บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาค้นคว้าแบบอิสระ เรื่องการพัฒนาระบบการจองการใช้บริการสปา โดยใช้เว็บเซอร์วิสตามหัวข้องานวิจัย ได้ทำการศึกษาทฤษฎีและหลักการต่างๆที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางนำมาประยุกต์ใช้ สามารถแบ่งออกเป็นหัวข้อต่างๆ ได้ดังต่อไปนี้

2.1 เว็บเซอร์วิส (Web Services)

(W3C Working Group Note, 2004) เว็บเซอร์วิส คือ แอพพลิเคชั่นหรือโปรแกรมที่ทำงานอย่างใดอย่างหนึ่งในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอพพลิเคชั่นอื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นสื่อในการแลกเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้งานส่วนใด ๆ ก็ได้ในแพลตฟอร์มที่แตกต่าง ซึ่งได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่างแอพพลิเคชั่นกับแอพพลิเคชั่นในปัจจุบัน

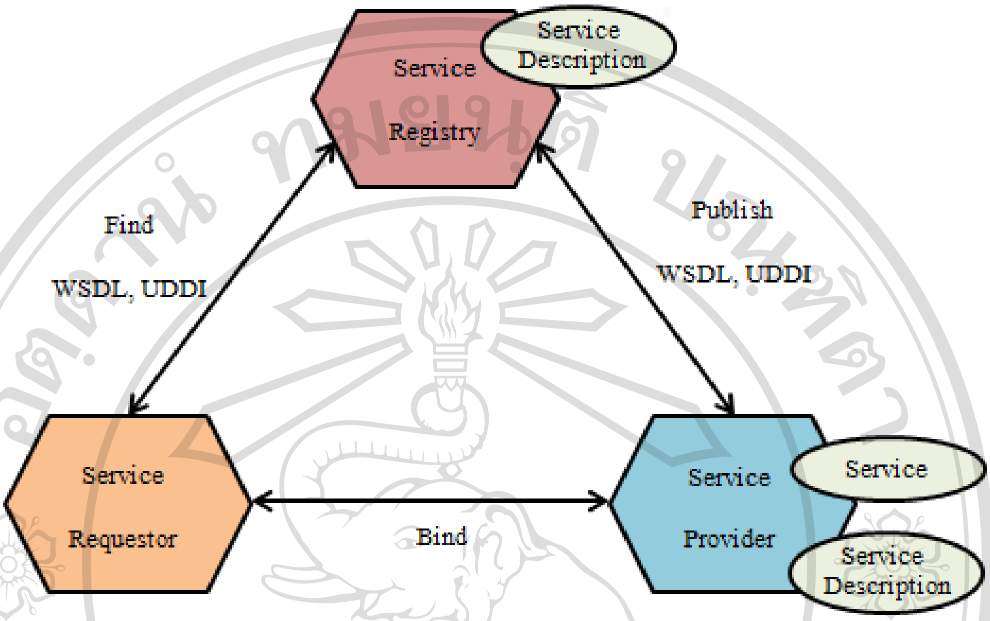
Web Service ช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอพพลิเคชันที่ต่างกันเป็นไปโดยง่าย โดยแอพพลิเคชันนั้นๆ สามารถเขียนด้วย Java และทำงานอยู่บน Sun Solaris Application Server หรืออาจจะเขียนด้วย C++ และทำงานอยู่บน Windows NT หรืออาจจะเขียนด้วย Perl และรันอยู่บนเครื่อง Linux ซึ่งมาตรฐานของ Web Service ทำให้อินเทอร์เฟซของแอพพลิเคชันเหล่านี้ ถูกอธิบายโดย WSDL และทำให้อยู่ในมาตรฐานของ UDDI หลังจากนั้น จึงสามารถติดต่อสื่อสารถึงกันโดย XML ผ่าน SOAP อินเตอร์เฟซ

Web Service สามารถถูกเรียกใช้ภายในองค์กรเองหรือจากภายนอกองค์กร โดยผ่านไฟร์วอล์ ดังนั้นจึงมีองค์กรใหญ่ๆ มากมาย กำลังพัฒนาระบบที่มีอยู่ของตน ให้เข้ากับ Web Service ซึ่งนับเป็นการลงทุนที่คุ้มค่า เนื่องจาก Web Service สามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กรได้อีกทางหนึ่ง

นอกจากนั้น Web Service ยังสามารถใช้ร่วมกับ Web Application โดยส่งผ่านข้อมูลทางอินเตอร์เน็ตได้อีกด้วยซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้าหรือหุ้นส่วน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.1.1 Web Service Model



รูปที่ 2.1 โครงสร้างสถาปัตยกรรม Web Service

1) Requestor

คือใครก็ตามที่ต้องการเรียกใช้บริการจาก Provider ซึ่งสามารถค้นหาบริการที่ต้องการได้จาก UDDI registry หรือ Service Registry หรือติดต่อจาก Provider โดยตรง

2) Registry

ทำหน้าที่เป็นตัวกลางให้ Provider มาลงทะเบียนไว้ โดยใช้ WSDL ไฟล์ บอกรายละเอียดของบริษัทและบริการที่มีให้ ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้

3) Provider

เป็นผู้ให้บริการ มีหน้าที่ในการเปิดบริการเพื่อรองรับการขอใช้บริการจาก Requestor ที่เรียกเข้ามาขอใช้

2.1.2 เทคโนโลยีที่ใช้ในการพัฒนา Web Services

1) XML (Extensible Markup Language)

(W3C Working Group Note, 2004) เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตอย่างรวดเร็วผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ใน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

การสร้าง เว็บเพจที่สามารถแสดงผลได้โดยโปรแกรมเบราวเซอร์ แต่ XML จะใส่ tags ได้อย่างอิสระแล้วทำการส่ง XML ชุดนี้ไป ประมวลผลยังแอพพลิเคชั่นใด ๆ ที่สามารถใช้ข้อมูลใน XML นี้ XML เป็นภาษาที่มีลักษะเป็น tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกันทำให้ข้อมูลระหว่าง tag สามารถนำไปประมวลผลต่อ

2) SOAP (Simple Object Access Protocol)

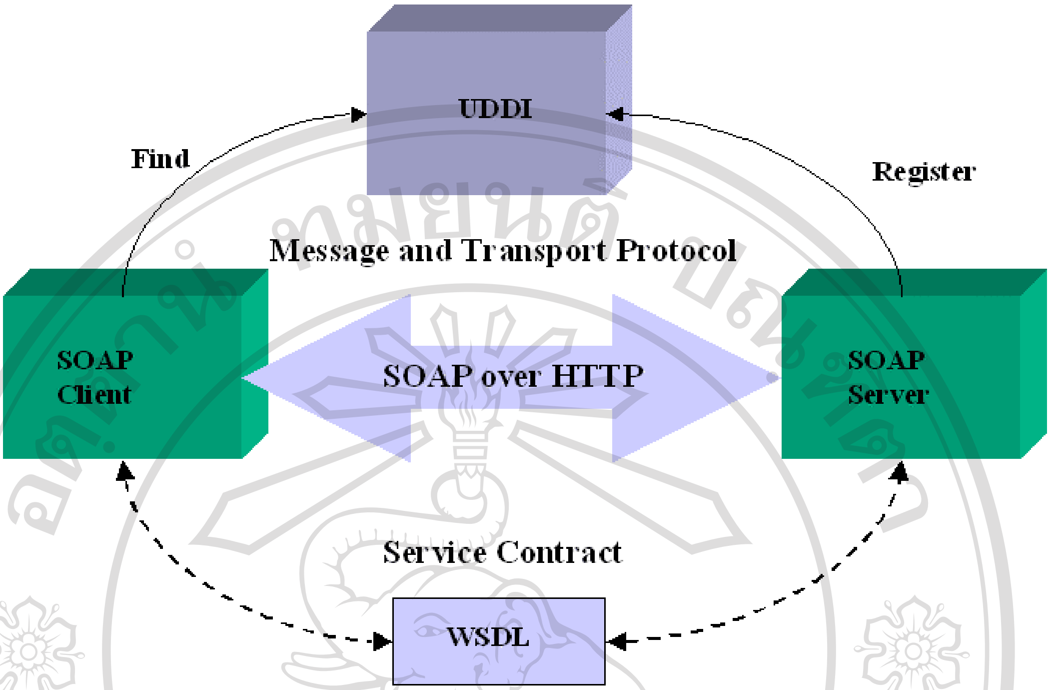
(VishyKasar, 2004) เป็นภาษา XML เพื่อทำหน้าที่ในการส่งข้อมูลผ่านโปรโตคอล (Message Protocol) สำหรับการแลกเปลี่ยนข้อมูลระหว่างผู้ให้บริการและผู้ใช้บริการ โดย SOAP จะใช้ HTTP Protocol ช่วยในการส่งข้อมูลอีกที จึงเปรียบ SOAP เป็นจดหมาย และ HTTP Protocol เป็นคนส่งจดหมาย การใช้สามารถทำได้สองวิธี คือแบบ Remote Procedure Call (RPC) และแบบ Document

RPC-Style SOAP คือการมอง เว็บเซอร์วิสให้เป็นประหนึ่งวัตถุ (Object) โดยภายใน Request จะมี Method Name ที่ใช้เรียก Parameter และ Method จะกระทำการบน Server และส่ง XML Response กลับไปให้ Client

Document-Style SOAP ใช้ XML Schemas เป็นตัวกำหนดรูปแบบของ Request และ Response ซึ่งจะมีความยืดหยุ่นกว่าแบบ RPC เนื่องจากรูปแบบจะถูกปรับเปลี่ยนตาม XML Schema

จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของ และเป็นโปรโตคอลที่ทำงานกับโปรโตคอลอื่นหลายชนิดการพัฒนาก็อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved



รูปที่ 2.2 การทำงานของ SOAP

ที่มา : VishyKasar (2004)

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส เราต้องการให้แอพพลิเคชั่นมีการทำงานกับแอพพลิเคชั่นที่ทำงานอยู่ในเครื่องอื่นโดยผ่านทางเครือข่ายซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่าง Object ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM,EJB หรือ CORBAนั้น ไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอล HTTP

เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้น ต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย ยกเว้น CORBA ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy Server ด้วย เนื่องจากโดยปกติเซอร์เวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอลHTTP ออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

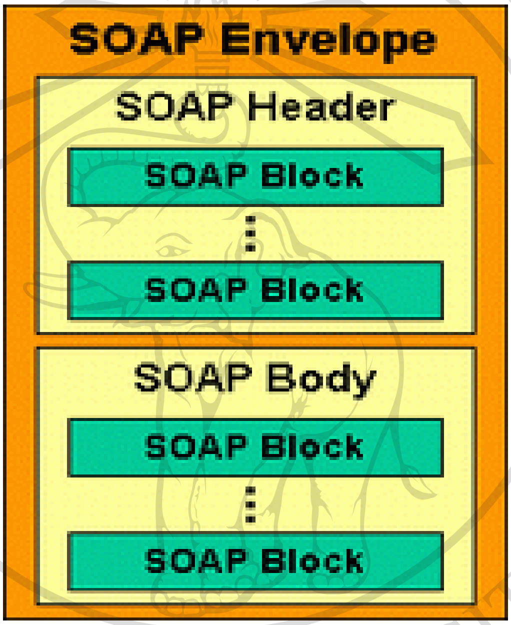
ดังนั้น ทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำบริการเว็บเซอร์วิสก็คือให้ทำงานอยู่บน โปรโตคอล HTTP ไปเลย ซึ่ง SOAP นอกจากจะทำงานบนโปรโตคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

โครงสร้างของ SOAP

(W3C Working Group Note, 2001) เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลัก ดังนี้คือ

1. SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
2. SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้
3. SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส



รูปที่ 2.3 โครงสร้างของเอกสาร SOAP
ที่มา : W3C Working Group Note (2001)

จะเห็นว่า SOAP Envelope ก็เป็นเอกสารรูปแบบ XML ทั่วไปแต่SOAP มีการกำหนดโครงสร้างของเอกสารเป็นส่วนๆ โดยอธิบายว่าส่วนใดมีหน้าที่อะไรมีข้อมูลอะไรอยู่ในส่วนนั้นและใครคือผู้ที่จะต้องสนใจในส่วนนั้น ๆและกล่าวถึงว่าส่วนใดจำเป็นที่ต้องมีในเอกสารและส่วนใดเป็นส่วนที่เพิ่มเติมอาจจะมีหรือไม่มีส่วนนั้นก็ได้ขึ้นอยู่ความจำเป็นในการใช้งานทำให้การพัฒนาโปรแกรมมีทิศทางที่ค้อนข้างชัดเจนแต่สามารถเพิ่มขยายได้ในอนาคตอีก

3) WSDL (Web Services Description Language)

เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของ Web Services และวิธีการติดต่อกับ Web Servicesปัจจุบัน WSDL เป็นภาษา ที่อยู่ในการดูแลของ W3C (World Wide Web Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชันที่ใช้งานอยู่ใน ปัจจุบันคือ WSDL 1.1WSDL

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

คือ มาตรฐานสำหรับการประกาศขั้นตอนที่จำเป็นในการเรียกใช้เซอร์วิส ผ่าน SOAP (Simple Object Access Protocol)

โครงสร้างเอกสาร WSDL

WSDL เป็นภาษาที่อยู่ในความดูแลขององค์กร W3C (World Wide Web Consortium) version ที่มีอยู่ในปัจจุบัน คือ WSDL 1.1 ในการใช้งานจริง หากเราสร้างบริการ Web Services ก็จะมีเครื่องมือช่วยสร้างเอกสาร WSDL สำหรับ Web Services อย่างอัตโนมัติจุดภายในเอกสารที่เราควรรู้เกี่ยวกับการติดต่อและเรียกใช้บริการของ Web Services มีจุดที่ควรรู้ ดังนี้

ตารางที่ 2.1 อธิบายความหมายส่วนประกอบต่างๆ ของเอกสาร WSDL

Element	Definition
<port Type>	เป็นส่วนที่สำคัญที่สุดในWSDL element อธิบาย operations ที่ web service มีให้บริการและ messages ที่เกี่ยวข้อง เทียบได้กับ function library หรือ module หรือ class ในการเขียนโปรแกรม
<operation>	อธิบาย method ที่ให้บริการ Web Services หนึ่งจะมี method จำนวนกี่ method ก็ได้
<message>	อธิบาย data elements ของ operation แต่ละ message อาจมีมากกว่าหนึ่งส่วน เทียบได้กับ parameter ของ function ในการเขียนโปรแกรม
<types>	อธิบายชนิดข้อมูลที่ web serviceใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema syntax ในการระบุชนิดข้อมูล
<binding>	อธิบาย format ของ message และ protocol details ในแต่ละ port
<service>	สำหรับ web server จะมี Web Services จำนวนกี่บริการก็ได้ และ ชื่อ Web Services ก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน

ตามทฤษฎีแล้ว ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการ Web Services ได้มากกว่า 1 บริการโดยแต่ละ Web Services จะมี port สื่อสารเฉพาะตัวซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

4) UDDI (Universal Description, Discovery, and Integration)

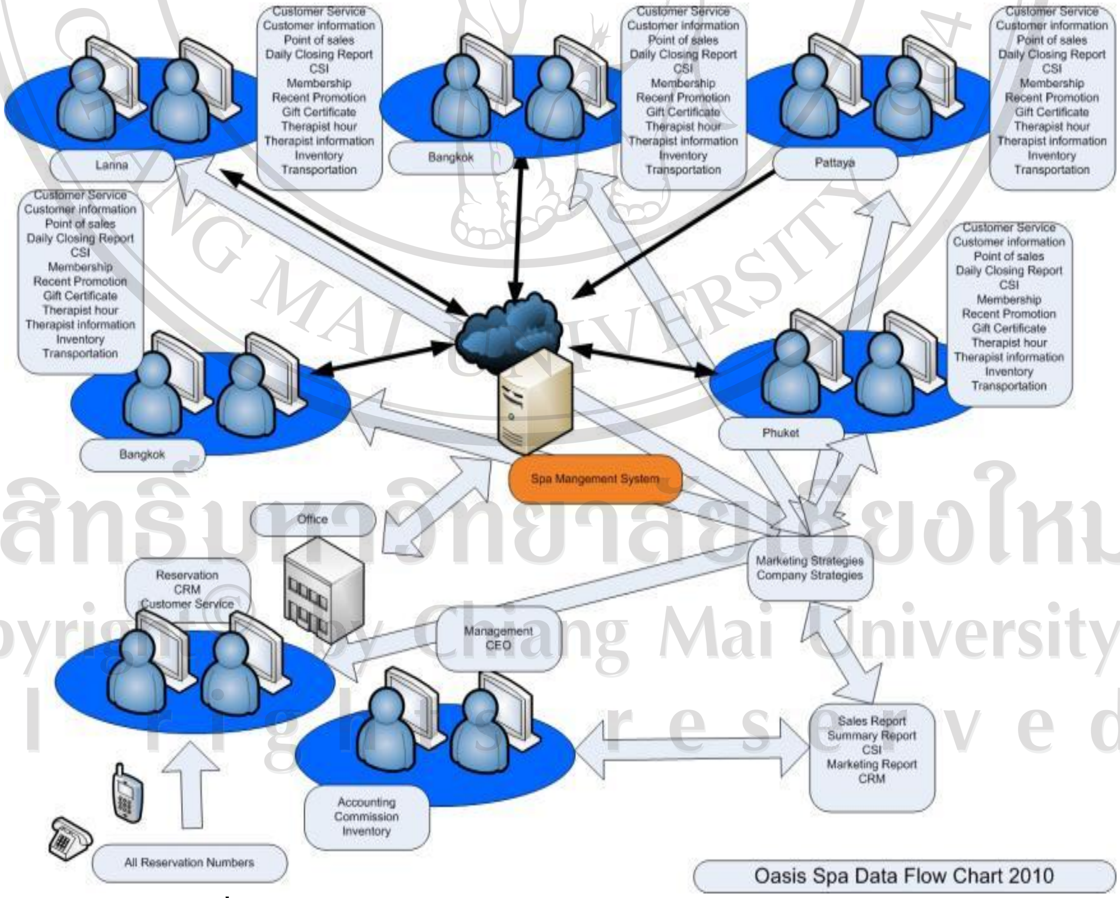
(W3C Working Group Note, 2004) เป็นมาตรฐานที่ให้ชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP ที่สามารถนำมาใช้ในการพัฒนาเป็นตัวแทนของผู้ให้บริการ (Service broker) UDDI ใช้สำหรับค้นหา Serviceและเมื่อได้มาแล้ว UDDI ยังจัดหา

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ข้อตกลงในวิธีการที่จะใช้งานเปรียบได้กับสมุดหน้าเหลืองเป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัทไอบีเอ็ม บริษัทไมโครซอฟต์ และบริษัทอารีบา (Ariba) ปัจจุบันมีบริษัทที่ร่วมกันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัทซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐาน B2B Interoperability

2.2 ระบบการจัดการสปา (Spa Management System)

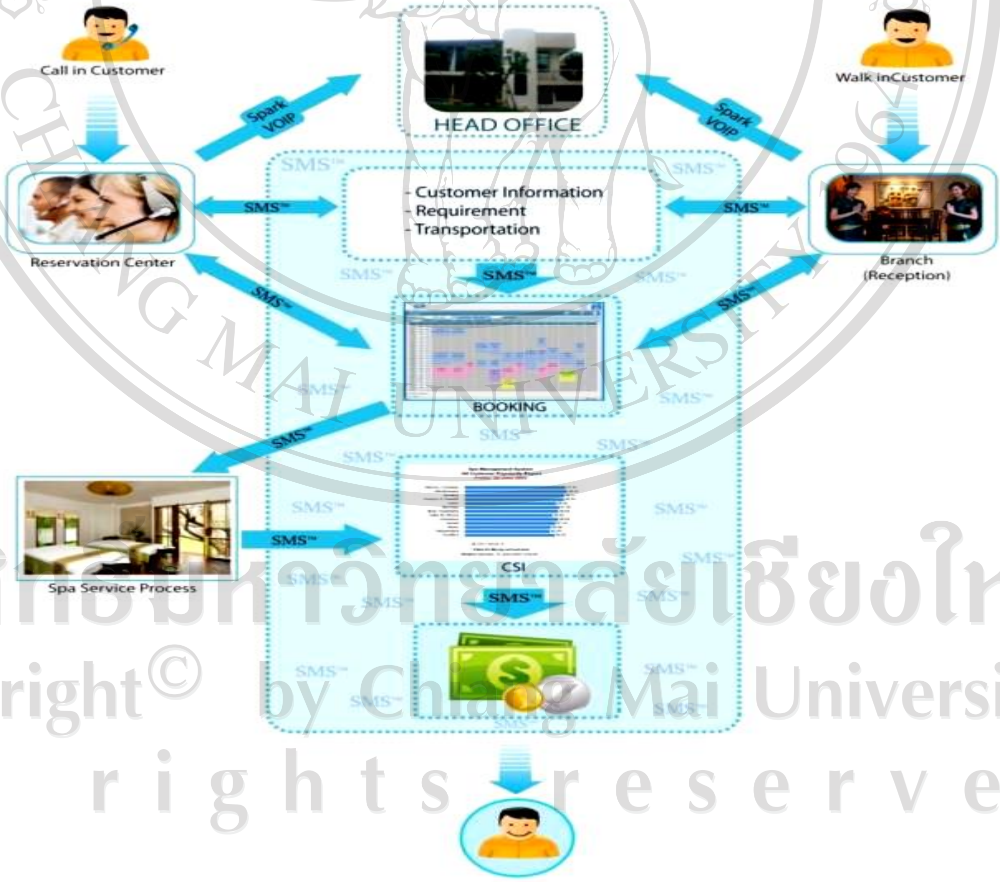
(The Oasis Spa Thailand, 2010) ระบบการจัดการสปา มีโครงสร้างการพัฒนาลักษณะ Software Application สำหรับระบบบริหารจัดการธุรกิจเดย์สปา (Day Spa) โดยมีชื่อว่า Spa Management System (SMS) ที่มีการรับส่ง ประมวลผล และ แสดงผลข้อมูลแบบ Real Time สามารถแสดงผลของข้อมูลที่มีการเปลี่ยนแปลงได้พร้อมกันสำหรับทุกเครื่องคอมพิวเตอร์ที่มีการต่อเชื่อมเข้ากับระบบ โดยใช้โครงข่ายอินเตอร์เน็ตในการเชื่อมต่อ (Web-Based application) มีระบบประมวลผลกลาง (Server) ไม่จำเป็นต้องติดตั้งโปรแกรมใด ๆ เพิ่มเติมเข้าไปในเครื่องคอมพิวเตอร์ผู้ใช้แต่ละคนเข้าถึงระบบจัดการสปาที่ใช้งานผ่านทาง Web-Browser ที่มีอยู่ในเครื่องได้ทันที



รูปที่ 2.4 แผนผัง Spa Management System Data Flow Chart 2010

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ระบบสามารถตอบสนองต่อกิจการสปาได้อย่างครบถ้วน ครอบคลุมตามความต้องการทางธุรกิจ ที่เอื้อต่อการจัดการสปาครอบคลุมตั้งแต่ ระบบจองสปา (Reservation) ระบบคำนวณเงิน (Sales-Receipt) ระบบสรุปยอดเงิน (Daily Closing Report) ระบบสมาชิก (Membership Database) ระบบการแสดงข้อมูลเพื่อการวิเคราะห์ทางการตลาด (Marketing Analysis Report) ระบบรายงานข้อมูลต่าง ๆ (Executive Reports) ระบบคอมมิชชั่น (Commission payment) ระบบข้อมูลบัตรกำนัล (Gift Certificate) ระบบสินค้าคงคลัง (Inventory) ระบบรถรับส่งลูกค้า (Transportation) ระบบฐานข้อมูลลูกค้าและผู้ที่เกี่ยวข้อง (Customers and Agents) และระบบฐานข้อมูลที่ปลอดภัย โดยการเข้ารหัส (Encrypt) แบบ Secured Socket Lay (SSL) เพื่อป้องกันและรักษาความลับของข้อมูลผู้ใช้ Secured Data Environment ในระบบ Spa Management System สูงสุดในใช้งานระบบในประเทศและต่างประเทศผ่านเครือข่ายอินเตอร์เน็ต พร้อมด้วยการพัฒนาฟังค์ชั่นพิเศษที่พัฒนานาเพื่อให้ผู้ใช้ เช่น Quick Search ที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้อย่างรวดเร็ว



รูปที่ 2.5 แผนผังการทำการของ Spa Management System ร่วมกับการทำงานของสปา

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.3 มาตรฐานอุตสาหกรรมซอฟต์แวร์ ISO 29110 VSE

2.3.1 ความหมาย VSE

(สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม, 2551) VSE ย่อมาจาก Very Small Entity ที่มีคำนิยามคือ กลุ่มหรือหน่วยที่มีขนาดเล็กมากใช้ในการให้นิยามกลุ่มผู้ประกอบการขนาดเล็กถึงเล็กมากที่มีบุคคลากรไม่เกิน 25 คน ซึ่งถือว่าเป็นกลุ่มผู้ประกอบการหรือหน่วยที่เป็นกำลังการผลิตที่สำคัญและในอุตสาหกรรมซอฟต์แวร์ไทย เป็นกลุ่มที่กำลังเริ่มต้นและต้องการการพัฒนาที่เป็นระบบเพื่อให้สอดรับกับการเติบโตของอุตสาหกรรมที่กำลังขยายตัว

2.3.2 ความหมาย ISO 29110

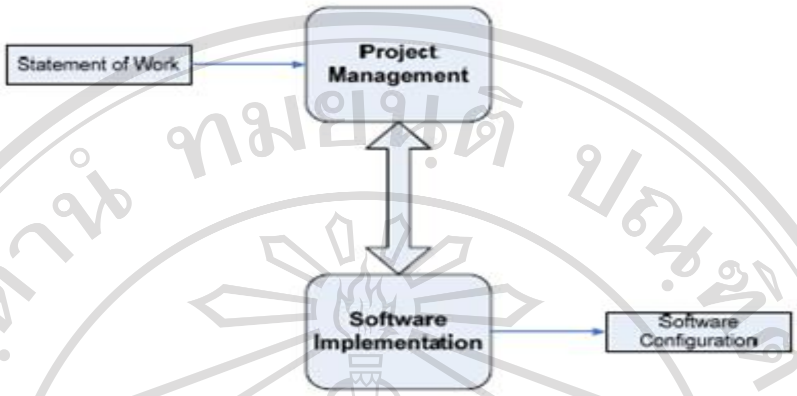
มาตรฐานวิศวกรรมซอฟต์แวร์สำหรับองค์กรขนาดกลางและเล็กที่เรียกว่า ISO29110SoftwareEngineeringLifecycle Profiles for Very Small Enterprises (VSE) ISO 29110 เป็นแนวคิดยุคใหม่ของ ISO ที่จะเน้นการเติบโตของอุตสาหกรรมขนาดกลางและเล็ก รวมทั้งผู้ประกอบใหม่ที่เข้ามาให้มีโอกาสในการแข่งขันตามแนวทางการพัฒนา ฯ Outsourcing ซึ่งในอดีตที่ผ่านมา มาตรฐานวิศวกรรมซอฟต์แวร์ได้ถูกทำให้เป็นเรื่องที่เข้าใจยากและมีความสลับซับซ้อนยุ่งยากในการปฏิบัติตาม ประกอบกับมาตรฐานซอฟต์แวร์ระดับสากลที่มี อยู่ในปัจจุบันจะเหมาะสมกับการปฏิบัติงานขององค์กรขนาดใหญ่ISO 29110 จึ่งถูกพัฒนาด้วยแนวคิดพื้นฐานเพื่อสนับสนุนองค์กรขนาดเล็กให้มีโอกาสในการปรับปรุงกระบวนการและรับรองคุณภาพในระดับสากล

2.3.3 กระบวนการของ ISO 29110

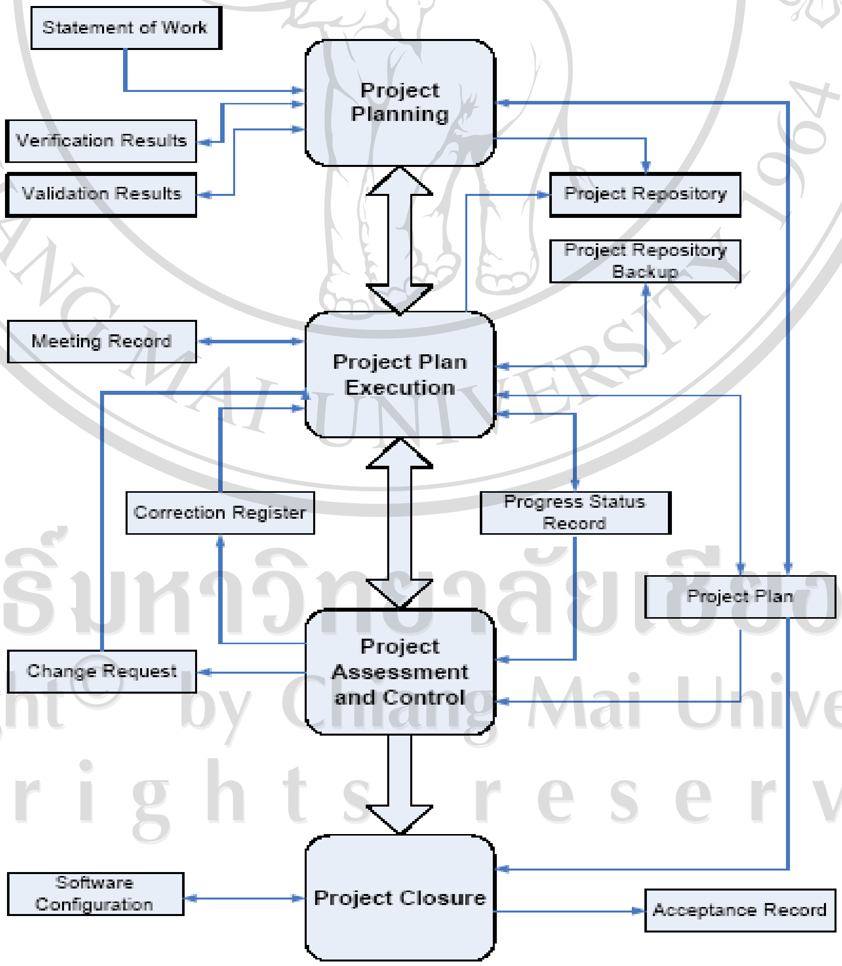
กระบวนการของ ISO 29110 จะเน้นให้ผู้ประกอบการซึ่งอาจจะเป็นผู้ประกอบการอิสระ ผู้ประกอบการขนาดเล็กที่มีบุคคลกรไม่เกิน 25 คน หรือหน่วยงานทางด้านซอฟต์แวร์ที่อยู่ในองค์กรขนาดใหญ่ให้มีกระบวนการในการพัฒนาซอฟต์แวร์ที่เป็นระบบ และเข่าสู่กระบวนการสากล โดยจะเป็นการเริ่มต้นในเชิงกิจกรรมของการปรับปรุงกระบวนการ หรือ SPI (Software Process Improvement) ทาง ISO 29110 ได้ให้ความสำคัญในกระบวนการที่จะต้องทำการปรับปรุงให้เป็นระบบและเป็นสากล 2 กระบวนการหลัก คือ กระบวนการด้านการบริหารโครงการ (Project Management) และกระบวนการด้านการสร้างซอฟต์แวร์ (Software Implementation) ซึ่งจะประกอบด้วยกระบวนการย่อยๆภายในอีกทั้งสองกระบวนได้ ถูกออกแบบให้เหมาะสมกับผู้ประกอบการขนาดเล็กจึงมีความเหมาะสมในการประยุกต์ ใช้ได้ทันที โดยได้กำหนดขนาดของกระบวนการให้เหมาะสมกับองค์กรขนาดเล็ก จึงไม่สร้างปัญหาในการปรับใช้งานให้เข้ากับองค์กร

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

1) กระบวนการด้านการบริหารโครงการ (Project Management)



รูปที่ 2.6 แผนภาพแสดงกิจกรรมหลัก (Basic Profile) ตามมาตรฐาน ISO/IEC 29110



รูปที่ 2.7 แผนภาพแสดงกิจกรรมของกระบวนการบริหารโครงการ ตามมาตรฐาน ISO 29110

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

สำหรับกระบวนการบริหารโครงการตามมาตรฐาน ISO29110 นั้น มีกิจกรรมหลักทั้งสิ้น 4 กิจกรรมหลัก ซึ่งประกอบไปด้วย

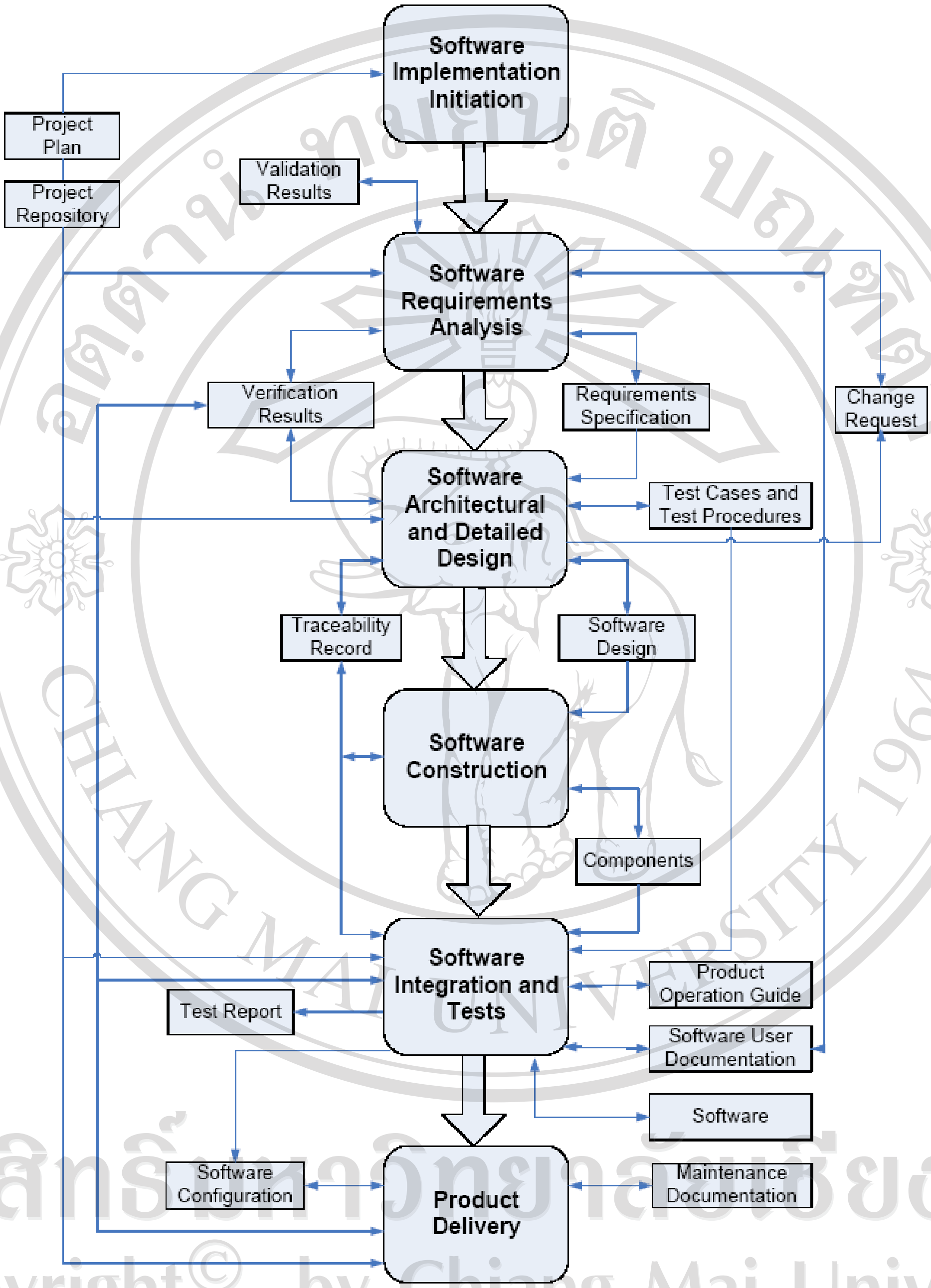
- PM.1 Project Planning
- PM.2 Project Plan Execution
- PM.3 Project Assessment and Control
- PM.4 Project Closure

โดยมีวัตถุประสงค์ในการดำเนินการตามกิจกรรมทั้งสิ้น 7 ประการ ซึ่งประกอบไปด้วย

- PM.O1. Project Plan จะต้องสร้างมาจากเอกสารความต้องการ โดยในแผนต้องมีการระบุทรัพยากรที่ต้องใช้และเวลาในการดำเนินการแต่ละขั้นตอน โดยประมาณ และแผนที่ได้จะต้องมีการรับทราบและยอมรับในแผน ก่อนจะเริ่มนำไปใช้งานในขั้นตอนถัดไป
- PM.O2. การดำเนินการโครงการจะต้องถูกติดตามความก้าวหน้าอย่างต่อเนื่องโดยเทียบเคียงกับแผนการดำเนินการโครงการและมีการบันทึกผลเก็บไว้ในแบบบันทึกการติดตามความก้าวหน้าโครงการ
- PM.O3. ขั้นตอนการร้องขอเปลี่ยนแปลง (Change Request) จะต้องมีการระบุวิธีการรับเรื่องและขั้นตอนการวิเคราะห์ผลกระทบ ทั้งในด้าน ต้นทุน เวลา และ ปัจจัยในทางเทคนิค
- PM.O4. มีการสรุปการประชุมระหว่างทีมพัฒนาและลูกค้า เพื่อกำหนดข้อตกลงในการติดตามความก้าวหน้าของโครงการ
- PM.O5. มีการระบุความเสี่ยง (Risk) ที่อาจจะเกิดในระหว่างการดำเนินการและจัดการโครงการ
- PM.O6. มีการกำหนดแผนการจัดการรุ่นซอฟต์แวร์ (Version Control Plan) โดยมีการกำหนด Baseline และมีการควบคุมการแก้ไขของทีมพัฒนา และการนำออกจากระบบเพื่อส่งมอบลูกค้า โดยในแผนต้องระบุสถานที่เก็บ และ โครงสร้างของสารบบ (Directory Structure)
- PM.O7. มีแผนการประกันคุณภาพซอฟต์แวร์ (Software Insurance) เพื่อให้มั่นใจกระบวนการดำเนินการ และผลการดำาเนินการมีความสอดคล้องกับแผนการดำเนินการและข้อตกลงความต้องการ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2) กระบวนการด้านการ สร้างซอฟต์แวร์ (Software Implementation)



รูปที่ 2.8 แผนภาพแสดงกิจกรรมในกระบวนการพัฒนาซอฟต์แวร์ตามมาตรฐาน ISO 29110

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

สำหรับกระบวนการพัฒนาซอฟต์แวร์ตามมาตรฐาน ISO29110 นั้น มีกิจกรรมหลักทั้งสิ้น 6 กิจกรรมหลัก ซึ่งประกอบไปด้วย

- SI.1 Software Implementation Initiation
- SI.2 Software Requirements Analysis

- SI.3 Software Architectural and Detailed Design
- SI.4 Software Construction
- SI.5 Software Integration and Tests
- SI.6 Product Delivery

โดยมีวัตถุประสงค์ในการดำเนินการตามกิจกรรมทั้งสิ้น 7 ประการ ประกอบไปด้วย

- SI.O1. งานในแต่ละกิจกรรมจะต้องได้รับการดำเนินการบรรลุเป้าหมายตามแผนการดำเนินการ
- SI.O2. ข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirement) จะต้องมีการกำหนดให้ชัดเจนถึงวิธีการวัดผลและข้อตกลงการทดสอบโดยวิธีการวัดผลดังกล่าวจะต้องได้รับการยอมรับจากลูกค้าเพื่อจัดทำเป็นเอกสารและเก็บไว้
- SI.O3. เอกสารสถาปัตยกรรมซอฟต์แวร์หรือเอกสารการออกแบบซอฟต์แวร์ที่จัดทำขึ้นจะต้องสามารถอธิบายรายละเอียดเกี่ยวกับซอฟต์แวร์ได้ดีรวมไปจนถึงมีความสัมพันธ์ที่ตรวจสอบย้อนกลับไปยังเอกสารความต้องการซอฟต์แวร์ได้
- SI.O4. ส่วนประกอบของซอฟต์แวร์ที่สร้างขึ้นจะต้องผ่านการทดสอบระดับหน่วย (Unit Test)และเป็นไปตามเอกสารความต้องการซอฟต์แวร์และเอกสารการออกแบบ
- SI.O5. มีการทดสอบการทำงานร่วมกันของส่วนประกอบทั้งหมดของซอฟต์แวร์ตาม Test Cases และ Test Procedures ที่ออกแบบไว้พร้อมบันทึกผลการทดสอบไว้ใน Test Report โดยข้อผิดพลาดทั้งหมดที่ตรวจพบจะต้องได้รับการแก้ไขให้เป็นไปตามเอกสารการออกแบบ
- SI.O6. โครงร่างซอฟต์แวร์ (Software Configuration) จะต้องตรงกับเอกสารความต้องการซอฟต์แวร์ที่เป็นไปตามข้อตกลงกับลูกค้าซึ่งรวมถึงคู่มือผู้ใช้ซึ่งทั้งหมดจะต้องถูกจัดเก็บใน Project Repository และเมื่อมีความต้องการเปลี่ยนแปลงรายการใด Repository แล้วนั้นจะต้องมีการเปิดคำขอเปลี่ยนแปลง
- SI.O7.การดำเนินงานทั้งหมดรวมถึงผลผลิตที่ได้จากงานทั้งหมดจะต้องได้รับการตรวจสอบว่าเป็นไปตามเกณฑ์มาตรฐานของ Input Product และ Output Product ของแต่ละกิจกรรมตามมาตรฐานกำหนดข้อผิดพลาดที่พบจะต้องได้รับการแก้ไขและบันทึกไว้ใน Validation Result หรือ Verification Result

และจากขั้นตอนดำเนินการตามวัตถุประสงค์ของมาตรฐาน ISO29110 จะได้เอกสารทั้งสิ้น 20 เอกสาร โดยมีรายละเอียดดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

<u>เอกสารที่ได้จากขั้นการบริหารโครงการ 8 เอกสาร</u>

- *Project Plan* คือ เอกสารและตารางเวลาโครงการ
- *Progress Status Record* คือ เอกสารบันทึกผลความก้าวหน้าโครงการ
- *Correction Register* คือ เอกสารบันทึกประเด็นปัญหาในโครงการพร้อมแนวทางแก้ไข
- *Meeting Record* คือ เอกสารบันทึกการประชุม
- *Change Request* คือ เอกสารขอเปลี่ยนแปลงความต้องการที่เกิดขึ้นระหว่างดำเนินการ
- *Verification Result* คือ เอกสารบันทึกผลการทบทวนเอกสาร
- *Validation Result* คือ เอกสารบันทึกผลการทบทวนเอกสารกับลูกค้า
- *Acceptance Record* คือ เอกสารการส่งมอบผลิตภัณฑ์ของโครงการ

<u>เอกสารที่ได้จากขั้นตอนการพัฒนาซอฟต์แวร์ 12 เอกสาร</u>

- *Requirement Specification* คือ เอกสารความต้องการซอฟต์แวร์
- *Software Design* คือ เอกสารการออกแบบซอฟต์แวร์
- *Traceability Record* คือ เอกสารแสดงความสัมพันธ์ของผลิตภัณฑ์
- *Software Component* คือ ผลิตภัณฑ์ซอฟต์แวร์
- *Software* คือ ผลิตภัณฑ์สำหรับส่งมอบให้กับลูกค้า
- *Test Cases and Test Procedures* คือ เอกสารจัดทำการทดสอบพร้อมวิธีการทดสอบ
- *Test Report* คือ เอกสารรายงานผลการทดสอบ
- *Software User Document* คือ เอกสารคู่มือการใช้งานระบบ
- *Product Operation Guideline* คือ เอกสารคู่มือการติดตั้งและการจัดการซอฟต์แวร์
- *Maintenance Document* คือ เอกสารกำหนดผลิตภัณฑ์และเครื่องมือที่ใช้ในโครงการ
- *Verification Result* คือ เอกสารบันทึกผลการทบทวนเอกสาร
- *Validation Result* คือ เอกสารบันทึกผลการทบทวนเอกสารกับลูกค้า

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.4 Extreme Programming (XP)

(Don Wells, 2009) Extreme Programming (XP) คือ กระบวนการทางด้านวิศวกรรมซอฟต์แวร์ที่มีชื่อเสียง ซึ่งผู้ที่นำเสนอวิธีการนี้ เชื่อว่าการปรับเปลี่ยนความต้องการในทุก ๆ ขั้นตอนในระหว่างทำโครงการฯ โดยสามารถเข้าถึงและทำได้ง่ายกว่าการกำหนดความต้องการตั้งแต่

แรกเริ่มและยังใช้ความพยายามในการควบคุมการเปลี่ยนแปลงความต้องการน้อยกว่า นอกจากนี้ยังเป็นวิธีที่ทำให้ผู้ใช้ และโปรแกรมเมอร์ได้ทำงานร่วมกันในทีม ซึ่งจะช่วยสร้างบรรยากาศในการสื่อสารให้เกิดขึ้นภายในทีมและการทำงานร่วมกันนี่เองที่จะช่วยแก้ปัญหาจำนวนมากที่เกิดขึ้นทั้งระหว่างและหลังการสร้างซอฟต์แวร์ได้

Extreme Programming นำเสนอแนวคิดในการแบ่งการพัฒนาออกเป็นกลุ่มของงานย่อย โดยแต่ละงานย่อย ควรจะทำเสร็จ ภายในระยะเวลาสั้น ๆ และควรมีการพบประระหว่างทีมพัฒนาในแต่ละวัน เพื่อทดสอบงานที่ทำเสร็จลงไปแล้ว โดยถ้ามีความเปลี่ยนแปลงเกิดขึ้นกับข้อกำหนดของซอฟต์แวร์ (Software Specification) จะต้องมีการจัดการอย่างเป็นกระบวนการเท่านั้น สำหรับตัวโครงการเองจะมีการพัฒนาแบบก้าวหน้าโดยโค้ดที่เขียนขึ้นต้องเพียงพอที่จะทำงานในระดับของฟังก์ชันรวมทั้งรายละเอียดทุกอย่าง (การออกแบบ คุณสมบัติและตัวโค้ด) จะต้องได้รับการดูแลควบคู่กันไปอย่างต่อเนื่อง เพื่อให้ได้ผลลัพธ์ตรงกับความต้องการของผู้ใช้มากที่สุด

XP เป็นการกำหนดกฎเกณฑ์ในการปฏิบัติงานวันต่อวันสำหรับผู้บริหารและนักพัฒนาระบบ กฎเกณฑ์ในการปฏิบัติเพื่อจะนำไปสู่กระบวนการพัฒนาที่ตอบสนองต่อความต้องการของผู้ใช้ให้ดียิ่งกว่าวิธีการแบบดั้งเดิม อีกทั้งยังทำให้การสร้างซอฟต์แวร์มีคุณภาพที่ดีขึ้นอีกด้วย

2.4.1 วัตถุประสงค์และเป้าหมาย

วัตถุประสงค์หลักของ XP คือการลดต้นทุนการเปลี่ยนแปลง ซึ่งสำหรับวิธีพัฒนาระบบแบบดั้งเดิมความต้องการของระบบจะถูกกำหนดเมื่อตอนเริ่มต้นโครงการ และบ่อยครั้งที่จะยึดติดกับความต้องการนั้น ซึ่งก็จะทำให้ต้นทุนการเปลี่ยนแปลงที่จะทำให้โปรแกรมเป็นไปตามความต้องการในขั้นตอนต่อมาสูงขึ้น

2.4.2 หลักการและเหตุผล

1) **Feedback** การตอบรับจะมีประโยชน์สูงสุดหากตอบสนองได้อย่างรวดเร็ว ระยะเวลาระหว่างการทำงานกับการทดสอบยิ่งใกล้กันเท่าไร ยิ่งเพิ่มความสำคัญของการตอบรับมากขึ้น โดยเป็นการย่นระยะเวลาผลสำเร็จของการพัฒนาให้เร็วขึ้น แทนที่จะยังอยู่ในช่วงกำลังพัฒนาระบบ

2) **Assuming Simplicity** การยอมรับความเรียบง่ายเป็นการแก้ไขปัญหาต่างๆ ซึ่ง XP จะไม่ยอมรับวิธีการแบบดั้งเดิมที่ให้ทำการวางแผนเพื่ออนาคตและเขียนโปรแกรมเพื่อนำกลับมาใช้ใหม่

3) **Incremental Changes** ในการสร้างโอกาสครั้งใหญ่ทั้งหมดในคราวเดียวอาจไม่ได้ผลอย่างที่คิดไว้ ยกตัวอย่างเช่น ในแบบ XP อาจมีการปล่อยระบบเล็กๆ ออกมาในทุก 3 อาทิตย์ เพื่อเป็นการสร้างทางเดินที่ละเล็กที่ละน้อยในการควบคุมการทำงานในการพัฒนาระบบให้มากขึ้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

4) Embracing Change เป็นการไม่ทำงานต่อต้านความเปลี่ยนแปลง แต่ให้ทำงานร่วมกับมัน ซึ่งการเคลื่อนไหวนี้ทำให้โครงการดำเนินงานต่อไปได้ การเปลี่ยนแปลงไม่ใช่สิ่งเลวร้าย

5) Encouraging Quality Work การกระตุ้นให้เกิดการทำงานอย่างสนุกสนาน ทำงานร่วมกับทีมได้เป็นอย่างดี ทำให้ผลงานออกมาอย่างมีคุณภาพ เป็นต้น

2.4.3 กิจกรรมในการพัฒนาซอฟต์แวร์

1) การพัฒนาซอฟต์แวร์ (Coding)

ประโยชน์ของ Coding นั้นมีอยู่หลากหลายมาก เช่น สามารถวาดแผนภาพ(Diagram) ที่ช่วยในการสร้าง Code นอกจากนี้ยังสามารถหาวิธีแก้ไขปัญหาที่เหมาะสม ตัวอย่างเช่น XP จะเลือกวิธีแก้ไขปัญหาที่เป็นไปได้ออกมาและจะเลือกเอาวิธีแก้ไขปัญหาที่เหมาะสมที่สุด โดยจะมีการตรวจสอบก่อนทำการเลือกวิธีแก้ไขปัญหาเพื่อให้ได้วิธีแก้ไขปัญหาที่เหมาะสมที่สุด Coding ยังช่วยในการสื่อสารความคิดที่เกี่ยวข้องกับปัญหาในการพัฒนาระบบโปรแกรมเมอร์ที่ทำงานกับปัญหาทางการพัฒนาระบบที่ซับซ้อนและพบว่าเป็นการยากที่จะถ่ายทอดความคิดของตนเองไปยังโปรแกรมเมอร์คนอื่นๆ โดยให้โปรแกรมเมอร์คนอื่นเข้าใจในสิ่งที่ตนถ่ายทอดไปในทิศทางเดียวกัน ซึ่ง Code ที่จะได้มาจะเคลียร์ สั้นกระชับและไม่สามารถที่จะแปลความหมายออกไปในทิศทางอื่นๆ ได้เลย ซึ่งโปรแกรมเมอร์คนอื่นๆ ก็สามารถจะ feedback ความคิดของตนเองโดย Coding ความคิดของพวกเขาลงไปได้เช่นกัน

2) การทดสอบซอฟต์แวร์ (Testing)

การทดสอบเป็นอีกกิจกรรมที่ใช้ในการพัฒนาระบบ มุมมองของ XP นั้น การทดสอบเป็นสิ่งที่จำเป็นซึ่งมีข้อแนะนำที่ใช้ในการทดสอบว่าในการทดสอบนั้นทำเพื่อตรวจสอบในเรื่องของการทำงานของชุดคำสั่ง (Coding, Functionality) ประสิทธิภาพการทำงาน(Performance) และความสอดคล้องในการทำงาน (Conformance)ซึ่งการทดสอบทั้งหลายนั้นจำเป็นที่จะต้องทำอยู่เสมอในระหว่างกระบวนการในการพัฒนาระบบ

สำหรับเหตุผลในการทำการทดสอบนั้นสามารถแบ่งออกเป็นสองประเภทคือประโยชน์ระยะสั้น (Short-term)เพื่อให้เกิดความมั่นใจในสิ่งที่โปรแกรมเมอร์กำลังทำอยู่ว่าจะไม่มีการหลุดออกจากกรอบแนวคิดที่ได้วางไว้ในตอนแรก และประโยชน์ระยะยาว (Long-term)สำหรับทำขึ้นเพื่อให้ระบบนั้นๆ ดำเนินงานของมันต่อไปได้อยากถูกต้อง

3) การฟัง (Listening)

กิจกรรมการฟังในระหว่างการสัมภาษณ์ ใน XP นั้น โปรแกรมเมอร์หรือนักพัฒนาระบบไม่จำเป็นจะต้องรู้อะไรเกี่ยวกับเรื่องของธุรกิจของผู้ใช้ แต่สิ่งที่จำเป็นก็คือนักพัฒนาระบบและโปรแกรมเมอร์ต้องเข้าใจในสิ่งที่ผู้ใช้ต้องการจริงๆ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

4) การออกแบบ (Designing)

ในการพัฒนาระบบนั้นไม่ได้ขึ้นอยู่แต่เพียงการเขียน Code ให้ดี การทดสอบหรือแม้แต่การรับฟังความต้องการที่แท้จริงของผู้ใช้เพียงเท่านั้น นักพัฒนาระบบบางคนสามารถทำงานไปได้ด้วยดีโดยปราศจากการออกแบบแต่นั่นหมายความว่าระบบที่กำลังพัฒนามีการติดขัดและเสียเวลา ซึ่งจะทำให้ระบบเกิดความซับซ้อนและยากที่จะดำเนินงานต่อ เพื่อที่จะหลีกเลี่ยงปัญหานี้สามารถทำได้โดยการออกแบบโครงสร้างไว้ใช้ในการจัดการกับแนวคิดที่ใช้ในการพัฒนาของระบบนั้นๆ การออกแบบที่ดี จะทำให้สามารถขยับขยายหรือเปลี่ยนแปลงส่วนต่างๆ ในระบบบางส่วนโดยจะไม่กระทบกับส่วนอื่นๆ ที่มีอยู่ในระบบอีกด้วย

2.4.4 ทรัพยากรที่ใช้ในการพัฒนาซอฟต์แวร์

การควบคุมทรัพยากรที่ไม่คงที่ของ XP ทั้ง 4 อย่าง จะช่วยให้กิจกรรม (Activities) ประสบความสำเร็จ ทรัพยากรทั้ง 4 ที่ต้องรวบรวมเพื่อทำให้โครงการเสร็จสมบูรณ์เหล่านั้น ได้แก่ เวลา ต้นทุน คุณภาพ และขอบเขต เมื่อทรัพยากรทั้ง 4 ถูกรวบรวมอย่างถูกต้องในการวางแผน จะเกิดความสมดุลระหว่างทรัพยากรและกิจกรรมที่จำเป็นต่อการดำเนินโครงการให้เสร็จสมบูรณ์ตามเวลาภายใต้ข้อบังคับต่างๆ และการบริหารโครงการนั้นเป็นสิ่งที่สำคัญยิ่ง การบริหารโครงการไม่ได้หมายถึงการจัดการงานกับทรัพยากรร่วมกันอย่างง่ายๆ เพียงเท่านั้น มันยังหมายความถึง การเผชิญกับรายจ่ายจำนวนมาก ซึ่งต้นทุนจะต้องถูกกำหนดไว้ก่อน การควบคุมทรัพยากรที่ไม่คงที่เหล่านี้สามารถอธิบายได้ดังนี้

1) เวลา (Time) เวลา คือ เวลาที่จะต้องเพียงพอต่อการดำเนินโครงการให้เรียบร้อย ซึ่งก็จะถูกแบ่งออกเป็นหลายๆ ส่วน เช่น เวลาที่จะรับฟังผู้ใช้ระบบเวลาที่จะออกแบบ เวลาที่จะเขียนโปรแกรม และเวลาที่จะทดสอบเป็นต้น ในการพัฒนาระบบ โปรแกรมเมอร์สามารถสร้างซอฟต์แวร์ที่มีคุณภาพได้ แต่อาจจะล้มเหลวในการรับฟังเสียงตอบรับจากผู้ใช้ระบบ การละทิ้งกิจกรรมในการรับฟังเพื่อบรรลุวัตถุประสงค์ในกิจกรรมอื่น จะส่งผลเสียเป็นอย่างมาก เพราะไม่ว่าจะอยู่ในธุรกิจใดก็ตามผู้ใช้ต่างต้องการความใส่ใจ นอกจากนี้ เวลาที่มากกว่าก็จะช่วยให้ได้ผลลัพธ์ที่ต้องการของผู้ใช้ระบบ บางส่วนชื่นชอบการทำงานที่เสร็จตรงตามเวลามากกว่าการขยายเวลาเพื่อเพิ่มส่วนประกอบอื่นๆ ของโปรแกรม

2) ต้นทุน (Cost) ต้นทุน คือ ตัวแปรที่สองที่สามารถพิจารณาปรับปรุงแก้ไข สังเกตได้ว่าต้นทุนสามารถทำให้โครงการเกิดความสมดุลได้ โดยจะต้องมีการพิจารณาเฉลี่ยทรัพยากรด้านต้นทุนให้ดี ดูว่าต้นทุนใดที่เพิ่มขึ้นแล้วเป็นประโยชน์ต่อการพัฒนาโดยปกติคนมักจะมองว่าการจ้างคนเพิ่มขึ้น งานจะเสร็จได้เร็วขึ้น แต่ในความเป็นจริงแล้วถ้าเราจ้างโปรแกรมเมอร์มากขึ้น อาจจะ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ทำให้การทำงานชนกับกลุ่มเดิมก็ได้ ดังนั้น การเพิ่มขึ้นไม่ได้หมายความว่าจะทำให้เวลาลดลงครึ่งหนึ่งเสมอไป ควรพิจารณาถึงการเพิ่มความต้องการด้านการสื่อสารและต้นทุนที่มองไม่เห็นอื่นๆ นอกจากนี้ หากคนที่เข้ามาใหม่เข้าไปอยู่ แล้วไม่รู้เรื่องโครงการหรือเรื่องทีมงานเดิม เขาจะดึงให้ทีมงานเดิมทำงานช้าลงไปอีก เพราะจะต้องเสียเวลามาดูแลคนที่มาใหม่ให้ทำงานได้คล่องขึ้น การทำงานล่วงเวลาก็จะไม่เป็นประโยชน์เช่นกัน นอกจากจะเพิ่มต้นทุนแล้ว ยังไม่ได้ทำให้ผลงานดีขึ้นด้วย โปรแกรมเมอร์ที่เหนื่อยล้าจะมีประสิทธิภาพต่ำกว่าโปรแกรมเมอร์ที่กระตือรือร้น พวกเขามักจะใช้เวลาในการดำเนินงานนานกว่าจะเสร็จสิ้น และมักก่อข้อผิดพลาดซึ่งทำให้ต้องเสียเวลาในการซ่อมแซมอีกด้วย

3) คุณภาพ (Quality) ระบบที่มีความสมบูรณ์จะให้ความสำคัญกับคุณภาพและวิธีการ (TQM และ 6 Sigma) ซึ่งช่วยให้มั่นใจได้ว่า คุณภาพของซอฟต์แวร์อยู่ในระดับที่สูง อย่างไรก็ตาม การเอาใจใส่ในเรื่องคุณภาพจะเป็นเรื่องที่ถูกคาดหวัง คุณภาพสามารถปรับปรุงได้ทั้งจากภายในและภายนอก คุณภาพภายในจะเกี่ยวข้องกับการทดสอบซอฟต์แวร์ เช่น ด้านการทำงานต่างๆ (โปรแกรมจะช่วยสนับสนุนงานอย่างไร) และการมีมาตรฐานเดียวกัน (ซอฟต์แวร์มีมาตรฐานและสามารถบำรุงรักษาได้หรือไม่) ส่วนคุณภาพภายนอก คือ ผู้ใช้ มองระบบว่าเป็นอย่างไร ผู้ใช้จะสนใจในส่วนการดำเนินงาน บางคำถามที่ผู้ใช้ระบบถามคือโปรแกรมสามารถเชื่อใจได้หรือไม่ ผลลัพธ์ที่ออกมามีประสิทธิภาพหรือไม่ ผลลัพธ์จะออกมาตรงเวลาหรือไม่ ซอฟต์แวร์ทำงานอย่างง่ายหรือไม่ ส่วนติดต่อผู้ใช้ง่ายต่อจะทำความเข้าใจและใช้งานหรือไม่เป็นต้น ในบางครั้ง XP ยอมให้ประเด็นเรื่องคุณภาพภายนอกบางอย่างตกไปบ้าง เพื่อให้ระบบสามารถเสร็จออกมาได้ตรงตามเวลา ผู้ใช้อาจจะมีข้อโต้แย้งเกี่ยวกับบั๊กของซอฟต์แวร์ ถ้าต้องการทำให้ทันเส้นตาย โปรแกรมก็อาจจะไม่สมบูรณ์แบบ หรือโปรแกรมเมอร์อาจแก้ไขโดยการออกเวอร์ชั่นใหม่ๆ ตามมาก็ได้

4) ขอบเขต (Scope) จะถูกกำหนดโดยผู้ใช้ ซึ่งจะเขียนความต้องการให้กับโปรแกรมเมอร์ จากนั้นโปรแกรมเมอร์จะตรวจสอบว่าความต้องการนั้น สามารถทำให้เกิดความพึงพอใจได้มากเท่าไร วิธี XP นี้จะช่วยรักษาคุณภาพและทำโครงการให้เสร็จสิ้นตรงตามเวลา อย่างไรก็ตาม นักวิเคราะห์ XP ที่ทำการขยายขอบเขตของ Project อาจจะทำสำเร็จได้ด้วยการเจรจากับผู้ใช้ระบบ 1 ครั้ง หรือมากกว่านั้นก็ได้

สรุปแล้วนักวิเคราะห์ XP สามารถควบคุมทรัพยากรสำคัญได้ทั้ง 4 อย่าง ทั้งเวลา ต้นทุน คุณภาพ และขอบเขต ซึ่งต้องมีตัววัดและให้ความสำคัญต่อการดำเนินโครงการให้เสร็จสิ้นตรงตามเวลา

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.4.5 แนวทางปฏิบัติ

XP สามารถแบ่งออกได้เป็น 12 แนวทางปฏิบัติและแบ่งออกเป็น 4 กลุ่มใหญ่ ดังนี้

<u>กลุ่มที่1</u>

1. Pair programming

Pair programming คือ code ทั้งหมดถูกสร้างจากนักพัฒนาระบบ 2 คนโดยทำงานบนเครื่องคอมพิวเตอร์เครื่องเดียวกัน คนที่หนึ่งจะเป็นคนที่เข้าใจรายละเอียดการ coding ส่วนอีกคนนั้น จะเป็นคนค่อยติดตามดูการทำงานของคนแรก และจะสลับกันทำหน้าที่กัน เป็นประจำ การทำ Pair programming ไม่ได้จำกัดคนสามารถ สับเปลี่ยนคนได้ตามความสามารถที่ทำได้

2. Planning game

วิธีการวางแผนใน XP เรียกว่า planning game โดยแบ่งออกเป็น 2 ส่วน

ส่วนที่หนึ่ง Release Planning จะสนใจว่าสิ่งที่เป็นความต้องการถูกนำมาใส่ไว้ก่อนที่จะนำส่ง ในส่วนนี้ทั้ง programmer และผู้ใช้จะมีส่วนร่วมกัน ทั้งนี้ Release planning ยังแบ่งออกเป็น 3 ระยะ คือ Exploration phase, Commitment Phase และ Steering Phase

1) Exploration Phase: ในส่วนนี้ ลูกค้าจะเขียนความต้องการในระบบลงใน Story card โดยมีวิธีการดังนี้
   - เก็บความต้องการของลูกค้าต่าง
   - เขียนความต้องใส่ลง User story card
   - แยกปัญหาต่างๆออกจากกัน
   - ประมาณเวลาการทำเพื่อให้ได้มาตาม User story card
2) Commitment Phase : เป็นส่วนของการตกลงกันในเรื่องของฟังชันการทำงานต่างๆ ผลตอบแทนและแผนตารางวันการทำงาน โดยมีส่วนประกอบด้วยกัน 4 ส่วนคือ
   - Sort by Value
   - Sort by Risk
   - Sort by velocity
   - Choose scope
3) Steering Phase : ในส่วนนี้แผนสามารถปรับเปลี่ยนและจะเพิ่มหรือลดความต้องการจากเดิมได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ส่วนที่สอง Iteration Planning ในส่วนนี้เป็นส่วนของ Developer ไม่เกี่ยวข้องกับลูกค้า แบ่งออกเป็น 3 ส่วนคือ Exploration Phrase, Commitment Phase, Steering Phase

1) Exploration Phase: ในส่วนนี้ความต้องการของลูกค้าจาก User story card จะถูกแปลงออกเป็นหน้าที่ต่างๆ และจะถูกบันทึกลงใน Task card แบ่งเป็น 3 ขั้นตอน
   - เก็บรวบรวม User story card
   - ยุบรวมและแบ่งงานออกเป็นส่วนๆ
   - ประมาณเวลาของงานในแต่ละส่วน
2) Commitment Phase: เป็นส่วนที่จะแบ่งงานให้กัน programmer ต่างๆพร้อมทั้งวันที่ต้องส่งงาน
3) Steering Phase: จะนำผลงานที่ได้มาเปรียบเทียบกับ User story card ว่าตรงกับที่ต้องการหรือไม่

3. Test Driven Development เป็นขั้นตอนการทดสอบงานที่ทำเสร็จในแต่ละส่วน

4. Whole Team การมีส่วนร่วมในทีมทั้งผู้ใช้และผู้ออกแบบ

<u>กลุ่มที่ 2</u> Continuous Processes

5. Continuous Integration การพัฒนาอย่างต่อเนื่อง แต่ให้ระวังการเพิ่มเติม เพราะอาจจะเป็นการเพิ่มปัญหาเข้าไป

6. Design improvementการออกแบบที่เรียบง่ายเพื่อการพัฒนาหรือปรับแต่งระบบในอนาคต

7. Small release สร้างความสัมพันธ์ต่างๆ ระหว่างส่วนต่างๆของโปรแกรมในน้อยที่สุดเพื่อง่ายต่อการพัฒนา

<u>กลุ่มที่3</u> Share Understanding

8. Coding Standard การเขียนโค้ด ต่างๆในรูปแบบที่เป็นมาตรฐาน

9. Corrective Code Ownership การที่อนุญาต ให้ผู้อื่นสามารถแก้ไขในส่วนของตนเองได้

10. Simple Design การใช้วิธีการต่างๆหรือแม้ ฟังชันการทำงานที่เรียบง่ายเพื่อง่ายต่อการเข้าใจ

11. System Metaphor การใส่ชื่อให้คล้ายกับหน้าที่การทำงานของส่วนนั้นๆ

<u>กลุ่มที่ 4</u> Programmer Welfare

12. Sustainable Pace การให้งานไม่ควรเกิน40 ชม ต่อ สัปดาห์

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.4.6 ขั้นตอนการปฏิบัติ

การพัฒนา Extreme Programming นั้น จะเข้ามามีส่วนร่วมและควบคุมกระบวนการพัฒนาระบบ ซึ่งจะมีปฏิสัมพันธ์ การทำซ้ำและเชื่อมประสานมากกว่าแบบ SDLC แต่ XP จะไม่แนะนำผู้พัฒนาในแต่ละช่วงการพัฒนา หากจะเป็นการเพิ่มขึ้นของกิจกรรมที่มีความสัมพันธ์กันที่บ่อย สังเกตได้จากหลายๆ ขั้นตอน ของวงจรสำหรับ XP จะมีขึ้นอย่างเป็นประจำทุกวัน ซึ่งค่อนข้างที่จะชัดเจนกว่า SDLC ที่มีกระบวนการที่ผ่านไปอย่างช้า และในแต่ละช่วงกิจกรรมนั้นๆ (ต้องการการวิเคราะห์, ทดสอบ และอื่นๆ) จะสำเร็จในแต่ละระยะที่ชัดเจนขึ้น กระบวนการ XP จะช่วยให้กิจกรรมเหล่านั้นสำเร็จลงได้ 5 ระยะของกระบวนการพัฒนา XP คือ

1. การสำรวจ
2. วางแผน
3. การวนลูปทำซ้ำๆกันโดยเริ่มที่ขั้นตอนแรก
4. การผลิต
5. การบำรุงรักษา

ขั้นตอนการพัฒนาจะเริ่มจากการตรวจสอบงานของผู้ใช้งานที่ระบุไว้ ระยะของการรวบรวมและสรุปจะถูกเรียกว่า Stand-up Meeting โดยอาจมีการตั้งคำถามสัก 2 -3 คำถามเกี่ยวกับการทำงานในแต่ละวัน เพื่อเป็นการช่วยอธิบายงานนั้นๆ และอาจจะซักถามนักพัฒนาคนอื่นๆ เพื่อช่วยในงานนั้นๆ

ขั้นตอนถัดไปคือ เขียนลงใน Task บน to do list ซึ่งในขั้นนี้อาจจะเขียน task case สำหรับสิ่งที่พยายามค้นหามัน เมื่อเสร็จแล้วก็ทดสอบการทำงานบางครั้งอาจจะล้มเหลว นักพัฒนาที่จับคู่กันจะต้องช่วยกันดูพิจารณาถึงtest case อื่นๆ และแก้ไขเกี่ยวกับข้อมูลที่ได้บันทึกไว้ และดำเนินการต่อไปเรื่อยๆใน test case และในขั้นสุดท้าย จะต้องบันทึกหนึ่งหัวข้อบน to do list ที่อาจจะต้องมีการปรับเปลี่ยนโครงสร้าง test case อื่นๆไปด้วย

นักพัฒนาจำเป็นต้องเก็บทุกๆ การปรับปรุงและเปลี่ยนแปลง และใช้ข้อมูลเหล่านั้นในทุกๆ Test case พร้อมระบุเหตุผลไว้ในทุกๆข้อผิดพลาดที่ไม่สามารถปฏิบัติการได้ด้วย

การเริ่มต้นกับ XP Development task หนึ่งในวิธีการนั้นก็คือการจับประเด็นให้ได้และแบ่งมันออกเป็นส่วนย่อยๆ โดยจะต้องมุ่งไปที่การแก้ไขปัญหาที่เร่งด่วนเป็นอันดับแรก และจะต้องทำการประยุกต์ใช้กลยุทธ์ XP Development กับการทำงานเพื่อแก้ไขปัญหาต่างๆโดยการทดสอบ coding listening, designing and integrating และจะประสบผลสำเร็จในทุกๆงานของ XP Development ในแต่ละงานเขียนโปรแกรม จากนั้นจะเกิดกระบวนการยอมรับในกระบวนการของระบบที่เพิ่มขึ้นและปัญหาที่ยากนั้นกลับง่ายและตรงไปตรงมากับแก่นที่จะนำไปสู่ความสำเร็จ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved