

PPT 1 Systems Analysis and Design การวิเคราะห์และออกแบบระบบ

Systems , Roles and Development Methodologies ระบบ หน้าที่ และวิธีการพัฒนา

วัตถุประสงค์

- เข้าใจถึงความต้องการการวิเคราะห์และออกแบบระบบในองค์กร
- ตระหนักถึงหน้าที่ในการวิเคราะห์ระบบว่าคืออะไร
- เข้าใจถึงปัจจัยในการพัฒนา 3 ข้อ
 - o SDLC (วงจรการพัฒนาระบบ)
 - o The agile approach (ความคล่องตัว)
 - o Object-oriented systems analysis and design (การวิเคราะห์และออกแบบระบบเชิงวัตถุ)

ข้อมูล ความสำคัญ

- ปัจจัยการนำไปสู่ความสำคัญหรือความล้มเหลวในธุรกิจ
- ความต้องการการจัดการที่ถูกต้อง
- สร้างความแตกต่างของข้อมูลจากการมอบหมาย ด้วยมือ

หัวข้อหลัก

- Fundamentals of different kinds of information systems (ความต่างพื้นฐานของระบบข้อมูล)
- Roles of system analysts (บทบาทของนักวิเคราะห์ระบบ)
- Phases in the systems development life cycle (SDLC) as they relate to Human-computer Interaction (HCI) factors (ขั้นตอนในวงจรการพัฒนาระบบที่สัมพันธ์กับ HCI)
- CASE tools
- Open source software (ทรัพยากรซอฟต์แวร์แบบเปิด)

ความต้องการของการวิเคราะห์และออกแบบระบบ

- การติดตั้งระบบโดยปราศจากการวางแผนที่เหมาะสม นำไปสู่ความไม่พอใจของ USER และเป็นการนำไปสู่การเลิกใช้ระบบ
- โครงสร้างที่ยึดการวิเคราะห์และออกแบบของระบบข้อมูล
- ชุดกระบวนการดำเนินอย่างเป็นระบบสู่การปรับปรุงธุรกิจ ตลอดจนการใช้งานระบบข้อมูลของคอมพิวเตอร์

หน้าที่ของนักวิเคราะห์ระบบ

- ต้องทำงานกับรายละเอียดของคนทั้งหมด และมีประสบการณ์การทำงานกับคอมพิวเตอร์
- 3 หน้าที่หลัก
 - o เป็นผู้ให้คำปรึกษา

- ผู้ชำนาญในด้านการสนับสนุน
- ตัวแทนการเปลี่ยนแปลง

ลักษณะของนักวิเคราะห์ระบบ

- แก้ปัญหาได้
- สื่อสารได้
- เก่ง เป็นมืออาชีพและมีจริยธรรม
- มีวินัยและแรงจูงใจ

SDLC

- ขั้นตอนการแก้ไขปัญหาธุรกิจ
- การพัฒนาที่เป็นวงจรเฉพาะของนักวิเคราะห์และกิจกรรมของผู้ใช้
- แต่ละขั้นตอนจะมีเอกลักษณ์ของกิจกรรมของผู้ใช้

7 ขั้นตอนของ SDLC

1. ระบุปัญหา โอกาส และวัตถุประสงค์
2. กำหนดความต้องการข้อมูลของมนุษย์
3. วิเคราะห์ระบบที่ต้องการ
4. ออกแบบการแนะนำระบบ
5. พัฒนา และ จัดเก็บเอกสารซอฟต์แวร์
6. ทดลองและรักษาระบบ
7. ดำเนินการและประเมินระบบ

ผสมผสานและพิจารณา HCI

- ความต้องการของนักวิเคราะห์ว่า ใครสามารถรวม HCI เข้ากับ กระบวนการพัฒนาระบบที่เพิ่มขึ้น
บริษัทจะเริ่มตระหนักถึงคุณภาพของระบบและคุณภาพชีวิตในการทำงานที่สามารถปรับปรุง โดยการพูดคุยเป็น
หลักและเข้าใจการเริ่มโปรเจค

ระบุปัญหา โอกาส และวัตถุประสงค์

กิจกรรม

- สัมภาษณ์การจัดการผู้ใช้
- สรุปลองค์ความรู้ที่ได้
- ประเมินขอบเขตของงาน
- จัดเก็บผล

ผลลัพธ์

- รายงานความเป็นไปได้ที่มีการนิยามปัญหาและการสรุปวัตถุประสงค์จากการจัดการที่สามารถช่วยการตัดสินใจหรือการดำเนินงานที่เหมาะสม

กำหนดความต้องการข้อมูลของมนุษย์

กิจกรรม

- สัมภาษณ์
- สุ่มตัวอย่างและลงทุน
- แบบสอบถาม
- สังเกต ตัดสินใจจากพฤติกรรมและสภาพแวดล้อม
- สร้างต้นแบบ
- ศึกษา ใคร อะไร ที่ไหน เมื่อไหร่ อย่างไร ทำไมของระบบปัจจุบัน

ผลลัพธ์

- นักวิเคราะห์เข้าใจถึงวิธีที่ผู้ใช้ประสบความสำเร็จในการทำงาน เมื่อมีการปฏิสัมพันธ์กับคอมพิวเตอร์
- รู้ถึงฟังก์ชันธุรกิจ
- ความสำเร็จเกิดจาก
 - o คน
 - o เป้าหมาย
 - o ข้อมูล
 - o การมีส่วนร่วม

วิเคราะห์ระบบที่ต้องการ

กิจกรรม

- สร้าง flow ข้อมูล กิจกรรม หรือ sequence diagrams (ไดอะแกรมแบบลำดับขั้น)
- พจนานุกรมข้อมูลที่ครบถ้วน
- วิเคราะห์องค์ประกอบที่ทำไป
- เตรียมตัวและนำเสนอ ข้อเสนอแนะระบบ

ผลลัพธ์

- ข้อเสนอแนะว่า คืออะไร ถ้าทำแบบนี้ ควรจะเป็นแบบไหน

ออกแบบการแนะนำระบบ

กิจกรรม

- ออกแบบขั้นตอนเพื่อเข้าถึงข้อมูล
- ออกแบบ interface
- ออกแบบระบบควบคุม
- ออกแบบ database
- ออกแบบขั้นตอนการ backup

ผลลัพธ์

- รูปแบบของระบบที่แท้จริง

พัฒนา และ จัดเก็บเอกสารซอฟต์แวร์

กิจกรรม

- วิเคราะห์ระบบกับโปรแกรมเมอร์ต้นฉบับที่พัฒนา
- ทำงานร่วมกับผู้ใช้ เพื่อพัฒนาเอกสารอย่างมีประสิทธิภาพ
- นักออกแบบ โค้ด และ ลบข้อความ error จากโปรแกรมคอมพิวเตอร์
- ซอฟต์แวร์ที่ช่วยเก็บไฟล์ ขั้นตอนแบบ manual และเว็บไซต์ช่วยถามตอบปัญหา

ผลลัพธ์

- โปรแกรมคอมพิวเตอร์
- เอกสารของระบบ

ทดลองและรักษาระบบ

กิจกรรม

- ทดสอบระบบข้อมูล
- รักษาระบบ
- เอกสารของการรักษา

ผลลัพธ์

- ปัญหา หรือถ้ามีปัญหา
- โปรแกรมอัปเดต
- เอกสาร

ดำเนินการและประเมินระบบ

กิจกรรม

- สอนผู้ใช้
- แผนการวิเคราะห์ที่ราบเรียบ แปลงจากระบบเก่าเป็นระบบใหม่
- ทบทวนและประเมินระบบ

ผลลัพธ์

- การฝึกอบรมบุคลากร
- ระบบที่ติดตั้งแล้ว

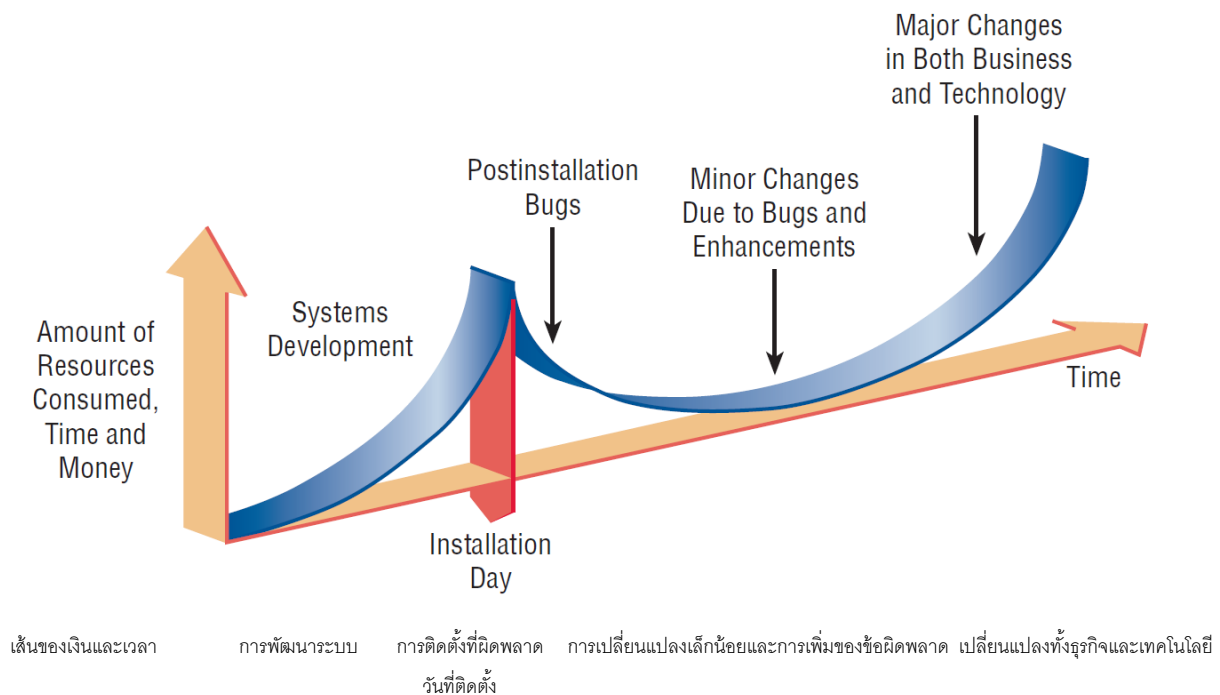
นักวิจัยบางคน ประเมินจำนวนเวลาที่ใช้ในการรักษาระบบ อาจจะมีมากกว่า 60% ของเวลาที่ใช้ในงานทั้งหมด

*สรุป คือ ใช้เวลาในการรักษาระบบ มากกว่า การทำงานหรือกิจกรรมของระบบ เป็น 60% ต่อ 40%

ผลกระทบของการรักษา

- การรักษาเกิดขึ้นจากเหตุผล 2 อย่าง
 - o การลบซอฟต์แวร์ผิดพลาด
 - o เพิ่มหรือเสริมจากที่มีอยู่
- ค่าใช้จ่ายต่อเนื่องในการรักษาจะมากกว่า การสร้างระบบใหม่ และเป็นไปได้ที่องค์กรต้องศึกษาระบบใหม่หมด

ทรัพยากรในการบริโภค



*สรุป เมื่อมีการเปลี่ยน ไม่ว่าจะการพัฒนา การเกิดข้อผิดพลาด จะต้องมีค่าใช้จ่าย(เงิน) ขณะที่เวลาผ่านไป

การเข้าถึง โครงสร้างการวิเคราะห์และออกแบบระบบ และ SDLC

- SDLC แบบดั้งเดิม
- ตัวอย่างของ SDLC
- การวิเคราะห์และออกแบบระบบเชิงวัตถุ

เครื่องมือตัวอย่าง

- คือผลผลิตสำหรับนักวิเคราะห์ระบบ เพื่อการปรับปรุงงานให้เป็นกิจวัตร ตลอดจนการใช้สนับสนุนแบบอัตโนมัติ

เหตุผลที่ใช้เครื่องมือตัวอย่าง

- นักวิเคราะห์เพิ่มขึ้น
- พัฒนาการสื่อสารของนักวิเคราะห์และผู้ใช้
- รวบรวมวงจรกิจกรรม

อธิบายหน้า 27

1. การออกแบบหน้าและรายงาน : หน้าแสดงผลว่ามีข้อมูลอะไรบ้าง
2. ไตรอะแกรมของระบบ : เป็นการวางแผนของระบบ
3. พจนานุกรมข้อมูล : บอกถึงคุณสมบัติว่า เก็บข้อมูลอะไรบ้าง
4. การจัดการโปรเจค : บอกว่าระบบต้องทำอะไร และได้รับอะไร

The agile approach (การเข้าถึงที่รวดเร็ว)

- พื้นฐาน
 - o คุณค่า (ผลลัพธ์)
 - o หลักการ
 - o หลักปฏิบัติ

Agile values

- การสื่อสาร
- ความง่าย
- การตอบกลับ
- ความกล้า

ทรัพยากร 4 ข้อของ Agile

1. เวลา
2. ค่าใช้จ่าย
3. คุณภาพ
4. ขอบเขต

5 หลักในการพัฒนา Agile

1. สำรวจ
2. วางแผน
3. เน้นย้ำ
4. ผลผลิต
5. การรักษา

กระบวนการพัฒนา Agile หน้า 32

1. เริ่มสำรวจ -> วางแผน -> เน้นย้ำ (ในรูปคือการวนซ้ำ 3 รอบ) -> ผลผลิต (เพิ่มขึ้นทีละก้าวหลังจากที่ผลผลิตถูกปล่อย ในรูป วนซ้ำ 5 รอบ) -> การรักษา -> กลับไปที่ การวางแผน

การวิเคราะห์และออกแบบระบบเชิงวัตถุ

- สลับการเข้าถึงโครงสร้างของ SDLC ให้สะดวกต่อการพัฒนาในการเปลี่ยนแปลงที่รวดเร็วของระบบ ในสภาพแวดล้อมของธุรกิจที่ยืดหยุ่น
- เริ่มวิเคราะห์จากส่วนที่เล็กของการติดตามระบบโดย การออกแบบและดำเนินงาน
- วิเคราะห์วงจรซ้ำๆ และดำเนินงานในส่วนต่อไป เป็นการซ้ำๆเพื่อให้โปรเจกต์สมบูรณ์
- ทดสอบระบบ

Unified Modeling Language (UML) Phases : ขั้นตอนการทำโมเดลมาตรฐานในการออกแบบ

**ง่าย ๆ ก็คือโปรแกรมที่วาดไดอะแกรม

- กำหนด use case
 - o Use case diagram (ตัวอย่างแผนภาพ)
 - o Use case scenarios (ตัวอย่างสถานการณ์)
- สร้าง UML
- พัฒนา Class
- วาด statechart (แผนภาพลำดับการทำการกิจกรรม)
- ปรับปรุง UML
- พัฒนาและจัดเก็บ

เลือก Method (เลือกรูปแบบ)

- SDLC
- Agile
- OO methodologies

ใช้ SDLC เมื่อ ?

- ระบบมีการพัฒนาและจัดเก็บโดยใช้ SDLC
- ทุกขั้นตอนมีความสำคัญ
- การจัดการที่สูงขึ้น รู้สึกได้มากกว่าความสะดวกและปลอดภัย
- ทรัพยากรและเวลาเพียงพอ
- การสื่อสารวิธีการทำงานของระบบใหม่

ใช้ Agile เมื่อ ?

- มีโครงการขององค์กรที่สนับสนุน Agile
- ต้องการพัฒนาแอปพลิเคชันได้อย่างรวดเร็วในสภาพแวดล้อมที่ยืดหยุ่น
- คุ้นเคยกับที่ผิดพลาด มีเวลาคิดว่าควรทำอะไร

- ลูกค้าพอใจในการปรับปรุง
- ผู้บริหารระดับสูงและนักวิเคราะห์กับหลักการ

ใช้ OO methodologies เมื่อ ?

- โมเดลมีปัญหา
- องค์กรมีการสนับสนุน
- ระบบค่อยๆเพิ่มระบบย่อย
- การนำซอฟต์แวร์ที่เคยเขียนไว้นำมาใช้ใหม่
- เป็นที่ยอมรับให้การรับมือกับปัญหา

Open Source software (ทรัพยากรแบบเปิด)

- ทางเลือกในการพัฒนาซอฟต์แวร์ดั้งเดิม ที่มีรหัสของผู้ใช้ซ่อนอยู่
- ไม่มีค่าใช้จ่าย แบ่งปันและปรับปรุงได้
- มีมากกว่าการสร้างซอฟต์แวร์ใหม่
- เช่น Linux Operating System , Apache Web Server , Mozilla Firefox

ประเภทของ Open source communities 4 อย่าง

- เฉพาะด้าน
- มาตรฐาน
- องค์กร
- พาณิชย

6 ส่วนที่สำคัญที่ต่างกันของ open source communities

- โครงสร้างทั่วไป
- สภาพแวดล้อม
- เป้าหมาย
- วิธีการ
- กลุ่มผู้ใช้
- การอนุญาต

เหตุผลที่เข้าใช้ open source communities

- พัฒนาและทดสอบได้รวดเร็ว
- เร็วกว่ามีผู้เชี่ยวชาญ
- ส่งเสริมความคิดสร้างสรรค์

- มีการประยุกต์
- ลดต้นทุน
- หนุนภาพลักษณ์
- มีการร่วมมือในการพัฒนา

ผลงานและความแตกต่างของ open source

- มีส่วนร่วมอย่างเปิดกว้างและแตกต่างด้วยเหตุผล ดังนี้
 - o ค่าใช้จ่าย
 - o การจัดการทรัพยากร
 - o เวลาในการปล่อยผลผลิตสู่ตลาด

เหตุผลในการมีส่วนร่วมของนักวิเคราะห์

- อยากรู้ อยากเห็นถึงประโยชน์
- เกิดการออกแบบโดยรวม ได้เป็น
 - o ผลผลิตที่เป็นกรรมสิทธิ์ (มีเจ้าของ)
 - o กระบวนการ
 - o องค์ความรู้
 - o สิ่งประดิษฐ์ด้านไอที

การออกแบบโดยรวม

- ผ่านมาออกแบบ โดยรวมสิ่งประดิษฐ์ไอทีเข้ากับ
 - o ชุมชนและโครงสร้างองค์กร
 - o องค์ความรู้
 - o การใช้งาน

สรุป

- ข้อมูลคือทรัพยากรสำคัญ
- ความถูกต้องของระบบเดิม ด้วยเทคโนโลยีใหม่
- หน้าที่และคุณภาพของนักวิเคราะห์
- พัฒนาระบบเป็นแบบวงจร
- เครื่องมือตัวอย่าง
- พัฒนาระบบ Agile
- พัฒนาระบบ O-O
- ระบบทรัพยากรแบบเปิด