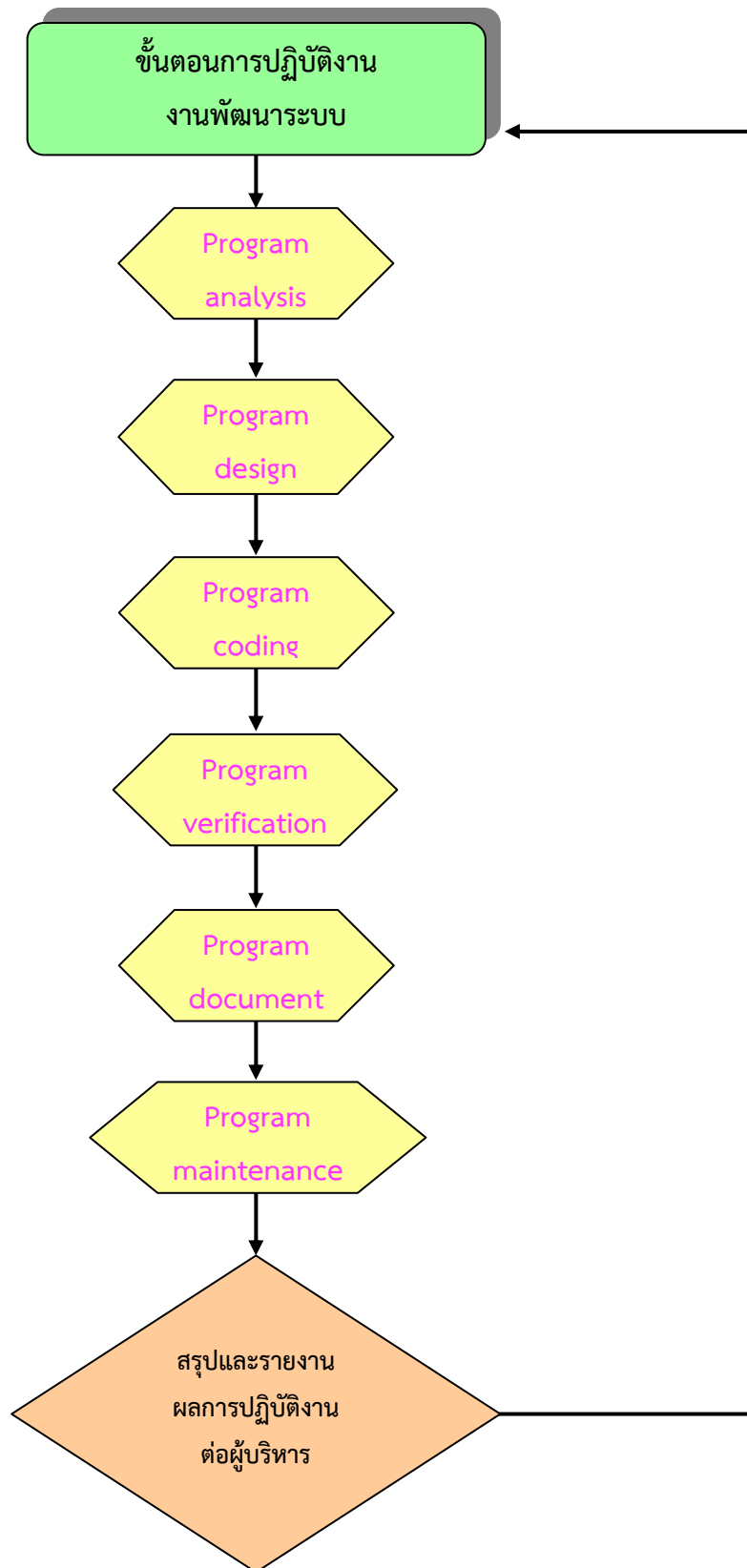


แผนปฏิบัติงานประจำปีงบประมาณ 2554 งานพัฒนาระบบสารสนเทศ สำนักศึกษาทั่วไป

ลำดับ	กิจกรรม/งานหลัก	ปี พ.ศ.2553			ปี พ.ศ. 2554								ผู้รับผิดชอบ	หมายเหตุ	
		ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.			ก.ย.
	1. งานพัฒนาระบบสารสนเทศ (ระบบใหม่)														
1	สำรวจความต้องการเทคโนโลยีสารสนเทศเข้ามาช่วยพัฒนางานของบุคลากรทุกๆ เดือน				←								→	สุลีสพร	
2	วิเคราะห์ถึงความเป็นไปได้ของระบบใหม่ที่มีการแจ้งความต้องการมา				←								→	สุลีสพร	
3	ออกแบบและพัฒนาระบบตามความต้องการที่ได้วิเคราะห์ถึงความเป็นไปได้แล้ว				←								→	สุลีสพร	
4	ทดสอบระบบที่พัฒนาเสร็จแล้ว				←								→	สุลีสพร	
5	จัดทำคู่มือระบบที่ทดสอบเรียบร้อยแล้ว				←								→	สุลีสพร	
	2. งานปรับปรุงและพัฒนาระบบสารสนเทศ(ระบบเก่า)													สุลีสพร	
1	สำรวจความต้องการเพิ่มเติมจากผลการประเมินและข้อเสนอแนะจากผู้ใช้ระบบ	←	←	←	←	←	←	←	←	←	←	←	←	สุลีสพร	ปฏิบัติงาน ทุกๆ 2 เดือน
2	วิเคราะห์ถึงความเป็นไปได้ของความต้องการของผู้ใช้	←	←	←	←	←	←	←	←	←	←	←	←	สุลีสพร	

7	รับสมัครนักเรียนเพื่อสอบแข่งขัน โครงการ MSU-Test(2554)											↔		สุสิทธิ์	
---	---	--	--	--	--	--	--	--	--	--	--	---	--	----------	--

แผนภูมิแสดงขั้นตอนการปฏิบัติงาน งานพัฒนาระบบสารสนเทศ สำนักศึกษาทั่วไป



ขั้นตอนการปฏิบัติงานของส่วนงานพัฒนาระบบ

งานพัฒนาระบบเป็นงานที่รับผิดชอบในส่วนการพัฒนาระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอนในรายวิชาศึกษาทั่วไป และเพื่อสนับสนุนการปฏิบัติงานให้สอดคล้องตามภารกิจหลักของหน่วยงาน ซึ่งมีขั้นตอนการพัฒนาระบบดังนี้

ระยะต่าง ๆ ของการเขียนโปรแกรม (The Stages of the Programming Process)

การเขียนโปรแกรมคอมพิวเตอร์ คือ วิธีการในการพัฒนาโปรแกรมคอมพิวเตอร์ซึ่งประกอบด้วยกลุ่มของคำสั่งซึ่งสั่งให้เครื่องคอมพิวเตอร์ประมวลผลหรือกิจกรรมต่างๆที่เกี่ยวข้องของการเขียนคำสั่งในภาษาโปรแกรมต่าง ๆ ซึ่งสามารถแบ่งได้เป็นหลายระยะคือ

1 Program analysis

เป็นระยะของการวิเคราะห์ถึงจุดประสงค์ของงานประยุกต์ โดยกำหนดถึงหน้าที่ต่าง ๆ ที่จะให้โปรแกรมทำงานได้

2 Program design

เป็นระยะของการวางแผนและออกแบบถึงคุณลักษณะของข้อมูลเข้า (input) ข้อมูลออก (output) หน่วยเก็บข้อมูล วิธีดำเนินการประมวลผล

3 Program coding

เป็นระยะของการเขียนคำสั่งภาษาโปรแกรมซึ่งเปลี่ยนจาก Program design เป็นโปรแกรมคอมพิวเตอร์ที่สมบูรณ์

4 Program verification

เป็นระยะของการตรวจทานทดสอบโปรแกรมที่เขียนขึ้นให้ถูกต้องและสมบูรณ์ตรงตามความต้องการของระบบ ซึ่งเรียกว่า debugging และ testing

5 Program documentation

เป็นระยะของการบันทึกรายละเอียดของการออกแบบและรายละเอียดของโปรแกรม โดยจัดทำเป็นคู่มือและเอกสารของระบบ

6 Program maintenance

เป็นระยะของการปรับปรุงหรือสร้างโปรแกรมคอมพิวเตอร์ เพื่อพัฒนาให้ดีขึ้นซึ่งอาจจะขยายขีดความสามารถหรือปรับปรุงให้ถูกต้องยิ่งขึ้น

1. การวิเคราะห์โปรแกรม (Program analysis)

การวิเคราะห์โปรแกรมเป็นขั้นตอนแรกในการเขียนโปรแกรมคอมพิวเตอร์โดยการวิเคราะห์ถึงหน้าที่ต่างๆ ของโปรแกรมโดยแบ่งเป็นงานหรือฟังก์ชันฟังก์ชันหนึ่งอาจปฏิบัติการได้ก็ต่อเมื่อปฏิบัติงานอีกฟังก์ชันหนึ่งเสร็จก่อน การวิเคราะห์ปัญหาต่างๆ ของโปรแกรมอาจเป็นปัญหาสั้นๆ พื้นฐานหรือปัญหาทาง

คณิตศาสตร์ซับซ้อน ซึ่งจะต้องกำหนดปัญหา (Problem Definition) และกำหนดรายละเอียดของปัญหา (Problem Specification) ในการปฏิบัติงานให้ชัดเจน ในกรณีทำงานประยุกต์เป็นงานประมวลผลข้อมูลการวิเคราะห์โปรแกรมควรวเคราะห์ถึงข้อกำหนดรายละเอียดของซอฟต์แวร์ (Software Specification) ในระยะของการออกแบบ (Design Stage) หรือความต้องการในรายละเอียดของโปรแกรม (Program Specification) อย่างเช่น

1. **Output** โดยวิเคราะห์ว่าผลลัพธ์ที่ต้องการคืออะไรบ้าง
2. **Input** โดยวิเคราะห์ว่าข้อมูลที่สามารถเรียกหาได้มีอะไรได้บ้าง
3. **Storage** โดยวิเคราะห์ว่าข้อมูลจะเก็บ (store) หรือดึง (retrieved) หรือแก้ไขในหน่วยเก็บข้อมูลอะไร
4. **Processing** โดยวิเคราะห์ถึงวิธีการประมวลผลต่าง ๆ ซึ่งอาจเป็นการคำนวณทางคณิตศาสตร์ การเปรียบเทียบและกรรมวิธีอื่น ๆ
5. **Control procedure** โดยวิเคราะห์วิธีการควบคุมการทำงานของโปรแกรม

2. การออกแบบโปรแกรม (Program design)

ระยะของการออกแบบโปรแกรมเป็นระยะของการวางแผนและออกแบบโดยระบุคุณลักษณะของข้อมูลเข้า (Input) ข้อมูลออก (Output) กรรมวิธีการประมวลผลกำหนดรายละเอียดของหน่วยเก็บข้อมูลและวิธีการควบคุมซึ่งค่าของความพยายาม (effort) ในการวิเคราะห์และออกแบบโปรแกรมขึ้นอยู่กับความซับซ้อนของงานประยุกต์และจำนวนของงานในระบบโดยปกติจะเป็นการกำหนดกฎเกณฑ์ทางตรรกะและคำสั่งที่ระบุถึงการปฏิบัติงานซึ่งเรียกว่าโมดูล (modules หรือ subdivisions) โดยแต่ละโมดูลจะมีส่วนของการกำหนดค่าเริ่มต้น (initialization) ข้อมูลเข้า (input) ประมวลผล (processing) และส่วนแสดงผล (output) และส่วนของการสิ้นสุดหรือเลิกใช้ (termination) โมดูลโปรแกรมส่วนมากมีโมดูลควบคุมใช้สำหรับตรวจสอบและควบคุมการทำงานต่าง ๆ เช่น

1. ลำดับของการประมวลผล (order of processing)
2. ขั้นตอนการทำงานซ้ำ ๆ (looping)
3. เงื่อนไขยกเว้น เช่น ข้อผิดพลาดต่าง ๆ (errors)
4. สิ่งเบี่ยงเบนจากการประมวลผลปกติ (other deviations from normal processing require)

3. การเขียนคำสั่งโปรแกรม (Program coding)

การเขียนคำสั่งโปรแกรมเป็นขั้นตอนในการแปลง (convert) ตรรกะที่ได้ออกแบบในระยะการออกแบบโปรแกรมให้เป็นกลุ่มของคำสั่งโปรแกรมภาษาเพื่อสั่งให้คอมพิวเตอร์ปฏิบัติตามโปรแกรมภาษาในปัจจุบันมีมากมายหลายภาษา ซึ่งเหมาะกับงานด้านต่าง ๆ ซึ่งแต่ละภาษามีการเขียนที่แตกต่างกัน ทั้งรูปแบบกฎเกณฑ์ต่างๆ ดังนั้นผู้เขียนควรศึกษารูปแบบและกฎเกณฑ์ต่างๆ เหล่านี้ก่อนโปรแกรมใดๆ ที่นิยมใช้ใน

ปัจจุบัน จะประกอบด้วยคำสั่งโครงสร้างพื้นฐาน 3 อย่างคือ

1. แบบลำดับ (Sequence)
2. แบบทางเลือก (Selection)
3. แบบวนรอบ (Loop หรือ Repetition)

ซึ่งเป็นส่วนสำคัญในการเขียนโปรแกรมโครงสร้างแบบบนลงล่าง (top-down structure) จะช่วยให้การเขียนโปรแกรมเป็นมาตรฐาน (standardizes) และเข้าใจได้ง่าย รวมทั้งการแก้ไขก็ง่ายขึ้นอีกด้วย

Sequence Structure

เป็นโครงสร้างลำดับ ซึ่งแสดงถึงลำดับของคำสั่งหรือการปฏิบัติงานกล่าวคือคำสั่งซึ่งอยู่ก่อนจะถูกปฏิบัติงานก่อนดังนั้นคำสั่งโปรแกรมซึ่งเก็บไว้ในคอมพิวเตอร์ก่อนถูกทำงานก่อน

Selection Structure

โครงสร้างทางเลือก หรือเรียกว่า decision หรือ IF-THEN-ELSE ก็ได้ เป็นโครงสร้างซึ่งแสดงทางเลือกของการทำงาน โดยขึ้นกับผลของเงื่อนไข โดยเงื่อนไขนี้ผลลัพธ์มี 2 ทางคือ จริง (True) และ เท็จ (False) ถ้าเงื่อนไขเป็นจริงจะกระทำอย่างหนึ่ง ถ้าเงื่อนไขเป็นเท็จจะกระทำอีกอย่างหนึ่ง

Repetition (Loop) Structure

โครงสร้างวนรอบ หรือเรียกว่า DO-WHILE หรือ DO-UNTIL ก็ได้ เป็นโครงสร้างที่กระทำหน้าที่หรือคำสั่งซึ่งขึ้นกับเงื่อนไข โดยการทำงานจะเป็นการทำงานซ้ำ ๆ กัน ซึ่งจะหยุดการทำงานวนรอบก็ต่อเมื่อเงื่อนไขเป็นเท็จ (False)

4. การตรวจสอบความถูกต้องของโปรแกรม (Program Verification)

การตรวจทานโปรแกรมโดยทั่วไปเรียกว่าdebuggingซึ่งเป็นระยะหนึ่งในการเขียนโปรแกรมรวมถึงการตรวจสอบ (checking) การทดสอบ (testing) และการทำให้ถูกต้อง (correction) เพราะการเขียนคำสั่งโปรแกรมใหม่ ๆ อาจเกิดข้อผิดพลาด (bugs) ได้ง่าย

4.1 ข้อผิดพลาดในการเขียนโปรแกรม (Programming Error)

ข้อผิดพลาดในการเขียนโปรแกรมสามารถแบ่งเป็น 3 ชนิดคือ
syntaxerrors,logicerrors และ system design errors

Syntax errors เป็นข้อผิดพลาดที่เกิดจากการเขียนคำสั่งโปรแกรมผิดรูปแบบไวยากรณ์ของภาษา

Logic errors เป็นข้อผิดพลาดที่เกิดจากการใช้ตรรกะผิดในโปรแกรม ซึ่งเงื่อนไขผิดมีผลให้การกระทำตามเงื่อนไขผิดไปด้วย

System design errors เป็นข้อผิดพลาดจากการออกแบบระบบทำให้ผลของการทำงานไม่เป็นที่ยอมรับของผู้ใช้ซึ่งความผิดพลาดนี้ อาจเกิดจากการสื่อสารระหว่างโปรแกรมเมอร์กับผู้วิเคราะห์ระบบหรือผู้ใช้ระบบไม่ดี

Syntax errors เป็นความผิดพลาดที่ค้นพบได้ง่ายกว่า Logic errors เพราะสามารถตรวจ

พบได้ในระหว่างการทำงานการแปลภาษา ส่วน logic errors จะตรวจพบเมื่อโปรแกรมทำงานจนได้ผลลัพธ์ไม่ถูกต้องเท่านั้น

4.2 การตรวจสอบ (checking)

การตรวจสอบโปรแกรมต้องมีขึ้นระหว่างการออกแบบโปรแกรม, การเขียนโปรแกรม, การตรวจทานโปรแกรมเพื่อให้แน่ใจว่าสามารถแก้ปัญหาได้ตรงตามความต้องการตามวัตถุประสงค์ที่ได้ตั้งไว้ เครื่องมือที่ใช้ในการออกแบบถูกต้องตามตรรกะในการประมวลผลรวมทั้งคำสั่งโปรแกรมที่เขียนขึ้นสามารถแปลได้โดยปราศจากข้อผิดพลาดต่างๆ และสามารถทำงานได้อย่างถูกต้องสมบูรณ์

4.3 Structured walkthroughs

เป็นเครื่องมือของการออกแบบ, เขียนคำสั่ง, ตรวจสอบข้อผิดพลาดของการเขียนโปรแกรมที่ดี ซึ่งเป็นวิธีการที่ผู้เขียนโปรแกรมแสดงผลงานให้โปรแกรมเมอร์อื่นๆตรวจสอบดูซึ่งการทำงานนี้เป็นแนวความคิดของการเขียนโปรแกรมเป็นทีมงานซึ่งมีการกำหนดให้พัฒนาโปรแกรมเดียวกันภายใต้การควบคุมของหัวหน้าโปรแกรมเมอร์ (chief programmer) โดยสมาชิกในทีมงานจะช่วยกันตรวจดูถึงการออกแบบและการเขียนคำสั่งเป็นช่วงๆ เป็นประจำของแต่ละโมดูล ซึ่งทำให้ค่าใช้จ่ายของการทวนสอบ (Verification) น้อยลง โดยสามารถพบข้อผิดพลาดได้ในระยะเริ่มแรกของการเขียนโปรแกรม โดยไม่ต้องคอยจนกระทั่งตรวจพบในระยะเวลาของการทดสอบโปรแกรม ซึ่งเป็นการทราบที่จะทราบว่าจุดใดที่เกิดข้อผิดพลาด ซึ่งมีผลทำให้ค่าใช้จ่ายสูงตามไปด้วย

4.4 การทดสอบ (testing)

การทดสอบเป็นการตรวจสอบการทำงานของโปรแกรมโดยใช้ข้อมูลทดสอบ (testdata) เพื่อตรวจสอบผลจากการทำงานการทดสอบที่ดีนั้นควรใช้ข้อมูลที่แตกต่างกันเพื่อให้สามารถกระทำได้ในทุกๆ เงื่อนไขในการทำงานของโปรแกรมรวมทั้งควรทดลองข้อมูลที่ไม่ว่างด้วยโปรแกรมที่ดีเมื่อใส่ข้อมูลผิดๆ จะไม่เกิด error แต่จะแสดงข้อความเพื่อเตือนเท่านั้นในการเขียนโปรแกรมโครงสร้างที่ใช้โปรแกรมภาษาระดับสูงนั้นผู้ทำการทดสอบโปรแกรมสามารถแบ่งการทดสอบโปรแกรมออกเป็นหลายๆ ซึ่งเรียกว่า โมดูล เพื่อสะดวกในการทดสอบซึ่งง่ายต่อการค้นหาข้อผิดพลาดและง่ายต่อการแก้ไขให้ถูกต้องด้วยเมื่อทดสอบโมดูลย่อยๆ เหล่านี้จนไม่มีข้อผิดพลาดจึงนำมาเป็นโปรแกรมหลักอีกครั้งหนึ่ง การทดสอบโปรแกรมกรณีที่ต้องให้โปรแกรมที่พัฒนาขึ้นมาแทนที่การประมวลผลเก่าซึ่งกำลังดำเนินการอยู่การทดสอบต้องกระทำขนานกันไปกับระบบเก่าซึ่งเรียกรวมการประมวลผลนี้ว่า parallel processing ดังนั้นการทดสอบระบบต้องทดสอบจนกว่าจะแน่ใจว่าสามารถกระทำแทนที่ระบบเก่าได้จึงจะนำเอาการปฏิบัติงานของระบบเก่าออกไปได้

5. เอกสารโปรแกรม (Program documentation)

เป็นเอกสารซึ่งบันทึกรายละเอียดของการออกแบบการเขียนคำสั่งซึ่งมีประโยชน์ต่อการวิเคราะห์ความผิดพลาดของโปรแกรม การแก้ไขปรับปรุงโปรแกรมหรือการรวมโปรแกรมกรณีเกิดการสูญหาย โดยเฉพาะโปรแกรมเมอร์หลักที่เขียนโปรแกรมเกิดลาออกไป ดังนั้นควรเก็บรวบรวมข้อมูลต่างๆ เอาไว้

6. การบำรุงรักษาโปรแกรม (Program maintenance)

ขั้นตอนสุดท้ายในการเขียนโปรแกรมคอมพิวเตอร์จะเริ่มหลังจากโปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นนั้นเป็นที่ยอมรับจากผู้ใช้งานและได้ปฏิบัติงานมาแล้วชั่วระยะเวลาหนึ่ง ต่อมาเกิดความจำเป็นบางอย่างเกิดขึ้น เช่น ต้องการปรับปรุงระบบให้มีประสิทธิภาพดีขึ้น หรือต้องการแก้ไขหน้าที่บางอย่างหรือต้องการขยายขีดความสามารถของโปรแกรม หรืออาจเกิดจากความผิดพลาดในโปรแกรมต้องการแก้ไขให้ถูกต้อง โดยอาจมีผลมาจากนโยบายของบริษัทที่เปลี่ยนไป หรือระเบียบทางราชการบังคับ หรือการแข่งขันทางธุรกิจ ฯลฯ การเปลี่ยนแปลงอาจเกิดขึ้นได้กับทั้งฮาร์ดแวร์และซอฟต์แวร์ของระบบ การบำรุงรักษาเป็นหน้าที่หลักของการประมวลผลข่าวสารของหน่วยงานเกี่ยวข้องกับการวิเคราะห์ การออกแบบ การเขียนคำสั่ง การตรวจสอบ การเปลี่ยนแปลงเอกสารให้ทันสมัย ซึ่งจะมี **Maintenance Programmers** เป็นผู้ที่รับผิดชอบหน้าที่บำรุงรักษาโปรแกรม ซึ่งในความเป็นจริงแล้วผู้เขียนโปรแกรมกับผู้ที่บำรุงรักษาโปรแกรมจะเป็นคนละทีมงานกัน ดังนั้นจึงเป็นการยากที่ต้องแก้ไขโปรแกรมที่ไม่ได้พัฒนาขึ้นมา ดังนั้นสิ่งสำคัญของการเขียนโปรแกรมโครงสร้างจะช่วยให้การเขียนโปรแกรมเป็นมาตรฐานและทำให้ง่ายต่อการอ่านและเข้าใจ