

รายการอ้างอิง

ภาษาไทย

- ประคอง กรรณสุด. สถิติเพื่อการวิจัยทางพฤติกรรมศาสตร์. กรุงเทพมหานคร: ภาควิชาวิจัยการศึกษา คณะครู
ศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2524.
- พรศรี ศรีธัญญาพร และ ยวดี วัฒนานนท์. สถิติและการวิจัยเบื้องต้น. กรุงเทพมหานคร: โรงพิมพ์สามเจริญพา
นิช, 2529.
- วันทนีย์ ชูศิลป์. ระเบียบวิธีการวิจัย. กรุงเทพมหานคร: สองแควการพิมพ์, 2524.
- วิเชียร เกตุสิงห์. สถิติวิเคราะห์สำหรับการวิจัย. กรุงเทพมหานคร, 2535.

ภาษาอังกฤษ

- Assawamekin, N., and Rivepiboon, W. A Development of Standard Software Development Process.
Proceedings of the National Electronics and Computer Technology Center (NECTEC) (April
1999): 217-225.
- Draft guidebook for MIL-STD-498. overview and tailoring. Washington DC: US Department of
Defence, 1995.
- ESA software engineering standards. ESA PSS-05-0. Netherlands: European Space Agency, 1991.
- Giakoumakis, E.A.; and Xylomenos, G. Evaluation and selection criteria for software requirements
specification standards. Software Engineering Journal (September 1996): 307-319.
- IEEE Std 830-1993: Recommended Practice for Software Requirements Specifications. New York:
The Institute of Electrical and Electronics Engineers, 1994.
- Mazza, C., Fairclough, J., Melton, B., De Pablo, D., Scheffer, A., Stevens, R., Jones, M., and Alvisi, G.
Software Engineering Guides. New York: Prentice Hall, 1996.
- Philip L. Weaver. Practical SSADM version 4. London: Pitman, 1994.
- Requirements DID, NASA-DID-P200. National Aeronautics and Space Agency, 1991.
- Roger S. Pressman. Software Engineering: A Practitioner's Approach. 3rd ed. New York: McGraw-
Hill, 1992.
- William M. Wilson. Writing Effective Natural Language Requirements Specifications. The Journal of
Defense Software Engineering (February 1999): 16-19.

ภาคผนวก

ภาคผนวก ก

ส่วนนี้เป็นแบบสอบถาม ซึ่งผู้วิจัยนำไปทำการตรวจสอบความสมเหตุสมผลเชิงเนื้อหาโดยการจัด
เวิร์กชอปในห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ก่อนที่จะนำแบบสอบถามนี้ไปทำพรีเทสต์

แบบสอบถามสำหรับผู้พัฒนาซอฟต์แวร์

ส่วนที่ 1 : ข้อมูลเกี่ยวกับหน่วยงาน

ชื่อหน่วยงาน.....

โทรศัพท์.....

1. หน่วยงานของท่านจัดอยู่ในกลุ่มใด

- บริษัทเอกชนที่ประกอบกิจการคอมพิวเตอร์โดยตรง และพัฒนาซอฟต์แวร์
 - ซอฟต์แวร์แฮนด์
 - ทั่วไป
- บริษัทเอกชนที่ไม่ได้ประกอบกิจการคอมพิวเตอร์โดยตรง แต่มีการพัฒนาซอฟต์แวร์
- หน่วยงานราชการ
- รัฐวิสาหกิจ
- อื่นๆ (ระบุ).....

2. ปัจจุบันหน่วยงานของท่านมีการพัฒนาซอฟต์แวร์ โดยมีที่มาจาก (ตอบได้มากกว่า 1 ข้อ)

ระบบ	ซอฟต์แวร์สำเร็จรูป (Software Package)	พัฒนาขึ้นเอง	จ้างบริษัทอื่นพัฒนา
CAD/CAM	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computer Aided Instruction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entertainment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Expert System	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GIS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Intranet/Internet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MIS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Point of Sales (POS)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scientifics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ระบบ	ซอฟต์แวร์สำเร็จรูป (Software Package)	พัฒนาขึ้นเอง	จ้างบริษัทอื่นพัฒนา
Simulation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Statistics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
กราฟฟิก (Graphic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
การเงิน (Financial)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
การท่องเที่ยว (Travelling)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
การผลิต (Manufacturing)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ควบคุมคุณภาพสินค้า (Q.C.)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
เครือข่าย (Network)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
เครื่องบิน (Airport)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ซื้อ-ขาย (Purchase-Sales)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ฐานข้อมูล (Database)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ธนาคาร (Bank)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
บัญชี (Account)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
บุคลากร (Personnel)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ประกันภัย (Insurance)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ระบบปฏิบัติการ (OS Systems)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
โรงพยาบาล (Hospital)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
โรงแรม (Hotel)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
สินค้าคงคลัง (Inventory)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ห้องสมุด (Library)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
อื่น ๆ (ระบุ)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
*			
*			
*			
*			

3. จำนวนพนักงานทั้งหมดในหน่วยงานของท่านโดยประมาณ.....คน

4. จำนวนพนักงานที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์โดยประมาณ.....คน

ส่วนที่ 2 : ข้อมูลเกี่ยวกับผู้ตอบ

1. ชื่อ-สกุล.....
2. อายุ.....ปี
3. E-mail address :
4. วุฒิการศึกษาสูงสุด
 - ต่ำกว่าปริญญาตรี
 - ปริญญาตรี
 - ปริญญาโท
 - ปริญญาเอก หรือสูงกว่าปริญญาเอก
5. ตำแหน่ง.....
6. หน้าที่ความรับผิดชอบ.....
7. ระยะเวลาทำงานในหน่วยงานปัจจุบัน.....ปี
8. ประสบการณ์ในการพัฒนาซอฟต์แวร์.....ปี

ส่วนที่ 3 : ข้อมูลเกี่ยวกับโครงการ

ชื่อโครงการ.....

1. จำนวนพนักงานที่ทำการพัฒนาซอฟต์แวร์ในโครงการนี้โดยประมาณ.....คน
2. ระยะเวลาที่ทำการพัฒนาซอฟต์แวร์ในโครงการนี้.....เดือน
3. ขนาดของโครงการ.....พันบรรทัดของโปรแกรม (KLOC)
4. หน่วยงานของท่านมีวิธีการใดที่ใช้ในการเก็บรวบรวมความต้องการของผู้ใช้หรือลูกค้าสำหรับการพัฒนาซอฟต์แวร์ (ตอบได้มากกว่า 1 ข้อ)
 - การสัมภาษณ์
 - แบบสอบถาม
 - ศึกษาจากระบบเก่า
 - Prototyping
 - อื่นๆ (ระบุ)
 -
 -
 -

5. ปัญหาหรืออุปสรรคในการเก็บรวบรวมความต้องการของผู้ใช้หรือลูกค้าเท่าที่ท่านพบ มีอะไรบ้าง

.....

.....

.....

.....

6. หน่วยงานของท่านมีการจัดทำ SRS ในการพัฒนาซอฟต์แวร์หรือไม่

ไม่มี

มี (ไปข้อ 8)

7. เหตุผลที่หน่วยงานของท่านไม่จัดทำ SRS

ระยะเวลาจำกัด

งบประมาณไม่เพียงพอ

ขาดความรู้

อื่นๆ (ระบุ)

.....

.....

8. หน่วยงานของท่านมีมาตรฐานที่ช่วยในการจัดทำ SRS หรือไม่

ไม่มีมาตรฐาน SRS ในหน่วยงาน

มีมาตรฐาน SRS ที่พัฒนาขึ้นเองในหน่วยงาน

อ้างอิงมาตรฐานที่มีอยู่แล้ว

IEEE

ESA

NASA

DoD

อื่นๆ (ระบุ)

.....

.....

9. หน่วยงานของท่านใช้ tools ในการจัดทำ SRS หรือไม่ (เช่น CASE tools, Word Processing, Microsoft Excel เป็นต้น)

ไม่ใช่

ใช้ (ระบุ)

.....

Contents	มี contents นี้ ในหน่วยงานหรือไม่		ควรมี contents นี้ อยู่ใน SRS หรือไม่			
	มี	ไม่มี	ไม่ควรมี	ควรมี		
				น้อย	ปานกลาง	มาก
50.Availability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
51.Security	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
52.Maintainability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
53.Portability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Section 4 : Supporting information						
54.Table of contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
55.Notes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
56.Appendixes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
57.Index	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

11. ท่านคิดว่า template ของ SRS ในข้อ 10 ควรมี contents ได้เพิ่มเติมหรือไม่

ไม่มี

มี (ระบุ).....

.....

.....

.....

.....

12. ท่านคิดว่าการแบ่ง section ของ template เหมาะสมหรือไม่

ไม่เหมาะสม เพราะ.....

.....

.....

เหมาะสม

13. ข้อเสนอแนะเพิ่มเติม (ถ้ามี).....

.....

.....

.....

.....

ภาคผนวก ข

ส่วนนี้เป็นแบบสอบถาม ซึ่งผู้วิจัยทำการสอบถามผู้มีส่วนร่วมในการพัฒนาซอฟต์แวร์หรือผู้บริหารโครงการซอฟต์แวร์ เพื่อนำมาเป็นแนวทางในการจัดทำ SRS

แบบสอบถามสำหรับผู้มีส่วนร่วมในการพัฒนาซอฟต์แวร์หรือผู้บริหารโครงการซอฟต์แวร์

ส่วนที่ 1 : ข้อมูลเกี่ยวกับองค์กร

ชื่อองค์กร.....

โทรศัพท์.....

1. องค์กรของท่านจัดอยู่ในกลุ่มใด

บริษัทเอกชน

หน่วยงานราชการ

รัฐวิสาหกิจ

อื่นๆ (ระบุ).....

2. จำนวนพนักงานทั้งหมดในองค์กรของท่านโดยประมาณ

ต่ำกว่า 30 คน

30-50 คน

51-100 คน

มากกว่า 100 คนขึ้นไป

3. จำนวนพนักงานที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์โดยประมาณ

ต่ำกว่า 10 คน

10-30 คน

31-50 คน

มากกว่า 50 คนขึ้นไป

4. องค์กรของท่านมีมาตรฐานที่ช่วยในการจัดทำ Software Requirements Specification (SRS) หรือไม่ (ตอบได้มากกว่า 1 ข้อ)

ไม่มีมาตรฐาน SRS ในองค์กร

มีมาตรฐาน SRS ที่พัฒนาขึ้นเองในองค์กร

อ้างอิงมาตรฐานที่มีอยู่แล้ว

IEEE

ESA

NASA

DoD

อื่นๆ (ระบุ).....

.....
.....

5. องค์กรของท่าน มีการพัฒนาซอฟต์แวร์หรือว่าจ้างบริษัทอื่นพัฒนาซอฟต์แวร์ประเภทใดบ้าง (ตอบได้มากกว่า 1 ข้อ)

ซอฟต์แวร์	ตัวอย่าง	พัฒนา ขึ้นเอง	ซื้อหรือจ้างบริษัท อื่นพัฒนา	ไม่มี
1. Banking	Automatic Teller Machine (ATM)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Data Communication	Internet, Intranet, Network, File Transfer Protocol (FTP), telnet, electronic mail, World Wide Web (WWW), Electronic Data Interchange (EDI)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Data Management	Database, Data Mining, Data Warehouse	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Education	Computer Aided Instruction (CAI)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Engineering & Scientific	Simulation, Computer Aided Design (CAD)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Finance/Accounting	Account Receivable (A/R), Account Payable (A/P), Financial	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Groupware	Lotus Notes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Insurance & Real Estate	Insurance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Language Processors	Dictionary, Word Processor, Word Segmentation, EDI's Translator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Manufacturing	Computer Aided Manufacturing (CAM), Material Resource Planning (MRP), Manufacturing Resource Planning II (MRP II), Quality Control (Q.C.)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. Medical & Healthcare	Hospital	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Office Automation	Microsoft Office, Fax	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. Payroll & Personnel	Personnel, Human resource	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. Programming Aids	McAfee, Norton	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. Sales & Distribution	Point Of Sales (POS), Inventory, Transportation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ซอฟต์แวร์	ตัวอย่าง	พัฒนา ตัวเอง	ชื่อหรือจ้างบริษัท อื่นพัฒนา	ไม่มี
16.System Programs	Operating System (OS), Compiler, Interpreter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17.Miscellaneous	Library, Airport, Hotel, etc.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ส่วนที่ 2 : ข้อมูลเกี่ยวกับผู้ตอบ

- ชื่อ-สกุล.....
- อายุ

<input type="checkbox"/> ต่ำกว่า 25 ปี	<input type="checkbox"/> 25-35 ปี
<input type="checkbox"/> 36-45 ปี	<input type="checkbox"/> มากกว่า 45 ปีขึ้นไป
- ที่อยู่ (สถานที่ติดต่อโดยสะดวก).....
 โทรศัพท์.....
 E-mail address :
- วุฒิการศึกษาสูงสุด

<input type="checkbox"/> ต่ำกว่าปริญญาตรี	<input type="checkbox"/> ปริญญาตรี
<input type="checkbox"/> ปริญญาโท	<input type="checkbox"/> ปริญญาเอก
- ชื่อฝ่าย/แผนก..... ตำแหน่ง.....
- หน้าที่ความรับผิดชอบ.....

- ระยะเวลาทำงานในองค์กรปัจจุบัน

<input type="checkbox"/> ต่ำกว่า 3 ปี	<input type="checkbox"/> 3-6 ปี
<input type="checkbox"/> 7-10 ปี	<input type="checkbox"/> มากกว่า 10 ปีขึ้นไป
- ประสบการณ์ในการพัฒนาซอฟต์แวร์

<input type="checkbox"/> ต่ำกว่า 3 ปี	<input type="checkbox"/> 3-6 ปี
<input type="checkbox"/> 7-10 ปี	<input type="checkbox"/> มากกว่า 10 ปีขึ้นไป
- จำนวนโครงการซอฟต์แวร์ที่เคยพัฒนา

<input type="checkbox"/> ต่ำกว่า 5 โครงการ	<input type="checkbox"/> 5-10 โครงการ
<input type="checkbox"/> 11-20 โครงการ	<input type="checkbox"/> มากกว่า 20 โครงการขึ้นไป

ส่วนที่ 3 : ข้อมูลเกี่ยวกับโครงการ

ชื่อโครงการ.....

โครงการนี้จัดอยู่ในซอฟต์แวร์ประเภท.....

(อ้างอิงจากส่วนที่ 1 ในข้อ 5)

1. จำนวนพนักงานที่ทำการพัฒนาซอฟต์แวร์ในโครงการนี้

ต่ำกว่า 10 คน

10-30 คน

31-50 คน

มากกว่า 50 คนขึ้นไป

2. ระยะเวลาที่ทำการพัฒนาซอฟต์แวร์ในโครงการนี้

ต่ำกว่า 6 เดือน

6-12 เดือน

13-24 เดือน

มากกว่า 24 เดือนขึ้นไป

3. ขนาดของโครงการ (ตอบได้มากกว่า 1 ข้อ)

- งบประมาณในส่วนของการพัฒนาซอฟต์แวร์

ต่ำกว่า 500,000 บาท

500,000-2,000,000 บาท

2,000,001-10,000,000 บาท

มากกว่า 10,000,000 บาทขึ้นไป

- จำนวนคนที่พัฒนาซอฟต์แวร์เฉลี่ยต่อเดือน (Man Month : MM)

ต่ำกว่า 10 MM

10-30 MM

31-100 MM

มากกว่า 100 MM ขึ้นไป

- จำนวนหน้าจอ (Screen)

ต่ำกว่า 20 หน้าจอ

20-50 หน้าจอ

51-100 หน้าจอ

มากกว่า 100 หน้าจอขึ้นไป

- จำนวนรายงาน (Report)

ต่ำกว่า 20 รายงาน

20-50 รายงาน

51-100 รายงาน

มากกว่า 100 รายงานขึ้นไป

4. ในโครงการนี้ ท่านใช้วิธีการใดในการเก็บรวบรวมความต้องการของผู้ใช้หรือลูกค้า (ตอบได้มากกว่า 1 ข้อ)

การสัมภาษณ์

แบบสอบถาม

ศึกษาจากระบบเก่า

Prototyping

อื่นๆ (ระบุ)

.....

.....

.....

5. ปัญหาหรืออุปสรรคในการเก็บรวบรวมความต้องการของผู้ใช้หรือลูกค้าเท่าที่ท่านพบ

.....

.....

.....

.....

6. โครงการนี้มีการจัดทำ SRS ในการพัฒนาซอฟต์แวร์หรือไม่

ไม่จัดทำ (ไปข้อ 7)

จัดทำ (ไปข้อ 8)

7. เหตุผลที่โครงการนี้ไม่จัดทำ SRS

ระยะเวลาจำกัด

งบประมาณไม่เพียงพอ

ไม่มีความจำเป็นต้องทำ

อื่นๆ (ระบุ)

.....

.....

8. โครงการนี้มีมาตรฐานที่ช่วยในการจัดทำ SRS หรือไม่ (ตอบได้มากกว่า 1 ข้อ)

ไม่มีมาตรฐาน SRS ในองค์กร

มีมาตรฐาน SRS ที่พัฒนาขึ้นเองในองค์กร

อ้างอิงมาตรฐานที่มีอยู่แล้ว

IEEE

ESA

NASA

DoD

อื่นๆ (ระบุ).....

.....

.....

9. โครงการนี้ใช้ CASE tools ช่วยในการจัดทำ SRS หรือไม่

ไม่ใช่

ใช่ (ระบุ).....

.....

.....

.....

10. กรุณาตอบคำถามโดยทำเครื่องหมาย ✓ ในช่อง ซึ่งคำถามมี 2 ข้อ คือ มีเนื้อหา (contents) นี้ใน SRS ของโครงการนี้หรือไม่ (ในกรณีที่โครงการนี้มีการจัดทำ SRS) และท่านคิดว่าควรมีเนื้อหานี้อยู่ใน SRS หรือไม่ ถ้ามีควรอยู่ในระดับใด

Contents	มีเนื้อหานี้ ใน SRS หรือไม่		ควรมีเนื้อหานี้อยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี			
				น้อย	ปาน กลาง	มาก	
Section 1 : Introduction							
1. Purpose	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงจุดประสงค์ของการเขียน SRS หรือความตั้งใจที่เฉพาะเจาะจงของผู้ที่จะใช้ SRS ฉบับนี้
2. Scope	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงชื่อ หน้าที่ของซอฟต์แวร์ SRS ต้องมีความสอดคล้องกับ System Requirements Specification (SyRS) และอธิบายถึงการนำเอาซอฟต์แวร์ไปประยุกต์ใช้ให้บรรลุวัตถุประสงค์ ประโยชน์ และเป้าหมายที่ตั้งไว้
3. Definitions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงคำจำกัดความของคำศัพท์ต่างๆ ที่ใช้ใน SRS เพื่อให้เข้าใจความหมายของคำศัพท์ที่ตรงกัน
4. Acronyms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงคำย่อของคำศัพท์ต่างๆ โดยนำเอาอักษรตัวแรกของคำศัพท์มาย่อ เช่น SRS ย่อมาจาก Software Requirements Specification
5. Abbreviations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงคำย่อของคำศัพท์ต่างๆ โดยไม่มีกฎเกณฑ์ในการย่อ เช่น Std ย่อมาจาก Standard
6. References	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงเอกสารอ้างอิงที่ใช้ใน SRS ประกอบด้วย ชื่อ หมายเลขรายงาน (ถ้ามี) วันที่ สถานที่ พิมพ์ และแหล่งที่มา
7. Overview	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงเนื้อหาโดยรวมที่อยู่ใน SRS เป็นการมองภาพรวมของเอกสาร และอธิบายว่า SRS มีการจัดระเบียบของเอกสารอย่างไร
Section 2 : Overall description							
8. User interfaces	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการต่อประสานระหว่างซอฟต์แวร์กับผู้ใช้ เช่น รูปแบบหน้าจอ โครงร่างของรายงาน เนื้อหาของรายงานหรือเมนู เป็นต้น

Contents	มีเนื้อหาใน SRS หรือไม่		ควรมีเนื้อหาอยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี			
				น้อย	ปานกลาง	มาก	
9.Hardware interfaces	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการต่อประสานระหว่างซอฟต์แวร์กับส่วนประกอบของฮาร์ดแวร์ เช่น number of ports, instruction set, devices, protocols เป็นต้น
10.Software interfaces	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการต่อประสานระหว่างซอฟต์แวร์กับระบบอื่นๆ โดยซอฟต์แวร์แต่ละชนิด ประกอบด้วย ชื่อ, mnemonic, version, แหล่งที่มา
11.Communications interfaces	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการต่อประสานระบบสื่อสารในรูปแบบต่างๆ เช่น local network protocols เป็นต้น
12.Memory constraints	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงลักษณะและข้อจำกัดของหน่วยความจำหลัก และหน่วยความจำรอง
13.Operations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการดำเนินการใช้งานซอฟต์แวร์แบบธรรมดา และแบบกรณีพิเศษ เช่น การดำเนินการของผู้ใช้เพื่อเริ่มต้นใช้ระบบ ฟังก์ชันสนับสนุนการประมวลผลข้อมูล เป็นต้น
14.Site adaptation reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงที่ตั้งของการติดตั้งซอฟต์แวร์ เช่น พิจารณาถึง config, platform, จำนวน node และ port ที่ใช้กับซอฟต์แวร์นี้
15.Product functions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุหน้าที่หลักของซอฟต์แวร์ เพื่อสร้างความเข้าใจให้กับผู้ใช้หรือลูกค้า โดยอาจใช้ข้อความหรือรูปภาพก็ได้
16.User characteristics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงลักษณะโดยทั่วไปของผู้ใช้ รวมถึง ระดับการศึกษา ประสบการณ์ การชำนาญทางเทคนิค
17.Qualification provisions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงกระบวนการที่จะนำมาพิจารณาซอฟต์แวร์ว่าซอฟต์แวร์มีความเหมาะสมในการใช้งานหรือไม่ เช่น ค่าใช้จ่าย ความสะดวก ระยะเวลา เป็นต้น
18.Requirements traceability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงความสามารถในการตรวจสอบความต้องการกลับไปได้ว่าความต้องการนั้นๆ มีจุดกำเนิดมาจากใคร เพื่อใช้อ้างอิงต่อไปในอนาคต

Contents	มีเนื้อหาใน SRS หรือไม่		ควรมีเนื้อหาอยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี			
				น้อย	ปานกลาง	มาก	
19.Constraints	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงคำอธิบายและข้อจำกัดในการพัฒนาซอฟต์แวร์ที่ควรคำนึงถึง คือ ข้อจำกัดทางฮาร์ดแวร์ หน้าที่ใช้การตรวจสอบระบบ (ทำ log file) หน้าที่ใช้การควบคุมระบบ (เช่น ทำ polling) ความเชื่อถือได้ ความปลอดภัย เป็นต้น
20.Assumptions and dependencies	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงปัจจัยต่างๆ ที่ส่งผลกระทบต่อความต้องการที่ระบุไว้ใน SRS เช่น ตั้งข้อสันนิษฐานว่าจะใช้ ORACLE 8.0 กับซอฟต์แวร์นี้ แต่ในความเป็นจริง ORACLE 8.0 ยังพัฒนาไม่เสร็จ ทำให้ซอฟต์แวร์ไม่สามารถใช้กับ ORACLE 8.0 ได้ ดังนั้นความต้องการที่ระบุไว้ใน SRS ต้องได้รับการเปลี่ยนแปลงด้วย
21.Apportioning of reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงความต้องการที่ผู้ใช้หรือลูกค้ายอมให้เกิดความล่าช้าหรือเป็นความต้องการที่คาดว่าจะไม่สามารถพัฒนาเสร็จในเวอร์ชันนี้ และจะทำการพัฒนาในเวอร์ชันต่อไปในอนาคต
Section 3 : Specific requirements							
22.External interfaces	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงรายละเอียดของข้อมูลที่เข้าไปยังระบบ และข้อมูลที่ออกจากระบบทั้งหมด ประกอบด้วย ชื่อของตัวข้อมูล (item) จุดประสงค์ แหล่งที่มาของข้อมูล ช่วงของข้อมูลที่ถูกต้อง รูปแบบหน้าจอ รูปแบบวินโดวส์ รูปแบบของข้อมูล รูปแบบของคำสั่งงาน ข้อความจบ (end message)
23.Functional reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงหน้าที่ของซอฟต์แวร์ โดยอธิบายในลักษณะของประโยคที่ว่า "ระบบควรจะ..." ประกอบด้วย การตรวจสอบความถูกต้องของข้อมูลเข้า ลำดับที่แน่นอนของการดำเนินการ การตอบสนองต่อเหตุการณ์ที่ผิดปกติ ความสัมพันธ์ระหว่างข้อมูลเข้าและข้อมูลออก

Contents	มีเนื้อหาใน SRS หรือไม่		ควรมีเนื้อหาอยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี			
				น้อย	ปานกลาง	มาก	
24.Performance reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงค่าที่ใช้วัดประสิทธิภาพของซอฟต์แวร์ โดยใช้กับตัวแปรที่สามารถวัดค่าได้ เช่น จำนวนของผู้ใช้ เวลาที่ใช้ในการโต้ตอบระหว่างผู้ใช้กับระบบ (response time) ปริมาณงาน (throughput) เป็นต้น
25.Logical database reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงความต้องการเชิงตรรกะของข้อมูลที่จะจัดเก็บในฐานข้อมูล ประกอบด้วย ชนิดของข้อมูล ความถี่ของการใช้งาน วิธีในการเข้าถึงข้อมูล (SELECT, UPDATE, INSERT) ระบุ data entity ความสัมพันธ์ระหว่าง entity
26.Operational reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงวิธีการดำเนินงานของซอฟต์แวร์และวิธีการติดต่อระหว่างผู้ใช้กับระบบ เช่น โครงร่างของหน้าจอ ข้อความผิดพลาด ระบบช่วยเหลือ เป็นต้น ซึ่งเนื้อหาในส่วนนี้จะเป็ประโยชน์ในการนิยามถึง semantics และ syntax ของคำสั่ง
27.Verification reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงวิธีการตรวจสอบความถูกต้องของซอฟต์แวร์ โดยอาจใช้วิธีการจำลองระบบ การทดสอบสภาพแวดล้อมจริง โดยใช้ข้อมูลที่จำลองขึ้นมาหรือข้อมูลจริง และทดสอบการต่อประสานไปยังระบบอื่นๆ ภายนอก
28.Acceptance reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงข้อกำหนดในการตรวจรับซอฟต์แวร์ เพื่อดูว่าซอฟต์แวร์สามารถทำหน้าที่ได้ตรงตามความต้องการของผู้ใช้หรือลูกค้าหรือไม่
29.Documentation reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงเอกสารที่ใช้ในการพัฒนาซอฟต์แวร์ ซึ่งอยู่นอกเหนือจากเอกสารที่กำหนดไว้แล้วในมาตรฐาน เช่น รูปแบบของเอกสารคู่มือผู้ใช้ซอฟต์แวร์ (Software User Manual) เป็นต้น
30.Safety reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงความปลอดภัยในระบบ เช่น ระบุถึงฟังก์ชันที่วิกฤต เพื่อลดโอกาสที่จะทำให้เกิดความเสียหายจากความผิดพลาดของซอฟต์แวร์ ซึ่งอาจเป็นอันตรายต่อคนหรือทรัพย์สิน
31.Environment reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงสภาพแวดล้อมในการติดตั้งซอฟต์แวร์ เช่น อุณหภูมิ ความชื้น ลม ฝน ฝุ่น สัญญาณรบกวน สัญญาณแม่เหล็กไฟฟ้า ความสะอาดในห้องปฏิบัติการคอมพิวเตอร์ เป็นต้น

Contents	มีเนื้อหาใน SRS หรือไม่		ควรมีเนื้อหาอยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี			
				น้อย	ปานกลาง	มาก	
32.Computer hardware resource utilization reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ภายในระบบ เพื่อให้เกิดประโยชน์สูงสุดในการใช้งาน เช่น ใช้โปรเซสเซอร์ หน่วยความจำ อุปกรณ์ข้อมูลนำเข้าหรือออก อุปกรณ์หน่วยความจำรอง อุปกรณ์การสื่อสารข้อมูล อุปกรณ์เครือข่าย เป็นต้น
33.Computer communication reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงการสื่อสารข้อมูลที่ใช้ภายในซอฟต์แวร์
34.Software quality factors	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงคุณลักษณะของซอฟต์แวร์ ได้แก่ ความถูกต้อง (correct) ไม่มีความคลุมเครือ (unambiguous) ความสมบูรณ์ (complete) ความต้องกัน (consistent) ความคงที่ (stability) ทวนสอบได้ (verification) แก้ไขได้ (modification) และติดตามได้ (traceable) ในหัวข้อนี้ไม่รวมถึงคุณลักษณะหลักๆ ที่เกี่ยวกับคุณภาพ (เช่น ความเชื่อถือได้ การบำรุงรักษา ความปลอดภัย) โดยอาจแสดงคุณลักษณะในรูปแบบของเมตริกซ์ (metrics) ได้
35.Training-related reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงความต้องการในการฝึกอบรมสำหรับบุคคลที่เกี่ยวข้องในระบบ แบ่งเป็น การฝึกอบรมสำหรับบุคลากรในทีมพัฒนาซอฟต์แวร์ และการฝึกอบรมสำหรับผู้ใช้นหรือลูกค้า
36.Logistics-related reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงขั้นตอนต่างๆ ที่เกี่ยวข้องในการนำซอฟต์แวร์มาใช้งาน (implementation) การสนับสนุนการใช้ซอฟต์แวร์ (support) และการบำรุงรักษาซอฟต์แวร์ (maintenance) รวมถึงการจัดการทรัพยากร เพื่อดำเนินการตามขั้นตอนดังกล่าวด้วย
37.Packaging reqs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงสื่อที่ใช้ในการเก็บบันทึกซอฟต์แวร์ (เช่น ดิสก์ เทป ซีดีรอม) เอกสารประกอบการใช้งานของซอฟต์แวร์ และอาจรวมถึงตัวต่อประสานที่จำเป็นต้องใช้กับซอฟต์แวร์ (เช่น I/O drivers)

Contents	มีเนื้อหาใน SRS หรือไม่		ควรมีเนื้อหาอยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี			
				น้อย	ปานกลาง	มาก	
38.Implementation constraints	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงข้อบังคับที่ใช้ในการออกแบบ และการใช้ซอฟต์แวร์และฮาร์ดแวร์ร่วมกัน เพื่อสนับสนุนการใช้งานของซอฟต์แวร์ เช่น ซอฟต์แวร์ต้อง run โดยใช้ระบบฐานข้อมูล หรือซอฟต์แวร์ต้องมีหน่วยความจำ 512 Kbytes เป็นต้น
39.Standard compliance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงมาตรฐานที่มีใช้อยู่แล้วในองค์กร ซึ่งจะนำมาใช้ในการพัฒนาซอฟต์แวร์ เช่น รูปแบบรายงาน การตั้งชื่อข้อมูล เป็นต้น ถ้ามีการใช้มาตรฐานอื่น ควรระบุไว้ในส่วนนี้ด้วย
40.Reliability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงเวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้น เรียกว่า mean time before failure (MTBF) เพื่อสร้างความเชื่อถือได้ในการส่งมอบซอฟต์แวร์ให้กับลูกค้า
41.Availability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงปัจจัยที่จำเป็น เพื่อรองรับสภาพพร้อมใช้งานของซอฟต์แวร์โดยรวมในระดับที่กำหนดไว้ เช่น checkpoint, recovery, restart เป็นต้น
42.Security	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงเทคนิคต่างๆ เพื่อป้องกันเหตุที่เกิดจากความมั่วร้ายในการเข้าถึงข้อมูล เช่น กระบวนการในการเข้าสู่ระบบและป้อนรหัสผ่าน การป้องกันไวรัสคอมพิวเตอร์ การใช้ประโยชน์จากเทคนิค cryptographical techniques การเก็บ historical data การเข้าถึงข้อมูลแบบอ่านอย่างเดียว การสำรองและกู้คืนข้อมูล เป็นต้น
43.Maintainability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงระดับความยากง่ายในการแก้ไขความผิดพลาด (fault) และการปรับเปลี่ยนซอฟต์แวร์ตามความต้องการของผู้ใช้หรือลูกค้า โดยระบุเป็นตัววัดเชิงปริมาณ เรียกว่า mean time to repair a default (MTTR)
44.Portability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงระดับความยากง่ายในการย้ายซอฟต์แวร์เพื่อไปใช้งานบน platform อื่น เช่น การใช้ภาษาที่สามารถ run บนหลาย platform (เช่น JAVA) การใช้ compiler และ OS โดยเฉพาะ เพื่อใช้ซอฟต์แวร์กับเครื่องคอมพิวเตอร์อื่นๆ ได้

Contents	มีเนื้อหาใน SRS หรือไม่		ควรมีเนื้อหาอยู่ใน SRS หรือไม่			คำอธิบายของเนื้อหาใน SRS	
	มี	ไม่มี	ไม่ควรมี	ควรมี น้อย ปานกลาง มาก			
Section 4 : Supporting information							
45. Table of contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงสารบัญ เพื่อแสดงถึงหัวข้อต่างๆ ที่มีอยู่ใน SRS
46. Notes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงรายละเอียดต่างๆ ที่ต้องการเพิ่มเติมใน SRS เพื่อให้เกิดความเข้าใจมากขึ้น
47. Appendixes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงภาคผนวก เป็นส่วนที่บรรจุรายละเอียดที่ไม่ได้เขียนอยู่ใน SRS หรือเป็นส่วนที่อธิบายเนื้อหาเพิ่มเติมใน SRS
48. index	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ระบุถึงดัชนี เพื่อสะดวกในการค้นหาคำต่างๆ ที่ต้องการได้อย่างรวดเร็ว

11. ท่านคิดว่า template ของ SRS ในข้อ 10 ควรมี contents ใดเพิ่มเติมหรือไม่

ไม่มี

มี (ระบุ).....

12. ท่านคิดว่า การแบ่ง section ของ template เหมาะสมหรือไม่

ไม่เหมาะสม เพราะ.....

เหมาะสม

13. ข้อเสนอแนะเพิ่มเติมเกี่ยวกับแบบสอบถามนี้ (ถ้ามี).....

.....

.....

.....

.....

บัญชีคำย่อที่ปรากฏในแบบสอบถาม

A/P	:	Account Payable
A/R	:	Account Receivable
ATM	:	Automatic Teller Machine
CAD	:	Computer Aided Design
CAI	:	Computer Aided Instruction
CAM	:	Computer Aided Manufacturing
CASE	:	Computer Aided Software Engineering
DoD	:	The Department of Defense
EDI	:	Electronic Data Interchange
ESA	:	The European Space Agency
FTP	:	File Transfer Protocol
IEEE	:	The Institute of Electrical and Electronics Engineers
MM	:	Man Month
MRP II	:	Manufacturing Resource Planning II
MRP	:	Material Resource Planning
MTBF	:	Mean Time Before Failure
MTTR	:	Mean Time To Repair
NASA	:	The National Aeronautics and Space Administration
OS	:	Operating System
POS	:	Point Of Sales
Q.C.	:	Quality Control
reqs.	:	requirements
SRS	:	Software Requirements Specification
Std	:	Standard
SyRS	:	System Requirements Specification
WWW	:	World Wide Web

ภาคผนวก ค

ในส่วนนี้ แสดงถึงรายชื่อขององค์กรต่างๆ ที่ผู้วิจัยได้ทำการสำรวจข้อมูล ซึ่งจำแนกเป็น 4 กลุ่ม ได้แก่ บริษัทเอกชน หน่วยงานราชการ รัฐวิสาหกิจ และอื่นๆ

บริษัทเอกชน 42 แห่ง ได้แก่

1. Advance Info Service Public Company Limited (AIS)
2. Anderson Consulting Co.,Ltd
3. Berli-Jucker Public Co.,Ltd
4. Bernhof Consultant Group Co.,Ltd
5. CDGS Co.,Ltd
6. Compound Technology Inc.
7. Hewlett-Packard Co.,Ltd
8. IBM Thailand Co.,Ltd
9. Intersol Consulting Co.,Ltd
10. Kernel Computers and Communications Co.,Ltd
11. Lucent Technologies Inc. Thailand
12. ORACLE SYSTEM (THAILAND) CO.,LTD
13. Overseas merchandise inspection
14. Progress Software Co.,Ltd
15. Tetra Pak Processing (Thai) Ltd.
16. Thai information Technology Co.,Ltd
17. โรงพยาบาลนนทเวช (มหาชน) จำกัด
18. ธนาคารกสิกรไทย
19. ธนาคารทหารไทย
20. บริษัท เครือเจริญโภคภัณฑ์ จำกัด (มหาชน)
21. บริษัท เทเลคอมโฮลดิ้ง จำกัด
22. บริษัท เบคไทยกรุงเทพอุปกรณ์เคมีภัณฑ์ จำกัด
23. บริษัท เปเปอร์บ็อกซ์ (ประเทศไทย) จำกัด
24. บริษัท เมโทรซิสเต็ม คอร์ปอเรชั่น จำกัด
25. บริษัท เอเอสแอล จำกัด
26. บริษัท เอ็นซีอาร์ (ประเทศไทย) จำกัด
27. บริษัท เอ็มซีซอฟต์แวร์ (ประเทศไทย) จำกัด
28. บริษัท แลนด์แอนด์เฮาส์ จำกัด (มหาชน)
29. บริษัท ไทยวาโก้ จำกัด (มหาชน)
30. บริษัท ไมโครอิเล็กทรอนิกส์ จำกัด
31. บริษัท ไอโซแพ็ค จำกัด
32. บริษัท กฤษดานคร จำกัด (มหาชน)
33. บริษัท คิวแอนด์เอซอฟต์แวร์ จำกัด
34. บริษัท ซีพี เซเว่นอีเลฟเว่น จำกัด
35. บริษัท ดีไอเดีย คอร์ปอเรชั่น (ไทยแลนด์) จำกัด
36. บริษัท รอยเตอร์ (ประเทศไทย) จำกัด
37. บริษัท สิทธิผลคอมพิวเตอร์แอนด์คอนซัลแตนท์ จำกัด
38. บริษัท หลักทรัพย์เอกอรัญ จำกัด
39. บริษัท หลักทรัพย์กองทุนรวม จำกัด (มหาชน)
40. บริษัท หลักทรัพย์ธนสยาม จำกัด
41. บริษัท อาร์ไอเอส จำกัด
42. บริษัท อีเอสอาร์ไอ (ประเทศไทย) จำกัด

จากบริษัทเอกชน 42 แห่ง จำแนกมาตรฐาน SRS ออกเป็น

- ไม่มีมาตรฐาน SRS ในองค์กร 14 แห่ง
- มีมาตรฐาน SRS ที่พัฒนาขึ้นเองในองค์กร 25 แห่ง
- อ้างอิงมาตรฐานที่มีอยู่แล้ว 3 แห่ง ได้แก่ มาตรฐาน IEEE และ ITIL

หน่วยงานราชการ 9 แห่ง ได้แก่

- | | |
|--|--|
| 1. กรมการขนส่งทหาร | 6. สำนักเทคโนโลยีสารสนเทศ |
| 2. กรมทรัพยากรธรณี | จุฬาลงกรณ์มหาวิทยาลัย |
| 3. ธนาคารแห่งประเทศไทย | 7. สำนักงบประมาณ |
| 4. ศูนย์บริการวิชาการแห่งจุฬาลงกรณ์มหาวิทยาลัย | 8. สำนักงานโครงการส่วนพระองค์สมเด็จพระเทพฯ |
| 5. สถาบันราชภัฏสวนดุสิต | 9. สำนักงานคณะกรรมการกำกับหลักทรัพย์และตลาดหลักทรัพย์ (ก.ล.ต.) |

จากหน่วยงานราชการ 9 แห่ง จำแนกมาตรฐาน SRS ออกเป็น

- ไม่มีมาตรฐาน SRS ในองค์กร 4 แห่ง
- มีมาตรฐาน SRS ที่พัฒนาขึ้นเองในองค์กร 5 แห่ง

รัฐวิสาหกิจ 11 แห่ง ได้แก่

- | | |
|----------------------------------|---|
| 1. การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย | 7. ธนาคารรัตนสิน |
| 2. การไฟฟ้าส่วนภูมิภาค | 8. ธนาคารอาคารสงเคราะห์ |
| 3. การท่าอากาศยานแห่งประเทศไทย | 9. บริษัท การบินไทย จำกัด (มหาชน) |
| 4. การประปานครหลวง | 10. บริษัท วิทยุการบินแห่งประเทศไทย จำกัด |
| 5. การปิโตรเลียมแห่งประเทศไทย | 11. องค์การโทรศัพท์แห่งประเทศไทย |
| 6. ธนาคารเพื่อการเกษตรและสหกรณ์ | |

จากรัฐวิสาหกิจ 11 แห่ง จำแนกมาตรฐาน SRS ออกเป็น

- ไม่มีมาตรฐาน SRS ในองค์กร 4 แห่ง
- มีมาตรฐาน SRS ที่พัฒนาขึ้นเองในองค์กร 6 แห่ง
- อ้างอิงมาตรฐานที่มีอยู่แล้ว 1 แห่ง ได้แก่ มาตรฐาน International Civil Aviation Organization : ICAO

อื่นๆ 1 แห่ง ได้แก่

- ตลาดหลักทรัพย์แห่งประเทศไทย ไม่มีมาตรฐาน SRS ในองค์กร

ภาคผนวก ง

ในส่วนนี้ แสดงถึงแผ่นแบบของ SRS ในระดับเริ่มต้น ระดับปานกลาง และระดับผู้เชี่ยวชาญ พร้อมทั้งแสดงถึงแผ่นแบบของ SRS ในระดับเริ่มต้น ระดับปานกลาง และระดับผู้เชี่ยวชาญ สำหรับบริษัทเอกชน หน่วยงานราชการ และรัฐวิสาหกิจทั้งภาษาไทยและภาษาอังกฤษ

แผ่นแบบของ SRS ในระดับเริ่มต้น ระดับปานกลาง และระดับผู้เชี่ยวชาญที่เป็นภาษาไทย

แผ่นแบบของ SRS ในระดับเริ่มต้น

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

สารบัญญ (Table of contents)

- บทนำ (Introduction)
 - จุดประสงค์ (Purpose)
 - ขอบเขต (Scope)
 - คำจำกัดความ (Definitions)
- รายละเอียดโดยรวม (Overall description)
 - แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - การต่อประสานกับผู้ใช้ (User interfaces)
 - หน้าที่ของซอฟต์แวร์ (Product functions)
 - ข้อจำกัด (Constraints)
- ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - การต่อประสานภายนอก (External interfaces)
 - ความต้องการด้านหน้าที่ (Functional requirements)
 - ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - บทนำ (Introduction)
 - ข้อมูลนำเข้า (Inputs)
 - กระบวนการ (Processing)
 - ข้อมูลนำออก (Outputs)

- 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
- 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
- 3.3 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.3.1 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
- 3.4 ลักษณะประจำ (Attributes)
 - 3.4.1 ความมั่นคง (Security)
- 3.5 ความต้องการอื่นๆ (Other requirements)
 - 3.5.1 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.5.2 ความต้องการด้านการทดสอบ (Verification requirements)
 - 3.5.3 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.5.4 ความต้องการด้านเอกสาร (Documentation requirements)
- 4. ข้อมูลสนับสนุน (Supporting information)

แผ่นแบบของ SRS ในระดับปานกลาง

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....

สารบัญ (Table of contents)

- 1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพจน์ (Acronyms)
 - 1.5 คำย่อ (Abbreviations)
 - 1.6 เอกสารอ้างอิง (References)
 - 1.7 ภาพรวม (Overview)
- 2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)

- 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
- 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
- 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
- 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
- 2.1.6 การดำเนินการ (Operations)
- 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
- 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
- 2.3 ข้อจำกัด (Constraints)
- 2.4 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
- 2.5 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
- 2.6 ความสามารถในการติดตามความต้องการ (Requirements traceability)
- 2.7 การแบ่งสับปันส่วนความต้องการ (Apportioning of requirements)
- 3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 - ⋮
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)
 - 3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.5 ลักษณะประจำ (Attributes)
 - 3.5.1 ความเชื่อถือได้ (Reliability)
 - 3.5.2 สภาพพร้อมใช้งาน (Availability)
 - 3.5.3 ความมั่นคง (Security)
 - 3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)
 - 3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)

3.6 ความต้องการอื่นๆ (Other requirements)

- 3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
- 3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)
- 3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
- 3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)
- 3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
- 3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
- 3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)
- 3.6.8 ความต้องการด้านการทวนสอบ (Verification requirements)
- 3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)
- 3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)
- 3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)
- 3.6.12 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)

4. ข้อมูลสนับสนุน (Supporting information)

- 4.1 บันทึก (Notes)
- 4.2 ภาคผนวก (Appendixes)
- 4.3 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับผู้เชี่ยวชาญ

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....
 สารบัญ (Table of contents)

- 1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพนัย (Acronyms)

- 1.5 ค่าย่อ (Abbreviations)
- 1.6 เอกสารอ้างอิง (References)
- 1.7 ภาพรวม (Overview)
2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
 - 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
 - 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
 - 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
 - 2.1.6 การดำเนินการ (Operations)
 - 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
 - 2.3 ลักษณะของผู้ใช้ (User characteristics)
 - 2.4 ข้อจำกัด (Constraints)
 - 2.5 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
 - 2.6 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
 - 2.7 ความสามารถในการติดตามความต้องการ (Requirements traceability)
 - 2.8 การแบ่งสัดส่วนความต้องการ (Apportioning of requirements)
3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)

3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)

3.5 ลักษณะประจำ (Attributes)

3.5.1 ความเชื่อถือได้ (Reliability)

3.5.2 สภาพพร้อมใช้งาน (Availability)

3.5.3 ความมั่นคง (Security)

3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)

3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)

3.6 ความต้องการอื่นๆ (Other requirements)

3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)

3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)

3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)

3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)

3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)

3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)

3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)

3.6.8 ความต้องการด้านการทวนสอบ (Verification requirements)

3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)

3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)

3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)

3.6.12 ความต้องการด้านสภาพแวดล้อม (Environment requirements)

3.6.13 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)

4. ข้อมูลสนับสนุน (Supporting information)

4.1 บันทึก (Notes)

4.2 ภาคผนวก (Appendixes)

4.3 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับเริ่มต้น ระดับปานกลาง และระดับผู้เชี่ยวชาญ สำหรับบริษัท เอกชน หน่วยงานราชการ และรัฐวิสาหกิจที่เป็นภาษาไทย

แผ่นแบบของ SRS ในระดับเริ่มต้นสำหรับบริษัทเอกชน

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....
สารบัญ (Table of contents)

1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
 - 2.3 ข้อจำกัด (Constraints)
3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 -
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.3 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.3.1 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)

- 3.4 ลักษณะประจำ (Attributes)
 - 3.4.1 ความมั่นคง (Security)
- 3.5 ความต้องการอื่นๆ (Other requirements)
 - 3.5.1 ความต้องการด้านการทดสอบ (Verification requirements)
 - 3.5.2 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.5.3 ความต้องการด้านเอกสาร (Documentation requirements)
- 4. ข้อมูลสนับสนุน (Supporting information)

แผ่นแบบของ SRS ในระดับปานกลางสำหรับบริษัทเอกชน

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....
 สารบัญ (Table of contents)

- 1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพจน์ (Acronyms)
 - 1.5 คำย่อ (Abbreviations)
 - 1.6 เอกสารอ้างอิง (References)
 - 1.7 ภาพรวม (Overview)
- 2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
 - 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
 - 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
 - 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
 - 2.1.6 การดำเนินการ (Operations)
 - 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)

- 2.3 ข้อจำกัด (Constraints)
- 2.4 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
- 2.5 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
- 2.6 ความสามารถในการติดตามความต้องการ (Requirements traceability)
- 2.7 การแบ่งสัดส่วนความต้องการ (Apportioning of requirements)
- 3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 - ...
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)
 - 3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.5 ลักษณะประจำ (Attributes)
 - 3.5.1 ความเชื่อถือได้ (Reliability)
 - 3.5.2 สภาพพร้อมใช้งาน (Availability)
 - 3.5.3 ความมั่นคง (Security)
 - 3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)
 - 3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)
 - 3.6 ความต้องการอื่นๆ (Other requirements)
 - 3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
 - 3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)
 - 3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
 - 3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)

- 3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
 - 3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
 - 3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.6.8 ความต้องการด้านการทวนสอบ (Verification requirements)
 - 3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)
 - 3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)
 - 3.6.12 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)
4. ข้อมูลสนับสนุน (Supporting information)
- 4.1 บันทึก (Notes)
 - 4.2 ภาคผนวก (Appendixes)
 - 4.3 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับผู้เชี่ยวชาญสำหรับบริษัทเอกชน หน่วยงานราชการ และรัฐวิสาหกิจ

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....

สารบัญ (Table of contents)

- 1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพจน์ (Acronyms)
 - 1.5 คำย่อ (Abbreviations)
 - 1.6 เอกสารอ้างอิง (References)
 - 1.7 ภาพรวม (Overview)
- 2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)

- 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
- 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
- 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
- 2.1.6 การดำเนินการ (Operations)
- 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
- 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
- 2.3 ลักษณะของผู้ใช้ (User characteristics)
- 2.4 ข้อจำกัด (Constraints)
- 2.5 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
- 2.6 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
- 2.7 ความสามารถในการติดตามความต้องการ (Requirements traceability)
- 2.8 การแบ่งสัดส่วนความต้องการ (Apportioning of requirements)
- 3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 - ...
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)
 - 3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.5 ลักษณะประจำ (Attributes)
 - 3.5.1 ความเชื่อถือได้ (Reliability)
 - 3.5.2 สภาพพร้อมใช้งาน (Availability)
 - 3.5.3 ความมั่นคง (Security)
 - 3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)
 - 3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)

3.6 ความต้องการอื่นๆ (Other requirements)

- 3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
- 3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)
- 3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
- 3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)
- 3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
- 3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
- 3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)
- 3.6.8 ความต้องการด้านการทวนสอบ (Verification requirements)
- 3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)
- 3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)
- 3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)
- 3.6.12 ความต้องการด้านสภาพแวดล้อม (Environment requirements)
- 3.6.13 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)

4. ข้อมูลสนับสนุน (Supporting information)

- 4.1 บันทึก (Notes)
- 4.2 ภาคผนวก (Appendixes)
- 4.3 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับเริ่มต้นสำหรับหน่วยงานราชการ

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....

สารบัญ (Table of contents)

- 1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)

2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การดำเนินการ (Operations)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.1.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.1.1.1 บทนำ (Introduction)
 - 3.1.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.1.1.3 กระบวนการ (Processing)
 - 3.1.1.4 ข้อมูลนำออก (Outputs)
 - 3.1.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 - ...
 - 3.1.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.2 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.2.1 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.3 ลักษณะประจำ (Attributes)
 - 3.3.1 ความมั่นคง (Security)
 - 3.4 ความต้องการอื่นๆ (Other requirements)
 - 3.4.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
 - 3.4.2 ความต้องการด้านการดำเนินการ (Operational requirements)
 - 3.4.3 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.4.4 ความต้องการด้านการทวนสอบ (Verification requirements)
 - 3.4.5 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.4.6 ความต้องการด้านเอกสาร (Documentation requirements)
4. ข้อมูลสนับสนุน (Supporting information)
 - 4.1 ภาคผนวก (Appendixes)
 - 4.2 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับปานกลางสำหรับหน่วยงานราชการ

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....
 สารบัญ (Table of contents)

1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพจน์ (Acronyms)
 - 1.5 คำย่อ (Abbreviations)
 - 1.6 เอกสารอ้างอิง (References)
 - 1.7 ภาพรวม (Overview)
2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
 - 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
 - 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
 - 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
 - 2.1.6 การดำเนินการ (Operations)
 - 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
 - 2.3 ลักษณะของผู้ใช้ (User characteristics)
 - 2.4 ข้อจำกัด (Constraints)
 - 2.5 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
 - 2.6 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
 - 2.7 ความสามารถในการติดตามความต้องการ (Requirements traceability)
3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)

- 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
- 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
- ...
- 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
- 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
- 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)
 - 3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
- 3.5 ลักษณะประจำ (Attributes)
 - 3.5.1 ความเชื่อถือได้ (Reliability)
 - 3.5.2 สภาพพร้อมใช้งาน (Availability)
 - 3.5.3 ความมั่นคง (Security)
 - 3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)
 - 3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)
- 3.6 ความต้องการอื่นๆ (Other requirements)
 - 3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
 - 3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)
 - 3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
 - 3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)
 - 3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
 - 3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
 - 3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.6.8 ความต้องการด้านการทวนสอบ (Verification requirements)
 - 3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)
 - 3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)

3.6.12 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)

4. ข้อมูลสนับสนุน (Supporting information)

- 4.1 บันทึก (Notes)
- 4.2 ภาคผนวก (Appendixes)
- 4.3 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับเริ่มต้นสำหรับรัฐวิสาหกิจ

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....

สารบัญ (Table of contents)

1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 เอกสารอ้างอิง (References)
2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
 - 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
 - 2.2 ข้อจำกัด (Constraints)
3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)

- 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
- 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
- 3.3 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.3.1 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
- 3.4 ลักษณะประจำ (Attributes)
 - 3.4.1 ความมั่นคง (Security)
- 3.5 ความต้องการอื่นๆ (Other requirements)
 - 3.5.1 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
 - 3.5.2 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.5.3 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.5.4 ความต้องการด้านเอกสาร (Documentation requirements)
- 4. ข้อมูลสนับสนุน (Supporting information)
 - 4.1 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับปานกลางสำหรับรัฐวิสาหกิจ

ชื่อโครงการ

เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์:

โดย:

วันที่:

.....
 สารบัญ (Table of contents)

- 1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพจน์ (Acronyms)
 - 1.5 คำย่อ (Abbreviations)
 - 1.6 เอกสารอ้างอิง (References)
 - 1.7 ภาพรวม (Overview)

2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interfaces)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
 - 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
 - 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
 - 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
 - 2.1.6 การดำเนินการ (Operations)
 - 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
 - 2.3 ข้อจำกัด (Constraints)
 - 2.4 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
 - 2.5 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
 - 2.6 ความสามารถในการติดตามความต้องการ (Requirements traceability)
 - 2.7 การแบ่งสับปันส่วนความต้องการ (Apportioning of requirements)
3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 - ...
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)
 - 3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.5 ลักษณะประจำ (Attributes)
 - 3.5.1 ความเชื่อถือได้ (Reliability)
 - 3.5.2 สภาพพร้อมใช้งาน (Availability)

- 3.5.3 ความมั่นคง (Security)
- 3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)
- 3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)
- 3.6 ความต้องการอื่นๆ (Other requirements)
 - 3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
 - 3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)
 - 3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
 - 3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)
 - 3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
 - 3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
 - 3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.6.8 ความต้องการด้านการทวนสอบ (Verification requirements)
 - 3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)
 - 3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)
 - 3.6.12 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)
- 4. ข้อมูลสนับสนุน (Supporting information)
 - 4.1 บันทึก (Notes)
 - 4.2 ภาคผนวก (Appendixes)
 - 4.3 ดัชนี (Index)

แผ่นแบบของ SRS ในระดับเริ่มต้น ระดับปานกลาง และระดับผู้เชี่ยวชาญที่เป็นภาษาอังกฤษ

Template of Software Requirements Specification (SRS) in beginner level

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.2 Product functions
 - 2.3 Constraints
3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
 - 3.2.2 Functional requirement 2
 - ⋮
 - ⋮
 - 3.2.n Functional requirement n
 - 3.3 Design constraints
 - 3.3.1 Implementation constraints
 - 3.4 Attributes
 - 3.4.1 Security
 - 3.5 Other requirements
 - 3.5.1 Safety requirements
 - 3.5.2 Verification requirements
 - 3.5.3 Acceptance requirements
 - 3.5.4 Documentation requirements
4. Supporting information

Template of Software Requirements Specification (SRS) in intermediate level

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
 - 1.4 Acronyms
 - 1.5 Abbreviations
 - 1.6 References
 - 1.7 Overview
2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.1.2 Hardware interfaces
 - 2.1.3 Software interfaces
 - 2.1.4 Communications interfaces
 - 2.1.5 Memory constraints
 - 2.1.6 Operations
 - 2.1.7 Site adaptation requirements
 - 2.2 Product functions
 - 2.3 Constraints
 - 2.4 Assumptions and dependencies
 - 2.5 Qualification provisions
 - 2.6 Requirements traceability
 - 2.7 Apportioning of requirements
3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements

- 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
- 3.2.2 Functional requirement 2

- 3.2.n Functional requirement n
- 3.3 Performance requirements
- 3.4 Design constraints
 - 3.4.1 Standards compliance
 - 3.4.2 Implementation constraints
- 3.5 Attributes
 - 3.5.1 Reliability
 - 3.5.2 Availability
 - 3.5.3 Security
 - 3.5.4 Maintainability
 - 3.5.5 Portability
- 3.6 Other requirements
 - 3.6.1 Logical database requirements
 - 3.6.2 Operational requirements
 - 3.6.3 Computer hardware resource utilization requirements
 - 3.6.4 Computer communication requirements
 - 3.6.5 Software quality factors
 - 3.6.6 Logistics-related requirements
 - 3.6.7 Safety requirements
 - 3.6.8 Verification requirements
 - 3.6.9 Acceptance requirements
 - 3.6.10 Documentation requirements
 - 3.6.11 Packaging requirements
 - 3.6.12 Training-related requirements
- 4. Supporting information
 - 4.1 Notes

4.2 Appendixes

4.3 Index

Template of Software Requirements Specification (SRS) in expert level

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions

1.4 Acronyms

1.5 Abbreviations

1.6 References

1.7 Overview

2. Overall description

2.1 Product perspective

2.1.1 User interfaces

2.1.2 Hardware interfaces

2.1.3 Software interfaces

2.1.4 Communications interfaces

2.1.5 Memory constraints

2.1.6 Operations

2.1.7 Site adaptation requirements

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

2.6 Qualification provisions

2.7 Requirements traceability

- 2.8 Apportioning of requirements
- 3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
 - 3.2.2 Functional requirement 2
 - ...
 - 3.2.n Functional requirement n
 - 3.3 Performance requirements
 - 3.4 Design constraints
 - 3.4.1 Standards compliance
 - 3.4.2 Implementation constraints
 - 3.5 Attributes
 - 3.5.1 Reliability
 - 3.5.2 Availability
 - 3.5.3 Security
 - 3.5.4 Maintainability
 - 3.5.5 Portability
 - 3.6 Other requirements
 - 3.6.1 Logical database requirements
 - 3.6.2 Operational requirements
 - 3.6.3 Computer hardware resource utilization requirements
 - 3.6.4 Computer communication requirements
 - 3.6.5 Software quality factors
 - 3.6.6 Logistics-related requirements
 - 3.6.7 Safety requirements
 - 3.6.8 Verification requirements
 - 3.6.9 Acceptance requirements
 - 3.6.10 Documentation requirements

- 3.6.11 Packaging requirements
- 3.6.12 Environment requirements
- 3.6.13 Training-related requirements

4. Supporting information

- 4.1 Notes
- 4.2 Appendixes
- 4.3 Index

แม่แบบของ SRS ในระดับเริ่มต้น ระดับปานกลาง และระดับผู้เชี่ยวชาญ สำหรับบริษัท เอกชน หน่วยงานราชการ และรัฐวิสาหกิจที่เป็นภาษาอังกฤษ

Template of Software Requirements Specification (SRS) in beginner level for private company

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
- 2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.2 Product functions
 - 2.3 Constraints
- 3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs

- 3.2.1.3 Processing
- 3.2.1.4 Outputs
- 3.2.2 Functional requirement 2
- ...
- 3.2.n Functional requirement n
- 3.3 Design constraints
 - 3.3.1 Implementation constraints
- 3.4 Attributes
 - 3.4.1 Security
- 3.5 Other requirements
 - 3.5.1 Verification requirements
 - 3.5.2 Acceptance requirements
 - 3.5.3 Documentation requirements
- 4. Supporting information

Template of Software Requirements Specification (SRS) in intermediate level for private company

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
 - 1.4 Acronyms
 - 1.5 Abbreviations
 - 1.6 References
 - 1.7 Overview
- 2. Overall description
 - 2.1 Product perspective

- 2.1.1 User interfaces
- 2.1.2 Hardware interfaces
- 2.1.3 Software interfaces
- 2.1.4 Communications interfaces
- 2.1.5 Memory constraints
- 2.1.6 Operations
- 2.1.7 Site adaptation requirements
- 2.2 Product functions
- 2.3 Constraints
- 2.4 Assumptions and dependencies
- 2.5 Qualification provisions
- 2.6 Requirements traceability
- 2.7 Apportioning of requirements
- 3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
 - 3.2.2 Functional requirement 2
 - .
 - .
 - 3.2.n Functional requirement n
 - 3.3 Performance requirements
 - 3.4 Design constraints
 - 3.4.1 Standards compliance
 - 3.4.2 Implementation constraints
 - 3.5 Attributes
 - 3.5.1 Reliability
 - 3.5.2 Availability
 - 3.5.3 Security
 - 3.5.4 Maintainability

3.5.5 Portability

3.6 Other requirements

- 3.6.1 Logical database requirements
- 3.6.2 Operational requirements
- 3.6.3 Computer hardware resource utilization requirements
- 3.6.4 Computer communication requirements
- 3.6.5 Software quality factors
- 3.6.6 Logistics-related requirements
- 3.6.7 Safety requirements
- 3.6.8 Verification requirements
- 3.6.9 Acceptance requirements
- 3.6.10 Documentation requirements
- 3.6.11 Packaging requirements
- 3.6.12 Training-related requirements

4. Supporting information

- 4.1 Notes
- 4.2 Appendixes
- 4.3 Index

*Template of Software Requirements Specification (SRS) in expert level
for private company, government unit and state enterprises*

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
 - 1.4 Acronyms
 - 1.5 Abbreviations

- 1.6 References
- 1.7 Overview
- 2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.1.2 Hardware interfaces
 - 2.1.3 Software interfaces
 - 2.1.4 Communications interfaces
 - 2.1.5 Memory constraints
 - 2.1.6 Operations
 - 2.1.7 Site adaptation requirements
 - 2.2 Product functions
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies
 - 2.6 Qualification provisions
 - 2.7 Requirements traceability
 - 2.8 Apportioning of requirements
- 3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
 - 3.2.2 Functional requirement 2
 - ...
 - 3.2.n Functional requirement n
 - 3.3 Performance requirements
 - 3.4 Design constraints
 - 3.4.1 Standards compliance
 - 3.4.2 Implementation constraints

3.5 Attributes

- 3.5.1 Reliability
- 3.5.2 Availability
- 3.5.3 Security
- 3.5.4 Maintainability
- 3.5.5 Portability

3.6 Other requirements

- 3.6.1 Logical database requirements
- 3.6.2 Operational requirements
- 3.6.3 Computer hardware resource utilization requirements
- 3.6.4 Computer communication requirements
- 3.6.5 Software quality factors
- 3.6.6 Logistics-related requirements
- 3.6.7 Safety requirements
- 3.6.8 Verification requirements
- 3.6.9 Acceptance requirements
- 3.6.10 Documentation requirements
- 3.6.11 Packaging requirements
- 3.6.12 Environment requirements
- 3.6.13 Training-related requirements

4. Supporting information

- 4.1 Notes
- 4.2 Appendixes
- 4.3 Index

Template of Software Requirements Specification (SRS) in beginner level for government unit

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.1.2 Operations
 - 2.2 Product functions
3. Specific requirements
 - 3.1 Functional requirements
 - 3.1.1 Functional requirement 1
 - 3.1.1.1 Introduction
 - 3.1.1.2 Inputs
 - 3.1.1.3 Processing
 - 3.1.1.4 Outputs
 - 3.1.2 Functional requirement 2
 -
 -
 - 3.1.n Functional requirement n
 - 3.2 Design constraints
 - 3.2.1 Implementation constraints
 - 3.3 Attributes
 - 3.3.1 Security
 - 3.4 Other requirements
 - 3.4.1 Logical database requirements
 - 3.4.2 Operational requirements
 - 3.4.3 Safety requirements
 - 3.4.4 Verification requirements
 - 3.4.5 Acceptance requirements
 - 3.4.6 Documentation requirements
4. Supporting information
 - 4.1 Appendixes
 - 4.2 Index

Template of Software Requirements Specification (SRS) in intermediate level for government unit

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
 - 1.4 Acronyms
 - 1.5 Abbreviations
 - 1.6 References
 - 1.7 Overview
2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.1.2 Hardware interfaces
 - 2.1.3 Software interfaces
 - 2.1.4 Communications interfaces
 - 2.1.5 Memory constraints
 - 2.1.6 Operations
 - 2.1.7 Site adaptation requirements
 - 2.2 Product functions
 - 2.3 User characteristics
 - 2.4 Constraints
 - 2.5 Assumptions and dependencies
 - 2.6 Qualification provisions
 - 2.7 Requirements traceability
3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements

- 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
- 3.2.2 Functional requirement 2
- ...
- 3.2.n Functional requirement n
- 3.3 Performance requirements
- 3.4 Design constraints
 - 3.4.1 Standards compliance
 - 3.4.2 Implementation constraints
- 3.5 Attributes
 - 3.5.1 Reliability
 - 3.5.2 Availability
 - 3.5.3 Security
 - 3.5.4 Maintainability
 - 3.5.5 Portability
- 3.6 Other requirements
 - 3.6.1 Logical database requirements
 - 3.6.2 Operational requirements
 - 3.6.3 Computer hardware resource utilization requirements
 - 3.6.4 Computer communication requirements
 - 3.6.5 Software quality factors
 - 3.6.6 Logistics-related requirements
 - 3.6.7 Safety requirements
 - 3.6.8 Verification requirements
 - 3.6.9 Acceptance requirements
 - 3.6.10 Documentation requirements
 - 3.6.11 Packaging requirements
 - 3.6.12 Training-related requirements
- 4. Supporting information
 - 4.1 Notes

4.2 Appendixes

4.3 Index

Template of Software Requirements Specification (SRS) in beginner level for state enterprises

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
 - 1.4 References
2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.1.2 Hardware interfaces
 - 2.1.3 Software interfaces
 - 2.2 Constraints
3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
 - 3.2.2 Functional requirement 2
 -
 - 3.2.n Functional requirement n

- 3.3 Design constraints
 - 3.3.1 Implementation constraints
- 3.4 Attributes
 - 3.4.1 Security
- 3.5 Other requirements
 - 3.5.1 Logistics-related requirements
 - 3.5.2 Safety requirements
 - 3.5.3 Acceptance requirements
 - 3.5.4 Documentation requirements
- 4. Supporting information
 - 4.1 Index

Template of Software Requirements Specification (SRS) in intermediate level for state enterprises

Project name

Development tools:

Prepared by:

Date:

.....

Table of contents

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions
 - 1.4 Acronyms
 - 1.5 Abbreviations
 - 1.6 References
 - 1.7 Overview
- 2. Overall description
 - 2.1 Product perspective
 - 2.1.1 User interfaces
 - 2.1.2 Hardware interfaces
 - 2.1.3 Software interfaces
 - 2.1.4 Communications interfaces

- 2.1.5 Memory constraints
- 2.1.6 Operations
- 2.1.7 Site adaptation requirements
- 2.2 Product functions
- 2.3 Constraints
- 2.4 Assumptions and dependencies
- 2.5 Qualification provisions
- 2.6 Requirements traceability
- 2.7 Apportioning of requirements
- 3. Specific requirements
 - 3.1 External interfaces
 - 3.2 Functional requirements
 - 3.2.1 Functional requirement 1
 - 3.2.1.1 Introduction
 - 3.2.1.2 Inputs
 - 3.2.1.3 Processing
 - 3.2.1.4 Outputs
 - 3.2.2 Functional requirement 2
 - .
 - .
 - 3.2.n Functional requirement n
 - 3.3 Performance requirements
 - 3.4 Design constraints
 - 3.4.1 Standards compliance
 - 3.4.2 Implementation constraints
 - 3.5 Attributes
 - 3.5.1 Reliability
 - 3.5.2 Availability
 - 3.5.3 Security
 - 3.5.4 Maintainability
 - 3.5.5 Portability
 - 3.6 Other requirements
 - 3.6.1 Logical database requirements
 - 3.6.2 Operational requirements

- 3.6.3 Computer hardware resource utilization requirements
- 3.6.4 Computer communication requirements
- 3.6.5 Software quality factors
- 3.6.6 Logistics-related requirements
- 3.6.7 Safety requirements
- 3.6.8 Verification requirements
- 3.6.9 Acceptance requirements
- 3.6.10 Documentation requirements
- 3.6.11 Packaging requirements
- 3.6.12 Training-related requirements

4. Supporting information

- 4.1 Notes
- 4.2 Appendixes
- 4.3 Index

ภาคผนวก จ

ในส่วนนี้ แสดงถึงคำอธิบายของเนื้อหาใน SRS และตัวอย่างในการเขียน SRS ของแต่ละเนื้อหาทั้งภาษาไทยและภาษาอังกฤษ

คำอธิบายของเนื้อหาใน SRS ที่เป็นภาษาไทย

เอกสารข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification: SRS) คือเอกสารที่ถูกพัฒนาขึ้นมา หลังจากทราบความต้องการของผู้ใช้หรือลูกค้าแล้ว ซึ่งผู้พัฒนาซอฟต์แวร์เขียนรายละเอียดต่างๆ เกี่ยวกับซอฟต์แวร์ในเอกสารนี้ เพื่ออธิบายถึงหน้าที่และการทำงานของซอฟต์แวร์ โดยจะต้องครอบคลุมถึงความต้องการของผู้ใช้หรือลูกค้าด้วย

ส่วนที่ 1 : บทนำ (Introduction)

1. จุดประสงค์ (Purpose)

ระบุถึงจุดประสงค์ของการเขียน SRS หรือความตั้งใจที่เฉพาะเจาะจงของผู้ที่จะใช้ SRS ฉบับนี้

2. ขอบเขต (Scope)

ระบุถึงขอบเขตของซอฟต์แวร์ ประกอบด้วย ชื่อของซอฟต์แวร์ (เช่น ระบบ ABC) หน้าที่ของซอฟต์แวร์ SRS ต้องมีความสอดคล้องกับเอกสารระดับสูง (เช่น System Requirements Specification : SyRS) และอธิบายถึงการนำเอาซอฟต์แวร์ไปประยุกต์ใช้ให้บรรลุวัตถุประสงค์ ประโยชน์ที่คาดว่าจะได้รับ และเป้าหมายที่ตั้งไว้

3. คำจำกัดความ (Definitions)

ระบุถึงคำจำกัดความของคำศัพท์ต่างๆ ที่ใช้ใน SRS เพื่อให้เข้าใจความหมายของคำศัพท์ที่ตรงกัน อาจระบุคำจำกัดความไว้ในส่วนของภาคผนวก หรือใช้อ้างอิงไปยังเอกสารอื่นๆ ได้

4. รัสพจน์ (Acronyms)

ระบุถึงคำย่อของคำศัพท์ต่างๆ ที่ใช้ใน SRS โดยนำเอาอักษรตัวแรกของคำศัพท์มาย่อ เพื่อให้สะดวกในการใช้งาน (เช่น SRS ย่อมาจาก Software Requirements Specification) อาจระบุคำย่อนี้ไว้ในส่วนของภาคผนวก หรือใช้อ้างอิงไปยังเอกสารอื่นๆ ได้

5. คำย่อ (Abbreviations)

ระบุถึงคำย่อของคำศัพท์ต่างๆ ที่ใช้ใน SRS โดยไม่มีกฎเกณฑ์ในการย่อ เพื่อให้สะดวกในการใช้งาน (เช่น Std ย่อมาจาก Standard) อาจระบุคำย่อนี้ไว้ในส่วนของภาคผนวก หรือใช้อ้างอิงไปยังเอกสารอื่นๆ ได้

6. เอกสารอ้างอิง (References)

ระบุถึงเอกสารอ้างอิงที่ใช้ใน SRS โดยเอกสารแต่ละชุด ประกอบด้วย ชื่อ หมายเลขรายงาน (ถ้ามี) (เช่น IEEE Std 830-1993) วันที่ สถานที่พิมพ์ พร้อมทั้งระบุแหล่งที่มาของเอกสารอ้างอิงด้วย

7. ภาพรวม (Overview)

ระบุถึงเนื้อหาโดยรวมที่อยู่ใน SRS เป็นการมองภาพรวมของเอกสาร และอธิบายว่า SRS มีการจัดระเบียบของเอกสารอย่างไร

ส่วนที่ 2 : รายละเอียดโดยรวม (Overall description)

8. การต่อประสานกับผู้ใช้ (User interfaces)

ระบุถึงการต่อประสานระหว่างซอฟต์แวร์กับผู้ใช้ เช่น รูปแบบหน้าจอ โครงร่างของรายงาน (page layouts) เนื้อหาของรายงานหรือเมนู การกำหนดคีย์พิเศษของแป้นพิมพ์ (เช่น ฟังก์ชันคีย์) เป็นต้น

9. การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)

ระบุถึงการต่อประสานระหว่างซอฟต์แวร์กับส่วนประกอบของฮาร์ดแวร์ เช่น number of ports, instruction set, devices และ protocols เป็นต้น

10. การต่อประสานกับซอฟต์แวร์ (Software interfaces)

ระบุถึงการต่อประสานระหว่างซอฟต์แวร์กับระบบอื่นๆ เช่น การเชื่อมโยงระหว่างระบบบัญชีลูกหนี้กับระบบบัญชีแยกประเภททั่วไป โดยซอฟต์แวร์แต่ละชนิด ประกอบด้วย ชื่อ ชื่อย่อ(mnemonic) หมายเลขรุ่น (version) และแหล่งที่มา

11. การต่อประสานกับระบบสื่อสาร (Communications interfaces)

ระบุถึงการต่อประสานกับระบบสื่อสารในรูปแบบต่างๆ เช่น local network protocols เป็นต้น

12. ข้อจำกัดของหน่วยความจำ (Memory constraints)

ระบุถึงลักษณะและข้อจำกัดของหน่วยความจำหลักและหน่วยความจำรอง

13. การดำเนินการ (Operations)

ระบุถึงการดำเนินการใช้งานซอฟต์แวร์แบบปกติและแบบกรณีพิเศษ เช่น การดำเนินการของผู้ใช้เพื่อเริ่มต้นใช้ระบบ ระยะเวลาของการดำเนินการแบบ interactive ระยะเวลาของการดำเนินการที่ไม่มีการสนใจเครื่องหรือระบบ ฟังก์ชันสนับสนุนการประมวลผลข้อมูล (เช่น การเรียงลำดับข้อมูล) การดำเนินการสำรองและกู้คืนข้อมูล เป็นต้น

14. ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)

ระบุถึงที่ตั้งของการติดตั้งซอฟต์แวร์ รวมทั้งวิธีการดำเนินการที่ใช้กับซอฟต์แวร์ เช่น พิจารณาถึง config เพื่อให้ซอฟต์แวร์เข้ากับเครื่องคอมพิวเตอร์นั้นๆ ได้ พิจารณาถึง platform จำนวน node และ port ที่ใช้กับซอฟต์แวร์นี้ เป็นต้น ถ้าที่ตั้งอยู่ในสภาพแวดล้อมที่ไม่เหมาะสม อาจเปลี่ยนแปลงสภาพแวดล้อมให้เหมาะสมได้

15. หน้าที่ของซอฟต์แวร์ (Product functions)

ระบุถึงหน้าที่หลักของซอฟต์แวร์ เพื่อสร้างความเข้าใจให้กับผู้ใช้หรือลูกค้าและคนอื่นที่อ่านเอกสารนี้เป็นครั้งแรก โดยอาจใช้ข้อความหรือรูปภาพก็ได้

16. ลักษณะของผู้ใช้ (User characteristics)

ระบุถึงลักษณะโดยทั่วไปของผู้ใช้ รวมถึงระดับการศึกษา ประสบการณ์ และการชำนาญทางเทคนิค

17. ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)

ระบุถึงกระบวนการที่จะนำมาพิจารณาซอฟต์แวร์ว่าซอฟต์แวร์มีความเหมาะสมในการใช้งานหรือไม่ เช่น ความสะดวก ค่าใช้จ่าย ระยะเวลา เป็นต้น

18. ความสามารถในการติดตามความต้องการ (Requirements traceability)

ระบุถึงความสามารถในการตรวจสอบความต้องการกลับไปได้ว่าความต้องการนั้นๆ มีจุดกำเนิดมาจากใคร มีจุดประสงค์อะไร เพื่อใช้อ้างอิงต่อไปในอนาคต เช่น เมื่อมีการเปลี่ยนแปลงความต้องการ ทำให้ผู้พัฒนาซอฟต์แวร์ต้องทำการ update ซอฟต์แวร์ ซึ่งผู้พัฒนาซอฟต์แวร์สามารถตรวจสอบความต้องการนั้นๆ กลับไปได้ว่าใครเป็นผู้พัฒนาก่อนหน้านี้ เพื่อสอบถามข้อมูลต่างๆ จากผู้นั้นได้

19. ข้อจำกัด (Constraints)

ระบุถึงคำอธิบายและข้อจำกัดในการพัฒนาซอฟต์แวร์ที่ควรคำนึงถึง คือ นโยบายต่างๆ ไป (เช่น หน่วยความจำ ฮาร์ดดิสก์) ข้อจำกัดทางฮาร์ดแวร์ การต่อประสานไปยังระบบอื่น การปฏิบัติการแบบขนาน หน้าที่การตรวจสอบระบบ (เช่น ทำ log file) หน้าที่การควบคุมระบบ (เช่น ทำ polling) ความต้องการในเรื่อง higher-order language signal handshake protocols (เช่น XON-XOFF, ACK-NACK) ความต้องการความเชื่อถือได้ ความวิกฤตในการประยุกต์ใช้ระบบ (เช่น มี transaction เข้ามาในระบบมากๆ) และพิจารณาถึงความปลอดภัยและความมั่นคง

20. ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)

ระบุถึงปัจจัยต่างๆ ที่ส่งผลกระทบต่อความต้องการที่ระบุไว้ใน SRS เช่น ตั้งข้อสันนิษฐานว่าจะใช้ระบบการจัดการฐานข้อมูล ORACLE 8.0 กับซอฟต์แวร์นี้ แต่ในความเป็นจริง เนื่องจาก ORACLE 8.0 ยังพัฒนาไม่เสร็จ ทำให้ซอฟต์แวร์ที่พัฒนาขึ้นมาไม่สามารถใช้กับระบบการจัดการฐานข้อมูล ORACLE 8.0 ได้ เป็นต้น ดังนั้นความต้องการที่ระบุไว้ใน SRS ต้องได้รับการเปลี่ยนแปลงด้วย

21. การแบ่งสัดส่วนส่วนความต้องการ (Apportioning of requirements)

ระบุถึงความต้องการที่ผู้ใช้หรือลูกค้ายอมให้เกิดความล่าช้า หรือเป็นความต้องการที่คาดว่าจะไม่สามารถพัฒนาเสร็จในเวอร์ชันนี้ และจะทำการพัฒนาซอฟต์แวร์ในเวอร์ชันต่อไปในอนาคต เพื่อให้ทำหน้าที่ครอบคลุมตามความต้องการของผู้ใช้หรือลูกค้าได้อย่างครบถ้วน

ส่วนที่ 3 : ความต้องการที่เฉพาะเจาะจง (Specific requirements)

22. การต่อประสานภายนอก (External interfaces)

ระบุถึงรายละเอียดของข้อมูลที่เข้าไปยังระบบและข้อมูลที่ออกจากระบบทั้งหมด ประกอบด้วย ชื่อของตัวข้อมูล (item) จุดประสงค์ แหล่งที่มาของข้อมูล ช่วงของข้อมูลที่ถูกต้อง (valid) หน่วยของข้อมูล (เช่น จำนวนเพิ่มข้อมูล จำนวน record จำนวนเอกสาร) ระยะเวลา ความสัมพันธ์ระหว่างข้อมูลเข้าและข้อมูลออก รูปแบบหน้าจอ รูปแบบวินโดวส์ รูปแบบของข้อมูล รูปแบบของคำสั่งงาน ข้อความจบ (end messages)

23. ความต้องการด้านหน้าที่ (Functional requirements)

ระบุถึงหน้าที่ของซอฟต์แวร์ว่าซอฟต์แวร์ต้องทำงานอย่างไร เมื่อมีการรับข้อมูลเข้า ทำการประมวลผล และสร้างข้อมูลออก โดยอธิบายในลักษณะของประโยคที่ว่า "ระบบควรที่จะ....." ประกอบด้วย การตรวจสอบ

ความถูกต้องของข้อมูลเข้า ลำดับที่แน่นอนของการดำเนินการ การตอบสนองต่อเหตุการณ์ที่ผิดปกติ (เช่น การเกิด overflow การจัดการด้านการสื่อสาร การจัดการในส่วนที่เป็น error และสามารถแก้ไขได้) ความสัมพันธ์ระหว่างข้อมูลเข้าและข้อมูลออก (เช่น ลำดับของข้อมูลเข้าและข้อมูลออก การแปลงข้อมูลเข้าไปเป็นข้อมูลออก)

24. ความต้องการด้านประสิทธิภาพ (Performance requirements)

ระบุถึงค่าที่ใช้วัดประสิทธิภาพของซอฟต์แวร์ โดยใช้กับตัวแปรที่สามารถวัดค่าได้ เช่น จำนวนของผู้ใช้ ขนาดของข้อมูลที่อยู่ในฐานข้อมูล เวลาที่ใช้ในการโต้ตอบระหว่างผู้ใช้กับระบบ (response time) ปริมาณงาน (throughput) เป็นต้น

25. ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)

ระบุถึงความต้องการเชิงตรรกะของข้อมูลที่จะจัดเก็บในฐานข้อมูล ประกอบด้วย ชนิดของข้อมูล ความถี่ของการใช้งาน วิธีในการเข้าถึงข้อมูล (เช่น SELECT, UPDATE, INSERT) ระบุ data entity และความสัมพันธ์ระหว่าง entity integrity constraints (เช่น เมื่อมีการ UPDATE ข้อมูล ต้องทำการ UPDATE ข้อมูลทุกตารางที่สัมพันธ์กัน) data retention requirements (เช่น กำหนดระยะเวลาในการเก็บข้อมูลไว้จนกว่าไม่ใช่ข้อมูลนั้นหรือลบข้อมูลนั้นทิ้ง)

26. ความต้องการด้านการดำเนินการ (Operational requirements)

ระบุถึงวิธีการดำเนินงานของซอฟต์แวร์และวิธีการติดต่อระหว่างผู้ใช้กับระบบ เช่น โครงร่างของหน้าจอ ข้อความผิดพลาด ระบบช่วยเหลือ เป็นต้น ซึ่งเนื้อหาในส่วนนี้จะจะเป็นประโยชน์ในการนิยามถึง semantics และ syntax ของคำสั่ง

27. ความต้องการด้านการทวนสอบ (Verification requirements)

ระบุถึงวิธีการตรวจสอบความถูกต้องของซอฟต์แวร์ โดยอาจใช้วิธีการจำลองระบบ (simulation/emulation) การทดสอบในสภาพแวดล้อมจริง โดยใช้ข้อมูลที่จำลองขึ้นมาหรือข้อมูลจริง และทำการทดสอบการต่อประสานไปยังระบบอื่นๆ ภายนอก

28. ความต้องการด้านการยอมรับ (Acceptance requirements)

ระบุถึงข้อกำหนดในการตรวจรับซอฟต์แวร์เพื่อดูว่าซอฟต์แวร์สามารถทำหน้าที่ได้ตรงตามความต้องการของผู้ใช้หรือลูกค้าหรือไม่

29. ความต้องการด้านเอกสาร (Documentation requirements)

ระบุถึงเอกสารที่ใช้ในการพัฒนาซอฟต์แวร์ ซึ่งอยู่นอกเหนือจากเอกสารที่กำหนดไว้แล้วในมาตรฐาน เช่น รูปแบบโดยละเอียดของเอกสารคู่มือผู้ใช้ซอฟต์แวร์ (Software User Manual) เป็นต้น

30. ความต้องการด้านความปลอดภัย (Safety requirements)

ระบุถึงความปลอดภัยในระบบ เช่น ระบุถึงฟังก์ชันที่วิกฤต เพื่อลดโอกาสที่จะทำให้เกิดความเสียหายจากความผิดพลาดของซอฟต์แวร์ ซึ่งอาจเป็นอันตรายต่อคนหรือทรัพย์สิน

31. ความต้องการด้านสภาพแวดล้อม (Environment requirements)

ระบุถึงสภาพแวดล้อมในการติดตั้งซอฟต์แวร์ เช่น อุณหภูมิ ความชื้น ลม ฝน ฝุ่น สัญญาณรบกวน สัญญาณแม่เหล็กไฟฟ้า ความสะอาดภายในห้องปฏิบัติการคอมพิวเตอร์ เป็นต้น เพื่อเป็นการอำนวยความสะดวกในการใช้ซอฟต์แวร์และฮาร์ดแวร์

32. ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)

ระบุถึงการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ภายในระบบ เพื่อให้เกิดประโยชน์สูงสุดในการใช้งาน เช่น ความสามารถในการใช้โปรเซสเซอร์ หน่วยความจำ อุปกรณ์ข้อมูลนำเข้าหรือออก อุปกรณ์หน่วยความจำรอง อุปกรณ์การสื่อสารข้อมูล อุปกรณ์เครือข่าย เป็นต้น

33. ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communications requirements)

ระบุถึงการสื่อสารข้อมูลที่ใช้ภายในซอฟต์แวร์

34. ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)

ระบุถึงคุณลักษณะ (attribute) ของซอฟต์แวร์ที่ทำให้ซอฟต์แวร์มีความเหมาะสมในการใช้งาน ได้แก่ ความถูกต้อง (correct) ไม่มีความคลุมเครือ (unambiguous) ความสมบูรณ์ (complete) ความดั่งกัน (consistent) ความคงที่ (stability) สามารถทวนสอบได้ (verification) สามารถแก้ไขได้ (modification) และสามารถติดตามได้ (traceable) ในหัวข้อนี้ จะไม่รวมถึงคุณลักษณะหลักๆ ที่เกี่ยวกับคุณภาพ (เช่น ความเชื่อถือได้ ความสามารถในการบำรุงรักษา ความปลอดภัย) โดยอาจจะแสดงถึงคุณลักษณะเหล่านี้ในรูปแบบของเมตริกซ์ (metrics) ได้

35. ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)

ระบุถึงความต้องการในการฝึกอบรมสำหรับบุคคลที่เกี่ยวข้องในระบบ แบ่งเป็น 2 ประเภท คือ การฝึกอบรมสำหรับบุคลากรในทีมพัฒนาซอฟต์แวร์และการฝึกอบรมสำหรับผู้ใช้หรือลูกค้า การฝึกอบรมสำหรับบุคลากรในทีมพัฒนาซอฟต์แวร์ แบ่งการฝึกอบรมเป็น 2 ส่วน ได้แก่ ด้านเทคนิค (เช่น การใช้โปรแกรม การใช้เครื่องมือเพื่อทดสอบโปรแกรม การใช้เทอร์มินัล การต่อประสานไปยังระบบอื่นที่เกี่ยวข้องกับซอฟต์แวร์) และด้านไม่ใช่เทคนิค (เช่น การจัดการ procedures การควบคุมในเรื่องเอกสาร) การฝึกอบรมสำหรับผู้ใช้หรือลูกค้า มีการอบรมเกี่ยวกับการติดตั้งซอฟต์แวร์ การใช้ซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์ และการเปลี่ยนแปลงแก้ไขซอฟต์แวร์

36. ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)

ระบุถึงขั้นตอนต่างๆ ที่เกี่ยวข้องในการนำซอฟต์แวร์มาใช้งาน (implementation) การสนับสนุนการใช้ซอฟต์แวร์ (support) และการบำรุงรักษาซอฟต์แวร์ (maintenance) รวมถึงการจัดหาทรัพยากร เพื่อดำเนินการตามขั้นตอนดังกล่าวด้วย

37. ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)

ระบุถึงสื่อที่ใช้ในการเก็บบันทึกซอฟต์แวร์ (เช่น ดิสก์ เทป ซีดีรอม) เอกสารประกอบการใช้งานของซอฟต์แวร์ และอาจรวมถึงตัวต่อประสานที่จำเป็นต้องใช้กับซอฟต์แวร์ (เช่น I/O drivers)

38. ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)

ระบุถึงข้อบังคับที่ใช้ในการออกแบบและการใช้ซอฟต์แวร์และฮาร์ดแวร์ร่วมกัน เพื่อสนับสนุนการใช้งานของซอฟต์แวร์ที่พัฒนา เช่น ใช้ตัวแปลภาษาโดยเฉพาะ หรือ ซอฟต์แวร์ต้อง run โดยใช้ระบบฐานข้อมูล หรือ ซอฟต์แวร์ต้องมีขนาดพอดีกับหน่วยความจำ 512 Kbytes เป็นต้น

39. มาตรฐานที่มีใช้ในองค์กร (Standard compliance)

ระบุถึงกฎระเบียบหรือมาตรฐานที่มีใช้อยู่แล้วในองค์กร ซึ่งจะนำมาใช้ในการพัฒนาซอฟต์แวร์ เช่น รูปแบบรายงาน การตั้งชื่อข้อมูล เป็นต้น ถ้ามีการใช้มาตรฐานอื่น ควรระบุไว้ในส่วนนี้ด้วย

40. ความเชื่อถือได้ (Reliability)

ระบุถึงเวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้น เรียกว่า mean time before failure (MTBF) เพื่อสร้างความเชื่อถือได้ในการส่งมอบซอฟต์แวร์ให้กับลูกค้า

41. สภาพพร้อมใช้งาน (Availability)

ระบุถึงปัจจัยที่จำเป็นต้องมี เพื่อรองรับถึงสภาพพร้อมใช้งานของซอฟต์แวร์โดยรวมในระดับที่กำหนดไว้ เช่น checkpoint, recovery, restart เป็นต้น

42. ความมั่นคง (Security)

ระบุถึงวิธี หรือเทคนิคต่างๆ เพื่อป้องกันเหตุที่เกิดจากความมุงร้ายในการเข้าถึงข้อมูล เช่น กระบวนการในการเข้าสู่ระบบและป้อนรหัสผ่าน ข้อจำกัดในการใช้งานระบบของผู้ใช้ (เช่น ผู้ใช้สามารถป้อนรหัสผ่านได้ไม่เกิน 3 ครั้ง) การป้องกันไวรัสคอมพิวเตอร์ การใช้ประโยชน์จากเทคนิควิทยาการเข้ารหัสลับ (cryptographical techniques) การทำ log file หรือเก็บ historical data การจำกัดการสื่อสารระหว่างบางพื้นที่ของโปรแกรม การตรวจสอบความสมบูรณ์ของข้อมูล (เช่น เมื่อมีการ UPDATE ข้อมูล ต้องตรวจสอบว่ามีการ UPDATE ข้อมูลทุกตารางที่สัมพันธ์กันหรือไม่) การเข้าถึงข้อมูลแบบอ่านอย่างเดียว การสำรองและกู้คืนข้อมูล เป็นต้น

43. ความสามารถในการบำรุงรักษา (Maintainability)

ระบุถึงระดับความยากง่ายในการแก้ไขความผิดพลาด (fault) และการปรับเปลี่ยนซอฟต์แวร์ตามความต้องการของผู้ใช้หรือลูกค้า โดยระบุเป็นตัววัดเชิงปริมาณ เรียกว่า mean time to repair a fault (MTTR)

44. ความสามารถในการใช้ได้หลายระบบ (Portability)

ระบุถึงระดับความยากง่ายในการย้ายซอฟต์แวร์ เพื่อไปใช้งานบน platform อื่น เช่น เปรอ์เซนต์ของการใช้คำสั่งที่ทำงานได้เฉพาะบน platform ที่กำหนด การใช้ภาษาที่สามารถ run บนหลาย platform (เช่น JAVA) การใช้ compiler และระบบปฏิบัติการโดยเฉพาะ เพื่อสามารถใช้ซอฟต์แวร์กับเครื่องคอมพิวเตอร์อื่นๆ ได้ เป็นต้น

ส่วนที่ 4 : ข้อมูลสนับสนุน (Supporting information)

45. สารบัญ (Table of contents)

ระบุถึงสารบัญ เพื่อแสดงถึงหัวข้อต่างๆ ที่มีอยู่ใน SRS

46. บันทึก (Notes)

ระบุถึงรายละเอียดต่างๆ ที่ต้องการเพิ่มเติมใน SRS เพื่อให้เกิดความเข้าใจมากขึ้น

47. ภาคผนวก (Appendixes)

ระบุถึงภาคผนวก เป็นส่วนที่บรรจุรายละเอียดที่ไม่ได้เขียนอยู่ใน SRS หรือเป็นส่วนที่อธิบายเนื้อหาเพิ่มเติมใน SRS อาจแบ่งแยกเป็นส่วนๆ ได้ เช่น template ต่างๆ คำอธิบายของการศึกษาการวิเคราะห์ค่าใช้จ่าย

จ่าย หรือผลลัพธ์ของการสำรวจผู้ใช้ ข้อมูลที่ช่วยให้ผู้อ่านอ่าน SRS ได้เข้าใจมากขึ้น รูปแบบของข้อมูลเข้าและข้อมูลออก คำอธิบายของการแก้ไขปัญหาที่คาดว่าจะเกิดขึ้นหลังจากใช้ซอฟต์แวร์แล้ว เป็นต้น

48. ดัชนี (Index)

ระบุถึงดัชนี เพื่อสะดวกในการค้นหาคำต่างๆ ที่ต้องการได้อย่างรวดเร็ว

ตัวอย่างในการเขียน SRS ของแต่ละเนื้อหาที่เป็นภาษาไทย

ส่วนที่ 1 : บทนำ (Introduction)

1. จุดประสงค์ (Purpose)

- จุดประสงค์ของ SRS ฉบับนี้ เพื่อระบุถึงความต้องการในการพัฒนาระบบควบคุมดาวเทียม โดยใช้แนวความคิดของซอฟต์แวร์
- SRS ฉบับนี้ อธิบายถึงความต้องการด้านหน้าที่และความต้องการด้านประสิทธิภาพในระบบย่อยของระบบสั่งซื้อสินค้า
- เพื่อพัฒนาเครื่องมือช่วยวิศวกรรมความต้องการ (Computer Aided Requirements Engineering: CARE) ซึ่งเป็นการลดความคลุมเครือ สร้างความชัดเจน ความตึงกัน ความถูกต้อง และวิธีการในการแก้ปัญหาต่างๆ ใน SRS

2. ขอบเขต (Scope)

- ใน SRS ฉบับนี้ อธิบายถึงการพัฒนาซอฟต์แวร์สำเร็จรูป 2 โปรแกรม ได้แก่ โปรแกรมแรก คือ โปรแกรมสำหรับการสื่อสารดาวเทียม และโปรแกรมที่สอง คือ โปรแกรมสำหรับการหาตำแหน่งของดาวเทียม
- ระบบย่อยของระบบสั่งซื้อสินค้า เป็นส่วนประกอบหนึ่งของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอปซี คำอธิบายของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติอยู่ในเอกสารที่มีชื่อเรื่องว่า "คำอธิบายการออกแบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ ครั้งที่ 4 ตุลาคม 1996"
- ข้อกำหนดฉบับนี้ ใช้เป็นพื้นฐานในการออกแบบซอฟต์แวร์ ซึ่งสามารถสนับสนุนการทำงานของเครื่องคอมพิวเตอร์ได้ถึง 100 เครื่อง โดยผู้ใช้ได้รับการฝึกอบรมในการใช้เครื่องคอมพิวเตอร์เหล่านี้
- เพื่อพัฒนาเครื่องมือ ซึ่งประยุกต์ใช้มาตรฐาน IEEE เป็นพื้นฐาน ได้แก่ มาตรฐาน IEEE Std 830-1993 และ มาตรฐาน IEEE Std 1233-1996

3. คำจำกัดความ (Definitions)

- ลูกค้า -- เป็นผู้ที่ต้องการซอฟต์แวร์ที่ได้รับการพัฒนาแล้ว
- ผู้พัฒนา -- เป็นผู้ที่ทำหน้าที่พัฒนาซอฟต์แวร์ เพื่อให้ซอฟต์แวร์มีหน้าที่ตรงตามความต้องการของผู้ใช้หรือลูกค้า
- ผู้ใช้ -- เป็นผู้ที่ใช้ประโยชน์จากซอฟต์แวร์ที่สมบูรณ์แล้ว ซึ่งอาจเป็นคนเดียวกับลูกค้าก็ได้

4. รัสพจน์ (Acronyms)

- DBLT -- Destination Bandwidth Lookup Table

- EPM -- Engineering Practices Manual
- IEEE -- Institute of Electrical and Electronics Engineers
- SRS -- Software Requirements Specification

5. คำย่อ (Abbreviations)

- <Comm Msg> -- Command Message
- Kbps -- Kilobytes per second
- OP Code -- Operational Code
- Std -- Standard

6. เอกสารอ้างอิง (References)

- EPM for Software, Lockheed Martin, Orlando FL, 1982.
- IEEE Std 610.12-1990, IEEE Standard Glossary for Software Engineering Terminology (ANSI).
- IEEE Std-830, IEEE New York, 1984 and 1993.

7. ภาพรวม (Overview)

- ในส่วนนี้ อธิบายถึงรายละเอียดต่างๆ และความต้องการของซอฟต์แวร์ที่จะทำการพัฒนา ใน SRS ฉบับนี้ได้มีการจัดระเบียบของเอกสารโดยอ้างอิงจากมาตรฐาน IEEE Std 830 ในปี 1984 และ 1993 และอ้างอิงจากมาตรฐาน Lockheed Martins Software EPM เพื่อผลิตเอกสารในการออกแบบซอฟต์แวร์
- ผลลัพธ์ของงานวิจัยนี้ เพื่อสร้างเครื่องมือวิศวกรรมความต้องการ หรือที่เรียกว่า 'CARE' ซึ่งมีหน้าที่การทำงาน คือ แก้ไขความต้องการ เพิ่มความต้องการ ปรับให้ทันสมัย (update) ลบความต้องการและแสดงภาพรวมแบบลำดับขั้นตอนในรูปแบบของต้นไม้

ส่วนที่ 2 : รายละเอียดโดยรวม (Overall description)

8. การต่อประสานกับผู้ใช้ (User interfaces)

- การต่อประสานกับผู้ใช้ของระบบควบคุมดาวเทียม ได้แก่ หน้าจอคอมพิวเตอร์ คีย์บอร์ดหรือเมาส์ และสายอากาศของผู้ส่ง (sender antenna)
- CARE มีการออกแบบเป็น Graphic User Interface : GUI โดยเครื่องมือนี้มีเมนูบาร์ และไอคอนสำหรับรับคำสั่งต่างๆ จากผู้ใช้ เพื่อให้ผู้ใช้ที่มีความรู้หรือทักษะทางด้านคอมพิวเตอร์น้อยมากได้ใช้ระบบนี้อย่างสะดวก

9. การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)

- การต่อประสานกับฮาร์ดแวร์ คือ การต่อประสานกับด้ามจับของสายอากาศ ดังนั้น จึงไม่มีโปรโตคอลเครือข่ายในการจำลองระบบนี้
- CARE จะพิมพ์รายงานออกทางเครื่องพิมพ์ ดังนั้น เครื่องคอมพิวเตอร์และเครื่องพิมพ์ต้องมีการต่อประสานทางด้านฮาร์ดแวร์อย่างง่าย

10. การต่อประสานกับซอฟต์แวร์ (Software interfaces)

- การต่อประสานกับซอฟต์แวร์ของระบบควบคุมดาวเทียม คือการต่อประสานกับระบบผู้ส่ง (sender) และระบบควบคุมภาคพื้นดิน (Ground Control)
- ระบบย่อยของระบบสั่งซื้อสินค้าจะต่อประสานกับระบบย่อย ดังต่อไปนี้
 - ก) คลังสินค้า (Warehouse)
 - ข) การบัญชี (Accounting)
 - ค) การตลาดและการขาย (Marketing and sales)
 - ง) สมุดแจ้งรายการสินค้า (Catalogue)
 - จ) ควบคุมเครือข่าย (Network control)
- เนื่องจาก CARE เป็นเวอร์ชันแรก ดังนั้น จึงไม่มีการติดต่อไปยังซอฟต์แวร์อื่น แต่ในอนาคต CARE ควรจะติดต่อไปยังเครื่องมือช่วยอื่นๆ เพื่อให้มีการใช้ข้อมูลร่วมกันได้

11. การต่อประสานกับระบบสื่อสาร (Communications interfaces)

- การสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ต่างๆ ใช้โปรโตคอล IEEE-488
- การต่อประสานกับระบบสื่อสารของระบบควบคุมดาวเทียม คือการสื่อสารซอฟต์แวร์ด้วยดาวเทียม
- CARE เป็นเวอร์ชันแรก จะดำเนินงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล ดังนั้น จึงไม่มีความจำเป็นในการต่อประสานซอฟต์แวร์กับระบบสื่อสาร

12. ข้อจำกัดของหน่วยความจำ (Memory constraints)

- ซอฟต์แวร์นี้ต้องการเนื้อที่ในฮาร์ดดิสก์ไม่ต่ำกว่า 1 กิกะไบต์
- CARE ต้องการหน่วยความจำมากกว่า 32 เมกกะไบต์ขึ้นไป

13. การดำเนินการ (Operations)

- เครื่องคอมพิวเตอร์ที่ใช้ในระบบสั่งซื้อสินค้าจะเปิดแบบเชื่อมต่อตรง (on line) เป็นเวลา 16 ชั่วโมงต่อ 1 วัน
- สัญญาณเตือนจะส่งเสียงดังที่เครื่องคอมพิวเตอร์ เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้น จากนั้นจะทำการสำรองข้อมูล เพื่อนำข้อมูลไปใช้ในภายหลังได้
- เพิ่มข้อมูลของเหตุการณ์ปกติ ใช้เป็นต้นแบบในการสร้างมาตรฐานของรายงานและสามารถเข้าถึงข้อมูลโดยมีผู้มีสิทธิ์เท่านั้น ระเบียบต่างๆ ของเหตุการณ์ปกติ ควรจะแสดงบนหน้าจอคอมพิวเตอร์ หรือพิมพ์ออกทางเครื่องพิมพ์

14. ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)

- ในการเพิ่มคำสั่งใหม่เข้าไปในระบบนั้น จะไม่ส่งผลกระทบกับการทดสอบคำสั่งที่มีอยู่เดิม
- ถ้ามีการเตรียมที่ตั้งของการติดตั้งซอฟต์แวร์ รายละเอียดต่างๆ จะปรากฏที่นี่
- ไม่สามารถทำให้เป็นประโยชน์ได้ (Not applicable : N/A)

15. หน้าที่ของซอฟต์แวร์ (Product functions)

- โปรแกรมสำเร็จรูปนี้จะแบ่งหน้าที่หลักออกเป็น 2 หน้าที่ ซึ่งจะทำงานกับระบบควบคุมดาวเทียม โดยการถ่ายทอดข้อความระหว่างกัน หน้าที่เหล่านี้ คือ

ก) การส่งผ่านข้อมูล (Data transmission)

ข) ยานพาหนะที่ใช้ซ่อมรบในที่ว่างระหว่างดวงดาว (Maneuverability of the spacecraft)

- CARE มีหน้าที่การทำงาน คือ แก้ไขความต้องการ เพิ่มความต้องการ ปรับให้ทันสมัย ลบความต้องการ และแสดงภาพรวมแบบลำดับขั้นตอนในรูปแบบของต้นไม้

16. ลักษณะของผู้ใช้ (User characteristics)

- ลักษณะที่เฉพาะเจาะจงของผู้ใช้ คือ ผู้ใช้ต้องมีความสามารถในการโต้ตอบกับเครื่องคอมพิวเตอร์ โดยการใช้คีย์บอร์ดหรือเมาส์ได้
- ลักษณะโดยทั่วไปของผู้ใช้ในระบบสั่งซื้อสินค้า มีดังนี้
 - ก) จบการศึกษาระดับมัธยมปลาย พร้อมทั้งได้รับการฝึกอบรมเพิ่มเติมเป็นเวลา 1-2 ปี
 - ข) ผู้ใช้ไม่มีความชำนาญในการใช้เครื่องคอมพิวเตอร์เหมือนกับนักคอมพิวเตอร์มืออาชีพ
 - ค) ผู้ใช้ควรได้รับการฝึกอบรมในการใช้เครื่องคอมพิวเตอร์อย่างน้อย 40 ชั่วโมงต่อสัปดาห์
- ผู้ใช้ CARE แบ่งออกเป็น 2 กลุ่ม ได้แก่ กลุ่มแรก คือ ลูกค้า เป็นผู้ที่มีความรู้ทางด้านคอมพิวเตอร์ น้อยมาก และกลุ่มที่สอง คือ วิศวกรซอฟต์แวร์ นักวิเคราะห์ระบบ หรือผู้จัดการโครงการ ซึ่งเป็นผู้ที่มีความชำนาญในด้านคอมพิวเตอร์

17. ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)

- ผู้ใช้ควรจะปรับขนาดของตัวอักษรก่อนที่จะพิมพ์ เพื่อเป็นการประหยัดกระดาษ
- เวลาที่ใช้ในการสำรองข้อมูลไม่ควรจะมากกว่า 6 ชั่วโมงต่อวัน

18. ความสามารถในการติดตามความต้องการ (Requirements traceability)

- SRS ฉบับนี้ สามารถตรวจสอบความต้องการกลับไปยังข้อกำหนดระบบของระบบควบคุมดาวเทียม
- ข้อกำหนดนี้ สามารถตรวจสอบความต้องการหรือมีการทดสอบความต้องการกลับไปยังข้อกำหนดระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี

19. ข้อจำกัด (Constraints)

- ระบบสั่งซื้อสินค้าจะได้รับการประมวลผลภายใต้ระบบปฏิบัติการ Zebra 7 release 4 และใช้ระบบปฏิบัติการเครือข่าย GIRAFE-12 release 3 นอกจากนี้ ระบบสั่งซื้อสินค้าจะเขียนด้วยภาษา C++ และเครื่องคอมพิวเตอร์ที่ใช้กับระบบนี้ คือ UNIGRAPH model A-3 สำหรับคีย์บอร์ดที่ใช้กับเครื่องคอมพิวเตอร์นี้ คือ คีย์บอร์ดมาตรฐาน QWERTY
- CARE ต้องได้รับการพัฒนาให้เสร็จสมบูรณ์ภายใน 6 เดือน
- CARE ต้องใช้ Windows เป็นโปรแกรมพื้นฐาน
- ข้อจำกัดของกระบวนการพัฒนาซอฟต์แวร์ คือ ปัญหาต่างๆ ที่เกิดขึ้นจากซอฟต์แวร์ จะไม่มีปัญหาที่เกิดขึ้นจากฮาร์ดแวร์เพราะในระบบนี้ไม่มีการระบุถึงความต้องการด้านฮาร์ดแวร์ ถ้าความต้องการของผู้ใช้หรือลูกค้าเปลี่ยนแปลงไป SRS ต้องได้รับการเปลี่ยนแปลงด้วย

20. ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)

- มีการตั้งข้อสันนิษฐานเกี่ยวกับลูกค้าว่าต้องมีความรู้ทางด้านการสื่อสารดาวเทียมและการออกแบบซอฟต์แวร์ที่ดีพอ ถ้ามีการเปลี่ยนแปลงการออกแบบซอฟต์แวร์ SRS จะต้องได้รับการเปลี่ยนแปลงด้วย

- CARE จะดำเนินงานภายใต้ระบบปฏิบัติการ Windows 95 หรือ Windows NT และดำเนินงานบนเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นเพนเทียม

21. การแบ่งสัณปันส่วนความต้องการ (Apportioning of requirements)

- การเพิ่มเติมความต้องการหรือรายละเอียดต่างๆ จะปรากฏบนหน้าจอคอมพิวเตอร์ ซึ่งผู้ดำเนินการกำลังใช้งานอยู่ เช่น อัตราการเปลี่ยนแปลง จำนวน ที่ตั้ง ขนาด และการแสดงความผิดพลาด โดยผู้เชี่ยวชาญเป็นผู้กำหนดสิ่งเหล่านี้

ส่วนที่ 3 : ความต้องการที่เฉพาะเจาะจง (Specific requirements)

22. การต่อประสานภายนอก (External interfaces)

- ฟังก์ชัน X และ Y จะมีการแลกเปลี่ยนข้อมูลซึ่งกันและกัน
- มีการส่งผ่านเชื้อเพลิงที่เหลืออยู่ไปยังระบบควบคุมภาคพื้นดินทุกๆ 15 วินาที
- การต่อประสานภายนอก จะรวมถึงการบำรุงรักษาซอฟต์แวร์ ผู้ดำเนินการจะรายงานและตีความหมายของเหตุการณ์ต่างๆ ให้กับลูกค้าทางโทรศัพท์

23. ความต้องการด้านหน้าที่ (Functional requirements)

- การวัดขนาดของอุปกรณ์ต่างๆ
- หน้าที่ของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า จะยอมรับชนิดของบัตรเครดิต หมายเลขบัตรเครดิต และวันที่หมดอายุของบัตรเครดิต
- ผู้ดำเนินการจะป้อนคำสั่งต่างๆ เพื่อตรวจสอบความถูกต้องของบัตรเครดิต

24. ความต้องการด้านประสิทธิภาพ (Performance requirements)

- การวัดขนาดของอุปกรณ์ต่างๆ มีความถูกต้องแม่นยำ 10%
- การประมวลผลรายการเปลี่ยนแปลง (transaction) ประมาณ 95% จะใช้เวลาต่ำกว่า 1 วินาที
- รายงานบัญชี มีความถูกต้องแม่นยำเท่ากับหนึ่งหน่วยการบัญชี (one accounting unit)
- โปรแกรมจะทำนายความสูงของดาวเทียม 10 เมตร โดยใช้เวลากายใน 7 วัน
- ประสิทธิภาพของระบบควบคุมดาวเทียมจะแตกต่างกัน ขึ้นอยู่กับเวลาของการลำดับเหตุการณ์ (Time Ordered Event Sequence : TOES) เพื่อนำไปใช้ในการจำลองระบบ
- เมื่อผู้ใช้ร้องขอไปยังระบบ ระบบจะโต้ตอบภายในเวลา 1 วินาที การโต้ตอบนี้ จะวัดจากการป้อนข้อมูลหรือการกดแป้นคีย์บอร์ดครั้งสุดท้าย เพื่อให้ข้อมูลไปปรากฏบนหน้าจอคอมพิวเตอร์

25. ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)

- การเข้าถึงฐานข้อมูลแบบอ่านอย่างเดียว และการแสดงรายละเอียดต่างๆ ไปยังผู้ดำเนินการ รายละเอียดเหล่านี้ จะได้รับการถ่ายทอดไปยังลูกค้าทางโทรศัพท์ด้วย
- ระเบียบของแฟ้มข้อมูลที่มีความผิดพลาดเกิดขึ้น จะเก็บข้อมูลไว้ภายใน 6 เดือน หลังจาก 6 เดือนแล้ว ควรเก็บข้อมูลลงบนเทปเป็นเวลา 3 ปี หลังจากนั้น จึงจะทำลายข้อมูลนั้นทิ้ง

26. ความต้องการด้านการดำเนินการ (Operational requirements)

- การป้อนข้อมูลในระบบสั่งซื้อสินค้า ถ้าข้อมูลที่ป้อนเข้าไปในระบบมีความผิดพลาดเกิดขึ้น ระบบจะโต้ตอบด้วยการแสดงข้อความผิดพลาด เช่น มีอักขระมากเกินไป มีอักขระน้อยเกินไป มีอักขระที่ไม่ถูกต้อง เป็นต้น
- ระบบสั่งซื้อสินค้าจัดการระบบช่วยเหลือให้กับผู้ใช้ เมื่อมีสถานการณ์ที่ผิดปกติในการติดต่อระหว่างผู้ใช้กับระบบเกิดขึ้น

27. ความต้องการด้านการทวนสอบ (Verification requirements)

- ผลลัพธ์ของโปรแกรม จะใช้เวลาภายใน 20 วินาทีในการประมวลผลเหตุการณ์ X 60% และจะใช้เวลาภายใน 30 วินาทีในการประมวลผลเหตุการณ์ X 99%

28. ความต้องการด้านการยอมรับ (Acceptance requirements)

- ระบุถึงข้อจำกัดในการตรวจรับซอฟต์แวร์
- ระบุถึงการตรวจสอบความถูกต้องในการส่งมอบซอฟต์แวร์ เพื่อดูว่าซอฟต์แวร์สามารถทำหน้าที่ได้ตรงตามความต้องการของผู้ใช้หรือลูกค้าหรือไม่

29. ความต้องการด้านเอกสาร (Documentation requirements)

- เอกสารในการออกแบบ ได้แก่ ข้อกำหนดของการออกแบบซอฟต์แวร์ (Software Design Specification : SDS) โดยใช้มาตรฐาน ANSI/IEEE 1016-1987
- เอกสารของซอฟต์แวร์ จะรวมถึง ข้อกำหนดในการออกแบบ ซึ่งอ้างอิงจากมาตรฐาน IEEE 830-1993 และชุดของการประมวลผลกรณีทดสอบ (test cases) ที่มีผลลัพธ์เหมาะสม
- เอกสารการทดสอบซอฟต์แวร์ จะรวมถึง แผนการทดสอบระบบ (test plan) การสร้างกรณีทดสอบ และผลลัพธ์จากการสร้างกรณีทดสอบ
- เอกสารการบำรุงรักษาซอฟต์แวร์ ควรจะรวมถึงการบำรุงรักษาซอฟต์แวร์ระหว่างขั้นตอนการดำเนินการ

30. ความต้องการด้านความปลอดภัย (Safety requirements)

- ระบบควรจะทำให้แน่ใจว่าข้อมูลจะไม่ได้รับความเสียหาย เมื่อมีความผิดพลาดเกิดขึ้นกับระบบ
- สถานะของเหตุการณ์ที่ผิดปกติจะบันทึกลงในรายงานประจำวันของระบบนั้น หลังจากมีการตัดสินใจกับเหตุการณ์ที่ผิดปกติแล้ว ผู้ใช้จะสิ้นสุดการใช้งาน
- เลือกระเบียน (fields) จากรายงานที่มีความผิดพลาดเกิดขึ้น และนำระเบียนเหล่านี้พิมพ์ใส่กระดาษ (hard copy)

31. ความต้องการด้านสภาพแวดล้อม (Environment requirements)

- ในการติดตั้งซอฟต์แวร์ ควรจะมีอุณหภูมิห้องระหว่าง 18-25 องศาเซลเซียส
- ไม่ควรติดตั้งซอฟต์แวร์ใกล้ห้องเครื่องปรับอากาศ

32. ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)

- เนื้อที่ว่างในฮาร์ดดิสก์ 5 เมกกะไบต์ขึ้นไป

- โปรแกรมทั้งหมดจะได้รับการประมวลผลภายในโควตาของผู้ใช้ที่กำหนดไว้

33. ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communications requirements)

- การสื่อสารข้อมูลระหว่างระบบควบคุมดาวเทียมและระบบผู้ส่ง ระบบผู้ส่งจะส่งผ่านข้อความไปยังผู้รับ เมื่อต้นทางคือระบบผู้ส่ง
- การสื่อสารข้อมูลระหว่างระบบควบคุมดาวเทียมและระบบควบคุมภาคพื้นดิน ถ้าต้นทางคือ ระบบควบคุมภาคพื้นดิน จะทำการส่งข้อความเพื่อกำหนดการดำเนินการที่จะใช้ในระบบนี้

34. ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)

- SRS มีความต้องการกัน เช่น รูปแบบของรายงาน ถ้าในข้อกำหนดหนึ่ง อธิบายความต้องการในรูปแบบของตาราง ในข้อกำหนดอื่นๆ ต้องอธิบายความต้องการในรูปแบบของตารางด้วย
- SRS สามารถตรวจสอบความถูกต้องของซอฟต์แวร์ เช่น ผลลัพธ์ของโปรแกรม จะใช้เวลาภายใน 20 วินาทีในการประมวลผลเหตุการณ์ X 60% และใช้เวลาภายใน 30 วินาทีในการประมวลผลเหตุการณ์ X 100%

35. ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)

- ผู้พัฒนาซอฟต์แวร์ควรจะได้รับ การฝึกอบรมเกี่ยวกับภาษาที่ใช้เขียนโปรแกรม ก่อนที่จะทำการพัฒนาระบบเอบีซี
- ผู้พัฒนาซอฟต์แวร์จะฝึกอบรมให้กับผู้ใช้ในเรื่องการบำรุงรักษาซอฟต์แวร์และการเปลี่ยนแปลงแก้ไขของระบบเอบีซี

36. ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)

- ผู้ใช้สามารถดาวน์โหลดเวอร์ชันใหม่ของซอฟต์แวร์จากเว็บไซต์
- ผู้ใช้สามารถสอบถามปัญหาต่างๆ จากผู้พัฒนาซอฟต์แวร์ทางโทรศัพท์
- ผู้ใช้ควรจะสามารถสำรองข้อมูลบนเทป

37. ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)

- ซอฟต์แวร์นี้ จะบรรจุภัณฑ์ลงบนซีดีรอมและเทป สำหรับส่งมอบซอฟต์แวร์ให้กับลูกค้า
- การบรรจุภัณฑ์นี้ นอกเหนือจากซอฟต์แวร์ที่จะส่งมอบให้กับลูกค้าแล้ว ยังรวมถึงเอกสารต่างๆ ด้วย เช่น เอกสารข้อกำหนดความต้องการซอฟต์แวร์ เอกสารคู่มือผู้ใช้ซอฟต์แวร์ เอกสารการออกแบบซอฟต์แวร์ เอกสารการทดสอบซอฟต์แวร์ เอกสารการบำรุงรักษาซอฟต์แวร์ เป็นต้น

38. ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)

- ข้อบังคับที่ใช้ในการออกแบบ คือ ใช้ UNIGRAPH model A-3
- CARE ได้รับการออกแบบโดยใช้ระเบียบวิธีเชิงวัตถุในการพัฒนาซอฟต์แวร์ ซึ่งใช้เครื่องหมายหรือสัญลักษณ์ของ Coad และ Yourdon ในการระบุและแสดงแผนผังเชิงวัตถุ พร้อมทั้งใช้ระบบการจัดการฐานข้อมูล (Database Management Systems : DBMS) เป็นฐานข้อมูลเชิงสัมพันธ์ (Relational database)

39. มาตรฐานที่มีใช้ในองค์กร (Standard compliance)

- การออกแบบและการพัฒนาซอฟต์แวร์ จะอ้างอิงจากมาตรฐาน ANSI/IEEE 1016-1987 และอ้างอิงจากมาตรฐานของบริษัทเอบีซี ถ้ามาตรฐานทั้งสองเกิดความขัดแย้งกัน จะประยุกต์ใช้มาตรฐานของบริษัทเอบีซีเป็นหลักในการพัฒนาซอฟต์แวร์
- มาตรฐานที่ใช้ในการพัฒนาซอฟต์แวร์ แบ่งเป็น 2 ประเภท ได้แก่ มาตรฐานกระบวนการ และมาตรฐานผลิตภัณฑ์ ตัวอย่างของมาตรฐานผลิตภัณฑ์ คือ รูปแบบของแฟ้มข้อมูล และรูปแบบของรายงาน สำหรับตัวอย่างของมาตรฐานกระบวนการ คือ มาตรฐานการรับประกันผลิตภัณฑ์ และกระบวนการทางการบัญชี

40. ความเชื่อถือได้ (Reliability)

- เวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้น จะต้องมากกว่า 5 วันขึ้นไป
- ภายในระยะเวลา 6 เดือน เวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้นโดยรวมเท่ากับ 4 สัปดาห์

41. สภาพพร้อมใช้งาน (Availability)

- ความสามารถของซอฟต์แวร์ที่คำนึงถึง คือ สภาพพร้อมใช้งานของซอฟต์แวร์ไม่ต่ำกว่า 98% ซึ่งสามารถทำงานได้ภายใน 48 ชั่วโมง และสภาพพร้อมใช้งานไม่ต่ำกว่า 75% สำหรับทำงานได้ทุก ๆ 3 ชั่วโมง
- ผู้ใช้จะจัดหาสภาพพร้อมใช้งานของซอฟต์แวร์โดยเฉลี่ย 98% ภายในระยะเวลาการทำงาน 1 ปีขึ้นไป และจะต้องไม่เคยต่ำกว่า 50% ของระยะเวลาการทำงานต่อ 1 สัปดาห์
- ระบบมีสภาพพร้อมใช้งานตลอด 24 ชั่วโมงต่อ 1 วัน
- เครื่องคอมพิวเตอร์ที่ใช้ในระบบสั่งซื้อสินค้า จะมีสภาพพร้อมใช้งาน 99.9%

42. ความมั่นคง (Security)

- ระบบสั่งซื้อสินค้าจะอนุญาตให้เฉพาะผู้ที่มีสิทธิ์หรืออำนาจในการเข้าถึงข้อมูลในระบบนี้เท่านั้น ผู้ดำเนินการ เสมียน และผู้จัดการสามารถเข้าถึงข้อมูล แฟ้มข้อมูล และระบบย่อยอื่นๆ ตลอดจนโปรแกรมประยุกต์ของระบบสั่งซื้อสินค้า
- ระบบนี้ จะอนุญาตเฉพาะผู้จัดการที่มีอำนาจในการเข้าถึงข้อมูลเท่านั้น ซึ่งผู้จัดการสามารถทำหน้าที่ในการสร้างรายงานได้ ถ้ามีบุคคลใดพยายามเข้าถึงข้อมูลครบ 3 ครั้ง ระบบจะปฏิเสธการเข้าถึงข้อมูลของบุคคลนั้น พร้อมทั้งจะจดบันทึกลงใน transaction log ด้วย

43. ความสามารถในการบำรุงรักษา (Maintainability)

- เวลาเฉลี่ยในการแก้ไขความผิดพลาดต้องไม่เกิน 1 สัปดาห์
- ภายในระยะเวลา 1 ปี เวลาเฉลี่ยในการแก้ไขความผิดพลาดโดยรวมเท่ากับ 1 วัน

44. ความสามารถในการใช้ได้หลายระบบ (Portability)

- ซอฟต์แวร์นี้ สามารถใช้งานได้ในสภาพแวดล้อม X และ Y
- ไม่มีสวอนใดส่วนหนึ่งของซอฟต์แวร์ที่เขียนด้วยภาษาแอสเซมบลี

- ถ้ามีการเปลี่ยนแปลงรหัสต้นทาง (source code) น้อยกว่า 2% จะไม่ส่งผลกระทบต่อผลการแปลโปรแกรม (compile) ใหม่อีกครั้งหนึ่งในซอฟต์แวร์นี้ เพื่อให้ดำเนินงานบนเครื่องคอมพิวเตอร์ X ได้

ส่วนที่ 4 : ข้อมูลสนับสนุน (Supporting information)

45. สารบัญ (Table of contents)

- บทที่ 1 วัฏจักรของการพัฒนาซอฟต์แวร์ 1-3
 - 1.1 บทนำ 1-3
 - 1.2 ขั้นตอนและกิจกรรมต่างๆ 1-3
 - 1.2.1 ขั้นตอนความต้องการของผู้ใช้ 1-5
- หัวข้อ
 - 1.0 บทนำ 1
 - 2.0 เอกสารที่เกี่ยวข้อง 2
 - 3.0 แนวคิดความต้องการ 3

46. บันทึก (Notes)

- หน้านี้มีไว้เพื่อบันทึกสิ่งที่ต้องการอธิบายเพิ่มเติมใน SRS

47. ภาคผนวก (Appendixes)

- ภาคผนวก ก : ตารางของการตรวจสอบความต้องการ (Requirements traceability matrix)
- ภาคผนวก ข : สารบัญของเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- ภาคผนวก ค : แผ่นแบบ (template) ของ SRS

48. ดัชนี (Index)

- ความต้องการด้านหน้าที่, 44
- วัฏจักรของการพัฒนาซอฟต์แวร์, 1-3
- เอกสารข้อกำหนดความต้องการซอฟต์แวร์ รูปที่ 2, รูปที่ 6, รูปที่ 7

คำอธิบายของเนื้อหาใน SRS ที่เป็นภาษาอังกฤษ

Software Requirements Specification (SRS) is a document that gives the developer's view of the problem rather than the user's. The developers are defining 'what' the software must do. The SRS must cover all the requirements stated in the user or client requirements.

Section 1 : Introduction

1. Purpose

This subsection should accomplish the following:

- Delineate the purpose of the particular SRS

- Specify the intended audience for the SRS

2. Scope

This subsection should

- Identify the software product(s) to be produced by name (for example, Host DBMS, Report Generator etc.)
- Explain what the software product(s) will, and, if necessary, will not do
- Describe the application of the software being specified, including relevant benefits, objectives, and goals
- Be consistent with similar statements in higher-level specifications (for example, the system requirement specification), if they exist

3. Definitions

This subsection should provide the definitions of all terms required to interpret properly the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

4. Acronyms

This subsection should provide acronyms required to interpret properly the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

5. Abbreviations

This subsection should provide abbreviations required to interpret properly the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

6. References

This subsection should

- Provide a complete list of all documents referenced elsewhere in the SRS
- Identify each document by title, report number (if applicable), date, and publishing organization
- Specify the sources from which the references can be obtained

7. Overview

This subsection should

- Describe what the rest of the SRS contains
- Explain how the SRS is organized

Section 2 : Overall description

8. User interfaces

This should specify the following:

- The logical characteristics of each interface between the software product and its users. This includes those configuration characteristics (e.g., required screen formats, page or window layouts, content of any reports or menus, or availability of programmable function keys) necessary to accomplish the software requirements.
- All the aspects of optimizing the interface with the person who must use the system. This may simply comprise a list of do's and don'ts on how the system will appear to the user. One example may be a requirement for the option of long or short error messages. Like all others, these requirements should be verifiable, for example, "a clerk typist grade 4 can do function X in Z min after 1 h of training" rather than "a typist can do function X." (This may also be specified in the Software System Attributes under a section titled Ease of Use.)

9. Hardware interfaces

This should specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (number of ports, instruction sets, etc.) It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full screen support as opposed to line by line.

10. Software interfaces

This should specify the use of other required software products (for example, a data management system, an operation system, or a mathematical package), and interfaces with other application systems (for example, the linkage between an accounts receivable system and a general ledger system). For each required software product, the following should be provided:

- Name
- Mnemonic
- Specification number
- Version number
- Source

11. Communications interfaces

This should specify the various interfaces to communications such as local network protocols, etc.

12. Memory constraints

This should specify any applicable characteristics and limits on primary and secondary memory.

13. Operations

This should specify the normal and special operations required by the user such as

- The various modes of operations in the user organization; for example user-initiated operations
- Periods of interactive operations and periods of unattended operations
- Data processing support functions
- Backup and recovery operations

14. Site adaptation requirements

This could

- Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode, for example, grid values, safety limits, etc.
- Specify the site or mission-related features that should be modified to adapt the software to a particular installation.

15. Product functions

This subsection of SRS should a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.

Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher level specification (if one exists) that allocates particular functions to the software product. Note that for the sake of clarity:

- The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.
- Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product but simply shows the logical relationships among variables.

16. User characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements but rather should provide the reasons why certain specific requirements are later specified in section 3 of the SRS.

17. Qualification provisions

The process of determining whether a system or component is suitable for operational use.

18. Requirements traceability

An SRS is traceable if the origin of each of its requirements is clear and if it facilitates the referencing of each requirement in future development or enhancement documentation.

19. Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer's options. These include

- Regulatory policies
- Hardware limitations (for example, signal timing requirements)
- Interfaces to other applications
- Parallel operation
- Audit functions
- Control functions
- Higher-order language requirements
- Signal handshake protocols (for example, XON-XOFF , ACK-NACK)
- Reliability requirements
- Criticality of the application
- Safety and security considerations

20. Assumptions and dependencies

This subsection of the SRS should list each of the factors that effect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

21. Apportioning of requirements

This subsection of the SRS should identify requirements that may be delayed until future versions of the system.

Section 3 : Specific requirements

22. External interfaces

This should be a detailed description of all inputs into and outputs from the software system. It should include both content and format as follows:

- Name of item

- Description of purpose
- Source of input or destination of output
- Valid range, accuracy and/or tolerance
- Units of measure
- Timing
- Relationships to other inputs/outputs
- Screen formats/organization
- Window formats/organization
- Data formats
- Command formats
- End messages

23. Functional requirements

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as 'shall' statements starting with The system shall...

These include

- a) Validity checks on the inputs
- b) Exact sequence of operations
- c) Responses to abnormal situations, including
 - 1) Overflow
 - 2) Communication facilities
 - 3) Error handling and recovery
- d) Effect of parameters
- e) Relationship of outputs to inputs, including
 - 1) Input/Output sequences
 - 2) Formulas for input to output conversion

It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does not imply that the software design will also be partitioned that way.

24. Performance requirements

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include:

- The number of terminals to be supported

- The number of simultaneous users to be supported
- Amount and type of information to be handled

Performance requirements specify numerical values for measurable variables used to define a function (e.g. rate, frequency, capacity, speed and accuracy).

25. Logical database requirements

This should specify the logical requirements for any information that is to be placed into a database. This may include:

- Types of information used by various functions
- Frequency of use
- Accessing capabilities
- Data entities and their relationships
- Integrity constraints
- Data retention requirements

26. Operational requirements

These specify how the system will run and how it will communicate with the human operators. Operational requirements include all user interfaces, man-machine or human-computer interaction requirements. Examples are: the screen layout, the content of error messages, help systems etc. It is often useful to define the semantics and syntax of commands.

27. Verification requirements

These specify the constraints on how the software is to be verified. The verification requirements might include requirements for simulation, emulation, live tests with simulated inputs, live tests with real inputs, and interfacing with the testing environment. Simulation is a model that behaves or operates like a given system when provided a set of controlled inputs.

28. Acceptance requirements

These specify the constraints on how the software is to be validated.

29. Documentation requirements

These specify project-specific requirements for documentation in addition to those contained in these standards (e.g. the detailed format of the Software User Manual).

30. Safety requirements

These specify any requirements to reduce the possibility of damage that can follow from software failure. Safety requirements may identify critical functions whose failure may be hazardous to people or property.

31. Environment requirements

Physical conditions such as temperature, humidity and cleanliness, within which a computer system operates; or, the support and utility software and hardware.

32. Computer hardware resource utilization requirements

This should specify the use of computer hardware resource utilization such as processor capability, memory, input/output devices, secondary devices, communication devices, network devices etc.

33. Computer communications requirements

This should specify the various interfaces to communications within software.

34. Software quality factors

These specify attributes of the software that ensure that it will be fit for its purpose (other than the major quality attributes of reliability, maintainability and safety, which should always be specified). Where appropriate, software quality attributes should be specified in measurable terms (i.e. with the use of metrics).

35. Training-related requirements

This paragraph defines the developer's training responsibilities. Two categories of training, internal training for the project team, and external training for the requestor and the requestor's customer, are described.

An outline for a detailed discussion of training follows:

Types of Training

A. Internal Training

1. Technical

- a. Programming language (s)
- b. Use of test tools
- c. Use of terminals
- d. The development system hardware
- e. Interfaces to related systems
- f. The scope of work
- g. The baseline design

2. Non-technical

- a. Management procedures
- b. Change control procedures
- c. Documentation control
- d. Reporting requirements

e. Clerical procedures

B. External Training

1. Installing the software system
2. Using the system
3. Maintaining and modifying the system

36. Logistics-related requirements

This paragraph considers the logistical characteristics that are required for the implementation, support and maintenance of the system.

37. Packaging requirements

This paragraph describes software packaging considerations such as transportation media; documentation shipped with software when released; special features required to satisfy varied interfaces (e.g., different I/O drivers for different hardware configurations).

38. Implementation constraints

Describe general implementation constraints on the design and implementation of the software, such as the use of GFE, COTS, or use of specific compilers, etc. If existing software is required to be used or modified, include such requirements here.

39. Standard compliance

This subsection should specify the requirements derived from existing standards or regulations. They may include

- Report format
- Data naming
- Accounting procedures
- Audit tracing

40. Reliability

These specify the acceptable mean time interval between failures of the software, averaged over a significant period (MTTF). They may also specify the minimum time between failures that is ever acceptable. Reliability requirements may have to be derived from the user's availability requirements.

41. Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

Availability may specify

- Mean and minimum capacity (e.g. all terminals)
- Start and end times of availability (e.g. from 0900 to 1730 daily)

- Time period for averaging availability (e.g. 1 year)

42. Security

This should specify the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to

- Utilize certain cryptographical techniques
- Keep specific log or history data sets
- Assign certain functions to different modules
- Restrict communications between some areas of the program
- Check data integrity for critical variables

43. Maintainability

These specify how easy it is to repair faults and adapt the software to new requirements. The ease of performing these tasks should be stated in quantitative terms, such as mean time to repair a fault (MTTR). They may include constraints imposed by the potential maintenance organization. Maintainability requirements may be derived from the user's availability and adaptability requirements.

44. Portability

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include

- Percentage of components with host-dependent code
- Percentage of code that is host dependent
- Use of a proven portable language
- Use of a particular compiler or language subset
- Use of a particular operating system

Section 4 : Supporting information

45. Table of contents

The table of contents is quite important and should follow general compositional practices.

46. Notes

Use this section to present information that aids in understanding the information provided in previous sections, and which is not contractually binding.

47. Appendixes

The appendixes are not always considered part of the actual requirements specification and are not always necessary. They may include

- Sample I/O formats, descriptions of cost analysis studies, or results of user surveys

- Supporting or background information that can help the readers of the SRS
- A description of the problems to be solved by the software
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements

48. Index

The index is quite important and should follow general compositional practices.

ตัวอย่างในการเขียน SRS ของแต่ละเนื้อหาที่เป็นภาษาอังกฤษ

Section 1 : Introduction

1. Purpose

- The purpose of this SRS is to define the requirements, which will enable the development of a Satellite Control System using a software approach.
- This SRS describes the function and performance requirements allocated to the order-entry subsystem.
- To develop Computer Aided Requirements Engineering (CARE) that can reduce unambiguous and make a clear, consistent, precise statement and way to solved the problem in a SRS.

2. Scope

- There will be two software packages developed using this SRS. The first will be a communication package. The second software package will be a positioning package.
- The order-entry subsystem is a component of the ABC Corporation's automated mail-order system. The automated mail-order system is described in a document entitled Automated Mail-Order System Design Description Release 4 October 1996.
- This specification is intended to form the basis for design of a software product that will support up to 100 special-purpose order-entry workstations manned by trained operators.
- To develop tool, which has report base on Applied IEEE standards: IEEE Std 830-1993 and IEEE Std 1233-1996.

3. Definitions

- Communication Packet -- An eighty character message string that is to be relayed from a source to a destination.
- Customer -- The person, or persons, who pays for the product and usually decides the requirements.

- User -- A person who utilizes the products or services of a system, or a person who controls and monitors the hardware and software of a system (i.e. an end user, an operator, or both).

4. Acronyms

- DBLT -- Destination Bandwidth Lookup Table
- EPM -- Engineering Practices Manual
- IEEE -- Institute of Electrical and Electronics Engineers
- SRS -- Software Requirements Specification

5. Abbreviations

- <Comm Msg> -- Command Message
- Kbps -- Kilobytes per second
- OP Code -- Operational Code
- Std -- Standard

6. References

- EPM for Software, Lockheed Martin, Orlando FL, 1982.
- IEEE Std 610.12-1990, IEEE Standard Glossary for Software Engineering Terminology (ANSI).
- IEEE Std-830, IEEE New York, 1984 and 1993.

7. Overview

- The following sections contain a description of the software being developed and its requirements. This SRS is organized in accordance with IEEE Std number 830. It follows both the 1984&1993 versions of this Std. Additional information was also taken from Lockheed Martins Software EPM, which is the manual that Lockheed Martin follows to produce new software designs.
- The outcome of these research will be a Requirements Engineering tool called CARE that have functions to edit, insert, update, delete requirements data and show viewpoints hierarchy in tree format.

Section 2 : Overall description

8. User interfaces

- The user interfaces of this control system are the computer screen, keyboard/mouse, and sender antenna.

- CARE will be Graphic User Interface (GUI) application. This application will have menus bar and icons for receive command from users. All the aspects of optimizing the interfaces with the users whom have a little computer skill to use the system.

9. Hardware interfaces

- Hardware interfaces are handled beneath abstract interfaces. Therefore no networking protocols are incorporated into this emulation.
- CARE will printout report to printer so it has simple hardware interface between personal computer and printer.

10. Software interfaces

- The software of the control system interfaces with both the sender and Ground Control.
- The order-entry subsystem shall interface with the following subsystems:
 - a) Warehouse
 - b) Accounting
 - c) Marketing and sales
 - d) Catalogue
 - e) Network control
- CARE first version will not connect to other software. But in the future CARE would be connecting to other CASE tools for sharing requirements data.

11. Communications interfaces

- Communications between the computer and remote instruments shall be via the IEEE-488 protocol.
- Since the communications interface is mostly the software communicating with the satellite.
- CARE first version will run on standalone personal computer so it does not need communication interface.

12. Memory constraints

- This software requires at least 1 GB of harddisk space.
- CARE requires memory more than 32 MB.

13. Operations

- The order-entry workstation is a key subsystem. Each operating workstation shall be on line 16 hours per day.
- An alarm signal shall be transmitted to alert the on-duty manager's workstation that an emergency incident has occurred. A duplicate copy of the recording associated with the incident shall also be forwarded immediately to the on-duty manager's workstation.

- The incident file is the source for three standard reports and is accessible only to authorized personnel. Incident records should be displayed on screen or printed in hard copy.

14. Site adaptation requirements

- It shall be possible to add new commands without retesting existing commands.
- If any site preparation is needed, it will appear here.
- Not applicable. (N/A)

15. Product functions

- This software package will adopt two basic functions, that when working in harmony will control a satellite system along with the relaying of messages. These functions are:
 - a) Data transmission
 - b) Maneuverability of the spacecraft
- CARE will have functions to edit, insert, update, delete requirements data and show viewpoints hierarchy in tree format.

16. User characteristics

- The only specific characteristic that a user must have is the ability to interact with a computer, using the keyboard and/or mouse.
- The order-entry operators typically have the following personal characteristics:
 - a) High school graduate frequently with one to two years additional academic training.
 - b) Operators are not computer professionals.
 - c) Computer-literate with minimum one 40-hour week hands-on training on workstation.
- User of CARE separate into 2 groups, first is clients who have a little computer skill and the second is software engineer or system analyst or project manager who have excellent computer skill.

17. Qualification provisions

- Users should adjust font size before printing to save paper cost.
- Each daily backup time should not last more than 6 hours.

18. Requirements traceability

- This SRS traces requirements to the satellite control system system specification.
- This specification traces requirements and requirement testing to the ABC automated mail-order system system specification.

19. Constraints

- The order-entry software product shall execute under release 4 of the Zebra 7 Operating System. The network operating system is GIRAFE-12 release 3. The order-entry software product code shall be written in C++. The order-entry workstations shall be UNIGRAPH, model A-3. A special-purpose keyboard is associated with each workstation. This keyboard is used as adjunct to the standard QWERTY keyboard.
- CARE must develop complete in 6 months.
- CARE must be windows base application.
- The only constrains that might limit the development process at this time are software problems. There are no hardware problems, because there are no hardware requirements to follow at this juncture. If at a later time the requirements are modified then the SRS will be updated accordingly.

20. Assumptions and dependencies

- It is assumed the customer is proficient in satellite communications and the software is being designed in such a fashion. If it is found that this is not the case, then changes in the design will be implemented and the SRS will have to change accordingly.
- CARE will run under Windows 95 or Windows NT operating system and run on Pentiums compatible computers.

21. Apportioning of requirements

- Additional requirements dealing with the physical characteristics of the information shown on the screen to operators, such as rate of change, amount, location, size, and error indications, have been imposed by human factors specialists.

Section 3 : Specific requirements

22. External interfaces

- Functions X and Y shall exchange instrument data.
- Transmit the amount of fuel remaining to the ground control system every fifteen seconds.
- External interfaces shall be maintained with telephone-based reports as interpreted and entered by the operator.

23. Functional requirements

- Calibrate the instrument data.
- Select all calibration stars brighter than the 6th magnitude.
- The order-enter input function shall accept credit card type, number, and expiration date.

- The operator shall enter a single keystroke command to activate the credit card validate menu.

24. Performance requirements

- Calibrate the instrument data to an accuracy of 10%.
- 95% of the transactions shall be processed in less than 1 second.
- The accuracy of accounting reports shall be one accounting unit.
- The program shall predict the satellite's altitude to within 10 metres, seven days in advance.
- The performance of the control system varies depending on the Time Ordered Event Sequence (TOES) chosen to simulate the system.
- All operator requests shall be responded to within one second. Response shall be measured from the last entry keystroke to the appearance of an on-screen reply.

25. Logical database requirements

- An arbitrary DBLT shall be initialized consistent with the characteristics of raw data sources. The DBLT shall be updated upon request from Ground Control via <T&C>. Also the maximum fuel amount and fuel consumption rate shall be input parameters to this prototype.
- The function accesses databases on a read-only basis and presents information to the operator that is to be relayed to the customer via telephone.
- An accident file record will be retained on-line for 6 months. After expiration of 6 months, the record will be archived on tape for 3 years, after which it will be destroyed.

26. Operational requirements

- The operation of the control system shall be TOES 1. Message sequences are composed of combinations of the following control events and data flow at system interfaces.
- The order-entry input function shall recognize the following data entry errors and respond with appropriate error messages such as too many characters, too few characters, invalid characters., etc.
- The order-entry software product is also expected to provide a help function to enable operators to deal with unusual situations.

27. Verification requirements

- The output of the program shall be given within 20s of event X, 60% of the time; and shall be given within 30s of event X, 99% of the time.

28. Acceptance requirements

- Acceptance constrain the design of the product.

- Validation confirms whether the user requirements are satisfied when the software is delivered.

29. Documentation requirements

- Design documentation shall include a complete listing for the order-entry input application product and a complete Software Design Specification (SDS) per ANSI/IEEE Standard 1016-1987.
- Software product documentation shall include a design specification as described in IEEE 830-1993, a commented listing (one comment line per three executable lines), and a set of executable test cases with appropriate results.
- Testing documentation shall include test plan, test cases, and test case results.
- Maintenance documentation should include all of detailed agreements regarding the maintenance of the software during the operational phase by the system developer or, as in many cases, by a third party.

30. Safety requirements

- The system should ensure that no data is lost when a power failure occurs.
- The emergency status entry shall signal completion of the incident report record. The operator shall terminate the call after making the emergency condition decision.
- Selected fields from the emergency report should be produced in hard copy to be faxed to the driver.

31. Environment requirements

- The temperature of software installed room should be between 18-25 Celsius degree.
- Don't install the software near air-condition controlled room.

32. Computer hardware resource utilization requirements

- 5 Mbytes of disk space are available.
- All programs shall execute with standard user quotas.

33. Computer communications requirements

- Communications between satellite control system and sender shall pass the message along to the receiver when the source is sender.
- Satellite control system communicates with Ground Control. If Ground Control is the source the software dissects the message to determine which operation and which sink is to be used.

34. Software quality factors

- An SRS is consistency. Example, the format of an output report should be described in one requirement as tabular and in another as tabular too.

- An SRS is verifiable. Example, output of the program shall be produced within 20s of event X 60% of the time; and shall be produced within 30s of event X 100% of the time.

35. Training-related requirements

- The training course in programming language should be held for programmer before developing the ABC system.
- Developers will train users how to maintain and modify the ABC system.

36. Logistics-related requirements

- Users can download new version of software from web site.
- Users can inquire problems from developers via telephone.
- Users should backup data on tape.

37. Packaging requirements

- This software product will be packaged on CD-ROM and tape when the software is delivered to clients.
- This packaging has documentation shipped with software when released such as Software Requirements Specification (SRS), Software User Manual, Design document, Testing document, Maintenance document.,etc.

38. Implementation constraints

- UNIGRAPH model A-3 characteristics and limitations.
- CARE will design by use Object-oriented methodology to be the direction to development. Coad and Yourdon Object-oriented notations have been used to define and represent object diagram. Relational database has been used to be a DBMS.

39. Standard compliance

- Design and development shall conform to ANSI/IEEE Standard 1016-1987, and ABC corporate standards for software development. In the event of conflict, ABC Corporation's software development standards shall apply.
- Two kinds of standards can be specified: process standards and product standards. Examples of product standards are export file formats and legal report formats. Examples of the process standards are product assurance standards and accounting procedures to be followed.

40. Reliability

- The minimum time between severe failures shall be in excess of 5 days.
- The MTBF for severe failures shall be 4 weeks, averaged over 6 months.

41. Availability

- All essential capabilities shall be at least 98% available in any 48 hour period and at least 75% available in every 3 hour period.
- The user shall be provided with 98% average availability over 1 year during working hours and never less than 50% of working hours in any one week.
- System is available 24 hours per day.
- Order-entry workstations shall have an availability of 99.9%.

42. Security

- Order-entry product software is allocated the responsibility for ensuring that only authorized order-entry operations, clerks, and managers can access the order-entry workstation software and access files and other subsystems through the order-entry workstation's order-entry application software.
- Only authorized on-duty managers shall be permitted access to the report generation function. Three unsuccessful access attempts shall be recorded in the transaction log, and access shall be denied after the third unsuccessful attempt.

43. Maintainability

- The time to repair shall never exceed 1 week.
- The MTTR shall be 1 day averaged over a 1 year.

44. Portability

- The software shall be portable between environments X and Y.
- No part of the software shall be written in assembler.
- It shall be possible to recompile this software to run on computer X without modifying more than 2% of the source code.

Section 4 : Supporting information

45. Table of contents

-	CHAPTER 1 THE SOFTWARE LIFE CYCLE	1-3
	1.1 INTRODUCTION	1-3
	1.2 PHASES, ACTIVITIES AND MILESTONES	1-3
	1.2.1 UR phase: user requirements definition	1-5
-	Section	
	1.0 INTRODUCTION.....	1
	2.0 RELATED DOCUMENTATION.....	2
	3.0 REQUIREMENTS APPROACH AND TRADEOFFS.....	3

46. Notes

- This page is provided for notes that you wish to make in using this SRS.

47. Appendixes

- APPENDIX A : REQUIREMENTS TRACEABILITY MATRIX
- APPENDIX B : SRD table of contents
- APPENDIX C : MANDATORY PRACTICES

48. Index

- functional requirements, 44
- life cycle, 1-3
- Software Requirements Specification (SRS) Fig 2, Fig 6, Fig 7

ภาคผนวก จ

ในส่วนนี้ แสดงถึงตัวอย่างในการเขียน SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซีทั้งภาษาไทยและภาษาอังกฤษ เพื่อใช้เป็นแนวทางสำหรับผู้ที่จะจัดทำ SRS

ตัวอย่างในการเขียน SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซีที่เป็นภาษาไทย

ระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี (ABC Corporation's automated mail-order system)

ส่วนที่ 1 : บทนำ (Introduction)

1. จุดประสงค์ (Purpose)

SRS ฉบับนี้ อธิบายถึงความต้องการด้านหน้าที่และความต้องการด้านประสิทธิภาพในระบบย่อยของระบบสั่งซื้อสินค้า

2. ขอบเขต (Scope)

ข้อกำหนดฉบับนี้ ใช้เป็นพื้นฐานในการออกแบบซอฟต์แวร์ ซึ่งสามารถสนับสนุนการทำงานของเครื่องคอมพิวเตอร์ได้ถึง 100 เครื่อง โดยผู้ใช้ได้รับการฝึกอบรมในการใช้เครื่องคอมพิวเตอร์เหล่านี้

3. คำจำกัดความ (Definitions)

- ลูกค้า -- เป็นผู้ที่ต้องการซอฟต์แวร์ที่ได้รับการพัฒนาแล้ว
- ผู้ใช้ -- เป็นผู้ที่ใช้ประโยชน์จากซอฟต์แวร์ที่สมบูรณ์แล้ว ซึ่งอาจเป็นคนเดียวกับลูกค้าก็ได้

4. รัสพจน์ (Acronyms)

- IEEE -- Institute of Electrical and Electronics Engineers
- SDS -- Software Design Specification
- SRS -- Software Requirements Specification

5. คำย่อ (Abbreviations)

- CID -- customer identification number
- PID -- product identification number
- Std -- Standard

6. เอกสารอ้างอิง (References)

- IEEE Std 610.12-1990, IEEE Standard Glossary for Software Engineering Terminology (ANSI).
- IEEE Std-830, IEEE New York, 1984 and 1993.

7. ภาพรวม (Overview)

ระบบย่อยของระบบสั่งซื้อสินค้า เป็นส่วนประกอบหนึ่งของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี คำอธิบายของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติอยู่ในเอกสารที่มีชื่อเรียกว่า "คำอธิบายการออกแบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติ ครั้งที่ 4 ตุลาคม 1996" ความต้องการที่ระบุในเอกสารฉบับนี้ได้รับการยอมรับเมื่อวันที่ 12 กันยายน 1995 และ ในส่วนที่ 3.8 ของข้อกำหนดนี้ สามารถตรวจสอบความต้องการ หรือมีการทดสอบความต้องการกลับไปยังข้อกำหนดระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี

ส่วนที่ 2 : รายละเอียดโดยรวม (Overall description)

8. การต่อประสานกับผู้ใช้ (User interfaces)

การต่อประสานไปยังระบบย่อยของระบบสั่งซื้อสินค้า จะติดต่อโดยใช้คีย์บอร์ด ซึ่งคีย์บอร์ดนี้ มีฟังก์ชันพิเศษเพื่อลดจำนวนครั้งของการกดแป้นคีย์บอร์ด ซึ่งจะส่งผลกระทบต่อความปลอดภัยของการเปลี่ยนแปลงของระบบสั่งซื้อสินค้า มาตรฐานของคีย์บอร์ด ใช้มาตรฐาน QWERTY สำหรับรายละเอียดต่างๆ อ้างอิงไปยังแผนผังระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี

9. การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)

ระบบย่อยของระบบสั่งซื้อสินค้าจะพิมพ์รายงานออกทางเครื่องพิมพ์ ดังนั้น เครื่องคอมพิวเตอร์และเครื่องพิมพ์ต้องมีการต่อประสานทางด้านฮาร์ดแวร์อย่างง่าย

10. การต่อประสานกับซอฟต์แวร์ (Software interfaces)

ระบบย่อยของระบบสั่งซื้อสินค้าจะต่อประสานกับระบบย่อย ดังต่อไปนี้

- คลังสินค้า (เขียนอย่างเดียว)
- การบัญชี (เขียนอย่างเดียว)
- การตลาดและการขาย (เขียนอย่างเดียว)
- สมุดแจ้งรายการสินค้า (เขียนอย่างเดียว)
- ความคุมเครือข่าย (อ่านและเขียน)

11. การต่อประสานกับระบบสื่อสาร (Communications interfaces)

การสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ต่างๆ ใช้โปรโตคอล IEEE-488

12. ข้อจำกัดของหน่วยความจำ (Memory constraints)

ซอฟต์แวร์นี้ต้องการเนื้อที่ในฮาร์ดดิสก์ไม่ต่ำกว่า 1 กิกะไบต์ และต้องการหน่วยความจำมากกว่า 32 เมกกะไบต์ขึ้นไป

13. การดำเนินการ (Operations)

- เครื่องคอมพิวเตอร์ที่ใช้ในระบบสั่งซื้อสินค้าจะเปิดแบบเชื่อมต่อตรง (on line) เป็นเวลา 16 ชั่วโมงต่อ 1 วัน
- สัญญาณเตือนจะส่งเสียงดังที่เครื่องคอมพิวเตอร์ เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้น จากนั้นจะทำการสำรองข้อมูล เพื่อนำข้อมูลไปใช้ในภายหลังได้

- เพิ่มข้อมูลของเหตุการณ์ปกติ ใช้เป็นต้นแบบในการสร้างมาตรฐานของรายงานและสามารถเข้าถึงข้อมูลโดยมีผู้มีสิทธิ์เท่านั้น ระเบียบต่างๆ ของเหตุการณ์ปกติ ควรจะแสดงบนหน้าจอคอมพิวเตอร์ หรือพิมพ์ออกทางเครื่องพิมพ์

14. ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)

ไม่สามารถใช้ให้เป็นประโยชน์ได้ (Not applicable : N/A)

15. หน้าที่ของซอฟต์แวร์ (Product functions)

ในระบบสั่งซื้อสินค้า ผู้ดำเนินการหรือเสมียนต้องทำงานหรือดำเนินงานตามขั้นตอน ดังต่อไปนี้

- (1) ยอมรับใบสั่งซื้อสินค้าที่ถูกต้องจากลูกค้า
- (2) แจ้งราคาสินค้าให้ถูกต้อง (รวมค่าภาษีและค่าส่งทางไปรษณีย์)
- (3) ยืนยันสถานะบัตรเครดิตของลูกค้า และมีการตรวจสอบบัตรเครดิตให้ถูกต้อง
- (4) ส่งข้อมูลเกี่ยวกับการสั่งซื้อสินค้าไปยังแผนกส่งของ (shipping)
- (5) ส่งใบเสร็จเก็บเงินไปยังแผนกลูกค้า
- (6) ปรับแก้ข้อมูลของลูกค้าให้ทันสมัย
- (7) เปิดเพิ่มข้อมูลของลูกค้าใหม่
- (8) เพิ่มข้อมูลของลูกค้าไปยังบัญชีรายชื่อผู้ที่สั่งซื้อสินค้าทางไปรษณีย์
- (9) ทำสมุดจดรายการเปลี่ยนแปลงในแต่ละรายการ
- (10) บันทึกการเปลี่ยนแปลงในแต่ละรายการลงบนเทป

16. ลักษณะของผู้ใช้ (User characteristics)

ลักษณะโดยทั่วไปของผู้ใช้ในระบบสั่งซื้อสินค้า มีดังนี้

- จบการศึกษาระดับมัธยมปลาย พร้อมทั้งได้รับการฝึกอบรมเพิ่มเติมเป็นเวลา 1-2 ปี
- มีการเลือกระบบสื่อสารที่สะดวกสบายที่สุดในการติดต่อ คือ การติดต่อสื่อสารทางโทรศัพท์
- ผู้ใช้มีความกระตือรือร้นและซื่อตรง
- ผู้ใช้ควรได้รับการฝึกอบรมในการใช้เครื่องคอมพิวเตอร์อย่างน้อย 40 ชั่วโมงต่อสัปดาห์
- ผู้ใช้ไม่ต้องมีความชำนาญในการใช้เครื่องคอมพิวเตอร์เหมือนกับนักคอมพิวเตอร์มืออาชีพ
- มีมาตรฐานในการออกแบบการต่อประสานระหว่างระบบกับผู้ดำเนินการ

17. ข้อกำหนดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)

- ผู้ใช้ควรจะปรับขนาดของตัวอักษรก่อนที่จะพิมพ์ เพื่อเป็นการประหยัดกระดาษ
- เวลาที่ใช้ในการสำรองข้อมูลไม่ควรจะมากกว่า 6 ชั่วโมงต่อวัน

18. ความสามารถในการติดตามความต้องการ (Requirements traceability)

ข้อกำหนดนี้ สามารถตรวจสอบความต้องการหรือมีการทดสอบความต้องการกลับไปยังข้อกำหนดระบบของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี

19. ข้อจำกัด (Constraints)

ระบบสั่งซื้อสินค้าจะได้รับการประมวลผลภายใต้ระบบปฏิบัติการ Zebra 7 release 4 และใช้ระบบปฏิบัติการเครือข่าย GIRAFE-12 release 3 นอกจากนี้ ระบบสั่งซื้อสินค้าจะเขียนด้วยภาษา C++ และเครื่อง

คอมพิวเตอร์ที่ใช้กับระบบนี้ คือ UNIGRAPH model A-3 สำหรับคีย์บอร์ดที่ใช้กับเครื่องคอมพิวเตอร์นี้ คือ คีย์บอร์ดมาตรฐาน QWERTY

20. ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)

- มีการตั้งข้อสันนิษฐานเกี่ยวกับลูกค้าว่าต้องมีความรู้ทางด้านกระบวนการสั่งซื้อสินค้าและการออกแบบซอฟต์แวร์ที่ดีพอ ถ้ามีการเปลี่ยนแปลงการออกแบบซอฟต์แวร์ SRS จะต้องได้รับการเปลี่ยนด้วย
- ระบบย่อยของระบบสั่งซื้อสินค้าจะดำเนินงานภายใต้ระบบปฏิบัติการ Windows 95 หรือ Windows NT และดำเนินงานบนเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นเพนเทียม

21. การแบ่งสัดส่วนส่วนความต้องการ (Apportioning of requirements)

การเพิ่มเติมความต้องการหรือรายละเอียดต่างๆ จะปรากฏบนหน้าจอคอมพิวเตอร์ ซึ่งผู้ดำเนินการกำลังใช้งานอยู่ เช่น อัตราการเปลี่ยนแปลง จำนวน ที่ตั้ง ขนาด และการแสดงความผิดพลาด โดยผู้เชี่ยวชาญเป็นผู้กำหนดสิ่งเหล่านี้ นอกจากนี้ ผู้เชี่ยวชาญยังกำหนดมาตรฐานของหน้าจอคอมพิวเตอร์ และพิจารณาถึงจำนวนครั้งที่ผู้ดำเนินการกดคีย์บอร์ดให้น้อยที่สุด เพื่อช่วยลดความผิดพลาดต่างๆ ที่อาจเกิดขึ้น โดยมีระบบช่วยเหลือแบบ on-line เพื่อเป็นแนวทางให้ผู้ดำเนินการแก้ไขปัญหาต่างๆ ที่เกิดจากความผิดพลาดของซอฟต์แวร์ได้

ส่วนที่ 3 : ความต้องการที่เฉพาะเจาะจง (Specific requirements)

22. การต่อประสานภายนอก (External interfaces)

การต่อประสานภายนอก จะรวมถึงการบำรุงรักษาซอฟต์แวร์ ผู้ดำเนินการจะรายงานและตีความหมายของเหตุการณ์ต่างๆ ให้กับลูกค้าทางโทรศัพท์ โดยจะมีการบำรุงรักษาซอฟต์แวร์ด้วยระบบ Philco RR model 3561 รายงานต่างๆ ที่เก็บบนกระดาด และเก็บลงดิสก์จะได้รับการบำรุงรักษา ซึ่งผลลัพธ์เหล่านี้ จะได้รับการควบคุมโดยบุคคลที่มีอำนาจในการจัดการข้อมูลเท่านั้น

23. ความต้องการด้านหน้าที่ (Functional requirements)

หน้าที่ของระบบสั่งซื้อสินค้า ประกอบด้วย 4 ส่วน ได้แก่

1. หน้าที่ของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า
 - 1.1 บทนำ

หน้าที่ในส่วนนี้ เป็นการเก็บรวบรวมข้อมูล และแสดงรายละเอียดต่างๆ ของข้อมูลบนหน้าจอคอมพิวเตอร์ที่ส่งผลกระทบต่อการใช้ของลูกค้า และการขายโดยผู้ดำเนินการ หน้าที่ในการเข้าถึงฐานข้อมูลเป็นแบบอ่านอย่างเดียว และการแสดงรายละเอียดต่างๆ ไปยังผู้ดำเนินการ รายละเอียดเหล่านี้ จะได้รับการถ่ายทอดไปยังลูกค้าทางโทรศัพท์ด้วย

1.2 ข้อมูลนำเข้า (Input)

มาตรฐานของข้อมูลนำเข้าในการป้อนข้อมูลไปยังระบบสั่งซื้อสินค้า จะใช้หมายเลขประจำตัวลูกค้า (customer identification number: CID) และหมายเลขประจำตัวสินค้า (product identification number: PID) โดย CID จะเป็นอักขระตัวอักษรผสมตัวเลข (alphanumeric) 6 หลัก และ PID จะเป็นอักขระตัวอักษรผสมตัวเลข 12 หลัก

ชนิดของข้อมูลนำเข้าประเภทที่สอง คือ รหัสของลูกค้าใหม่ ซึ่งลูกค้าใหม่จะไม่มี CID ผู้ดำเนินการจะป้อนข้อมูลลงในรายการเปลี่ยนแปลงของลูกค้าใหม่ เมนูจะแสดงหน้าจอของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า เพื่อให้กรอกชื่อลูกค้า ที่อยู่ และเบอร์โทรศัพท์ รายละเอียดเหล่านี้ จะยังคงเก็บอยู่ในหน่วยความจำจนกระทั่งทำรายการเปลี่ยนแปลงเสร็จสมบูรณ์

หน้าที่ของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้าจะยอมรับชนิดของบัตรเครดิต หมายเลขบัตรเครดิตและวันที่หมดอายุของบัตรเครดิต ผู้ดำเนินการจะป้อนคำสั่งต่างๆ เพื่อตรวจสอบความถูกต้องของบัตรเครดิต เมนูจะแสดงบัญชีรายชื่อชนิดของบัตรเครดิตที่ได้รับการยอมรับ ผู้ดำเนินการสามารถกดแป้นคีย์บอร์ดเพื่อเลือกชนิดของบัตรเครดิตได้ สำหรับหมายเลขบัตรเครดิต และวันที่หมดอายุของบัตรเครดิต จะเก็บอยู่ในระเบียบของลูกค้ารายนั้น

เมื่อกดฟังก์ชันคีย์ เมนูแจ้งรายการสินค้าจะปรากฏขึ้นมา โดยมีทางเลือกดังต่อไปนี้

- 1) Catalogue -- season, year (12 characters)
- 2) Flyer -- season, month, year, number (18 characters)
- 3) Home shopping -- week, month (10 characters)
- 4) Other

เมนูย่อยจะจัดหาทางเลือกต่างๆ ดังต่อไปนี้

- 1) PID (6 characters)
- 2) Product key words (50 characters)
- 3) Page number (6 characters)
- 4) Other

1.3 กระบวนการ (Processing)

ข้อมูลนำเข้าในการสั่งซื้อสินค้า จะใช้ PID เป็นคีย์ เพื่อที่จะดาวน์โหลดรายชื่อสินค้าที่เป็นข้อความและรูปภาพให้ปรากฏบนหน้าจอคอมพิวเตอร์ พร้อมทั้งจะดาวน์โหลดข้อมูลของลูกค้าที่สมบูรณ์โดยใช้ CID เป็นคีย์

การดาวน์โหลดข้อมูลทั้ง 2 ระเบียบ จะใช้เวลาน้อยกว่า 1 วินาที เวลาที่ใช้ในการดาวน์โหลดจะวัดจากการกดแป้นคีย์บอร์ดเพื่อป้อน PID หรือ CID ให้ไปปรากฏบนหน้าจอคอมพิวเตอร์

รายละเอียดบัตรเครดิตของลูกค้าใหม่ จะมีรูปแบบของข้อความที่เก็บรายละเอียดของบัตรเครดิต ได้แก่ type, number, expiration, date, operator ID, station ID, date, และ time ข้อความต่างๆ จะได้รับการตรวจสอบความถูกต้องของบัตรเครดิตภายในเวลา 1 วินาที พร้อมทั้งทำการโต้ตอบหรือแสดงผลภายใน 2 วินาที ผู้ดำเนินการสามารถอ่านรายละเอียดเหล่านี้ได้อย่างชัดเจน และไม่มี ความคลุมเครือเกิดขึ้น

1.4 ข้อมูลนำออก (Output)

หน้าที่ของข้อมูลนำออกในการสั่งซื้อสินค้า จะมีรูปแบบในการส่งผ่านข้อความผ่านเครือข่ายไปยังแฟ้มข้อมูล และระบบย่อย ดังต่อไปนี้

ไปยังระบบบัญชี เพื่อบันทึกลงในระเบียบเหล่านี้ คือ CID, PID, price, tax, shipping cost, date, time, operator ID

1.5 การจัดการความผิดพลาด (Error Handling)

การป้อนข้อมูลในระบบสั่งซื้อสินค้า ถ้าข้อมูลที่ป้อนเข้าไปในระบบมีความผิดพลาดเกิดขึ้น ระบบจะโต้ตอบด้วยการแสดงข้อความผิดพลาด เช่น มีอักขระมากเกินไป มีอักขระน้อยเกินไป มีอักขระที่ไม่ถูกต้อง เป็นต้น

2. หน้าทีของกระบวนการสั่งซื้อสินค้า

2.1 บทนำ

2.2 ข้อมูลนำเข้า

2.3 กระบวนการ

2.4 ข้อมูลนำออก

2.5 การจัดการความผิดพลาด

3. หน้าทีของข้อมูลนำออกในการสั่งซื้อสินค้า

3.1 บทนำ

3.2 ข้อมูลนำเข้า

3.3 กระบวนการ

3.4 ข้อมูลนำออก

3.5 การจัดการความผิดพลาด

4. หน้าที่ของระบบช่วยเหลือในการสั่งซื้อสินค้า

4.1 บทนำ

4.2 ข้อมูลนำเข้า

4.3 กระบวนการ

4.4 ข้อมูลนำออก

4.5 การจัดการความผิดพลาด

24. ความต้องการด้านประสิทธิภาพ (Performance requirements)

เมื่อผู้ใช้ร้องขอไปยังระบบ ระบบจะโต้ตอบภายในเวลา 1 วินาที การโต้ตอบนี้ จะวัดจากการป้อนข้อมูลหรือการกดแป้นคีย์บอร์ดครั้งสุดท้าย เพื่อให้ข้อมูลไปปรากฏบนหน้าจอคอมพิวเตอร์ และในการเข้าถึงฐานข้อมูลโดยทั่วไป ควรใช้เวลาภายใน 1 วินาที

25. ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)

- การเข้าถึงฐานข้อมูลแบบอ่านอย่างเดียว และการแสดงรายละเอียดต่างๆ ไปยังผู้ดำเนินการ รายละเอียดเหล่านี้ จะได้รับการถ่ายทอดไปยังลูกค้าทางโทรศัพท์ด้วย
- ระเบียบของแฟ้มข้อมูลที่มีความผิดพลาดเกิดขึ้น จะเก็บข้อมูลไว้ภายใน 6 เดือน หลังจาก 6 เดือนแล้ว ควรเก็บข้อมูลลงบนเทปเป็นเวลา 3 ปี หลังจากนั้น จึงจะทำลายข้อมูลนั้นทิ้ง

26. ความต้องการด้านการดำเนินการ (Operational requirements)

ระบบสั่งซื้อสินค้าจัดหาระบบช่วยเหลือให้กับผู้ใช้ เมื่อมีสถานการณ์ที่ผิดปกติในการติดต่อระหว่างผู้ใช้งานกับระบบเกิดขึ้น การป้อนข้อมูลในระบบสั่งซื้อสินค้า ถ้าข้อมูลที่ป้อนเข้าไปในระบบมีความผิดพลาดเกิดขึ้น ระบบจะโต้ตอบด้วยการแสดงข้อความผิดพลาด เช่น มีอักขระมากเกินไป มีอักขระน้อยเกินไป มีอักขระที่ไม่ถูกต้อง เป็นต้น

27. ความต้องการด้านการทวนสอบ (Verification requirements)

ผลลัพธ์ของโปรแกรม จะใช้เวลาภายใน 20 วินาทีในการประมวลผลเหตุการณ์ X 60% และจะใช้เวลาภายใน 30 วินาทีในการประมวลผลเหตุการณ์ X 99%

28. ความต้องการด้านการยอมรับ (Acceptance requirements)

- ระบุถึงข้อจำกัดในการตรวจรับซอฟต์แวร์
- ระบุถึงการตรวจสอบความถูกต้องในการส่งมอบซอฟต์แวร์ เพื่อดูว่าซอฟต์แวร์สามารถทำหน้าที่ได้ตรงตามความต้องการของผู้ใช้หรือลูกค้าหรือไม่

29. ความต้องการด้านเอกสาร (Documentation requirements)

- เอกสารในการออกแบบ ได้แก่ ข้อกำหนดของการออกแบบซอฟต์แวร์ (Software Design Specification : SDS) โดยใช้มาตรฐาน ANSI/IEEE 1016-1987
- เอกสารของซอฟต์แวร์ จะรวมถึง ข้อกำหนดในการออกแบบ ซึ่งอ้างอิงจากมาตรฐาน IEEE 830-1993 และเซตของการประมวลผลกรณีทดสอบ (test cases) ที่มีผลลัพธ์เหมาะสม
- เอกสารการทดสอบซอฟต์แวร์ จะรวมถึง แผนการทดสอบระบบ (test plan) การสร้างกรณีทดสอบ และผลลัพธ์จากการสร้างกรณีทดสอบ

- เอกสารการบำรุงรักษาซอฟต์แวร์ ควรจะรวมถึงการบำรุงรักษาซอฟต์แวร์ระหว่างขั้นตอนการดำเนินการ
30. ความต้องการด้านความปลอดภัย (Safety requirements)
- ระบบควรจะทำให้แน่ใจว่าข้อมูลจะไม่ได้รับความเสียหาย เมื่อมีความผิดพลาดเกิดขึ้นกับระบบ
 - สถานะของเหตุการณ์ที่ผิดปกติจะบันทึกลงในรายงานประจำวันของระบบนั้น หลังจากมีการตัดสินใจกับเหตุการณ์ที่ผิดปกตินี้แล้ว ผู้ใช้จะสิ้นสุดการใช้งาน
 - เลือกระเบียน (fields) จากรายงานที่มีความผิดพลาดเกิดขึ้น และนำกระเบียนเหล่านี้พิมพ์ใส่กระดาษ (hard copy)
31. ความต้องการด้านสภาพแวดล้อม (Environment requirements)
- ในการติดตั้งซอฟต์แวร์ ควรจะมีอุณหภูมิห้องระหว่าง 18-25 องศาเซลเซียส
 - ไม่ควรติดตั้งซอฟต์แวร์ใกล้ห้องเครื่องปรับอากาศ
32. ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
- เนื้อที่ว่างในฮาร์ดดิสก์ 5 เมกกะไบต์ขึ้นไป
 - โปรแกรมทั้งหมดจะได้รับการประมวลผลภายในโควตาของผู้ใช้ที่กำหนดไว้
33. ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communications requirements)
- การสื่อสารข้อมูลระหว่างระบบย่อยของระบบส่งข้อความและระบบคลังสินค้า ระบบคลังสินค้าจะส่งผ่านข้อความไปยังผู้รับ เมื่อต้นทางคือระบบคลังสินค้า
 - การสื่อสารข้อมูลระหว่างระบบย่อยของระบบส่งข้อความและระบบบัญชี ถ้าต้นทางคือ ระบบบัญชี จะทำการส่งข้อความเพื่อกำหนดการดำเนินการที่จะใช้ในระบบนี้
34. ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
- SRS มีความต้องการกัน เช่น รูปแบบของรายงาน ถ้าในข้อกำหนดหนึ่ง อธิบายความต้องการในรูปแบบของตาราง ในข้อกำหนดอื่นๆ ต้องอธิบายความต้องการในรูปแบบของตารางด้วย
 - SRS สามารถตรวจสอบความถูกต้องของซอฟต์แวร์ เช่น ผลลัพธ์ของโปรแกรม จะใช้เวลาภายใน 20 วินาทีในการประมวลผลเหตุการณ์ X 60% และใช้เวลาภายใน 30 วินาทีในการประมวลผลเหตุการณ์ X 100%
35. ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)
- ผู้พัฒนาซอฟต์แวร์ควรได้รับการฝึกอบรมเกี่ยวกับภาษาที่ใช้เขียนโปรแกรม ก่อนที่จะทำการพัฒนาระบบส่งข้อความทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี
 - ผู้พัฒนาซอฟต์แวร์จะฝึกอบรมให้กับผู้ใช้ในเรื่องการบำรุงรักษาซอฟต์แวร์และการเปลี่ยนแปลงแก้ไขของระบบส่งข้อความทางไปรษณีย์อัตโนมัติของบริษัทเอบีซี
36. ความต้องการด้านการเตรียมขั้นตอนต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
- ผู้ใช้สามารถดาวน์โหลดเวอร์ชันใหม่ของซอฟต์แวร์จากเว็บไซต์

- ผู้ใช้สามารถสอบถามปัญหาต่างๆ จากผู้พัฒนาซอฟต์แวร์ทางโทรศัพท์
- ผู้ใช้ควรจะสามารถสำรองข้อมูลบนเทป

37. ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)

- ซอฟต์แวร์นี้ จะบรรจุภัณฑ์ลงบนซีดีรอมและเทป สำหรับส่งมอบซอฟต์แวร์ให้กับลูกค้า
- การบรรจุภัณฑ์นี้ นอกเหนือจากซอฟต์แวร์ที่จะส่งมอบให้กับลูกค้าแล้ว ยังรวมถึงเอกสารต่างๆ ด้วย เช่น เอกสารข้อกำหนดความต้องการซอฟต์แวร์ เอกสารคู่มือผู้ใช้ซอฟต์แวร์ เอกสารการออกแบบซอฟต์แวร์ เอกสารการทดสอบซอฟต์แวร์ เอกสารการบำรุงรักษาซอฟต์แวร์ เป็นต้น

38. ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)

ข้อบังคับที่ใช้ในการออกแบบ คือ ใช้ UNIGRAPH model A-3

39. มาตรฐานที่มีใช้ในองค์กร (Standard compliance)

การออกแบบและการพัฒนาซอฟต์แวร์ จะอ้างอิงจากมาตรฐาน ANSI/IEEE 1016-1987 และอ้างอิงจากมาตรฐานของบริษัทเอบีซี ถ้ามาตรฐานทั้งสองเกิดความขัดแย้งกัน จะประยุกต์ใช้มาตรฐานของบริษัทเอบีซีเป็นหลักในการพัฒนาซอฟต์แวร์

40. ความเชื่อถือได้ (Reliability)

เวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้น จะต้องมากกว่า 5 วันขึ้นไป และภายในระยะเวลา 6 เดือน เวลาเฉลี่ยที่ซอฟต์แวร์สามารถทำงานได้ก่อนมีความผิดพลาดเกิดขึ้นโดยรวมเท่ากับ 4 สัปดาห์

41. สภาพพร้อมใช้งาน (Availability)

เครื่องคอมพิวเตอร์ที่ใช้ในระบบสั่งซื้อสินค้า จะมีสภาพพร้อมใช้งาน 99.9% ซึ่งสภาพพร้อมใช้งาน ได้แก่ ระบบปฏิบัติการ โปรแกรมประยุกต์ และฮาร์ดแวร์

42. ความมั่นคง (Security)

ระบบสั่งซื้อสินค้าจะอนุญาตให้เฉพาะผู้ที่มีสิทธิ์หรืออำนาจในการเข้าถึงข้อมูลในระบบนี้เท่านั้น ผู้ดำเนินการ เหมียน และผู้จัดการสามารถเข้าถึงข้อมูล เพิ่มข้อมูล และระบบย่อยอื่นๆ ตลอดจนโปรแกรมประยุกต์ของระบบสั่งซื้อสินค้า สำหรับระบบย่อยอื่นๆ เช่น คลังสินค้า การบัญชี เป็นต้น จะได้รับการป้องกันจากการเข้าถึงข้อมูลหรือการจัดการเพิ่มข้อมูลจากระบบย่อยของระบบสั่งซื้อสินค้า

43. ความสามารถในการบำรุงรักษา (Maintainability)

- เวลาเฉลี่ยในการแก้ไขความผิดพลาดต้องไม่เกิน 1 สัปดาห์
- เอกสารในการออกแบบ ได้แก่ ข้อกำหนดของการออกแบบซอฟต์แวร์ (Software Design Specification : SDS) โดยใช้มาตรฐาน ANSI/IEEE 1016-1987 และเอกสารการทดสอบซอฟต์แวร์ จะรวมถึง แผนการทดสอบระบบ (test plan) การสร้างกรณีทดสอบ และผลลัพธ์จากการสร้างกรณีทดสอบ

44. ความสามารถในการใช้ได้หลายระบบ (Portability)

- ซอฟต์แวร์นี้ สามารถใช้งานได้หลายแพลตฟอร์ม X และ Y
- ไม่มีสไลด์ส่วนหนึ่งของซอฟต์แวร์ที่เขียนด้วยภาษาแอสเซมบลี

- ถ้ามีการเปลี่ยนแปลงรหัสต้นทาง (source code) น้อยกว่า 2% จะไม่ส่งผลกระทบต่อถึงการแปลโปรแกรม (compile) ใหม่อีกครั้งหนึ่งในซอฟต์แวร์นี้ เพื่อให้ดำเนินงานบนเครื่องคอมพิวเตอร์ X ได้

ส่วนที่ 4 : ข้อมูลสนับสนุน (Supporting information)

45. สารบัญ (Table of contents)

ส่วนที่

1	บทนำ.....	1
2	รายละเอียดโดยรวม.....	2
3	ความต้องการที่เฉพาะเจาะจง.....	5
4	ข้อมูลสนับสนุน.....	13

46. บันทึก (Notes)

หน้านี้มีไว้เพื่อบันทึกสิ่งที่ต้องการอธิบายเพิ่มเติมใน SRS

47. ภาคผนวก (Appendixes)

- ภาคผนวก ก : ตารางของการตรวจสอบความต้องการ (Requirements traceability matrix)
- ภาคผนวก ข : สารบัญของเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- ภาคผนวก ค : แผ่นแบบ (template) ของ SRS

48. ดัชนี (Index)

- ความต้องการด้านหน้าที่, 44
- วัฏจักรของการพัฒนาซอฟต์แวร์, 3, 4
- เอกสารข้อกำหนดความต้องการซอฟต์แวร์, 1, 5, 7, 12

ตัวอย่างในการเขียน SRS ของระบบสั่งซื้อสินค้าทางไปรษณีย์อัตโนมัติของบริษัทเอบีซีที่เป็นภาษาอังกฤษ

ABC Corporation's automated mail-order system

Section 1 : Introduction

1. Purpose

This SRS describes the function and performance requirements allocated to the order-entry subsystem.

2. Scope

This specification is intended to form the basis for design of a software product that will support up to 100 special-purpose order-entry workstations manned by trained operators.

3. Definitions

- Customer -- The person, or persons, who pays for the product and usually decides the requirements.
- User -- A person who utilizes the products or services of a system, or a person who controls and monitors the hardware and software of a system (i.e. an end user, an operator, or both).

4. Acronyms

- IEEE -- Institute of Electrical and Electronics Engineers
- SDS -- Software Design Specification
- SRS -- Software Requirements Specification

5. Abbreviations

- CID -- customer identification number
- PID -- product identification number
- Std -- Standard

6. References

A list of all directly applicable references along with their release number, date, and source for obtaining copies.

- IEEE Std 610.12-1990, IEEE Standard Glossary for Software Engineering Terminology (ANSI).
- IEEE Std-830, IEEE New York, 1984 and 1993.

7. Overview

The order-entry subsystem is a component of the ABC Corporation's automated mail-order system. The automated mail-order system is described in a document entitled Automated Mail-Order System Design Description Release 4 October 1996. The requirements specified in this document have been allocated from the automated mail-order system system specification approved on September 12, 1995. Section 3.8 of this specification traces requirements and requirement testing to the ABC automated mail-order system system specification.

Section 2 : Overall description

8. User interfaces

The order-entry subsystem interfaces with the order-entry operator's keyboard. The keyboard is unique with special function keys that are made available to reduce the number of keystrokes that are required to effect common steps in order-entry transactions. The special-purpose keyboard is

adjunct to the standard QWERTY keyboard provided with the workstation. For more detail refer to system diagram of the ABC Corporation's automated mail-order system.

9. Hardware interfaces

The order-entry subsystem will printout report to printer so it has simple hardware interface between personal computer and printer.

10. Software interfaces

The order-entry subsystem shall interface with the following subsystems:

- Warehouse (write only)
- Accounting (write only)
- Marketing and sales (write only)
- Catalogue (write only)
- Network control (read and write)

11. Communications interfaces

Communications between the computer and remote instruments shall be via the IEEE-488 protocol.

12. Memory constraints

This software product requires at least 1 GB of harddisk space and requires memory more than 32 MB.

13. Operations

- The order-entry workstation is a key subsystem. Each operating workstation shall be on line 16 hours per day.
- An alarm signal shall be transmitted to alert the on-duty manager's workstation that an emergency incident has occurred. A duplicate copy of the recording associated with the incident shall also be forwarded immediately to the on-duty manager's workstation.
- The incident file is the source for three standard reports and is accessible only to authorized personnel. Incident records should be displayed on screen or printed in hard copy.

14. Site adaptation requirements

Not applicable (N/A)

15. Product functions

The order-entry software product is expected to provide support to the order-entry operator or clerk in the following tasks:

- (1) accepting customer orders correctly
- (2) quoting product price correctly (including tax and mailing)

- (3) confirming customer credit status and credit card validity
- (4) passing order details to shipping department
- (5) passing billing information to accounts receivable
- (6) updating customer files
- (7) opening new customer files
- (8) adding customer to mailing list
- (9) logging each transaction
- (10) writing to tape a log of each transaction

16. User characteristics

The order-entry operators typically have the following personal characteristics:

- High school graduate frequently with one to two years additional academic training.
- Selected for ability to deal successfully with public and comfortable with telephone communication.
- Highly motivated and conscientious.
- Computer-literate with minimum one 40-hour week hands-on training on workstation.
- Operators are not computer professionals.
- System-to-operator interface design standards.

17. Qualification provisions

- Users should adjust font size before printing to save paper cost.
- Each daily backup time should not last more than 6 hours.

18. Requirements traceability

This specification traces requirements and requirement testing to the ABC automated mail-order system system specification.

19. Constraints

The order-entry software product shall execute under release 4 of the Zebra 7 Operating System. The network operating system is GIRAFE-12 release 3. The order-entry software product code shall be written in C++. The order-entry workstations shall be UNIGRAPH, model A-3. A special-purpose keyboard is associated with each workstation. This keyboard is used as adjunct to the standard QWERTY keyboard.

20. Assumptions and dependencies

- It is assumed the customer is proficient in order-entry process and the software is being designed in such a fashion. If it is found that this is not the case, then changes in the design will be implemented and the SRS will have to change accordingly.

- The order-entry subsystem will run under Windows 95 or Windows NT operating system and run on Pentiums compatible computers.

21. Apportioning of requirements

Additional requirements dealing with the physical characteristics of the information shown on the screen to operators, such as rate of change, amount, location, size, and error indications, have been imposed by human factors specialists. A standard screen or panel has also been developed by human factors specialists, and all displays (soft copy) must conform. The need to minimize operator keystrokes and at the same time provide as many operator aids as possible to reduce or eliminate operator errors is an implicit requirement. A requirement for an on-line help menu to guide new operators or operators with unusual problems is now emerging from system analysts.

Section 3 : Specific requirements

22. External interfaces

External interfaces shall be maintained with telephone-based reports as interpreted and entered by the operator. An external interface shall be maintained with the Philco RR model 3561 recording system. Hard- and soft-copy reports shall be the only output maintained. These outputs shall be controlled by authorized management personnel.

23. Functional requirements

The order-entry function consists of the four components. The first component, order-entry input, performs the following functions.

1. Order-Entry Input Function

1.1 Introduction

This function gathers and displays all essential data required to effect a buy (by the customer) and sell (by the operator) decision. The function accesses databases on a read-only basis and presents information to the operator that is to be relayed to the customer via telephone. An implicit requirement is that the transaction be effected as quickly as possible without appearing to hurry the customer.

1.2 Input

Standard inputs to the order-entry input function shall be customer identification number (CID) and product identification number (PID). The CID shall be a 6 character alphanumeric value. The PID shall be a 12-character alphanumeric value.

A second input type (new customer code) shall be accepted by the order-entry input function. This input identifies a new customer without an assigned CID. A new-customer transaction shall be identified by the operator with a single keystroke. A menu shall be

provided by the order-entry input function to accept customer name, address, and telephone number. This information shall be retained in local memory until transaction completion.

The order-entry input function shall accept credit card type, number, and expiration date. The operator shall enter a single keystroke command to activate the credit card validate menu. The menu shall list acceptable credit card types. A single operator-generated keystroke shall be provided to select credit card type. Credit card number and expiration date shall be entered via keyboard. Credit card type and number are fields in a customer record.

A single function key shall be provided to bring up a catalogue menu with the following options:

- a. Catalogue – season, year (12 characters)
- b. Flyer – season, month, year, number (18 characters)
- c. Home shopping – week, month (10 characters)
- d. Other

A submenu shall be provided for a, b, and c with the following options:

- a. PID (6 characters)
- b. Product key words (50 characters)
- c. Page number (6 characters)
- d. Other

1.3 Processing

The order-entry input shall use the PID as a key field to download catalogue text and images for a specific product or family of products to the workstation. The order-entry input shall download a complete customer record using the CID as the key field.

The complete download of both records shall be accomplished in less than 1 second. Download time shall be measured from keystroke entry of PID or CID to on-screen appearance of the subject record.

Credit card information for new customers is normally entered at this point. The order-entry input function shall format a message containing credit card information (type, number, expiration date, operator ID, station ID, date, and time). The message shall be forwarded to the credit card validation file within 1 second. The reply shall be displayed within 2 seconds. The reply shall be unambiguous and clearly readable by the operator.

1.4 Output

The order-entry output function shall format and transmit over the network the following messages to files and subsystems:

To accounting subsystem, a single record with the following fields: CID, PID, price, tax, shipping cost, date, time, operator ID.

1.5 Error Handling

The order-entry input function shall recognize the following data entry errors and response with appropriate error messages.

- Too many characters
- Too few characters
- Invalid characters

2. Order Entry Process Function

2.1 Introduction

2.2 Input

2.3 Processing

2.4 Output

2.5 Error Handling

3. Order Entry Output Function

3.1 Introduction

3.2 Input

3.3 Processing

3.4 Output

3.5 Error Handling

4. Order Entry Help Function

4.1 Introduction

4.2 Input

4.3 Processing

4.4 Output

4.5 Error Handling

24. Performance requirements

All operator requests shall be responded to within one second. Response shall be measured from the last entry keystroke to the appearance of an on-screen reply. And general 1-second requirement on all database accesses, unless otherwise specified.

25. Logical database requirements

- The function accesses databases on a read-only basis and presents information to the operator that is to be relayed to the customer via telephone.
- An accident file record will be retained on-line for 6 months. After expiration of 6 months, the record will be archived on tape for 3 years, after which it will be destroyed.

26. Operational requirements

The order-entry software product is also expected to provide a help function to enable operators to deal with unusual situations. The order-entry input function shall recognize the following data entry errors and respond with appropriate error messages such as too many characters, too few characters, invalid characters., etc.

27. Verification requirements

The output of the program shall be given within 20s of event X, 60% of the time; and shall be given within 30s of event X, 99% of the time.

28. Acceptance requirements

- Acceptance constrain the design of the product.
- Validation confirms whether the user requirements are satisfied when the software is delivered.

29. Documentation requirements

- Design documentation shall include a complete listing for the order-entry input application product and a complete Software Design Specification (SDS) per ANSI/IEEE Standard 1016-1987.
- Software product documentation shall include a design specification as described in IEEE 830-1993, a commented listing (one comment line per three executable lines), and a set of executable test cases with appropriate results.
- Testing documentation shall include test plan, test cases, and test case results.
- Maintenance documentation should include all of detailed agreements regarding the maintenance of the software during the operational phase by the system developer or, as in many cases, by a third party.

30. Safety requirements

- The system should ensure that no data is lost when a power failure occurs.
- The emergency status entry shall signal completion of the incident report record. The operator shall terminate the call after making the emergency condition decision.
- Selected fields from the emergency report should be produced in hard copy to the driver.

31. Environment requirements

- The temperature of software installed room should be between 18-25 Celsius degree.
- Don't install the software near air-condition controlled room.

32. Computer hardware resource utilization requirements

- 5 Mbytes of disk space are available.
- All programs shall execute with standard user quotas.

33. Computer communications requirements

- Communications between order-entry subsystem and warehouse shall pass the data along to the receiver when the source is warehouse.
- Order-entry subsystem communicates with accounting. If accounting is the source the data to determine which operation and which sink is to be used.

34. Software quality factors

- An SRS is consistency. Example, the format of an output report should be described in one requirement as tabular and in another as tabular too.
- An SRS is verifiable. Example, output of the program shall be produced within 20s of event X 60% of the time; and shall be produced within 30s of event X 100% of the time.

35. Training-related requirements

- The training course in programming language should be held for programmer before developing the ABC Corporation's automated mail-order system.
- Developers will train users how to maintain and modify the ABC Corporation's automated mail-order system.

36. Logistics-related requirements

- Users can download new version of software from web site.
- Users can inquire problems from developers via telephone.
- Users should backup data on tape.

37. Packaging requirements

- This software product will be packaged on CD-ROM and tape when the software is delivered to clients.
- This packaging has documentation shipped with software when released such as Software Requirements Specification (SRS), Software User Manual, Design document, Testing document, Maintenance document., etc.

38. Implementation constraints

UNIGRAPH model A-3 characteristics and limitations.

39. Standard compliance

Design and development shall conform to ANSI/IEEE Standard 1016-1987, and ABC corporate standards for software development. In the event of conflict, ABC Corporation's software development standards shall apply.

40. Reliability

The minimum time between severe failures shall be in excess of 5 days and shall be 4 weeks, averaged over 6 months.

41. Availability

Order-entry workstations shall have an availability of 99.9%. Availability includes operating system, application software product, and workstation hardware.

42. Security

Order-entry product software is allocated the responsibility for ensuring that only authorized order-entry operations, clerks, and managers can access the order-entry workstation software and access files and other subsystems through the order-entry workstation's order-entry application software. Similarly, other subsystems shall be prevented from accessing and activating order-entry functions or manipulating files via the order-entry subsystem.

43. Maintainability

- The time to repair shall never exceed 1 week.

- Design documentation shall include a complete listing for the order-entry input application product and a complete SDS per ANSI/IEEE Standard 1016-1987. Testing documentation shall include test plan, test cases, and test case results.

44. Portability

- The software shall be portable between environments X and Y.
- No part of the software shall be written in assembler.
- It shall be possible to recompile this software to run on computer X without modifying more than 2% of the source code.

Section 4 : Supporting information

45. Table of contents

Section	
1.	Introduction..... 1
2.	Overall description..... 2
3.	Specific requirements..... 6
4.	Supporting information..... 14

46. Notes

This page is provided for notes that you wish to make in using this SRS.

47. Appendixes

- APPENDIX A : REQUIREMENTS TRACEABILITY MATRIX
- APPENDIX B : SRD table of contents
- APPENDIX C : MANDATORY PRACTICES

48. Index

- functional requirements, 44
- life cycle, 3,4
- Software Requirements Specification (SRS), 1, 5, 7, 12

ภาคผนวก ข

ในส่วนนี้ แสดงถึงบทความที่ตีพิมพ์จากวิทยานิพนธ์ และนำเสนอในงานประชุมวิชาการ NECTEC

การพัฒนากระบวนการกำหนดมาตรฐานการผลิตซอฟต์แวร์คอมพิวเตอร์
(A Development of Standard Software Development Process)

โดย

ดร.วันชัย รั้วไพบูลย์

น.ส.น้ำฝน อัครเมฆิน

ห้องปฏิบัติการวิศวกรรมซอฟต์แวร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

E-mail : Wanchai.R@chula.ac.th

Abstract

Requirements Engineering is an essential process in Software Development. As an evidence that the failure rate of Software project is increased due to the incompleteness of Requirements Specification.

This study concentrates on the standardization of Software Requirements Specification. Using International standard as a basis for Thai working culture. Expected that the result will upgrade Thai Software Industry in the future.

Key Words

Software Requirements, Software Development, Software Industry

บทคัดย่อ

วิศวกรรมความต้องการนับเป็นสิ่งจำเป็นสำหรับการพัฒนาซอฟต์แวร์ ตามความเป็นจริงที่ว่าซอฟต์แวร์ที่สร้างจากข้อกำหนดความต้องการที่ไม่ครบถ้วน จะมีผลทำให้โครงการไม่ประสบผลสำเร็จ

* รองศาสตราจารย์

** นิสิตมหาดบัณฑิต

ผู้วิจัยได้ให้ความสนใจในการแก้ปัญหาเอกสารความต้องการซอฟต์แวร์ที่ไม่เป็นมาตรฐาน โดยการประยุกต์มาตรฐานการทำเอกสารความต้องการของสถาบันในต่างประเทศ ให้เข้ากับวัฒนธรรมการทำงานของสังคมไทย เพื่อเป็นแนวทางสำหรับการปรับปรุงอุตสาหกรรมซอฟต์แวร์ในประเทศไทย

คำสำคัญ

ข้อกำหนดความต้องการซอฟต์แวร์ การพัฒนาซอฟต์แวร์ อุตสาหกรรมซอฟต์แวร์

บทนำ

นับแต่มีนวัตกรรมคอมพิวเตอร์เกิดขึ้นเมื่อราว 1960 การผลิตซอฟต์แวร์ จึงเกิดมีความจำเป็นขึ้นเพื่อประยุกต์นวัตกรรมขั้นนี้ให้เกิดประโยชน์สูงสุด การผลิตหรือการพัฒนาซอฟต์แวร์เป็นกระบวนการที่ซับซ้อนต้องการมโนทัศน์และจินตนาการที่เป็นระบบ รวมถึงการจัดการโครงการที่ทันสมัย จนในปัจจุบันเป็นที่ยอมรับว่า การผลิตซอฟต์แวร์เป็นกระบวนการทางวิศวกรรมสาขาหนึ่ง คือ วิศวกรรมซอฟต์แวร์ (Software Engineering)

ปัจจัยหนึ่งที่มีส่วนสำคัญในการพัฒนาซอฟต์แวร์ ก็คือข้อกำหนดความต้องการซอฟต์แวร์ยังมีลักษณะไม่ครบถ้วน หรือครอบคลุมความต้องการของผู้ใช้ได้ดีพอ ซึ่งเป็นผลต่อต้นทุนในการพัฒนาซอฟต์แวร์มาก [1] ดังนั้นจึงได้มีการกำหนดเป็นวิชาวิศวกรรมความต้องการ (Requirements Engineering) ขึ้นในหลักสูตรวิศวกรรมคอมพิวเตอร์ (Computer Engineering) และวิทยาศาสตร์คอมพิวเตอร์ (Computer Science) และได้มีความพยายามในการสร้างเครื่องมือช่วย (Computer Aided Software Engineering: CASE) เช่น ณัฐพล ลือพรหมชัย และ วันชัย ริวโพนุลย์ [2] ได้พัฒนา CARE: Computer Aided Requirements Engineering โดยใช้แนวทางการเก็บรวบรวมความต้องการตามหลักการ View Point ของ Sommerville

ในขณะที่มีมาตรฐานที่วาดด้วยกำหนดและการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ได้มีการกำหนดไว้แล้วในต่างประเทศก็ตาม คณะผู้วิจัยมีความสนใจในความเป็นมาของปัญหาดังกล่าว มีแนวความคิดในการวิจัยในเชิงปฏิบัติการ โดยทำการสำรวจ และรวบรวมข้อดีและข้อเสียของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้งภายในประเทศและต่างประเทศ และเมื่อพิจารณาประกอบกับมาตรฐานในต่างประเทศซึ่งเป็นที่ยอมรับกัน นำมาวิเคราะห์หาส่วนประกอบของเอกสารที่เหมาะสมสำหรับการพัฒนาซอฟต์แวร์ในประเทศไทย อันจะเป็นแนวทางในการความต้องการซอฟต์แวร์ให้ได้ครบถ้วนมากที่สุด

การวิจัยในส่วนการกำหนดความต้องการนี้ เป็นส่วนหนึ่งของการวิจัย เรื่องการพัฒนากระบวนการกำหนดมาตรฐานการผลิตซอฟต์แวร์คอมพิวเตอร์ ซึ่งได้วางขอบเขตไว้รวมถึงการทดสอบซอฟต์แวร์

แนวทางการวิจัย

ในการวิจัยเพื่อยกร่างคู่มือในการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อให้สามารถใช้งานได้ในทางปฏิบัติ โดยคำนึงถึงการใช้งานในประเทศเป็นเบื้องต้น เอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นผลจากการปฏิบัติตามคู่มือดังกล่าว ควรจะครอบคลุมเนื้อหา หรือองค์ประกอบต่างๆ ที่จำเป็นในการพัฒนาระบบซอฟต์แวร์โดยสามารถใช้เป็นเอกสารอ้างอิงในแต่ละขั้นตอนในการพัฒนา และในขั้นตอนการตรวจรับซอฟต์แวร์ในช่วงสุดท้ายด้วย

การวิจัยมีแนวทางในการยกร่างคู่มือดังกล่าว ดังต่อไปนี้

1. สํารวจและรวบรวมมาตรฐานที่มีใช้ในต่างประเทศที่เป็นที่ยอมรับ
2. สํารวจและรวบรวมมาตรฐานหรือแนวทางการเขียนเอกสารที่มีใช้ในประเทศ
3. วิเคราะห์และตีความเพื่อหาคุณลักษณะที่เป็นจุดเด่นของมาตรฐานหรือเอกสารที่รวบรวมได้
4. แจกแจงองค์ประกอบของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ตามมาตรฐานต่างๆ เป็นส่วนย่อยๆ
5. สร้างตารางแสดงความสัมพันธ์ระหว่างองค์ประกอบของข้อกำหนด และคุณลักษณะที่พึงจะได้รับ กรณีที่มีองค์ประกอบนั้นๆ ปรากฏในเอกสาร
6. คัดเลือกองค์ประกอบที่คาดว่า เป็นประโยชน์และสามารถใช้งานได้ในทางปฏิบัติในประเทศ สำหรับผู้พัฒนาซอฟต์แวร์ หรือผู้ที่มีหน้าที่ในการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์

ในการวิจัยนี้ได้ทำการสํารวจและรวบรวมมาตรฐาน ที่กล่าวถึงการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ โดยเลือกมาจากองค์กรมาตรฐานหรือหน่วยงานราชการ ในกลุ่มประเทศยุโรปและประเทศสหรัฐอเมริกา โดยมีรายการมาตรฐานที่คัดเลือกมาดังต่อไปนี้

1. มาตรฐาน IEEE
2. มาตรฐาน ESA
3. มาตรฐาน NASA
4. มาตรฐาน DoD

จากนั้นได้เลือกวิเคราะห์และสร้างตารางความสัมพันธ์ระหว่างองค์ประกอบที่ปรากฏในเอกสาร และคุณลักษณะที่ดีจากองค์ประกอบเหล่านั้นดังแสดงในตารางที่ ข.1 ซึ่งจะเห็นได้ว่าในแต่ละบรรทัดจะเป็นส่วนย่อยที่ปรากฏเป็นส่วนประกอบในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ รวบรวมมาจากมาตรฐานที่สํารวจมาจาก IEEE [3] และ ESA และในแต่ละคอลัมน์จะเป็นคุณลักษณะที่ดีที่คาดว่าพึงจะมีสำหรับเอกสารข้อกำหนดความต้องการซอฟต์แวร์ [4] โดยเป็นประเด็นที่ได้กล่าวไว้ในมาตรฐานดังกล่าวเช่นเดียวกัน

ตารางที่ ข.1 แสดงความสัมพันธ์ระหว่าง SRS Content และ Quality Criteria for Good SRS

SRS Contents \ Quality Criteria for Good SRS	Correct	Unambiguous	Complete	Consistent	Stability	Verification	Modification	Traceable
Purpose	[1]	[1]	[1]	[1,2]	-	-	-	-
Scope	[1]	[1]	[1]	[1,2]	-	-	-	-
Definition	[1]	[1]	[1]	[1,2]	-	-	-	-
Acronyms	[1]	[1]	[1]	[1,2]	-	-	-	-
Abbreviations	[1]	[1]	[1]	[1,2]	-	-	-	-
References	[1]	[1]	[1]	[1,2]	-	-	-	[1,2]
Overview	[1]	[1]	[1]	[1,2]	-	-	-	-
System interfaces	[1]	-	[1]	[1,2]	-	[1,2]	[1]	[1,2]
User interfaces	[1]	-	[1]	[1,2]	-	[1,2]	[1]	[1,2]
Hardware interfaces	[1]	-	[1]	[1,2]	-	[1,2]	[1]	[1,2]
Software interfaces	[1]	-	[1]	[1,2]	-	[1,2]	[1]	[1,2]
Communications interfaces	[1]	-	[1]	[1,2]	-	[1,2]	[1]	[1,2]
Memory constraints	[1]	-	[1]	[1]	[1]	[1]	[1]	[1,2]
Operations	[1]	[1]	[1]	[1]	-	[1]	[1]	[1]
Site adaptation requirements	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
User characteristics	[1]	[1]	[1]	[1]	-	-	-	-
Constraints	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Assumptions and dependencies	[1]	[1]	[1]	[1]	-	[1]	[1]	[1]
Apportioning of requirements	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Functional requirements	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Performance requirements	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Interface requirements	-	-	-	[2]	[2]	[2]	-	[2]
Operational requirements	-	-	-	[2]	[2]	[2]	-	[2]
Resource requirements	-	-	-	[2]	[2]	[2]	-	[2]
Verification requirements	-	-	-	[2]	[2]	[2]	-	[2]
Acceptance testing requirements	-	-	-	[2]	[2]	[2]	-	[2]
Documentation requirements	-	-	-	[2]	[2]	[2]	-	[2]
Security requirements	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Portability requirements	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Quality requirements	-	-	-	[2]	[2]	[2]	-	[2]
Reliability requirements	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Maintainability requirements	[1]	[1]	[1]	[1,2]	[1,2]	[1,2]	[1]	[1,2]
Safety requirements	-	-	-	[2]	[2]	[2]	-	[2]
External interfaces	[1]	-	[1]	[1,2]	-	[1]	[1]	[1]
Logical database requirements	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Design constraints	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Availability	[1]	-	[1]	[1]	-	[1]	-	[1]
Organizing the specific requirements	[1]	[1]	[1]	[1]	-	[1]	[1]	[1]
Table of contents	[1]	-	[1]	[1,2]	[1,2]	-	-	[1,2]
Index	[1]	-	[1]	[1,2]	[1,2]	-	-	[1,2]
Appendixes	[1]	-	[1]	[1,2]	[1,2]	-	-	[1,2]

[1] IEEE Std 830-1993

[2] ESA PSS-05-0

คำอธิบายในตารางที่ ซ.1

Purpose

- บรรยายถึงจุดประสงค์ของการจัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- ระบุความตั้งใจในการจัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์

Scope

- ระบุชื่อของผลิตภัณฑ์ซอฟต์แวร์ เช่น ระบบจัดการฐานข้อมูลแม่ข่าย (Host DBMS) ตัวสร้างรายงาน (Report Generator) เป็นต้น
- อธิบายว่าผลิตภัณฑ์ซอฟต์แวร์ทำหน้าที่อะไร และไม่ทำอะไร
- อธิบายถึงการประยุกต์ใช้ซอฟต์แวร์ที่กำหนดไว้ รวมถึงผลประโยชน์ที่เกี่ยวข้อง วัตถุประสงค์ และ เป้าหมาย
- มีความต้องการกับเอกสารระดับสูง เช่น เอกสารความต้องการของระบบ

Definition

- จัดหาคำจำกัดความของคำศัพท์ต่างๆ ที่ปรากฏอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อให้ตีความหมายได้ง่าย

Acronyms

- จัดหารหัสพจน์ของคำศัพท์ต่างๆ ที่ปรากฏอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อให้ตีความหมายได้ง่าย

Abbreviations

- จัดหาคำย่อของคำศัพท์ต่างๆ ที่ปรากฏอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เพื่อให้ตีความหมายได้ง่าย

References

- จัดหาเอกสารที่อ้างอิงอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- ระบุเอกสารแต่ละชุด โดยใช้ชื่อ หมายเลขรายงาน วันที่ และ สถานที่พิมพ์
- ระบุแหล่งที่มาของเอกสารอ้างอิงที่อยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

Overview

- อธิบายถึงสิ่งที่บรรจุอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์
- อธิบายว่าเอกสารข้อกำหนดความต้องการซอฟต์แวร์มีการจัดระเบียบได้อย่างไร

System interfaces

- ในการต่อประสานระบบนั้น ต้องระบุหน้าที่ของซอฟต์แวร์ เพื่อให้ความต้องการของระบบสำเร็จ และอธิบายถึงระบบที่จะต่อประสานไปยังระบบอื่นๆ ได้

User interfaces

- คุณลักษณะทางตรรกะของการต่อประสานระหว่างผลิตภัณฑ์ซอฟต์แวร์ และผู้ใช้ สิ่งนี้รวมถึงคุณลักษณะโครงแบบ เช่น รูปแบบหน้าจอ แผนผังหน้า (page layouts) หรือ แผนผังหน้า

ต่าง สารบัญของรายงานหรือรายการเลือก หรือสภาพพร้อมใช้งานของเป้าหมายหน้าทึ่ในโปรแกรม เป็นต้น

Hardware interfaces

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ควรระบุถึงคุณลักษณะทางตรรกะของการต่อประสานระหว่างผลิตภัณฑ์ซอฟต์แวร์และส่วนประกอบฮาร์ดแวร์ของระบบ สิ่งนี้รวมถึงคุณลักษณะโครงสร้าง เช่น จำนวนของช่องทาง ชุดของคำสั่งเครื่อง และอื่นๆ ซึ่งบางทีอาจรวมถึงอุปกรณ์ และโพรโทคอล

Software interfaces

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรระบุถึงการใช้ผลิตภัณฑ์ซอฟต์แวร์ เช่น ระบบการจัดการข้อมูล ระบบปฏิบัติการ หรือโปรแกรมสำเร็จเชิงคณิตศาสตร์ และการต่อประสานกับระบบประยุกต์อื่นๆ เช่น การเชื่อมโยงกันระหว่างระบบบัญชีลูกหนี้ และระบบบัญชีแยกประเภททั่วไป ซึ่งในแต่ละผลิตภัณฑ์ซอฟต์แวร์ ควรมีรายละเอียดดังต่อไปนี้
 - ชื่อ
 - นิยาม
 - หมายเลขเอกสาร
 - หมายเลขรุ่น
 - แหล่งที่มา

Communications interfaces

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรระบุถึงการต่อประสานการสื่อสารที่หลากหลาย เช่น โพรโทคอลเครือข่ายเฉพาะที่

Memory constraints

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรระบุถึงคุณลักษณะ และข้อจำกัดของหน่วยความจำหลัก และหน่วยความจำรอง

Operations

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ผู้ใช้ควรระบุถึงการปฏิบัติการแบบธรรมดา และการปฏิบัติการแบบพิเศษ เช่น
 - วิธีที่หลากหลายของการปฏิบัติการในองค์กรผู้ใช้ เช่น การปฏิบัติการสำหรับผู้เริ่มต้น
 - ระยะเวลาของการปฏิบัติการเชิงโต้ตอบ และระยะเวลาของการปฏิบัติการที่ไม่มีการเอาใจใส่
 - ฟังก์ชันสนับสนุนการประมวลผลข้อมูล
 - การปฏิบัติการสำรอง และการปฏิบัติการกู้

Site adaptation requirements

- กำหนดความต้องการของข้อมูล หรือลำดับการเริ่มต้น ซึ่งระบุที่ตั้ง ภารกิจ หรือ วิธีปฏิบัติการ เช่น ข้อจำกัดของความปลอดภัย

- ระบุที่ตั้ง หรือ ลักษณะของภารกิจที่สัมพันธ์กัน ซึ่งอาจได้รับการเปลี่ยนแปลง เพื่อที่จะตัดแปลงซอฟต์แวร์ในการติดตั้งโดยเฉพาะ

User characteristics

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรอธิบายถึงคุณลักษณะโดยทั่วไปของผู้ใช้ที่ใช้ผลิตภัณฑ์ รวมถึงระดับการศึกษา ประสบการณ์ และการชำนาญทางเทคนิค

Constraints

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรจัดหารายละเอียดโดยทั่วไปของสิ่งต่างๆ ซึ่งจำกัดทางเลือกของนักพัฒนาระบบ รวมถึง
 - นโยบายการกำหนดระเบียบ
 - ข้อจำกัดทางฮาร์ดแวร์
 - การต่อประสานไปยังการประยุกต์อื่นๆ
 - การปฏิบัติการแบบขนาน
 - หน้าที่การตรวจสอบ
 - หน้าที่ควบคุม
 - ความต้องการภาษาลำดับที่สูงกว่า
 - โพรโทคอลจับมือสัญญาณ เช่น XON-XOFF , ACK-NACK
 - ความต้องการความเชื่อถือได้
 - ความวิกฤตของการประยุกต์
 - การพิจารณาความปลอดภัยและความมั่นคง

Assumptions and dependencies

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรแสดงถึงปัจจัยที่ส่งผลกระทบต่อความต้องการในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ปัจจัยเหล่านี้ไม่อยู่ในข้อบังคับในการออกแบบของซอฟต์แวร์ แต่สามารถส่งผลกระทบต่อความต้องการในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เช่น ข้อสันนิษฐาน อาจระบุถึงระบบปฏิบัติการที่พร้อมใช้งานบนฮาร์ดแวร์สำหรับผลิตภัณฑ์ซอฟต์แวร์ ถ้าในความเป็นจริง ระบบปฏิบัติการไม่พร้อมที่จะใช้งาน ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ต้องได้รับการเปลี่ยนด้วย

Apportioning of requirements

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ควรระบุถึงความต้องการ ซึ่งอาจถูกหน่วยงานกระทั่งถึงรุ่นในอนาคตของระบบ

Functional requirements

- ระบุว่าซอฟต์แวร์ทำอะไร ระบุถึงจุดประสงค์ของซอฟต์แวร์ ความต้องการเชิงหน้าที่ ได้รับการแปลงจากรูปแบบเชิงตรรกะ ซึ่งรูปแบบเชิงตรรกะแปลงมาจากความต้องการของผู้ใช้ ซึ่งความต้องการเชิงหน้าที่ รวมถึง ประสิทธิภาพด้วย

Performance requirements

- ระบุถึงค่าของตัวเลขสำหรับตัวแปรที่สามารถวัดค่าได้ เช่น อัตรา ความถี่ ความจุ และความเร็ว เป็นต้น

Interface requirements

- ระบุถึงฮาร์ดแวร์ ซอฟต์แวร์ หรือฐานข้อมูลที่ใช้ในระบบ ความต้องการของการต่อประสาน แบ่งเป็นซอฟต์แวร์ ฮาร์ดแวร์ และการต่อประสานของการติดต่อ การต่อประสานซอฟต์แวร์ รวมถึงระบบปฏิบัติการ สภาพแวดล้อมของซอฟต์แวร์ รูปแบบของเพิ่มข้อมูล ระบบการจัดการฐานข้อมูล และซอฟต์แวร์อื่นๆ การต่อประสานฮาร์ดแวร์ ระบุถึงโครงสร้างของฮาร์ดแวร์ และการต่อประสานของการติดต่อ ระบุถึงข้อบังคับในการต่อประสานไปยังฮาร์ดแวร์อื่นหรือซอฟต์แวร์อื่น

Operational requirements

- ระบุถึงวิธีการดำเนินงานของระบบและวิธีการติดต่อระหว่างระบบกับคน ความต้องการของการดำเนินการ รวมถึงการต่อประสานผู้ใช้ เช่น แผนผังหน้าจอ ข้อความผิดพลาด ระบบช่วย เป็นต้น สิ่งนี้เป็นประโยชน์ เพื่อระบุความหมาย (semantics) และวากยสัมพันธ์ (syntax) ของคำสั่ง

Resource requirements

- ระบุถึงข้อจำกัดสูงสุดในการใช้ทรัพยากรทางกายภาพ เช่น กำลังของการประมวลผล หน่วยความจำหลัก เนื้อที่ว่างของดิสก์ เป็นต้น

Verification requirements

- ระบุถึงข้อจำกัดของซอฟต์แวร์ที่ได้รับการทวนสอบ ความต้องการของการทวนสอบ เป็นข้อจำกัดในแผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์ รวมทั้ง ความต้องการในการจำลอง และการเลียนแบบ การทดสอบข้อมูลเข้าที่จำลองขึ้นมา การทดสอบข้อมูลเข้าที่เป็นข้อมูลจริง การทดสอบสภาพแวดล้อมของการต่อประสาน

Acceptance requirements

- ระบุถึงข้อจำกัดของซอฟต์แวร์ที่ได้รับการตรวจสอบความสมเหตุสมผล ความต้องการของการทดสอบการยอมรับ เป็นข้อจำกัดในแผนการทวนสอบและความสมเหตุสมผลซอฟต์แวร์

Documentation requirements

- ระบุถึงความต้องการที่เฉพาะเจาะจงของโครงการพัฒนาซอฟต์แวร์ เกี่ยวกับเรื่องเอกสารที่มีอยู่ในมาตรฐาน เช่น รูปแบบของเอกสารคู่มือผู้ใช้ซอฟต์แวร์ เป็นต้น

Security requirements

- ระบุถึงความต้องการในการรักษาความมั่นคงให้กับระบบ เช่น การเข้าถึงข้อมูลแบบอ่านอย่างเดียว การเข้ารหัสผ่านไปยังระบบ การป้องกันไวรัสคอมพิวเตอร์ เป็นต้น ซึ่งการป้องกันทำโดยมีการสำรองข้อมูลเก็บไว้

Portability requirements

- ระบุถึงการเปลี่ยนแปลงแก้ไขซอฟต์แวร์โดยง่าย เพื่อประมวลผลบนเครื่องคอมพิวเตอร์เครื่องอื่น หรือบนระบบปฏิบัติการอื่น

Quality requirements

- ระบุถึงลักษณะประจำของซอฟต์แวร์ เพื่อให้แน่ใจว่าซอฟต์แวร์นั้นมีคุณสมบัติตรงตามจุดประสงค์ที่กำหนดไว้ เช่น ความเชื่อถือได้ ความสามารถในการบำรุงรักษา ความปลอดภัย ควรได้รับการระบุในซอฟต์แวร์ ถ้าคุณภาพของซอฟต์แวร์มีความเหมาะสม จะได้รับการระบุในรูปแบบของมาตรวัด เช่น ใช้ตัววัด (metrics)

Reliability requirements

- ระบุถึงช่วงเวลาเฉลี่ย (mean time interval) ระหว่างความผิดพลาดของซอฟต์แวร์กับระยะเวลาที่มีนัยสำคัญ สิ่งนี้จะระบุเวลาที่ต่ำที่สุดของความผิดพลาดที่เกิดขึ้น ความต้องการของความเชื่อถือได้ได้รับการแปลงจากสภาพพร้อมใช้งานของผู้ใช้

Maintainability requirements

- ระบุถึงวิธีที่แก้ไขความผิดพลาด และการปรับแต่งซอฟต์แวร์เมื่อมีความต้องการใหม่เกิดขึ้น การแสดงงานต่างๆ ทำได้ง่ายในรูปแบบเชิงปริมาณ เช่น เวลาเฉลี่ยที่จะแก้ไขความผิดพลาด (mean time to repair a fault : MTTR) รวมถึงข้อจำกัดต่างๆ ในองค์กร ความต้องการของความสามารถในการบำรุงรักษาได้รับการแปลงจากสภาพพร้อมใช้งานของผู้ใช้ และความสามารถในการปรับแต่งความต้องการ

Safety requirements

- เพื่อลดความเสียหายที่เกิดขึ้นจากความผิดพลาดของซอฟต์แวร์ ความต้องการของความปลอดภัยมีการระบุถึงฟังก์ชันที่วิกฤต ซึ่งเป็นอันตรายต่อคนหรือทรัพย์สิน

External interfaces

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรอธิบายถึงรายละเอียดของข้อมูลเข้าที่ไปยังระบบซอฟต์แวร์ทั้งหมด และข้อมูลออกจากซอฟต์แวร์ทั้งหมด สิ่งเหล่านี้ควรรวมถึงสารบัญ และรูปแบบดังต่อไปนี้
 - ชื่อของสิ่งต่างๆ
 - รายละเอียดของจุดประสงค์
 - แหล่งที่มาของข้อมูลเข้า หรือปลายทางของข้อมูลออก
 - พิสัยสมเหตุสมผลและความแม่นยำ
 - หน่วยของการวัด
 - เวลา
 - ความสัมพันธ์ที่ไปยังข้อมูลเข้าหรือข้อมูลออกอื่น
 - รูปแบบหน้าจอ
 - รูปแบบหน้าต่าง
 - รูปแบบข้อมูล
 - รูปแบบคำสั่งงาน
 - ข้อความจบ

Logical database requirements

- ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ควรระบุถึงความต้องการเชิงตรรกะสำหรับรายละเอียดต่างๆ ที่ได้รับการแทนที่ไปยังฐานข้อมูล รวมถึง
 - ชนิดของรายละเอียดที่ใช้โดยหน้าที่ที่หลากหลาย
 - ความถี่ของการใช้
 - ความสามารถในการเข้าถึง
 - เดต้าเอนทิตี และความสัมพันธ์ระหว่างข้อมูล
 - ข้อบังคับของบูรณภาพ
 - ความต้องการที่จะเก็บข้อมูลไว้

Design constraints

ควรระบุถึงความต้องการที่ได้รับจากมาตรฐานหรือกำหนดระเบียบที่มีอยู่ รวมถึง

- รูปแบบรายงาน
- การตั้งชื่อข้อมูล
- กระบวนการคำนวณ
- การตรวจสอบความถูกต้อง

และควรระบุถึงข้อจำกัดในการออกแบบที่กำหนดโดยมาตรฐานอื่น

Availability

- ควรระบุถึงปัจจัยที่ต้องการ เพื่อรับประกันสภาพพร้อมใช้งานของระบบ เช่น จุดตรวจสอบ การกู้ และเริ่มทำต่อ

Organizing the specific requirements

- เป็นการพิจารณาอย่างระมัดระวังในการจัดระเบียบความต้องการให้เข้าใจได้ง่าย ซึ่งไม่มีวิธีการจัดระเบียบที่ดีที่สุดสำหรับระบบทั้งหมด แบ่งเป็น 7 วิธี คือ System mode User class Objects Feature Stimulus Response และ Functional hierarchy

Table of contents

- ตารางของสารบัญ เป็นสิ่งสำคัญในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

Index

- ดัชนี เป็นสิ่งสำคัญในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

Appendixes

- ภาคผนวก เป็นส่วนที่บรรจุรายละเอียดเกี่ยวกับสิ่งที่ไม่ได้รับการพิจารณาในส่วนของเอกสารข้อกำหนดความต้องการที่แท้จริง และเป็นสิ่งที่ไม่จำเป็น รวมถึง
 - รูปแบบของข้อมูลเข้าและข้อมูลออก คำอธิบายของการศึกษาการวิเคราะห์ค่าใช้จ่าย หรือผลลัพธ์ของการสำรวจผู้ใช้
 - การสนับสนุนข้อมูล หรือ พื้นความรู้ เป็นสิ่งที่ช่วยให้ผู้อ่านอ่านเอกสารข้อกำหนดความต้องการซอฟต์แวร์ได้เข้าใจมากยิ่งขึ้น

- คำอธิบายของปัญหาที่ได้รับการแก้ไขโดยซอฟต์แวร์
- คำสั่งโปรแกรมสำเร็จพิเศษสำหรับรหัส และสื่อต่างๆ

เอกสารความต้องการซอฟต์แวร์ (Software Requirements Specification Document)

โดยทั่วไปความต้องการของผู้ใช้จะแสดงในรูปภาษาธรรมชาติ [4] ได้มีผู้พยายามที่จะกำหนดให้อยู่ในรูปแบบโครงสร้าง (Structural Language) [5] เพื่อลดความกำกวมและทำให้กระจ่างชัดมากขึ้น นอกจากนี้ การเขียนเอกสารนี้ในรูปแบบคณิตศาสตร์ (Formal Method) [6] เช่น Z หรือ VDM ก็เป็นการวิจัยที่กำลังเริ่มมีบทบาทมากขึ้น

สำหรับงานวิจัยนี้ หลังจากได้ข้อสรุปจากแบบสอบถามแล้วจะออกแบบเอกสารโดยใช้รูปแบบโครงสร้าง

ผลที่ได้จากการศึกษา

จากการศึกษาวิจัยที่ได้ในส่วนี้ได้กำหนดแบบสอบถาม (ซึ่งกำลังดำเนินการรวบรวมข้อมูล) สำหรับการกำหนดความต้องการซอฟต์แวร์ต่อไป

สรุปและข้อเสนอแนะ

การศึกษาวิจัยนี้ ได้ข้อสรุปเบื้องต้นสำหรับแนวทางการกำหนดความต้องการซอฟต์แวร์โดยรวบรวมมาจากมาตรฐานขององค์กรต่างประเทศต่าง ๆ เช่น IEEE, ESA

แบบสอบถามที่สร้างขึ้น จะเป็นเครื่องช่วยในการกำหนดความต้องการซอฟต์แวร์สำหรับอุตสาหกรรมซอฟต์แวร์ในประเทศไทยต่อไป

ผู้วิจัยหวังในความร่วมมือของท่านในการให้ข้อมูลเพื่อมาตรฐานการพัฒนาซอฟต์แวร์ของเรา

เอกสารอ้างอิง

- [1] Sommerville, Ian, Software Engineering, 5th Ed., Addison-Wesley, Wokingham, England, 1996.
- [2] N. Luepromchai and W. Rivepiboon, Computer Aided Requirements Engineering (CARE),
Proceeding NCSEC'98, Kasetsart Univ, Bangkok, Thailand, 1998
- [3] IEEE Std 830-1993, Recommended Practice for Software Requirements Specifications, 1994.
- [4] Kotonya, Gerald and Sommerville, Ian, Requirement Engineering: Processes and Techniques,
John Wiley & Sons, Baffins Lane, England, 1998.
- [5] William M. Wilson, Writing Effective Requirements Specifications, Proceeding Software
Technology Conference, Utah, April 1997.
- [6] D. C. Ince, An Introduction to Discrete Mathematics and Formal System Specification, Oxford
University Press, England, 1988.

ประวัติผู้วิจัย

นางสาวน้ำฝน อัครเมธิน เกิดเมื่อวันที่ 27 ธันวาคม พ.ศ. 2517 ที่กรุงเทพมหานครสำเร็จการศึกษา
ระดับปริญญาตรีวิทยาศาสตร์บัณฑิต เกียรตินิยมอันดับหนึ่ง สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยา
ศาสตร์ มหาวิทยาลัยหอการค้าไทย ในปีการศึกษา 2538 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา
2539

