บทที่ 2
สรุปสาระสำคัญจากเอกสารที่เกี่ยวข้อง

ในการพัฒนาระบบจัดการความรู้สำหรับโครงการศูนย์พัฒนาเด็กคริสตจักรเวียงทอง ได้ทำการศึกษาทฤษฎีและหลักการต่างๆที่สามารถนำมาประยุกต์เข้ากับงานได้ โดยแบ่งออกเป็นหัวข้อต่างๆ ได้ดังต่อไปนี้

1. การจัดการองค์ความรู้
2. ระบบค่าตอบแทน
3. Balanced Score Card
4. Software configuration Management
5. Version Control
6. Double Buffer ในคอมพิวเตอร์กราฟิก
7. วงจรชีวิตการพัฒนาซอฟต์แวร์

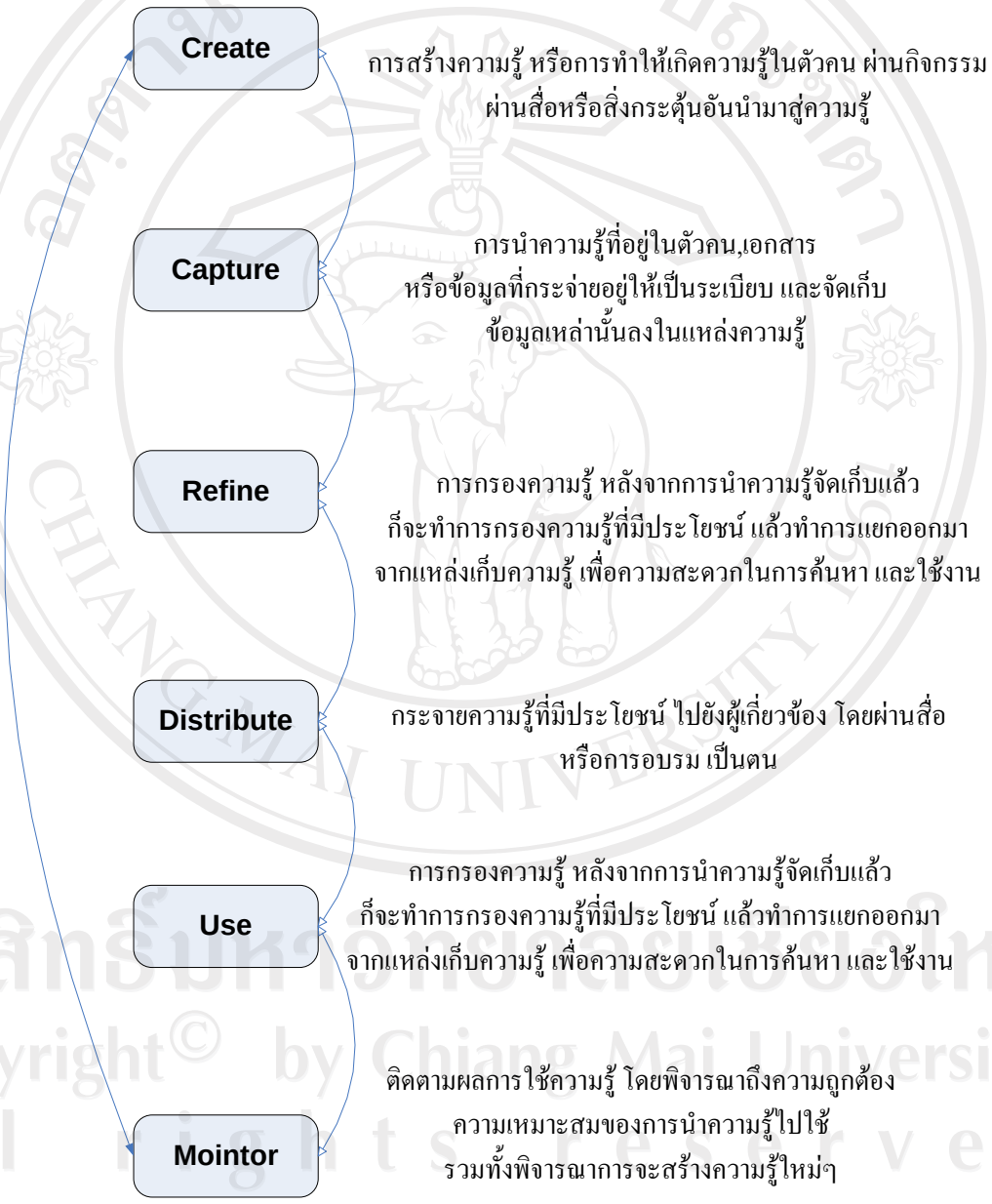
2.1 การจัดการองค์ความรู้ (Knowledge Management)

2.1.1 ความรู้ที่ชัดแจ้ง (Explicit Knowledge) ซึ่งเป็นความรู้ที่สามารถสังเกตเห็นได้ง่ายและสามารถจะจัดเก็บเป็นข้อมูล เช่น เอกสารข้อมูลของเด็กที่เข้าร่วมโครงการ, ข้อมูลการเข้าร่วมกิจกรรมของเด็กแต่ละคนเป็นต้น ซึ่งหากเก็บความรู้ประเภทนี้ลงคลังข้อมูลแล้ว (Knowledge Repository) จะทำให้พนักงานในองค์กรสามารถที่จะนำความรู้เหล่านี้มาใช้ หรือประยุกต์ใช้งานจากเอกสารเหล่านี้ได้ เจนเนตร มณีนาค (2546)

2.1.2 ความรู้ที่ฝังอยู่ในตัวคน (Tacit Knowledge) เป็นความรู้ในแบบที่ซ่อนเร้นอยู่ในสมองของพนักงานแต่ละคน ความรู้ประเภทนี้มักเป็นความรู้ที่ได้มาจากประสบการณ์ของผู้เชี่ยวชาญ และถือว่าเป็นส่วนที่ทรงคุณค่าที่สุดของความรอบรู้ ซึ่งมาจากตัวผู้เชี่ยวชาญ การนำมาซึ่งความรู้ประเภทนี้ จะต้องได้รับการถ่ายทอดผ่านผู้เชี่ยวชาญ เช่นจากการสัมภาษณ์ การลงไปฝึกปฏิบัติ ภายใต้การชี้แนะ (ของผู้เชี่ยวชาญ) หรือ โดยผ่านขั้นตอนที่เรียกกว่าการจับความรู้ (Knowledge Capture) ซึ่งในขั้นตอนนี้ วิศวกรองค์ความรู้ (Knowledge Engineer) จะเป็นผู้เก็บข้อมูลจาก ผู้เชี่ยวชาญ แล้วนำมาวิเคราะห์ให้ได้สาระขององค์ความรู้ ซึ่งท้ายสุดความรู้ต่างๆ จะได้ออกมาในรูปแบบของผังความรู้ (Knowledge Model) Morten T.Hensen (2548)

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.1.3 แนวคิด การจัดการความรู้ของ Turban ประกอบด้วย การสร้าง (Create) การจัดและเก็บ (Capture and store) การเลือกหรือกรอง (Refine) การกระจาย (Distribute) และการใช้ (Use) ซึ่งอาจสลับลำดับของกิจกรรมไปมา หรือเรียงลำดับการทำงานก็ได้ตามสถานการณ์ที่เกิดขึ้น ซึ่งเราสามารถอธิบายขั้นตอนการจัดการความรู้ของ Turban ภราดร จินดาวงศ์ (2549) ได้ดังนี้



รูป 2.1 แสดงการจัดการองค์ความรู้ของ Turban

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.1.4 Knowledge Repository

คือระบบฐานข้อมูล ที่ใช้จัดเก็บข้อมูลความรู้และเอกสาร ให้เป็นระบบหมวดหมู่ ง่ายต่อการค้นหา ง่ายต่อการใช้งาน การนำความรู้มาจัดทำเป็นคลังความรู้ จะทำให้เป็นแหล่งข้อมูลที่พนักงานในองค์กรสามารถใช้ค้นคว้าหาความรู้ได้อย่างไม่จำกัด ทั้งนี้ในคลังความรู้จะมีระบบค้นหา (Search System) ซึ่งเป็นเครื่องมือที่ให้ความสะดวกในการค้นหาข้อมูล และเอกสารต่างๆ

2.2 ระบบค่าตอบแทน (Reward System)

กิ่งพร ทองใบ (2545) บุคลากรต่างต้องการผลตอบแทนจูงใจ สำหรับการมีส่วนร่วมในกระบวนการแบ่งปันความรู้ ระบบค่าตอบแทน ค่าตอบแทนสามารถแบ่งออกเป็น ค่าตอบแทนที่เป็นตัวเงินเป็นค่าตอบแทนที่จ่ายเป็นเงินทางตรงและค่าตอบแทนทางอ้อม เช่น เงินเดือน, โบนัส หรือสวัสดิการ และ ค่าตอบแทนที่ไม่เป็นตัวเงิน ซึ่งอาจเป็นสิ่งของ สิทธิประโยชน์ที่นายจ้างจัดให้แก่พนักงาน ค่าตอบแทนชนิดนี้อาจไม่ใช่ตัวเงินสิ่งของแต่มีคุณค่าทางจิตใจ เช่นรู้สึกว่าตนมีค่าต่อองค์กร ความรู้สึกประสบความสำเร็จในหน้าที่การงานเป็นต้น องค์กรเองควรสร้างแรงจูงใจที่ชัดเจนเช่นเมื่อพนักงานมีการแบ่งปันความรู้ร่วมกับหน่วยงานอื่นนั้น องค์กรควรมีรางวัลจูงใจสำหรับการแบ่งปันความรู้นั้นๆ ของพนักงานเป็นต้น

ความภูมิใจในองค์กร การที่พนักงานมีความรู้สึกภูมิใจที่ได้ทำงานในองค์กรอาจสามารถเรียกได้ว่าเป็นค่าตอบแทนได้ เพราะการที่องค์กรมีชื่อเสียงหรือมีจุดเด่นจะทำให้พนักงานมีความรู้สึกรู้ภูมิใจที่ตนเป็นคนที่มีค่าที่ได้มีส่วนในการทำให้องค์กรมีชื่อเสียง ดังนั้นองค์กรหรือหน่วยงานเองจะต้องหมั่นพูดให้พนักงานมีความรู้สึกภูมิใจที่ได้ทำงานในองค์กรนั้นๆ

2.3 Balanced Score card

Nils-Goran Olve (2546) Balanced Score card มาจากแนวความคิดในการเชื่อมโยงการควบคุมการดำเนินงาน ระยะสั้น เข้ากับวิสัยทัศน์และกลยุทธ์ในระยะยาวขององค์กร โดยเป็นเครื่องมือที่ช่วยให้ผู้บริหารเห็นถึงการปฏิบัติงานในหลายๆ ด้านพร้อมๆ กัน จาก 4 มุมมองคือ มุมมองลูกค้า, มุมมองภายใน, มุมมองด้านนวัตกรรมและการเรียนรู้, มุมมองด้านการเงิน และสามารถใช้เป็นเครื่องมือวัดประสิทธิภาพขององค์กรได้ ซึ่งการจัดการองค์ความรู้จะมีส่วนช่วยพัฒนาการเรียนรู้ขององค์กรได้ดีขึ้น วัตถุประสงค์หลักอีกอย่างหนึ่ง คือการพัฒนาองค์กรให้เป็นองค์กรแห่งการเรียนรู้ เพื่อให้คงไว้ซึ่งความสามารถของการแข่งขันในอนาคต

ปัจจุบันองค์กรต่างๆให้ความสำคัญในการวัดผลสำเร็จ และการประกันคุณภาพการดำเนินงานมากขึ้น ดังนั้นจึงต้องมีการกำหนดดัชนีวัดคุณภาพและวัดความสำเร็จของการดำเนินงานโดยมีวัตถุประสงค์เพื่อใช้เป็นเครื่องมือช่วยให้ผู้บริหารได้ทราบถึงผลการดำเนินงาน และใช้เป็น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

ตัวช่วยสำหรับการตัดสินใจทั้งในระดับกลยุทธ์และระดับปฏิบัติการ ดัชนีจะทำหน้าที่เป็นตัวบ่งชี้ลักษณะหรือองค์ประกอบหลักของการดำเนินงานภายในองค์กรว่า ประสบความสำเร็จมากน้อยเพียงใดนอกจากนี้ยังสามารถใช้เพื่อเปรียบเทียบผลการดำเนินงานกับองค์กรอื่นๆ หรือในองค์กรเดียวกันแต่ต่างช่วงระยะเวลากันไป

Balanced Scorecard (BJC) พัฒนาขึ้นโดย Prof. Robert S. Kaplan และ Dr. David P. Norton แห่ง Harvard Business School เครื่องมือนี้ได้รับการออกแบบขึ้นในปี ค.ศ. 1996 (Kaplan และ Norton, 1996) เพื่อพัฒนาปรับปรุงระบบการประเมินผลการดำเนินงานขององค์กรธุรกิจ และต่อมาได้รับการนำไปใช้ในองค์กรทั้งที่เน้นกำไรและไม่หวังผลกำไรอย่างแพร่หลาย

การนำ Balanced Scorecard มาใช้ในระบบคุณภาพของหน่วยงานจะเป็นกลไกช่วยให้สามารถควบคุมการทำงานให้เป็นไปตามแผนเชิงกลยุทธ์ที่วางไว้ เพื่อบรรลุตามวัตถุประสงค์และพันธกิจขององค์กร โดยมีดัชนีวัดผลสำเร็จของการดำเนินงานที่วัดผลในแง่มุมต่างๆ อย่างสมดุลของมุมมองด้านการเงินและด้านที่ไม่ใช่การเงินที่จะช่วยให้องค์กรบริหารงาน และการดำเนินงานอย่างมั่นใจได้ว่าจะสามารถบรรลุวัตถุประสงค์เชิงกลยุทธ์ที่ตั้งไว้ ในขณะเดียวกันก็จะใช้เป็นเครื่องพิสูจน์ถึงความสามารถขององค์กรในการปฏิบัติภารกิจว่าเป็นไปอย่างมีคุณภาพ และบรรลุความคาดหวังของสังคมได้ในระดับใด และเพื่อวัดผลสำเร็จการนำองค์กรไปสู่การดำเนินงานอย่างมีประสิทธิภาพมากขึ้น

Balanced Scorecard เป็นเครื่องมือซึ่งเกิดจากแนวความคิดที่ช่วยให้องค์กร ในการแปลจากกลยุทธ์ให้เป็นการปฏิบัติ โดยเริ่มต้นที่วิสัยทัศน์ ภารกิจ และกลยุทธ์ขององค์กรซึ่งเป็นขั้นของการกำหนดปัจจัยสำคัญต่อความสำเร็จ และเป็นการสร้างดัชนีวัดผลสำเร็จ (Key Performance Indicators) ขึ้นเพื่อเป็นตัวบ่งชี้ถึงเป้าหมายและใช้วัดผลการดำเนินงานในส่วนที่สำคัญต่อกลยุทธ์ จึงถือได้ว่า Balanced Scorecard เป็นระบบการวัดผลการดำเนินงาน ที่ถ่ายทอดวิสัยทัศน์และกลยุทธ์ขององค์กรสู่การปฏิบัติ และสะท้อนการดำเนินงานในมุมมอง 4 ด้านหลัก คือ ด้านการเงิน ด้านลูกค้า ด้านกระบวนการภายใน และด้านการเรียนรู้และการพัฒนา

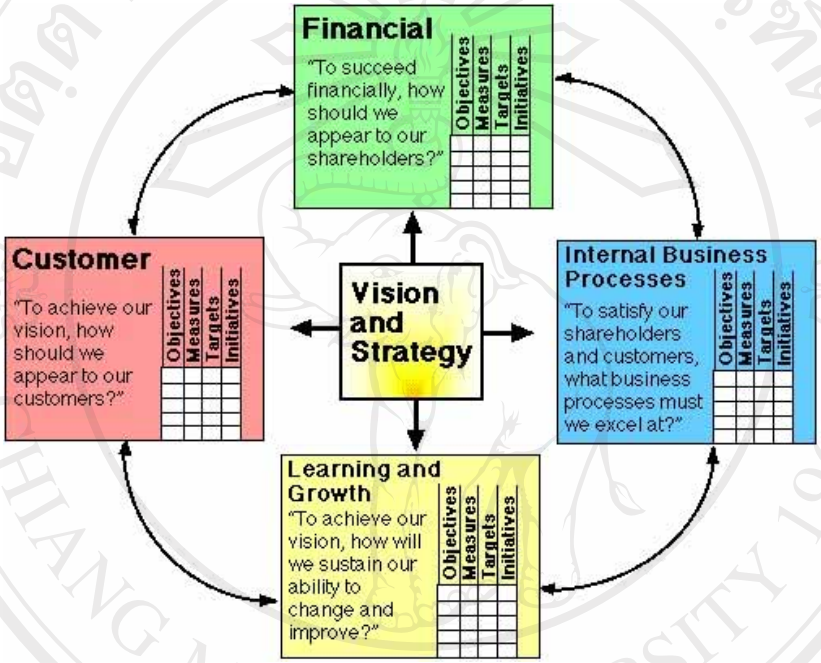
1. **The Learning and Growth Perspective** เป็นมุมมองด้านการเรียนรู้และการเติบโต เช่น การพัฒนาความรู้ความสามารถของพนักงาน, ความพึงพอใจของพนักงาน, การพัฒนาระบบอำนวยความสะดวกในการทำงาน เป็นต้น

2. **The Business Process Perspective** เป็นมุมมองด้านกระบวนการทำงานภายในองค์กรเอง เช่น การคิดค้นนวัตกรรมใหม่ ๆ, การจัดโครงสร้างองค์กรที่มีประสิทธิภาพ, การประสานงานภายในองค์กร, การจัดการด้านสายงานผลิตที่มีประสิทธิภาพ เป็นต้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

3. **The Customer Perspective** เป็นมุมมองด้านลูกค้า เช่น ความพึงพอของลูกค้า, ภาพลักษณ์, กระบวนการด้านการตลาด, การจัดการด้านลูกค้าสัมพันธ์ เป็นต้น

4. **The Financial Perspective** เป็นมุมมองด้านการเงิน เช่น การเพิ่มรายได้, ประสิทธิภาพในการผลิตที่มีต้นทุนต่ำและมีการสูญเสียระหว่างผลิตน้อย, การหาแหล่งเงินทุนที่มีต้นทุนต่ำ เป็นต้น



รูป 2.2 แสดงความสัมพันธ์ทั้ง 4 มุมมองของ Balanced Score Card

การวัดผลสำเร็จธุรกิจตามที่เคยทำกันมาแต่เดิมนั้น มักมุ่งเน้นที่ข้อมูลทางการเงินและการบัญชีจากภายนอก ซึ่งเป็นวิธีการที่เริ่มไม่ได้รับความนิยมในยุคปัจจุบัน องค์กรต้องการดัชนีวัดความสำเร็จที่ครอบคลุมมากกว่า เพื่อใช้เป็นเครื่องมือในการวางแผนอย่างมีประสิทธิภาพสำหรับองค์กรในยุคข้อมูลข่าวสาร

ประโยชน์ที่องค์กรจะได้รับจากการใช้ Balanced Scorecard

- ช่วยทุกคนในองค์กรมองเห็นวิสัยทัศน์ขององค์กรได้ชัดเจน
- ช่วยให้ทุกหน่วยงานในองค์กรปฏิบัติงานได้สอดคล้อง ตามแผนที่วางไว้ยิ่งขึ้น
- สามารถใช้เป็นกรอบในการกำหนดแนวทางการทำงานทั่วทั้งองค์กร
- ช่วยให้มีการจัดแบ่งงบประมาณและทรัพยากรต่าง ๆ สำหรับแต่ละกิจกรรมได้อย่างเหมาะสม

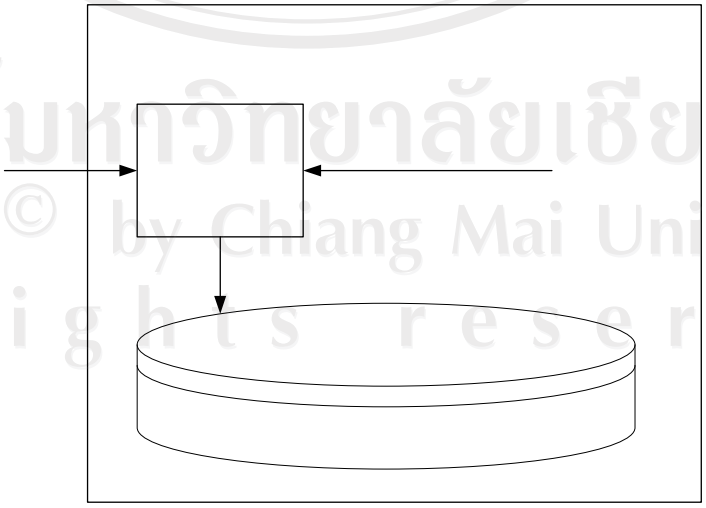
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- เป็นการรวมแผนกลยุทธ์ของทุกหน่วยงานเข้ามาไว้ด้วยกัน ด้วยแผนธุรกิจขององค์กร ทำให้แผนกลยุทธ์ทั้งหมดมีความสอดคล้องกัน
- สามารถวัดผลได้ทั้งลักษณะเป็นทีมและตัวบุคคล

2.4 Software Configuration Management

ในงานวิศวกรรมซอฟต์แวร์กิจกรรมด้าน Software Configuration Management (SCM) เป็นกิจกรรมเกี่ยวกับการควบคุมและบริหารการเปลี่ยนแปลงในขั้นตอนการสร้างซอฟต์แวร์ ภายในกิจกรรม SCM จะประกอบไปด้วยกลุ่มของกิจกรรมต่างๆ ที่มีวัตถุประสงค์สำหรับการควบคุม-บริหารการเปลี่ยนแปลง รายงานและการเชื่อมความสัมพันธ์ภายในกิจกรรมต่างๆเข้าด้วยกัน ทั้งยังเกี่ยวข้องกับการเลือกใช้กลไกการควบคุมต้นฉบับ Version Control สำหรับการผลิตซอฟต์แวร์อีกด้วย เราสามารถกำหนดกิจกรรมต่างๆของการทำ SCM ไว้ดังนี้

1. ขั้นตอนการระบุข้อมูลหรือสิ่งที่มีความสำคัญที่จะต้องเก็บไว้ในคลังข้อมูล Identification โดยขั้นตอนนี้จะต้องระบุว่าเอกสารอะไรที่จะใช้สำหรับเก็บลงคลังข้อมูล(Configuration Item) เช่น รหัส, เอกสารการออกแบบต่างๆ เอกสารการวางแผนหรือแม้กระทั่งสัญญาต่างๆ (เอกสารทั้งหมดที่เกี่ยวข้องกับวิศวกรรมซอฟต์แวร์ในการพัฒนาระบบ) จากนั้นนำเอกสารเหล่านี้เก็บลงคลังข้อมูลเพื่อว่าเอกสารเหล่านี้จะสามารถติดตามและสามารถค้นหาได้จากระบบคลังข้อมูล ทั้งนี้ผู้บริหารจะต้องเลือกใช้เครื่องมือเกี่ยวกับการบริหารต้นฉบับมาใช้สำหรับระบบคลังข้อมูลเช่น SubVersion เป็นต้น เพื่ออำนวยความสะดวกในการบริหาร จัดการข้อมูลต่างๆในคลังข้อมูล ดูการบริหารต้นฉบับในข้อที่ 2.5



รูป 2.3 แสดงขั้นตอนการเก็บข้อมูลที่ระบุไว้ลงในระบบคลังข้อมูล

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2. กำหนด Input โดยระบุว่าเริ่มต้นเอกสารหรือข้อมูลต่างๆจะถูกเก็บไว้ที่ไหนโดยอ้างอิงจากแผนโครงการ Project Plan เพราะจากแผนโครงการจะทำให้รู้ว่าจะต้องใช้ข้อมูล เอกสารหรือชิ้นส่วนรายการไหนในการสร้างซอฟต์แวร์ จากนั้นเราก็จะสามารถระบุได้ว่ารายการเหล่านั้นถูกเก็บไว้ที่ไหน ภายใต้การควบคุมด้วยกิจกรรม SCM และถ้าเกิดเหตุการณ์ที่ต้องมีการเปลี่ยนแปลงจะต้องระบุว่าเริ่มจากส่วนไหน เช่น ถ้าเกิดการเปลี่ยนแปลงจะต้องเริ่มจากการกรอกฟอร์มลงทะเบียนการเปลี่ยนแปลง เป็นต้น
3. กำหนด Output คือระบบผลลัพธ์ของแต่ละรายการที่ได้จากการลงทะเบียน SCM
4. อธิบาย Process ต่างๆ โดยขั้นตอนนี้จะต้องระบุว่าแต่ละกิจกรรมมีขั้นตอนอะไรบ้างและมีวิธีการอย่างไรเช่น
    a. กำหนดกิจกรรมบริหารการเปลี่ยนแปลง ต้องระบบว่ามีขั้นตอนอะไรบ้างตั้งแต่ ลงทะเบียนการเปลี่ยนแปลงไปจนถึง รายงานผลการเปลี่ยนแปลง
    b. กำหนดกิจกรรมการสร้าง Configuration Item ว่าต้องทำอะไรบ้าง
    c. กำหนดกิจกรรมการออกผลิตภัณฑ์ว่ามีขั้นตอนอะไรบ้าง
    d. กำหนดข้อตกลงการสร้างรหัสอ้างอิง (ข้อ5)

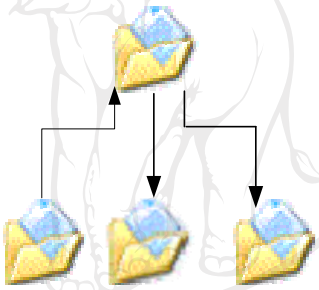
5. ระบบรหัสที่จะใช้อ้างอิงสำหรับแต่แฟ้มหรือข้อมูลต่างๆในคลังข้อมูล (Unique Identification) เช่น การตั้งรหัสสำหรับแต่ละรายการ, การกำหนดกฎการตั้งชื่อ (Naming Conversion) ไปจนถึงการระบบรายชื่อการสร้างแฟ้ม
6. กำหนดหน้าที่และผู้มีสิทธิ์ (Authorization) การกำหนดสิทธิ์จะอ้างอิงตามเอกสารแผนประกันคุณภาพ (Quality assurance Plan) โดยกำหนดว่าใครมีสิทธิ์อะไรและมีหน้าที่รับผิดชอบอะไรบ้าง
7. กำหนดกฎ (Roles) เช่นกฎการตั้งชื่อแฟ้มข้อมูล ในกรณีที่ต้องการสร้างแฟ้มใหม่, กฎการใช้งาน Tool ต่างๆในการทำ SCM เป็นต้น
8. เชื่อมความสัมพันธ์ของแต่ละกิจกรรมที่สัมพันธ์กันเข้าด้วยกัน งานอีกอย่างหนึ่งของการทำ SCM คือการระบุว่ากิจกรรมไหนสัมพันธ์กัน และติดตามว่าเอกสารหรือรายการต่างๆในกิจกรรมนั้นๆ ได้เชื่อมต่อกับกิจกรรมอื่นหรือไม่ เช่น เมื่อต้องการแก้ไขข้อผิดพลาดของระบบซึ่งเริ่มจากกิจกรรมบริหารการเปลี่ยนแปลง จากนั้นก็จะเชื่อมไปยังกิจกรรมการพัฒนาระบบและสุดท้ายลงมาที่รายงานผลการเปลี่ยนแปลง ซึ่งสิ่งเหล่านี้จะเป็นงานของกิจกรรม SCM

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.5 Version Control

คือการบริหารแฟ้มข้อมูลที่เป็นเรื่องเดียวกัน และที่มีข้อมูลที่แตกต่างกันโดยการแยกออกเป็นฉบับ ซึ่งเป็นเทคนิคที่นิยมใช้กันในกลุ่มผู้พัฒนาซอฟแวร์เพื่อใช้ในการบริหารแฟ้มรหัสหรือ แฟ้มข้อมูลดิจิตอลต่างๆ การระบุหมายเลขฉบับ(Version Number) นั้นมี 2 ประเภท คือ ถ้าเป็นแฟ้มส่วนตัวของผู้แก้ไขจะเรียกว่าหมายเลขฉบับปรับปรุง (Revision) ส่วนถ้าแฟ้มที่แก้ไขระบุให้เป็นสาธารณะจะเรียกว่าฉบับจริง (Version)

Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato (2008) ได้อธิบายการทำงานของการบริหาต้นฉบับโดยเริ่มจากการนำข้อมูลต่างๆมาเก็บไว้ในคลังข้อมูลกลาง (Repository) ซึ่งในคลังข้อมูลนี้จะเก็บแฟ้มข้อมูลโครงสร้างแบบต้นไม้ (File System Tree) ในลักษณะของ Directory และทุกครั้งที่มีคนเขียนหรืออ่านข้อมูลลงในแฟ้มข้อมูลการเขียนข้อมูลหรือการอ่านข้อมูลจะถูกบันทึกไว้ในคลังข้อมูลดังรูปที่ 2.4

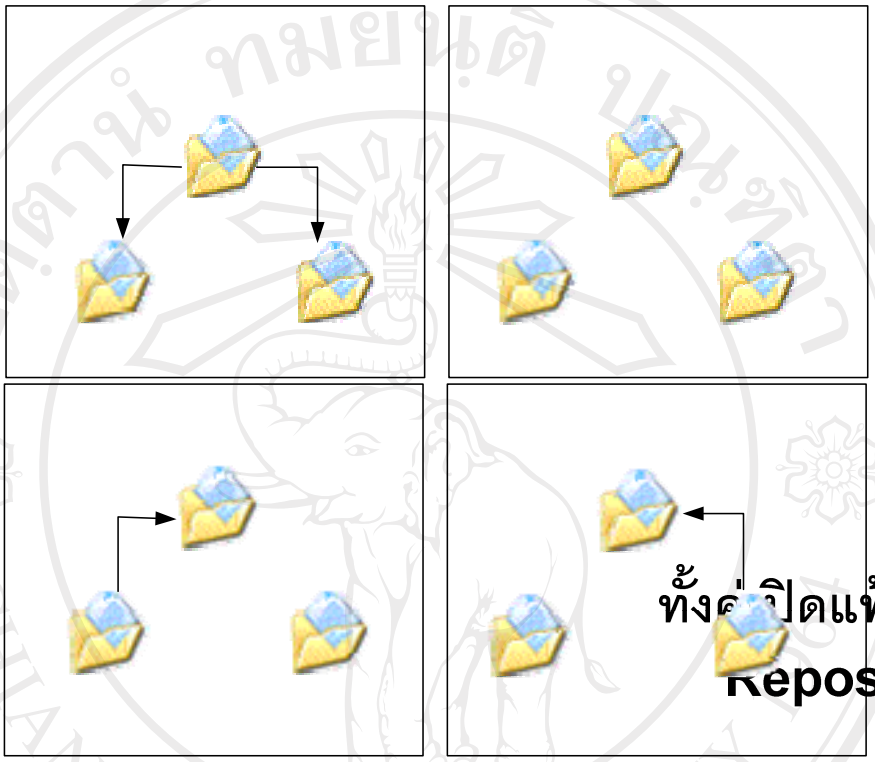


รูป 2.4 แสดงการเขียนและอ่านข้อมูลในคลังข้อมูล

ดังนั้นทุกครั้งที่มีการแก้ไขข้อมูลในแฟ้มข้อมูลข้อมูลเหล่านั้นจะถูกบันทึกไว้ เมื่อผู้ใช้งานต้องการอ่านแฟ้มข้อมูลในคลังข้อมูล ผู้ใช้สามารถเปิดข้อมูลเก่าของแฟ้มนั้นๆขึ้นมาดูได้ ทั้งยังสามารถเปรียบเทียบการเปลี่ยนแปลงของข้อมูลในแฟ้มข้อมูลนั้นๆได้อีกด้วย สิ่งเหล่านี้ถือเป็นหัวใจหลักของการบริหารต้นฉบับ วัตถุประสงค์หลักของการบริหารต้นฉบับคือการให้ผู้ใช้สามารถแก้ไขแฟ้มข้อมูลร่วมกัน ซึ่งปกติแล้วการใช้ข้อมูลร่วมกัน Share Data มักเกิดปัญหาการเขียนข้อมูลทับกับข้อมูลซึ่งแสดงในรูปที่ 2.5 ดังนี้ สมมติว่ามีพนักงาน 2 คนคือ Harry และ Sally ทั้งคู่ต้องการที่จะแก้ไขแฟ้มข้อมูลเดียวกันในคลังข้อมูลเป็นการแก้ไขข้อมูลพร้อมๆกัน ถ้า Harry บันทึกข้อมูลที่ตัวเองได้แก้ไขลงในคลังก่อน จากนั้นในเวลาไล่เลียกัน Sally ได้แก้ไขข้อมูลของตัวเองทับลงไปในคลังข้อมูล ถ้าหากไม่มีการทำการบริหารต้นฉบับ ข้อมูลของ Harry ก็จะถูกทับด้วยข้อมูลของ Sally แต่ในกรณีที่มีการบริหารต้นฉบับข้อมูลของ Harry จะถูกบันทึกเป็น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

version หนึ่งและของ Sally จะถูกบันทึกเป็น version ล่าสุดซึ่งในกรณีนี้จะเกิดปัญหาคือข้อมูลที่ Harry ได้แก้ไขไปนั้นไม่ได้ถูกใส่เข้าไปใน Version ล่าสุดพร้อมกับของ Sally ซึ่งความเป็นจริงนั้นข้อมูลล่าสุดจะต้องเป็นข้อมูลของทั้ง Harry และ Sally



รูป 2.5 ตัวอย่างการอ่านและเขียนข้อมูลลงคลังข้อมูล

2.5.1 กลไกการ Lock-Modify-Unlock

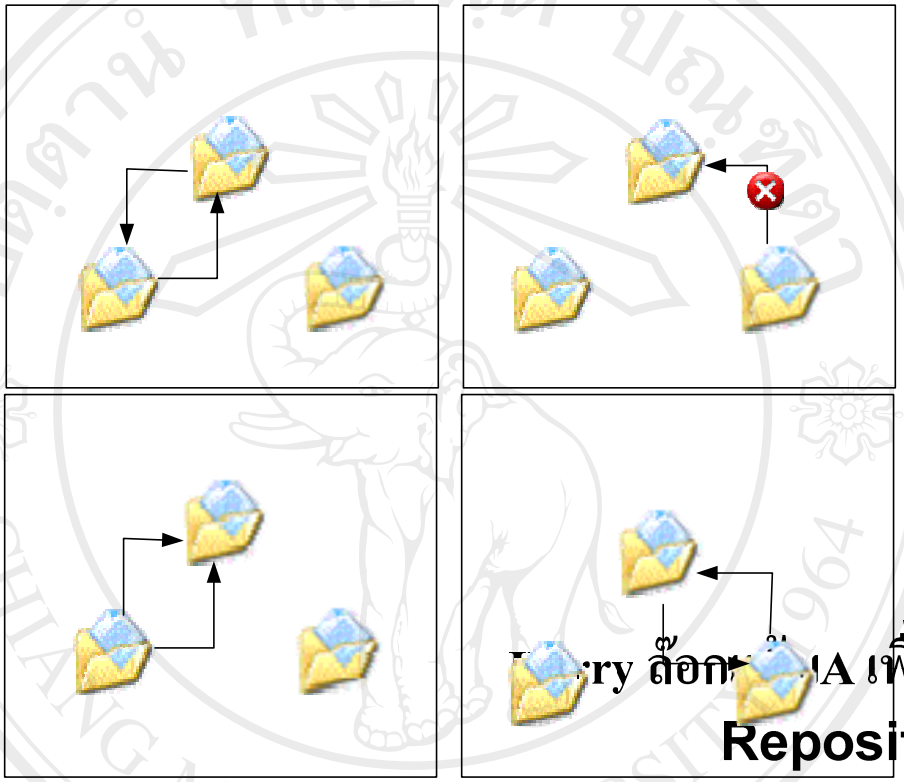
เพื่อแก้ไขปัญหาที่กล่าวข้างต้นจะใช้กลไกการ Lock-Modify-Unlock ซึ่งเราสามารถอธิบายกลไกนี้โดยใช้ตัวอย่างเดิมของ Harry และ Sally ดังรูปที่ 2.6 กล่าวคือเมื่อ Harry ต้องการแก้ไขแฟ้มข้อมูลในคลังเขาจะทำการ Lock แฟ้มข้อมูลนั้นแล้วทำการอ่านข้อมูล เมื่อ Sally ต้องการเข้าถึงข้อมูลที่ Harry ได้ Lock ไว้ก็จะไม่สามารถทำได้ดังนั้นเมื่อ Harry แก้ไขข้อมูลเสร็จแล้วทำการเขียนข้อมูลลงแล้วทำการ Unlock แล้ว Sally จึงจะสามารถทำการแก้ไขข้อมูลต่อไป เมื่อ Sally แก้ไขข้อมูลเสร็จสุดท้ายเราก็จะได้ข้อมูลล่าสุดที่มีข้อมูลของทั้ง Harry และ Sally

จุดด้อยของการใช้กลไก lock-modify-unlock ดังนี้

- จากตัวอย่างถ้าหาก Harry ลืม Unlock จะทำให้ไม่มีใครสามารถเข้าถึงข้อมูลที่ Harry ได้ทำการ Lock เอาไว้ได้ ยิ่งกว่านั้นถ้าหาก Harry ออกไปข้างนอกแล้วลืม ปลดล็อกแฟ้มนั้นก็แสดงว่าวันนั้นทั้งวันจะไม่มีใครสามารถเข้าถึงแฟ้มที่ Harry ล็อกไว้ได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- การ Lock อาจมีผลในด้านความปลอดภัยเช่นถ้าหากมีแฟ้ม 2 แฟ้มที่มีผลต่อกันแล้ว Harry Lock แล้วแก้ไขแฟ้มที่ 1 ขณะที่ Sally ทำการแก้ไขแฟ้มที่ 2 ซึ่งในกรณีนี้ทั้งคู่จะสามารถแก้ไขข้อมูลทั้ง 2 แฟ้มได้แต่เนื่องจาก แฟ้มทั้ง 2 นี้มีความสัมพันธ์กันซึ่งอาจทำให้เกิดปัญหาการสูญเสียข้อมูลสำคัญได้



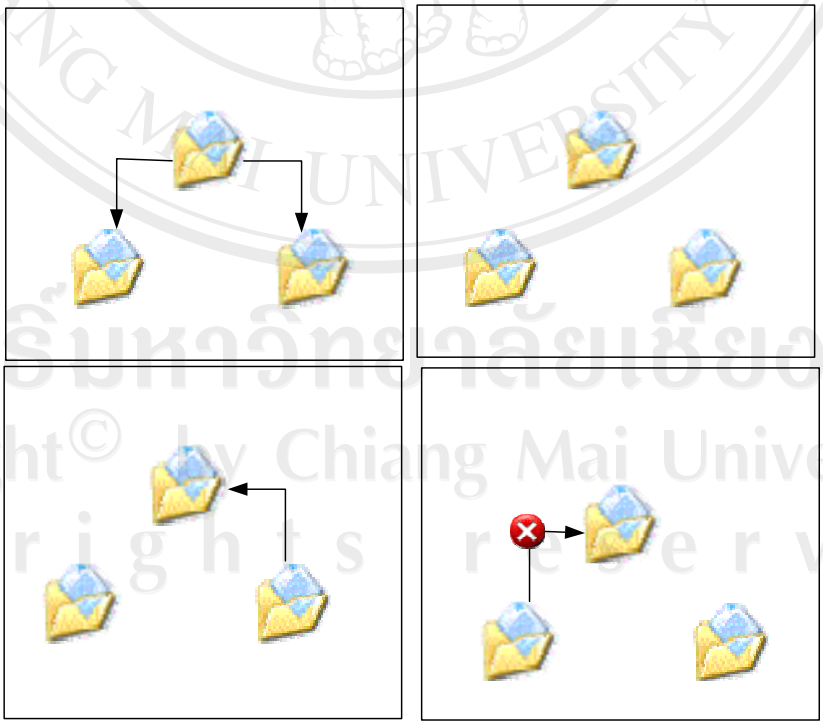
รูป 2.6 แสดงการเขียนและอ่านข้อมูลในคลังข้อมูล

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

2.5.2 กลไกการ Copy-Modify-Merge

เพื่อแก้ไขจุดด้อยของกลไกการ Lock-Modify-Unlock จึงมีคนคิดค้นวิธี Copy-Modify-Merge ในกรณีนี้ผู้แก้ไขข้อมูลสามารถ Copy ข้อมูลออกมาจากคลังเป็นข้อมูลสำหรับแก้ไขของตนเอง(Working Copy) เป็นข้อมูลส่วนตัวจากนั้นผู้แก้ไขจะทำการแก้ไขข้อมูล ซึ่งจะไม่ถูกบันทึกลงในคลังข้อมูล จนกว่าผู้แก้ไขจะทำการรวม (Merge) ข้อมูลที่แก้ไขลงในคลังข้อมูล

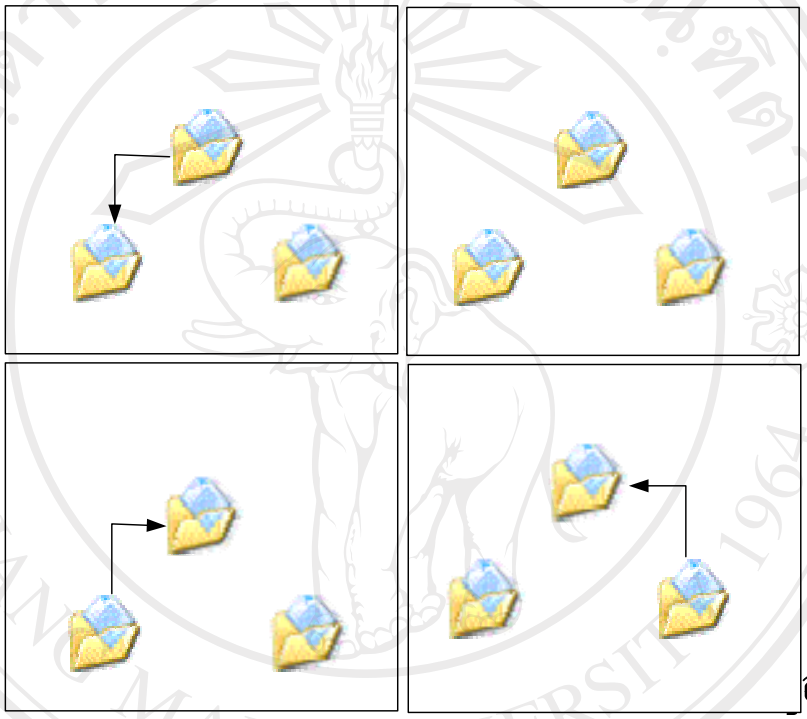
ซึ่งเราสามารถอธิบายกลไกนี้ โดยใช้ตัวอย่างเดิมของ Harry และ Sally กล่าวคือเมื่อ Harry ต้องการแก้ไขแฟ้มข้อมูลในคลังเขาจะทำการคัดลอกแฟ้มข้อมูลจากคลัง แล้วจากนั้นก็ทำการแก้ไขข้อมูลส่วนตัวนั้นในเครื่องตัวเอง ขณะที่ Sally ก็ทำเหมือนกับ Harry คือคัดลอกแฟ้มจากคลังแล้วแก้ไขข้อมูลที่คัดลอกมาในเครื่องของตัวเองเมื่อ Sally แก้ไขข้อมูลเสร็จ ก็ทำการ Update ข้อมูลของตนเองลงคลังข้อมูล จากนั้น Harry ก็ Update ข้อมูลของตัวเองแต่ครั้งนี้ระบบจะแจ้งเตือน Harry ว่าข้อมูลของเขาไม่ใช่ข้อมูลล่าสุดแล้ว (เพราะก่อนหน้านั้นทั้ง Harry และ Sally แก้ไขข้อมูลต้นฉบับเดียวกัน เมื่อระบบตรวจสอบว่าข้อมูลในคลังตรงกับต้นฉบับที่ Harry คัดลอกมาแก้ไขหรือไม่) ดังนั้นแทนที่ระบบจะทำการบันทึกทับข้อมูลที่ Sally เคยแก้ไขไว้ ก็จะถามว่าต้องการรวมข้อมูล (Merge) หรือไม่ ถ้า Harry ต้องการรวมข้อมูลระบบก็จะทำการรวมข้อมูลของ Harry ลงไป ซึ่งก็จะทำให้ได้ข้อมูลทั้งของ Harry และ Sally ลงในแฟ้มที่ทั้งคู่ได้แก้ไข



รูป 2.7 รูปแสดงตัวอย่างการ Copy-Modify-Merge

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

จากตัวอย่างข้างต้นถ้าหากว่าข้อมูลที่ Sally แก้ไขอยู่ในตำแหน่งเดียวกันกับข้อมูลที่ Harry ได้แก้ไขไว้ซึ่งเราเรียกเหตุการณ์นี้ว่า Conflict กลไกนี้สามารถแก้ไขปัญหานี้ได้โดยระบบจะทำการแสดงความแตกต่างของแฟ้มทั้ง 2 แฟ้มคือข้อมูลที่ Sally แก้ไขกับข้อมูลที่ Harry แก้ไขแล้วให้ Harry เป็นคนตัดสินใจว่าจะใช้ข้อมูลของใครจากนั้นค่อยทำการบันทึกลงในคลังข้อมูลดังแสดงในรูป 2.8



ลในคลังกับของตัวเอ

เพื่อทำการ รวมข้อมูล (Merge)

รูป 2.8 รูปแสดงการแก้ไขปัญหาการ Conflict

Repository

ด้วยวิธีนี้จะช่วยแก้ไขจุดด้อยของวิธีการ Lock-Modify-Unlock และยังทำให้เราสามารถแก้ไขแฟ้มข้อมูลในคลังข้อมูลพร้อมๆกัน ทั้งยังสามารถนำแฟ้มในคลังข้อมูลออกมาแก้ไขที่เครื่องของตัวเองได้โดยไม่ยึดติดกับคลังข้อมูลระหว่างการแก้ไขข้อมูลอีกด้วย

Read

2.6 Double Buffer ในคอมพิวเตอร์กราฟิก

A

Harry

A

Sally

บางคนเรียกว่า Ping-Pong Buffering เป็นเทคนิคที่ใช้กันอย่างแพร่หลายในวงการคอมพิวเตอร์กราฟิก ในการแก้ไขปัญหา การกระพริบของหน้าจอที่เกิดขึ้นในขั้นตอนของการวาดรูป อันเนื่องมาจากการใช้คำสั่งการแสดงรูปเร็วเกินไป เร็วกว่า Refresh Rate ของจอภาพจึงทำให้เกิดการกระพริบของจอภาพ ซึ่งโดยปกติแล้วจอภาพ Monitor จะสามารถวาดภาพประมาณ 60

Harry บันทึกข้อมูลต้นฉบับใหม่ของท้

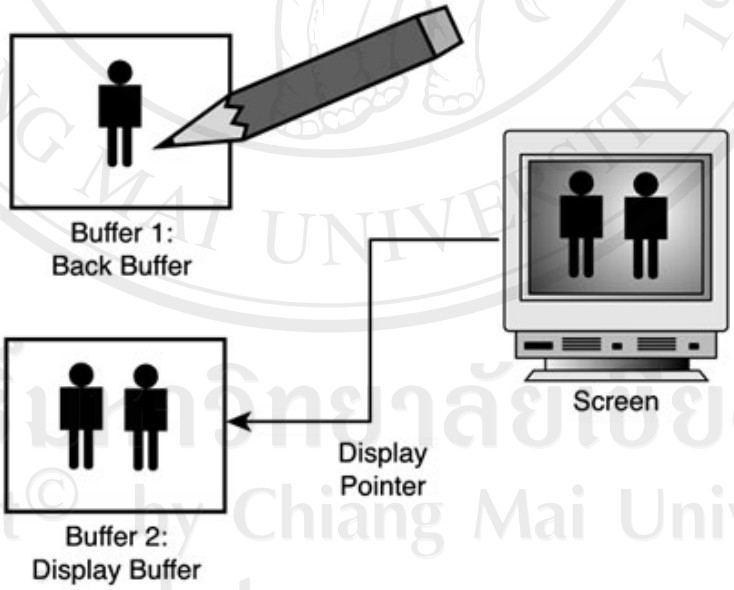
Repository

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

จอภาพต่อ 1 วินาทีโดยประมาณ ดังนั้นการใช้คำสั่งวาดภาพก่อนที่จอภาพ จะแสดงรูปเสร็จนั้นจะทำให้เกิดการกระพริบของหน้าจอ หรือทำให้เกิดภาพเละ (ugly artifacts)

การใช้ซอฟต์แวร์ในการแก้ไขปัญหานี้ โดยการใช้เนื้อที่หน่วยความหลัก (RAM) ในการเก็บข้อมูลที่จะใช้ในการวาดลงในจอภาพ แทนที่จะเป็นหน่วยความจำของจอภาพเอง (VRAM) ด้วย เหตุนี้เราจึงสามารถใช้หน่วยความจำหลักเปรียบเสมือนหน่วยความจำของจอภาพ กล่าวคือ เราสามารถเขียนข้อมูลที่ต้องการแสดงผล ลงในหน่วยความจำก่อนจากนั้นจึงทำการคัดลอกข้อมูลลงในหน่วยความจำของจอภาพ ซึ่งจะช่วยแก้ไขปัญหาการกระพริบและการเละของภาพได้ แต่ว่าการใช้ ซอฟต์แวร์ในการแก้ไขปัญหาจะทำให้หน่วยประมวลผลกลาง (CPU) ทำงานเยอะขึ้น

การใช้ฮาร์ดแวร์ในการแก้ไขปัญหานี้โดยการแบ่งหน่วยความจำของจอภาพ ออกเป็น 2 จอภาพ จอหนึ่งไว้สำหรับการแสดงผล (Display Buffer) ส่วนอีกเจอไว้สำหรับวาดข้อมูลลง (Back Buffer) เมื่อวาดเสร็จก็จะใช้คำสั่ง Page Flip นำจอที่วาดเสร็จนี้ออกแสดงออกทางจอภาพ ทำแบบนี้สลับไปมา เพราะฉะนั้นทุกครั้งที่จอหนึ่ง(หน่วยความจำ) แสดงผล(Display Buffer) อีก 1 จอ (หน่วยความจำ) ก็จะมีไว้สำหรับวาดข้อมูลลง(Back Buffer) เมื่อวาดเสร็จก่อนค่อยสั่งให้แสดงข้อมูลที่วาดนั้นแทน



รูป 2.9 แสดงความสัมพันธ์การทำ Double Buffer

การทำ Double Buffer ใน Java Applet นั้นเราสามารถจองหน่วยความจำสำหรับการวาด (Back Buffer) แล้วทำการเขียนข้อมูลที่ต้องการแสดงลงในหน่วยความจำที่จะแสดง จากนั้นก็สั่งแสดงผลข้อมูลของหน่วยความจำที่จองนั้น (Back Buffer) โดยมีขั้นตอนดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- สร้าง Back Buffer โดยเริ่มจากการสร้างตัวแปรแบบ Image ดังนี้
  - Image img;
- ด้วยจากนั้นใช้คำสั่ง createImage(int w, int h) เพื่อจองหน่วยความจำลงในตัวแปรที่สร้างไว้
  - img = createImage(800, 600)
- จากนั้นทำการสร้างตัวแปร Back Buffer
  - Graphics back_buffer;
- สั่งให้ Back Buffer ใช้ไปยังตำแหน่งของ หน่วยความจำที่เราจองไว้ ดังนี้
  - back_buffer =(Graphics2D)bufferImage.getGraphics();
- จากนั้นก็จะสามารถใช้ back_buffer ในการวาดภาพต่างๆแทนที่จะวาดจาก event paint() ของ Applet
- เมื่อวาดข้อมูลต่างๆลงใน Back Buffer ตามที่ต้องการแล้ว และถ้าเราต้องการให้แสดงข้อมูลใน Back Buffer ที่สร้างขึ้น โดยเราอาจสามารถวาดข้อมูลใน Back Buffer ใน Event paint ได้ เช่น

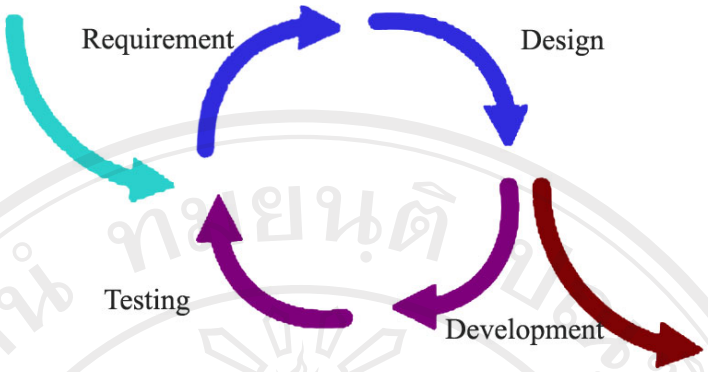
```
public void paint(Graphics g){
        g.drawImage(back_buffer,0,0,null);
}
```

การใช้เทคนิคของ Double Buffer จะสามารถแก้ไขปัญหาการกระพริบของจอภาพในขั้นตอนการวาดรูปเพื่อแสดงผลทางจอภาพ การค้นคว้าอิสระนี้ได้นำเทคนิค Double Buffer มาใช้สำหรับการแก้ไขปัญหาการกระพริบของจอภาพ ในส่วนของการวาดผังองค์ความรู้

2.7 วงจรชีวิตการพัฒนาซอฟต์แวร์

วงจรชีวิตการพัฒนาซอฟต์แวร์ Incremental Model เป็นวงจรชีวิตการพัฒนาที่แบ่งระบบใหญ่ออกเป็นระบบย่อย จากนั้นทำการพัฒนาทีละระบบย่อย จากนั้นนำแต่ละระบบย่อยมาประกอบรวมกัน (Integrate) ทำแบบนี้จนพัฒนาระบบเสร็จสิ้น โดยลักษณะวงจรชีวิตจะคล้ายกับ Waterfall Model และ Spiral Model รวมกัน โดยแบ่งระบบย่อยออกจากระบบใหญ่ แล้วทำการพัฒนาระบบย่อยที่ละระบบตาม Waterfall Model ดังรูป 2.10

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved



รูป 2.10 แสดงวงจรชีวิตการพัฒนาซอฟต์แวร์แบบ Incremental

วงจรการพัฒนาอาจสามารถพัฒนาเริ่มจากการแยกความต้องการของระบบออก โดยเริ่มพัฒนาจุดที่ง่ายสุดก่อนโดยพัฒนาให้ครบวงจรชีวิต จากนั้นค่อยๆพัฒนาจนครบความต้องการของผู้ใช้ (Full version) โดยทุกครั้งที่พัฒนาครบวงจรชีวิตหนึ่ง ก็จะได้ความสามารถของระบบเพิ่มขึ้นทำเช่นนี้จนความสามารถของระบบครบตามความต้องการของผู้ใช้

ประโยชน์จากการพัฒนาซอฟต์แวร์ โดยใช้วงจรชีวิตการพัฒนาซอฟต์แวร์แบบ Incremental คือผู้จ้างสามารถเห็นความก้าวหน้าของการพัฒนาระบบได้ การส่งมอบงานสามารถทำได้ง่ายขึ้น เพราะผู้จ้างมีส่วนในการติดตามความก้าวหน้าของระบบอยู่เสมอ และระบบที่พัฒนาจะสามารถพัฒนาได้ตรงตามความต้องการของผู้ใช้ที่มีการเปลี่ยนแปลงอยู่เสมอได้ดียิ่งขึ้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved