน็ตไม่มีใครเป็นเจ้าของอย่างแท้จริง และไม่มีใครสามารถควบคุมเครือข่ายอินเทอร์เน็ตได้ แต่สำหรับเครือข่ายอินทราเน็ตมีเจ้าของแน่นอน และถูกควบคุมโดยองค์กรหรือบุคคลผู้เป็นเจ้าของ

ภาษาพีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์ อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจที่มีความตอบโต้ได้อย่างรวดเร็ว

ชื่อของพีเอชพี

ภาษาพีเอชพี ในชื่อภาษาอังกฤษว่า PHP ซึ่งใช้เป็นคำย่อแบบกล่าวซ้ำ จากคำว่า PHP Hypertext Preprocessor หรือชื่อเดิม Personal Home Page

ตัวอย่างภาษาพีเอชพี

ภาษาพีเอชพี จะเป็นส่วนประกอบภายในเว็บเพจ โดยคำสั่งจะปรากฏระหว่าง `<?php ... ?>` การแสดงผลของพีเอชพี จะปรากฏในลักษณะHTML ซึ่งจะไม่ได้แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะCGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ใน

ยูนิกซ์หรือลินุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แพลช (โดยใช้ libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและเข้าสู่เอกสาร XML เรารองรับมาตรฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML

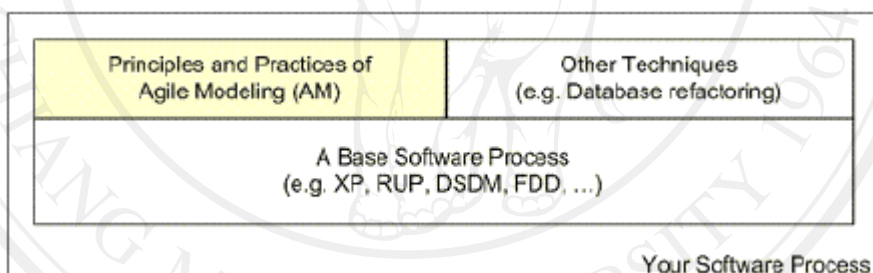
2.3 การพัฒนาซอฟต์แวร์แบบ AGILE

หลักการ Agile

- เน้นความพอใจให้ลูกค้า ลูกค้าชอบ มีการส่งมอบ Software อย่างต่อเนื่อง
- ยอมรับ requirement ที่เปลี่ยนแปลง
- มีการส่งมอบงานบ่อยๆ (ทุกๆ 2 สัปดาห์)
- ลูกค้าและผู้พัฒนาต้องทำงานร่วมกัน (โปรแกรมเมอร์ไปทำงานที่ site ลูกค้า) ต้องเจอกันทุกวันจนโปรเจกต์เสร็จ
- การทำงานต้องปล่อยให้ทีมงานมีอำนาจการตัดสินใจเองได้ ปล่อยให้เค้าทำงาน ไว้ใจกัน และทีมงานก็ต้องมีความรับผิดชอบระดับหนึ่ง
- การติดต่อกัน ต้องคุยซึ่งๆหน้า ห้ามอีเมลล์หรือโทร
- วัดความก้าวหน้าของงานที่ Software
- กระบวนการทำงาน ให้ทำไปเรื่อยๆ อย่าหิวหิว ค่อยๆทำ ส่งงานทีละนิด ช่วยทำให้คุณภาพชีวิตของผู้พัฒนาดีขึ้น
- ผู้พัฒนา สปอนเซอร์ ลูกค้า ต้องมีการทำไปเรื่อยๆ คงที่ ไม่เร็วเกินหรือช้าเกิน
- ทีมงานต้องให้ความสนใจกับเทคนิคต่างๆ มีการแชร์กัน
- เน้นความง่าย ออกแบบง่ายๆ ฟังก์ชัน ไม่ซับซ้อน ทำให้ดูแลแก้ไขง่ายเมื่อพบความเปลี่ยนแปลง
- ทีมมีความรับผิดชอบในกระบวนการของตัวเอง
- มีการนัดพบแลกเปลี่ยนกันสม่ำเสมอ

โมเดลของ Agile (AM : Agile Modeling)

- เลือกบางหลักการมาทำ
- เป็นวิธีหนึ่งที่จะเอาหลักการของ Agile มาจัดการกับเอกสารและระบบเดิมที่มีอยู่ได้
- ใน Agile ประกอบด้วย
 1. value ผลลัพธ์
 2. principle หลักการ
 3. practices วิธีปฏิบัติ
- ทั้งสามอย่างนี้เป็นส่วนหนึ่งในโมเดล Agile ที่สามารถนำมาพัฒนา SW ให้มีประสิทธิภาพ และเกิด overhead น้อย
- ให้มอง Agile เป็นส่วนขยายของกระบวนการพัฒนา SW แบบเดิมได้
 - ให้ Agile เข้าไปกำกับ ดูว่าของเดิมที่มีอยู่อันไหนสำคัญก็ทำ ไม่สำคัญก็ละ
 - นำ Agile มาจัดลำดับความสำคัญ ดูว่ากิจกรรมไหน ควรทำ ไม่ควรทำ



AM Value (ผลลัพธ์)

- เน้นติดต่อสื่อสาร
- เน้นความง่าย ไม่ซับซ้อน
- เน้น feedback จากลูกค้า
- เน้นความกล้าตัดสินใจ
- เน้นความเคารพกันและกัน

AM Core Principle (หลักการ)

- ง่าย ไม่มากเกินไป
- รับ requirement พร้อมเปลี่ยนแปลงได้ตลอดเวลา
- เน้นปัจจุบันเป็นหลัก

- ทำ model ตามความจำเป็นเท่านั้น
- พยายามใช้ multiple model มองหลายๆมุมมอง
- มีการตอบกลับเร็ว
- SW ถือเป็นจุดมุ่งหมายหลัก
- ให้เบกส์มีการระเบาะๆ
- AM Supplement Principle
 - เน้น content มากกว่า representation (ที่ใช้ UML เขียน) ไม่เน้นเครื่องมือ เน้นที่เนื้อหาข้างใน
 - ติดต่อกันอย่างเปิดเผย และตรงไปตรงมา

AM Core Practice (แนวทางปฏิบัติ /ลงมือทำ)

1. จัดประชุม รวบรวม Active stakeholder เท่านั้น บางมีอาจมี None stakeholder เข้ามาฟังได้ แต่ห้ามออกความคิดเห็น ห้ามถาม ห้ามติดต่อ ห้ามแสดงโอเคเดียว
2. นำ Artifact มาใช้ให้ถูกต้อง
 - Artifact คือชิ้นส่วนของงานที่เราทำระหว่างการพัฒนาาระบบเช่น อีแมลล์, source code,จดหมาย,ใบเชิญประชุม ถ้า Artifact ใดถูกเลือกมาใช้ในการทำงาน เรียกว่า "work products" และถ้า work products นี้ ถูกส่งมอบให้ลูกค้าเรียกว่า "Deliverable"
3. พยายามเป็นเจ้าของงาน สามารถทำงานแทนกันและกันได้
4. พยายามใช้โมเดลแบบคู่ขนาน จะได้มองต่างมุม เพื่อเก็บรายละเอียดของระบบให้ครบ
5. ทำให้เนื้อหาง่าย
6. พยายามวาดรูปไม่ให้ซับซ้อน
7. พยายามให้โมเดลเข้าถึงได้ทุกคน
8. สามารถเปลี่ยน Artifact หนึ่งไปอีกอันได้
9. ใช้โมเดลแบบเล็กก่อนค่อยขยาย
10. พยายามให้ผู้อื่นมีส่วนร่วมในการทำโมเดล
11. พิสูจน์ด้วยการลองเขียน code ดู (จาก code เริ่มต้นตั้งแต่แรก)
12. ใช้เครื่องมือง่ายๆในการทำงาน เช่น กระดาษ,กระดานดำ

2.4 Extreme Programming (XP)

Extreme Programming นำเสนอแนวคิดในการแบ่งการพัฒนาออกเป็นกลุ่มของงานย่อย โดยแต่ละงานย่อย ควรจะทำเสร็จ ภายในระยะเวลาสั้น ๆ และควรมีการพบปะระหว่างทีมพัฒนาในแต่ละวัน เพื่อทดสอบงาน ที่ทำเสร็จลงไปแล้ว โดยถ้ามีความเปลี่ยนแปลงเกิดขึ้นกับข้อกำหนดของซอฟต์แวร์ (Software Specification) จะต้องมีการจัดการอย่างเป็น กระบวนการเท่านั้น สำหรับตัวโครงการเองจะมีการพัฒนาแบบก้าวหน้า โดยโค้ดที่เขียนขึ้นต้องเพียงพอที่จะทำงาน ในระดับของฟังก์ชัน รวมทั้งรายละเอียดทุกอย่าง (การออกแบบ คุณสมบัติและตัวโค้ด) จะต้องได้รับการดูแลควบคู่กันไปอย่างต่อเนื่อง เพื่อให้ได้ผลลัพธ์ตรงกับความต้องการ ของ user มากที่สุด

XP เป็นการกำหนดกฎเกณฑ์ในการปฏิบัติงานวันต่อวันสำหรับผู้บริหารและนักพัฒนาระบบ กฎเกณฑ์ในการปฏิบัติที่ว่านี้หมายถึงการถือหุ่่น XP values และยังทำให้เป็นรูปธรรมยิ่งขึ้น ผู้เสนอยังเชื่ออีกว่า การใช้วิธีการปฏิบัติเหล่านี้ (ซึ่งเป็นวิธีการปฏิบัติแบบดั้งเดิมในของวิศวกรรมซอฟต์แวร์ซึ่งเรียกอีกอย่างว่า Extreme levels) นั้นจะนำไปสู่กระบวนการพัฒนาที่ตอบสนองต่อความต้องการของ user (agile) ได้ดียิ่งกว่าวิธีการแบบดั้งเดิม ทั้งยังทำให้การสร้างซอฟต์แวร์มีคุณภาพที่ดีขึ้นอีกด้วย

วัตถุประสงค์หลักของ XP คือการลดต้นทุนการเปลี่ยนแปลง ซึ่งสำหรับวิธีพัฒนาระบบแบบดั้งเดิมความต้องการของระบบจะถูกกำหนดเมื่อตอนเริ่มต้น โครงการ และบ่อยครั้งที่จะยึดติดกับความต้องการนั้น ซึ่งก็จะทำให้ต้นทุนการเปลี่ยนแปลงที่จะทำให้โปรแกรมเป็นไปตามความต้องการในขั้นตอนต่อมาสูงขึ้น (เป็นลักษณะทั่วไปของโครงการวิศวกรรมซอฟต์แวร์)

XP ตั้งเป้าที่จะลดต้นทุนการเปลี่ยนแปลงโดยการเสนอแนะ values, principles และ practices ซึ่งในการประยุกต์ใช้งาน XP โครงการพัฒนาระบบควรจะมีคามยืดหยุ่นมากยิ่งขึ้นต่อการเปลี่ยนแปลงความต้องการของ user