

บทที่ 2

ความเข้าใจเบื้องต้นเกี่ยวกับซอฟต์แวร์คอมพิวเตอร์

ในบทนี้จะได้อธิบายถึงความหมาย ประเภท สถานภาพทางกฎหมาย ลักษณะการได้มาซึ่งสิทธิในซอฟต์แวร์คอมพิวเตอร์ ตลอดจนลักษณะของความซ้ำชุดบกพร่องหรือความไม่ปลอดภัยในซอฟต์แวร์คอมพิวเตอร์ เพื่อเป็นความเข้าใจเบื้องต้นก่อนจะพิจารณาถึงปัญหาเกี่ยวกับกฎหมายว่าด้วยความรับผิดเพื่อความเสียหายอันเกิดจากสินค้าที่ไม่ปลอดภัย รายละเอียดจากการศึกษาแนวทางการแก้ไขปัญหาในต่างประเทศ และวิเคราะห์เปรียบเทียบกับกฎหมายของประเทศไทยต่อไป

2.1 ความหมายของซอฟต์แวร์คอมพิวเตอร์

คำว่า “ซอฟต์แวร์คอมพิวเตอร์” (computer software) หมายถึง ชุดคำสั่งหรือโปรแกรมที่เขียนขึ้นมาอย่างเป็นลำดับขั้นตอน เพื่อสั่งให้คอมพิวเตอร์ทำงานใดๆ ตามที่ต้องการได้ ทั้งนี้ ผู้ใช้งานคอมพิวเตอร์จะไม่สามารถใช้งานเครื่องคอมพิวเตอร์เพื่อสนับสนุนการทำงานในด้านต่างๆ ได้เลย หากไม่มีซอฟต์แวร์คอมพิวเตอร์ที่เป็นตัวกลางประสานการทำงานระหว่างผู้ใช้งานกับอุปกรณ์คอมพิวเตอร์ที่เกี่ยวข้อง และโดยที่ภายในซอฟต์แวร์คอมพิวเตอร์มักจะประกอบไปด้วยโปรแกรมย่อยหลายโปรแกรม ดังนั้น จึงมักเรียก “ซอฟต์แวร์” และ “โปรแกรม” คอมพิวเตอร์ ในความหมายเดียวกัน¹ ซอฟต์แวร์คอมพิวเตอร์หรือชุดคำสั่งดังกล่าวสามารถกำหนด และ/หรือควบคุมการทำงานของคอมพิวเตอร์ให้บังคับควบคุมกระบวนการต่างๆ รวมถึงเครื่องไม้เครื่องมือหรือเครื่องจักรกลในทางกายภาพได้ ไม่ว่าโดยตรง หรือโดยการที่นำมาใช้เป็นข้อมูล (input) สำหรับการทำงานของซอฟต์แวร์คอมพิวเตอร์อื่น²

¹ พณิศา พาณิชกุล, เทคโนโลยีสารสนเทศ, (กรุงเทพมหานคร: เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2549), น. 77.

² Gregory A. Stobbs, Software Patents, 2nd ed. (United States: ASPEN Publishers, Inc., 2006), p. 49.

ซอฟต์แวร์มักจะถูกนำไปเปรียบเทียบกับคำตรงข้ามกับส่วนต่างๆ ทางกายภาพของคอมพิวเตอร์ซึ่งเรียกว่าคอมพิวเตอร์ฮาร์ดแวร์ (computer hardware) เช่น หน่วยประมวลผลกลาง (CPU) ฮาร์ดดิสก์ (Harddisk) เมนบอร์ด (Mainboard) เป็นต้น อย่างไรก็ตาม คำว่า “ซอฟต์แวร์คอมพิวเตอร์” นั้นมักจะใช้กันในความหมายที่กว้างกว่า “โปรแกรมคอมพิวเตอร์” กล่าวคือ ในบางกรณี “ซอฟต์แวร์คอมพิวเตอร์” นอกจากจะหมายถึงตัวโปรแกรมคอมพิวเตอร์ซึ่งเป็นชุดคำสั่งและตัวแปรที่กำหนดการทำงานของคอมพิวเตอร์แล้ว ยังรวมถึงกระบวนการ (procedure) และเอกสารต่างๆ (documentation) ที่เกี่ยวข้องกับตัวโปรแกรมคอมพิวเตอร์นั้นๆ ด้วย³ เช่น แผนภาพ (flowcharts) คู่มือการใช้งาน (instruction manual) คำแนะนำเกี่ยวกับความต้องการของระบบ (system requirements) เป็นต้น อย่างไรก็ตาม ในวิทยานิพนธ์ฉบับนี้ ให้คำว่า “ซอฟต์แวร์คอมพิวเตอร์” หมายความว่า ชุดคำสั่งหรือโปรแกรมที่เขียนขึ้นมาอย่างเป็นลำดับขั้นตอน เพื่อความมุ่งหมายในการสั่งให้คอมพิวเตอร์ทำงานอย่างใดอย่างหนึ่ง โดยให้หมายความรวมถึงโปรแกรมคอมพิวเตอร์ด้วย แต่ไม่ให้ความหมายรวมถึงข้อมูลหรือข้อความซึ่งมุ่งสื่อสารกับมนุษย์ เช่น คู่มือการติดตั้ง คู่มือการใช้ คำเตือน ข้อกำหนดและเงื่อนไขในการใช้งาน เป็นต้น นอกจากนี้ ยังไม่ให้ความหมายรวมถึงข้อมูลเกี่ยวกับงานภาพยนตร์ เพลง ภาพเคลื่อนไหว หรืองานอื่นๆ ในทำนองเดียวกัน

ก่อนถึงกลางยุคทศวรรษ 1960 ซอฟต์แวร์คอมพิวเตอร์ยังไม่เป็นสินค้าที่แยกจำหน่ายต่างหากจากเครื่องคอมพิวเตอร์ โดยในขณะนั้นซอฟต์แวร์ถือเป็นส่วนหนึ่งของเครื่องคอมพิวเตอร์ และไม่สามารถแยกซื้อต่างหากได้ ผู้ผลิตคอมพิวเตอร์รายใหญ่ในสมัยนั้น เช่น IBM, Univac, Honeywell, RCA, Burroughs และ NCR ต่างนำซอฟต์แวร์รวมเข้าเป็นส่วนหนึ่งของการจำหน่ายเครื่องคอมพิวเตอร์ทั้งสิ้น ผู้ใช้คอมพิวเตอร์ทั่วไปในสมัยนั้นยังไม่มีความรู้ความสามารถเพียงพอในการเขียนโปรแกรมไว้ใช้งานเองและจำเป็นต้องใช้ซอฟต์แวร์และการบริการ (support) จากผู้จำหน่ายเครื่องคอมพิวเตอร์เท่านั้น ส่งผลให้ค่าใช้จ่ายและค่าบริการในการเขียนโปรแกรมในช่วงต้นถึงกลางทศวรรษที่ 1960 มีราคาแพงมาก จนกระทั่งประมาณปี 1955 – 1962 เจ้าของเครื่องคอมพิวเตอร์ส่วนบุคคล (personal computer) จึงเริ่มเรียนรู้ที่จะเขียนโปรแกรมคอมพิวเตอร์เองและมีการรวมกลุ่มผู้ใช้งานเพื่อแลกเปลี่ยนความรู้และช่วยเหลือซึ่งกันและกัน โดยกลุ่มผู้ใช้งาน

³ Retrieved from http://en.wikipedia.org/wiki/Computer_software, January 20, 2010.

กลุ่มหนึ่งซึ่งเรียกตัวเองว่า “SHARE” ได้เริ่มทำการจำหน่ายซอฟต์แวร์คอมพิวเตอร์ที่มีประโยชน์ต่อการใช้งานขึ้นอย่างเป็นระบบโดยไม่คิดค่าตอบแทน ต่อมาหลังจากกลางทศวรรษที่ 1960 จึงเริ่มมีผู้ผลิตและพัฒนาซอฟต์แวร์ (software house) ขึ้นเพื่อขายหรือจำหน่ายในทางธุรกิจแข่งขันกับผู้ผลิตเครื่องคอมพิวเตอร์⁴

ในช่วงสี่สิบปีที่ผ่านมา นับเป็นช่วงที่คอมพิวเตอร์ได้มีการเปลี่ยนแปลงทางด้านเทคโนโลยีไปมาก เทคโนโลยีได้มีการพัฒนาให้ดีขึ้น มีขนาดเล็กลง มีความสามารถสูงขึ้น น้ำหนักเบา พกพาติดตัวไปได้ ใช้พลังงานน้อย และมีความเร็วในการทำงานมากขึ้น ทำให้ราคาตัวเครื่องและอุปกรณ์ด้านฮาร์ดแวร์ถูกลงเป็นอย่างมาก ในขณะที่ตัวเครื่องและอุปกรณ์มีคุณภาพและประสิทธิภาพ ในทางกลับกัน เมื่อพิจารณาในส่วนของซอฟต์แวร์คอมพิวเตอร์แล้ว ค่าใช้จ่ายในการพัฒนาลับสูงขึ้นไป และยังคงเผชิญกับปัญหาในด้านการบำรุงรักษาซอฟต์แวร์ (maintenance) มากขึ้นด้วย โดยในเรื่องค่าใช้จ่ายดังกล่าว ได้มีผู้ทำการศึกษาเปรียบเทียบค่าใช้จ่ายของศูนย์คอมพิวเตอร์ในประเทศสหรัฐอเมริการะหว่างฮาร์ดแวร์และซอฟต์แวร์ในช่วงปี พ.ศ. 2503 – 2543 พบว่า ในปี พ.ศ. 2503 ค่าใช้จ่ายด้านฮาร์ดแวร์สูงมาก ด้านซอฟต์แวร์ต่ำ แต่ในปีหลังๆ ค่าใช้จ่ายด้านซอฟต์แวร์จะมากกว่าด้านฮาร์ดแวร์หลายเท่า โดยค่าใช้จ่ายในส่วนที่เพิ่มขึ้นของซอฟต์แวร์ไม่ใช่ค่าใช้จ่ายในการพัฒนาหรือการผลิตซอฟต์แวร์ หากแต่เป็นค่าใช้จ่ายในระยะซ่อมบำรุงซอฟต์แวร์ (software maintenance) คือ ในระยะดัดแปลงแก้ไข และปรับปรุงโปรแกรมให้ทันต่อการเปลี่ยนแปลงทางธุรกิจและการใช้งานในตลาด ค่าใช้จ่ายในข้อนี้ยิ่งนานวันยิ่งเพิ่มมากขึ้น⁵ ซึ่งข้อมูลดังกล่าวแสดงให้เห็นถึงนัยสองประการ คือ ประการแรก การก้าวเข้ามามีบทบาทสำคัญเพิ่มขึ้นของอุตสาหกรรมซอฟต์แวร์คอมพิวเตอร์ อีกประการหนึ่ง คือ การสะท้อนให้เห็นถึงคุณภาพในเรื่องวิธีการพัฒนาซอฟต์แวร์ที่ขาดการวิเคราะห์ ออกแบบ และการตรวจสอบที่ดี ทำให้ต้องเสียค่าใช้จ่ายในระยะยาวในสัดส่วนที่สูงขึ้น

สำหรับในประเทศไทย คอมพิวเตอร์และซอฟต์แวร์ถูกติดตั้งครั้งแรกในประเทศไทยที่คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2506 และไม่กี่ปีหลังจากนั้นเครื่องคอมพิวเตอร์และซอฟต์แวร์คอมพิวเตอร์ก็ได้ถูกนำมาใช้ในทางธุรกิจ อย่างไรก็ตาม ในขณะที่

⁴ Gregory A. Stobbs, *supra* note 2, pp. 38-39.

⁵ สานนท์ เจริญฉาย, การเขียนโปรแกรมและอัลกอริทึม (กรณีตัวอย่างภาษาซี)

(นนทบุรี: นิติธรรมการพิมพ์, 2552), น. 2-3.

นั้นไม่ปรากฏว่ารัฐบาลไทยมีนโยบายที่จะส่งเสริมผลักดันให้มีการพัฒนาเทคโนโลยีด้านซอฟต์แวร์คอมพิวเตอร์ขึ้นในประเทศอย่างเป็นรูปธรรม ทั้งๆ ที่อุตสาหกรรมซอฟต์แวร์คอมพิวเตอร์เป็นอุตสาหกรรมที่ไม่ก่อให้เกิดมลภาวะทางกายภาพ ใช้เครื่องจักรในการผลิตน้อยมาก และมีมูลค่าทางการตลาดเติบโตขึ้นอย่างรวดเร็ว

จากการสำรวจตลาดเทคโนโลยีสารสนเทศและการสื่อสารของประเทศไทยในปี พ.ศ. 2548 โดยสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (SIPA) และสมาคมอุตสาหกรรมเทคโนโลยีสารสนเทศไทย (ATCI) ในปี 2548 พบว่าตลาดซอฟต์แวร์คอมพิวเตอร์ในประเทศไทย มีมูลค่าถึง 41,435 ล้านบาท โดยมีอัตราการเจริญเติบโตร้อยละ 27 เมื่อเทียบกับปี 2547 ซึ่งถือว่ามีอัตราการเจริญเติบโตสูงที่สุดในตลาดเทคโนโลยีสารสนเทศและการสื่อสาร⁶ อย่างไรก็ตาม อุตสาหกรรมซอฟต์แวร์คอมพิวเตอร์ของไทยยังคงประสบปัญหาอยู่หลายประการ ซึ่งรวมถึงการขาดความเชื่อมั่นในเรื่องมาตรฐานและคุณภาพ และการละเมิดลิขสิทธิ์ในซอฟต์แวร์คอมพิวเตอร์⁷ ซึ่งผู้เขียนเชื่อว่า มาตรการทางกฎหมาย ทั้งความชัดเจนด้านกฎหมายสารบัญญัติ และการบังคับใช้กฎหมายน่าจะมีส่วนช่วยผลักดันสนับสนุนอุตสาหกรรมซอฟต์แวร์คอมพิวเตอร์ในประเทศไทยได้อีกทางหนึ่ง

2.2 ประเภทของซอฟต์แวร์คอมพิวเตอร์

ซอฟต์แวร์คอมพิวเตอร์ในปัจจุบันมีจำนวนมากมายับไม่ถ้วน เพื่อรองรับความต้องการในการใช้งานที่แตกต่างกัน ซึ่งเราอาจแบ่งประเภทของซอฟต์แวร์ได้ดังนี้

⁶ Retrieved from

[http://www.atsi.or.th/front/file/Thailand IT Market Outlook 2006.pdf](http://www.atsi.or.th/front/file/Thailand_IT_Market_Outlook_2006.pdf), December 10, 2008.

⁷ Aunya Singsangob, Computer Software and Information Licensing in Emerging Markets: the Need for a Viable Legal Framework, (Hague: Kluwer Law International, 2003), p. 33.

2.2.1 ซอฟต์แวร์มาตรฐานหรือซอฟต์แวร์สำเร็จรูป (Standard or Package Software) กับซอฟต์แวร์ที่พัฒนาขึ้นโดยเฉพาะเจาะจง (Bespoke or Customized Software)

2.2.1.1 ซอฟต์แวร์มาตรฐานหรือซอฟต์แวร์สำเร็จรูป (Standard or Package Software)

เป็นซอฟต์แวร์ที่นิยมใช้กันทั่วไปในธุรกิจ เช่น Microsoft Word, MS Excel, MySQL, Microsoft PowerPoint, Adobe Photoshop เป็นต้น ซึ่งเป็นโปรแกรมที่ออกแบบสำหรับวางขายทั่วไปในท้องตลาด ตอบสนองความต้องการของผู้ใช้งาน (users) เป็นจำนวนมาก โดยทั่วไปแล้วจะนำออกจำหน่ายผ่านทางเครือข่ายการจัดจำหน่าย ส่วนมากจะไม่มีการเจรจาและทำสัญญากันเป็นลายลักษณ์อักษรระหว่างคู่กรณีที่เกี่ยวข้อง แต่ผู้ผลิตมักจะเป็นผู้วางข้อกำหนดและเงื่อนไขเป็นสัญญามาตรฐานไว้ในตัวซอฟต์แวร์หรือบรรจุภัณฑ์⁸ ข้อดีของซอฟต์แวร์สำเร็จรูป คือ ผู้ใช้สามารถนำซอฟต์แวร์ดังกล่าวไปใช้งานได้โดยทำการติดตั้งกับอุปกรณ์คอมพิวเตอร์ โดยไม่ต้องเสียเวลาพัฒนาหรือปรับแต่งซอฟต์แวร์เองหรือจ้างนักเขียนโปรแกรมมาพัฒนาเพื่อการใช้งานสำหรับผู้ใช้เป็นกรณีเฉพาะ แต่มีข้อเสีย คือ ผู้ใช้ไม่สามารถแก้ไขตัวซอฟต์แวร์เองได้⁹ ซึ่งทำให้ตัวซอฟต์แวร์คอมพิวเตอร์อาจไม่เหมาะสมกับความมุ่งหมายของผู้ใช้งานเป็นการเฉพาะเจาะจง

2.2.1.2 ซอฟต์แวร์เพื่อการใช้งานโดยเฉพาะเจาะจง (Bespoke or Customized Software)

Bespoke Software ซึ่งเป็นซอฟต์แวร์ที่เขียนขึ้นใหม่เพื่อการใช้งานของบุคคลโดยเฉพาะเจาะจง เพื่อให้ผู้ใช้งานได้รับซอฟต์แวร์ที่เหมาะสมกับการใช้งานของตนมากที่สุด ซอฟต์แวร์จำพวกนี้มักจะมีราคาที่สูงมาก ส่วน Customized Software เป็นซอฟต์แวร์ที่ผู้จำหน่ายนำซอฟต์แวร์มาตรฐานของตนหรือของบุคคลที่สามมาดัดแปลงให้เหมาะสมกับการใช้งานของลูกค้าแต่ละราย

⁸ Ian J. Lloyd, Information Technology Law, 4th ed. (Oxford: Oxford University Press, 2004), p. 572.

⁹ พณิศา พาณิซกุล, อ้างแล้ว เชิงอรรถที่ 1, น. 91.

โดยทั่วไปแล้ว Bespoke Software หรือ Customized Software มักจะมีราคาสูงกว่า Standard or Package Software และมักจะมีการเจรจาเข้าทำสัญญากันเป็นลายลักษณ์อักษร ระหว่างคู่สัญญา¹⁰

ซอฟต์แวร์คอมพิวเตอร์ประเภทที่พัฒนาขึ้นโดยเฉพาะเจาะจงนี้ จะต้องมีการรวบรวมความต้องการของผู้ใช้ ก่อนที่ผู้เขียนโปรแกรม (programmer) จะเริ่มทำการเขียนโปรแกรมขึ้นมา เป็นซอฟต์แวร์ที่สามารถทำงานได้ตามความต้องการนั้น ซึ่งส่วนใหญ่เพื่อเป็นการตอบสนองความต้องการของธุรกิจที่ไม่สามารถหาซอฟต์แวร์สำเร็จรูปมาใช้งานได้อย่างตรงตามวัตถุประสงค์ของธุรกิจนั้นๆ เช่น ระบบงานบัญชี ระบบงานคลังสินค้า ระบบงานขาย ระบบงานห้องสมุด ระบบงานทะเบียนประวัติ ระบบบริหารบุคคล ระบบการเรียนการสอนทางไกลผ่านเว็บไซต์ ระบบพาณิชย์อิเล็กทรอนิกส์ (e-commerce) เป็นต้น จึงจำเป็นต้องว่าจ้างนักพัฒนาระบบหรือบริษัทพัฒนาระบบ ให้วิเคราะห์ความต้องการ (Analysis) ออกแบบระบบ (Design) เขียนโปรแกรม (Coding) และติดตั้งเพื่อใช้งาน (Implementation / Installation) ซอฟต์แวร์ประเภทนี้มีข้อดี คือ สามารถใช้งานได้ตรงตามความต้องการของผู้ใช้อย่างแท้จริง ส่วนข้อเสีย คือ ใช้ต้นทุนในการพัฒนาค่อนข้างสูง และต้องใช้เวลานาน¹¹

2.2.2 ซอฟต์แวร์ระบบ (System Software) กับ ซอฟต์แวร์ประยุกต์ (Application Software)

2.2.2.1 ซอฟต์แวร์ระบบ (System Software)

เป็นชุดคำสั่งหรือโปรแกรมที่เขียนขึ้นมาเพื่อใช้จัดการกับระบบ ทำหน้าที่ควบคุมการทำงานของคอมพิวเตอร์และอุปกรณ์ต่างๆ¹² ซอฟต์แวร์ระบบที่รู้จักกันโดยทั่วไป คือ ระบบปฏิบัติการ โปรแกรมอรรถประโยชน์ และตัวแปลภาษา

ระบบปฏิบัติการ (Operating System: OS) เป็นโปรแกรมที่ทำให้เครื่องคอมพิวเตอร์สามารถควบคุมและจัดการทรัพยากรต่างๆ ของเครื่องได้ ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ อุปกรณ์ต่อพ่วง เป็นรากฐานการทำงานของซอฟต์แวร์คอมพิวเตอร์อื่นๆหน้าที่หลักของซอฟต์แวร์ระบบปฏิบัติการมีสามประการ คือ ประการแรก เป็นตัวกลางระหว่างผู้ใช้งานซอฟต์แวร์ประยุกต์

¹⁰ *supra* note 8.

¹¹ พนิดา พาณิชกุล, อ้างแล้ว เจริญธรรมที่ 9, น. 91-92.

¹² เพ็งอ้าง, น. 78.

กับคอมพิวเตอร์ฮาร์ดแวร์ (user interface) ประการที่สอง ควบคุมการทำงานของระบบฮาร์ดแวร์ทั้งหมด (control system) ประการที่สาม บริหารทรัพยากรในระบบให้ได้ประสิทธิภาพสูงสุด (resource management) ตัวอย่างของระบบปฏิบัติการ คือ Windows XP, Window Vista, Mac OS, NetWare, Windows NT Server, UNIX, LINUX เป็นต้น

โปรแกรมอรรถประโยชน์ (Utility Program) เป็นโปรแกรมหรือซอฟต์แวร์คอมพิวเตอร์ที่ช่วยสนับสนุน เพิ่ม หรือขยายขีดความสามารถของโปรแกรมที่ใช้งานอยู่แล้วให้มีประสิทธิภาพมากขึ้น ซึ่งระบบปฏิบัติการ (OS) ส่วนใหญ่จะมีโปรแกรมอรรถประโยชน์มาให้ใช้งานอยู่แล้ว เช่น Windows Explorer, Window Media Player, Scandisk อย่างไรก็ตาม ผู้ใช้สามารถหาโปรแกรมอรรถประโยชน์มาใช้เพิ่มเติมได้จากท้องตลาด หรือดาวน์โหลดได้จากอินเทอร์เน็ต เช่น โปรแกรมสแกนไวรัส โปรแกรมบีบอัดไฟล์ข้อมูล เครื่องมือจัดการรูปแบบอักษร (Font Management) เป็นต้น

ตัวแปลภาษา (Language Translator) เป็นซอฟต์แวร์คอมพิวเตอร์ที่เขียนขึ้นมาเพื่อใช้แปลความหมายของคำสั่งในภาษาระดับสูงชนิดต่างๆ ให้เป็นภาษาเครื่องซึ่งเป็นภาษาเดียวที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ ภาษาระดับสูงมักจะถูกนำไปเขียนเพื่อสร้างเป็นซอฟต์แวร์คอมพิวเตอร์ต่างๆ มากมาย ซึ่งในระหว่างที่ทำการเขียน ผู้เขียนโปรแกรม (programmer) จำเป็นต้องใช้ตัวแปลภาษาเพื่อทดสอบโค้ดที่ตัวเองเขียนขึ้นว่าถูกต้องหรือไม่ ตัวแปลภาษานี้แบ่งออกได้เป็นสองชนิด คือ อินเตอร์พรีเตอร์ (Interpreter) และคอมไพเลอร์ (Compiler) โดยที่อินเตอร์พรีเตอร์ (Interpreter) ใช้สำหรับแปลคำสั่งในโปรแกรมที่เขียนด้วยภาษาระดับสูง (หรือที่เรียกว่า Source Code) ครั้งละหนึ่งคำสั่ง ให้เป็นภาษาเครื่อง (Object Code) แล้วนำไปประมวลผลและแสดงผลทันที หากไม่พบข้อผิดพลาด จะรับคำสั่งถัดไปในโปรแกรมเพื่อแปลประมวลผล และแสดงผลทันที ทำเช่นนี้เรื่อยไปจนกว่าจะจบโปรแกรม แต่หากพบข้อผิดพลาด อินเตอร์พรีเตอร์จะหยุดแปลคำสั่ง แล้วแจ้งข้อผิดพลาดด้วย Error Message ให้ผู้เขียนโปรแกรมทราบ ตัวอย่างภาษาที่ใช้อินเตอร์พรีเตอร์เป็นตัวแปลภาษา เช่น BASIC, dBase เป็นต้น ส่วนคอมไพเลอร์ (Compiler) จำทำงานโดยตรวจสอบคำสั่งทั้งหมดในโปรแกรมก่อน แล้วจึงแปลเป็นภาษาเครื่องในครั้งเดียว จากนั้นจะทำการประมวลผลและแสดงผลหรือออกทางจอภาพ หากพบข้อผิดพลาดในการเขียนคำสั่ง คอมไพเลอร์จะยังคงแปลคำสั่งอื่นต่อไปจนจบ แล้วรายงาน Error Message ให้โปรแกรมเมอร์ทราบในคราวเดียว โปรแกรมที่ใช้คอมไพเลอร์เป็นตัวแปลภาษา จะทำให้เครื่องทำงานเร็วกว่าโปรแกรมที่ใช้อินเตอร์พรีเตอร์ เนื่องจากคอมไพเลอร์จะให้เครื่องจัดเก็บ

Object Code ใ้ทุกครั้งที่มีการคอมไพล์ และแก้ไข ดังนั้น เมื่อแก้ไขโปรแกรมจนสมบูรณ์แล้ว จะทำให้ได้ Object ที่สมบูรณ์ด้วย ไม่ต้องแปลใหม่เมื่อต้องการใช้งาน¹³

2.2.2.2 ซอฟต์แวร์ประยุกต์ (Application Software)

เป็นซอฟต์แวร์ที่ออกแบบ พัฒนา หรือผลิตขึ้นเพื่อให้พร้อมและง่ายต่อการใช้งานของผู้ใช้ (users) คอมพิวเตอร์ สำหรับการใช้ประโยชน์ในการทำงานใดๆ โดยเฉพาะ ซึ่งอาจจะเป็นซอฟต์แวร์สำเร็จรูป (Package Software) ที่มีขายทั่วไปตามท้องตลาด หรืออาจเป็นซอฟต์แวร์พัฒนาขึ้นเพื่อผู้ใช้งานโดยเฉพาะเจาะจง (Bespoke Software / Customerized Software) ก็ได้ เช่น Microsoft Word, Microsoft Excel, SAP เป็นต้น

2.2.3 แชร์แวร์ (Shareware) ฟรีแวร์ (Freeware) ซอฟต์แวร์เสรี (Free Software) และโอเพนซอร์สซอฟต์แวร์ (Open Source Software)

2.2.3.1 แชร์แวร์ (Shareware)

เป็นซอฟต์แวร์ที่ในเบื้องต้นสามารถใช้งานได้โดยไม่เสียค่าใช้จ่ายภายใต้เงื่อนไขต่างๆ เช่น ให้ทดลองใช้ก่อนภายในระยะเวลาจำกัด หรือภายใต้ฟังก์ชันการใช้งานที่จำกัด โดยหากผู้ใช้งานถูกใจและประสงค์จะใช้งานต่อหลังจากสิ้นระยะเวลาทดลองใช้ หรือต้องการใช้ฟังก์ชันอื่นให้ได้ครบถ้วนก็จะต้องจ่ายเงินค่าตอบแทนการใช้ซอฟต์แวร์

2.2.3.2 ฟรีแวร์ (Freeware)

เป็นซอฟต์แวร์ที่ผู้สร้างหรือผู้พัฒนามีความประสงค์จะให้ใช้งานซอฟต์แวร์ที่ตนเองได้พัฒนาขึ้นโดยไม่คิดมูลค่า และทำการเผยแพร่ให้แก่ผู้ใช้ผ่านทางอินเทอร์เน็ต ผู้ใช้จะต้องดาวน์โหลดมาจัดเก็บไว้ในเครื่องคอมพิวเตอร์เพื่อทำการติดตั้งและใช้งานต่อไป แต่ทั้งนี้ จะไม่มีการเผยแพร่ Source Code ที่ใช้ในการพัฒนา

2.2.3.3 ซอฟต์แวร์เสรี (Free Software)

ในขณะที่ฟรีแวร์มีความหมายที่มุ่งเน้นในเรื่องของการให้ใช้งานโดยไม่คิดมูลค่า แต่ซอฟต์แวร์เสรีได้ถูกกำหนดคำนิยามให้มุ่งเน้นในเรื่องของเสรีภาพ ไม่ใช่ในเรื่องของราคา โดยฟรีซอฟต์แวร์หมายถึงซอฟต์แวร์ที่ผู้ใช้งานมีเสรีภาพในการใช้งาน คัดลอก จำหน่าย ศึกษาปรับเปลี่ยน ตัดแปลง และพัฒนาซอฟต์แวร์นั้นได้ โดยไม่ต้องได้รับอนุญาตใดๆ จากผู้พัฒนาอีก

¹³ เฟิงอ้าง, น. 88.

การได้มาซึ่งฟรีซอฟต์แวร์นั้น ซึ่งอาจได้มาโดยไม่เสียค่าตอบแทนหรือเสียค่าตอบแทนก็ได้ แต่สิ่งสำคัญ คือ ผู้ใช้งานจะต้องมีเสรีภาพในการคัดลอก จำหน่าย ศึกษา ปรับเปลี่ยน ตัดแปลง และพัฒนาซอฟต์แวร์นั้นต่อไป¹⁴ ปรัชญาและคำจำกัดความของซอฟต์แวร์เสรีนั้น มีที่มาจากมูลนิธิซอฟต์แวร์เสรี (Free Software Foundation) ซึ่งเป็นองค์กรที่ไม่แสวงหากำไร มีวัตถุประสงค์ในการธำรงรักษาเสรีภาพของผู้ใช้งานคอมพิวเตอร์ โดยการส่งเสริมการใช้และการพัฒนาซอฟต์แวร์คอมพิวเตอร์อย่างเสรี โดยได้ทำการพัฒนาซอฟต์แวร์ระบบปฏิบัติการ GNU (GNU Operating System)¹⁵

2.2.3.5 โอเพนซอร์สซอฟต์แวร์ (Open Source Software)

เป็นซอฟต์แวร์ที่บุคคลทั่วไปสามารถนำไปใช้งาน ศึกษา พัฒนาต่อ และเผยแพร่ได้อย่างเสรี ภายใต้เงื่อนไขที่กำหนด เป็นซอฟต์แวร์ที่เปิดเผย Source Code ให้ผู้อื่นนำไปพัฒนาต่อไป ทำให้เกิดความร่วมมือกันทำงานอย่างไร้พรมแดนผ่านเครือข่ายทางอินเทอร์เน็ต กลุ่มที่พัฒนา Open Source ในปัจจุบันเรียกตัวเองว่า Open Source Initiative ซึ่งเป็นองค์กรที่ไม่แสวงหากำไร ที่มีวัตถุประสงค์ในการให้ความรู้และปกป้องซอฟต์แวร์ประเภทที่ไม่สงวนสิทธิ์ในการเป็นเจ้าของเพื่อประโยชน์ของบุคคลใดบุคคลหนึ่ง (non-proprietary software)¹⁶ ซึ่งโดยทั่วไปแล้วหลักการของซอฟต์แวร์เสรีกับ Open Source นั้นมีส่วนคล้ายกันมาก แต่มีที่มาจากต่างองค์กรกัน

2.3 ภาษาคอมพิวเตอร์ (Computer Language)

ภาษาคอมพิวเตอร์ (Computer Language) เป็นภาษาที่ใช้เขียนชุดคำสั่งหรือเขียนโปรแกรมเพื่อสั่งให้คอมพิวเตอร์ทำตามคำสั่งงานที่ต้องการ โดยเป็นเครื่องสื่อสารระหว่างคนกับเครื่องคอมพิวเตอร์ คอมพิวเตอร์ในปัจจุบัน ระบบวงจรภายในเครื่องเป็นแบบดิจิทัล ซึ่งข้อมูลและคำสั่งที่เครื่องจะรับรู้และสามารถนำไปประมวลผลได้ จะต้องเป็นระบบเลขฐานสอง (Binary Code) ทั้งสิ้น นั่นคือประกอบไปด้วยสัญลักษณ์ 0 และ 1 ดังนั้น สัญลักษณ์ที่คอมพิวเตอร์เข้าใจและมนุษย์ใช้สื่อความหมายกับคอมพิวเตอร์ได้ จะต้องเป็นระบบเลขฐานสอง เราเรียกภาษา

¹⁴ Retrieved from <http://www.gnu.org/philosophy/free-sw.html>, May 31, 2010.

¹⁵ Peter Brown, "Free software is a matter of liberty, not price", Retrieved from <http://www.fsf.org/about/>, May 31, 2010.

¹⁶ Retrieved from <http://www.opensource.org/about>, May 31, 2010.

ดังกล่าวนี้ว่าภาษาเครื่อง (Machine Language / Object Code) ซึ่งเป็นภาษาเดียวเท่านั้นที่คอมพิวเตอร์เข้าใจได้โดยตรง ไม่ต้องมีการแปลภาษา แต่ภาษาเครื่องนี้เองกลับเป็นภาษาซึ่งยากสำหรับมนุษย์ที่จะนำมาใช้เขียนโปรแกรม จึงได้มีผู้คิดค้นภาษาที่มนุษย์จะใช้ติดต่อกับเครื่องได้ง่ายขึ้น จึงเกิดภาษาใหม่ขึ้นมากมาย¹⁷

ภาษาคอมพิวเตอร์อาจแบ่งประเภทตามระดับของภาษาได้ดังนี้¹⁸

2.3.1 ภาษาระดับต่ำ (Low-level Language/First Generation Language/1GL)

เป็นภาษาเครื่อง (Machine Language / Object Code) ใช้สัญลักษณ์ 0 กับ 1 ดังที่กล่าวมาข้างต้น เป็นภาษาที่คอมพิวเตอร์สามารถอ่านเข้าใจได้ทันทีโดยไม่ต้องผ่านตัวแปลภาษา ตัวอย่างภาษาเครื่อง คือ ASCII (American Standard Code for Information Interchange)

2.3.2 ภาษาสัญลักษณ์ (Symbol Language/Second Generation Language/2GL)

เป็นภาษาที่เริ่มพัฒนาสัญลักษณ์ใหม่เข้ามาแทน 0 และ 1 ในคำสั่งที่ต้องใช้บ่อยๆ เพื่อลดความยากและประหยัดเวลามากขึ้น ภาษาในประเภทนี้คือภาษาแอสเซมบลี (Assembly) เมื่อมีการใช้สัญลักษณ์ก็ต้องมีการแปลภาษาจากสัญลักษณ์ให้เป็นภาษาเครื่อง ตัวแปลภาษานี้มีชื่อว่า แอสเซมเบลอร์ (Assembler)

สัญลักษณ์ที่ใช้แทนตัวเลขดังกล่าวในภาษาสัญลักษณ์นี้ก็คือ ตัวอักษรภาษาอังกฤษนั่นเอง ซึ่งอาจจะเป็น 1 ตัวอักษร หรือเป็นกลุ่มตัวอักษรก็ได้ เพื่อใช้แทนคำสั่ง 1 คำสั่ง ตัวอย่างเช่น ใช้อักษร A แทนคำสั่ง add หรือตัวอักษร MVC แทนคำสั่ง Move เป็นต้น¹⁹

¹⁷ สานนท์ เจริญฉาย, อ้างแล้ว เชิงอรรถที่ 5, น. 13-14.

¹⁸ เพ็งอ้าง.

¹⁹ พนิดา พาณิชกุล, อ้างแล้ว เชิงอรรถที่ 9, น. 92.

2.3.3 ภาษาระดับสูง (High-level Language/Third Generation Language/3GL)

เป็นภาษาที่ใช้สัญลักษณ์เต็มรูปแบบ โดยมีวัตถุประสงค์เพื่อให้ง่ายต่อมนุษย์ที่จะนำไปเขียนโปรแกรม วิธีการสร้างภาษาระดับสูงก็คือ ผู้ประดิษฐ์ภาษาจะนำตัวเลข ตัวอักษร และเครื่องหมายต่างๆ ของภาษาลาติน มาสร้างเป็นภาษาคอมพิวเตอร์ โดยจะสร้างองค์ประกอบของภาษาอย่างครบถ้วน รวมทั้งองค์ประกอบที่ช่วยในการเขียนโปรแกรมในภาษานั้นๆ ให้สะดวกและง่าย กล่าวคือ นอกจากจะสามารถใช้ตัวอักษรแทนคำสั่ง 1 คำสั่งแล้ว ยังสามารถเขียนคำสั่งได้เป็นประโยคภาษาอังกฤษและเป็นกลุ่มคำที่ยาวและมีความหมายมากขึ้น ซึ่งใกล้เคียงกับการใช้ภาษาของมนุษย์ เช่น คำสั่ง Read, Write, Add a to b และคำสั่ง let $x = 0$ เป็นต้น ตัวอย่างภาษาประเภทนี้ คือ BASIC, PASCAL, FORTRAN, COBOL, C, PL/1 และ Ada เป็นต้น

เช่นเดียวกันกับภาษาสัญลักษณ์ ภาษาระดับสูงจะต้องมีตัวแปลภาษา เพื่อแปลคำสั่งหรือโปรแกรมที่ไม่ใช่ภาษาคอมพิวเตอร์ (Source Code) ให้เป็นภาษาเครื่อง (Machine Language/Object Code) จึงจะทำให้เครื่องสามารถเข้าใจได้

2.3.4 ภาษาระดับสูงมาก (Very High-level Language/Fourth Generation Language/4GL)

เป็นภาษาที่มีคำสั่งที่สามารถเขียนเป็นภาษาอังกฤษได้มากขึ้น และการเขียนคำสั่งก็ไม่จำเป็นจะต้องเขียนหลายบรรทัดเพื่อให้ได้งานอย่างหนึ่ง แต่สามารถระบุคำสั่ง 1 คำสั่ง เพื่อให้ได้งาน 1 งาน ผู้เขียนเพียงกำหนดความต้องการว่าจะให้โปรแกรมทำอะไรบ้างโดยไม่ต้องรู้ว่าทำอย่างไร การเขียนโปรแกรมจะง่ายและรวดเร็ว เช่น ระบุคำสั่ง “Average” ก็จะได้ผลลัพธ์ของ “Average” โดยที่ไม่ต้องระบุว่าคำนวณค่าเฉลี่ยนั้นอย่างไร เป็นต้น ภาษาประเภทนี้สามารถนำมาใช้เขียนคำสั่งเพื่อเชื่อมต่อกับฐานข้อมูลได้ด้วย ทำให้ผู้เขียนโปรแกรมสามารถจัดการกับข้อมูลในฐานข้อมูลได้ง่ายขึ้น นอกจากนี้ ยังมีการพัฒนาเครื่องมือช่วยในการเขียนโปรแกรมในรูปแบบลักษณะใหม่ที่ยั่งยืน เรียกว่า “Visual Programming ยกตัวอย่างกลุ่มภาษาประเภทดังกล่าว เช่น Visual Basic, Visual C, Visual FoxPro เป็นต้น

อย่างไรก็ตาม เนื่องด้วยการพัฒนาซอฟต์แวร์ประยุกต์ใช้ในงานด้านต่างๆ หากเป็นระบบงานขนาดใหญ่ ข้อมูลและสารสนเทศมีจำนวนมาก การเขียนโปรแกรมในลักษณะเดิมจะยังมีความซับซ้อนมากขึ้น จำนวนบรรทัดของคำสั่งก็จะมีมากตามไปด้วย ส่งผลให้การบำรุงรักษา

โปรแกรมและการแก้ไขเป็นไปด้วยความยากลำบาก ผู้เขียนโปรแกรมจึงหันมาใช้วิธีการเขียนโปรแกรมที่เรียกว่า “การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP)” ซึ่งใช้แนวคิดในการมองและจำลองปัญหาและตัวแปรต่างๆ เป็นวัตถุ (Object) และมองว่ากิจกรรมทั้งหมดที่เกิดขึ้นเกิดจากความสัมพันธ์และปฏิสัมพันธ์ระหว่างวัตถุ ดังนั้น โปรแกรมจึงประกอบไปด้วยกลุ่มของวัตถุ ซึ่งแต่ละวัตถุที่มีคุณสมบัติเหมือนกัน จะถูกจัดกลุ่มให้อยู่ในคลาส (Class) เดียวกัน โดยในแต่ละ Class สามารถถ่ายทอดลักษณะ (Inheritance) กันลงมาเป็นลำดับขั้น (Hierarchy) เพื่อใช้ในงานอื่นได้ นอกจากนี้ วัตถุในคลาสใดๆ สามารถถูกเรียกใช้จากคลาสอื่นได้ ทำให้ไม่จำเป็นต้องเขียนคำสั่งแบบเดิมซ้ำอีก เพียงแต่เขียนคำสั่งเรียกใช้เท่านั้นก็สามารถทำงานได้ จึงทำให้การพัฒนาโปรแกรมหรือซอฟต์แวร์คอมพิวเตอร์รวดเร็วขึ้น²⁰

2.3.5 ภาษาในยุคที่ห้า (Fifth Generation Language/5GL)

เป็นภาษาที่ใช้สำหรับซอฟต์แวร์เพื่อระบบผู้เชี่ยวชาญ (Expert Systems: ES) และปัญญาประดิษฐ์ (Artificial Intelligence: AI) หรือเรียกอีกอย่างหนึ่งว่า “ภาษารธรรมชาติ (Natural Language)” เนื่องจากการใช้ไวยากรณ์ที่มีโครงสร้างใกล้เคียงกับภาษาของมนุษย์ คือภาษาอังกฤษ มากที่สุด ดังนั้น ผู้เขียนโปรแกรม (programmer) จึงสามารถสร้างซอฟต์แวร์ได้ง่ายดาย โดยสามารถรองรับเทคโนโลยีปัญญาประดิษฐ์ต่างๆ โดยเฉพาะอย่างยิ่ง การทำให้เครื่องคอมพิวเตอร์สามารถเข้าใจคำสั่งที่ผู้ใช้สั่งด้วยเสียงพูด อย่างไรก็ตาม เทคโนโลยีดังกล่าวยังอยู่ในระหว่างการค้นคว้า ทดลอง และพัฒนาให้สมบูรณ์แบบ

2.4 ภาพรวมของขั้นตอนในการออกแบบพัฒนาซอฟต์แวร์คอมพิวเตอร์

เพื่อความเข้าใจถึงสภาพและลักษณะของการพัฒนาหรือออกแบบซอฟต์แวร์คอมพิวเตอร์ซึ่งมีความแตกต่างจากการผลิตสินค้าอุตสาหกรรมอื่นๆ ทั่วไป จึงขออธิบายภาพรวมของขั้นตอนการออกแบบพัฒนาโปรแกรมหรือซอฟต์แวร์คอมพิวเตอร์ ดังต่อไปนี้

²⁰ เฟื่องอ้าง, น. 93-94.

2.4.1 การวิเคราะห์ความต้องการและกำหนดคุณลักษณะของซอฟต์แวร์ (Collecting and analyzing requirement and defining specification)

จุดเริ่มต้นของการพัฒนาโปรแกรม คือ การกำหนดความต้องการของผู้ใช้ ซึ่งความต้องการของผู้ใช้นี้จะนำมาเป็นข้อกำหนดคุณลักษณะของซอฟต์แวร์ ซึ่งวิธีการที่จะทำให้ได้มาซึ่งความต้องการของผู้ใช้นั้น สามารถกระทำได้หลายวิธี เช่น ใช้เทคนิคการทำต้นแบบ (prototyping model) แล้วให้ผู้ใช้ได้ดูต้นแบบเปรียบเทียบกับความต้องการของเขา ใช้เทคนิคการแบ่งขั้นตอนการพัฒนาชุดคำสั่งเป็นขั้นตอน โดยให้ผู้ใช้ตรวจสอบในแต่ละขั้นตอน หรือใช้วิธีสัมภาษณ์และเขียนบันทึกความต้องการของผู้ใช้ไว้เป็นลายลักษณ์อักษรเพื่อเป็นเอกสารอ้างอิง เป็นต้น²¹

2.4.2 การออกแบบซอฟต์แวร์ (Design)

ขั้นตอนนี้เป็นการออกแบบข้อกำหนดและความต้องการรวมทั้งเงื่อนไขที่เกี่ยวข้อง ซึ่งอาจแบ่งเป็นการออกแบบขั้นตอนเบื้องต้น (preliminary design) และการออกแบบขั้นตอนในรายละเอียด (detail design)

การออกแบบขั้นตอนเบื้องต้น (preliminary design) เป็นขั้นตอนที่แสวงหาแนวทางในการพัฒนาชุดคำสั่ง ซึ่งได้เป็นสองส่วนที่สัมพันธ์กัน คือ การออกแบบเชิงแนวคิด (conceptual design) ที่เป็นการคิดพิจารณาถึงหน้าที่และองค์ประกอบของชุดคำสั่งว่าต้องทำอะไรบ้าง ได้ผลอย่างไร ตรงตามความต้องการของผู้ใช้หรือไม่ และ การออกแบบเชิงเทคนิค (technical design) ที่เป็นขั้นตอนออกโดยพิจารณาว่าจะทำชุดคำสั่งอย่างไร โดยนึกถึงเทคนิคที่ต้องใช้ และต้องมีข้อกำหนดทางเทคนิค

การออกแบบขั้นตอนในรายละเอียด (detail design) เป็นการลงรายละเอียดเพื่อกำหนดคุณสมบัติเฉพาะสำหรับการลงรหัสคำสั่ง เลือกแนวทางการเขียนคำสั่งในการสร้างโปรแกรมและการรวมโปรแกรมหรือซอฟต์แวร์เข้าเป็นระบบ

การออกแบบซอฟต์แวร์ที่ดี ควรมีคุณสมบัติให้สามารถปรับปรุงแก้ไขได้ทุกขั้นตอน และการแก้ไขนั้นๆ ให้มีผลกระทบกับส่วนอื่นๆ น้อยที่สุด เพราะไม่ว่าจะคิดออกแบบรูปแบบใดๆ ก็ตาม โอกาสที่มีข้อผิดพลาด (bug/error) ย่อมเกิดขึ้นได้เสมอ ดังนั้น การออกแบบจึงต้องมีลักษณะที่เอื้อ

²¹ สานนท์ เจริญฉาย, อ้างแล้ว เชิงอรรถที่ 5, น. 26-27.

ต่อการปรับปรุงได้ตลอดเวลา²² เช่น การใช้หลักการออกแบบหน่วยย่อยหรือมอดูล (Modularity) กล่าวคือ เป็นการพยายามทำให้แต่ละมอดูลมีความเป็นอิสระระหว่างกันให้มากที่สุด และให้ผลลัพธ์การทำงานของแต่ละมอดูลมีเพียงวัตถุประสงค์เดียว

2.4.3 การเขียนรหัสคำสั่ง (Coding or Programming)

ขั้นตอนนี้ก็คือ การสร้างและเขียนโปรแกรมนั่นเอง โดยเป็นการนำรายละเอียดจากขั้นตอนการออกแบบที่ผ่านมา มาทำการเขียนคำสั่ง (Coding) เพื่อสร้างโปรแกรม โดยจะมีการแจกจ่ายโปรแกรมย่อย หรือมอดูล (โปรแกรมย่อยมีการเรียกกันหลายชื่อตามแต่ลักษณะของภาษานั้นๆ เช่น ในภาษา C เรียกว่า “Function” ในภาษา PASCAL เรียกว่า “Procedure” เป็นต้น ส่วนมอดูลหมายถึง ส่วนประกอบของโปรแกรมซึ่งอาจอยู่ในรูปของโปรแกรมย่อยก็ได้ และสามารถทดสอบได้อย่างอิสระ โดยมอดูลนั้นมีการกิจอย่างใดอย่างหนึ่งโดยเฉพาะ แต่โดยปกติถ้ากล่าวถึงมอดูลอาจหมายถึง Function หรือ Procedure ก็ได้²³) ที่ออกแบบแล้วให้นักเขียนโปรแกรมเขียนคำสั่งตามข้อกำหนดคุณลักษณะของมอดูลนั้นๆ โดยใช้ภาษาคอมพิวเตอร์ ซึ่งมีหลากหลายภาษา แล้วแต่ว่าภาษาใดจะเหมาะสมกับงานประเภทใด ซึ่งจะต้องควบคุมผู้ลงรหัสให้ใช้มาตรฐานคุณลักษณะของแต่ละมอดูล ที่ผู้ออกแบบกำหนดไว้อย่างเคร่งครัด ซึ่งในการลงรหัสนิยมการสอดแทรกข้อความเพื่อบอกกล่าวประเด็นสำคัญบางอย่างไว้ในโปรแกรม ข้อความสอดแทรกนี้เรียกว่า “หมายเหตุ” (comment)²⁴

2.4.4 การทดสอบการทำงานและประเมินผลซอฟต์แวร์ (Testing & Evaluation)

ขั้นตอนนี้เป็นการทดสอบซอฟต์แวร์ภายใต้เงื่อนไขและข้อกำหนดที่ออกแบบไว้ โดยมีวัตถุประสงค์เพื่อค้นหาข้อผิดพลาดของคำสั่ง พบส่วนผิดพลาดทางตรรกะ และเปิดเผยส่วน

²² เพิ่งอ้าง, น. 30.

²³ วิทยา สุคตบวร, วิศวกรรมซอฟต์แวร์เบื้องต้น (กรุงเทพมหานคร: ซีเอ็ดยุคเคชั่น, 2551), น. 30.

²⁴ สานนท์ เจริญฉาย, อ้างแล้ว เชิงอรรถที่ 5, น. 28-29.

ผิดพลาดแล้วแก้ไขให้ถูกต้อง ซึ่งเทคนิคในการทดสอบอาจแบ่งออกได้เป็นสองลักษณะ คือ การวิเคราะห์แบบสถิตินี้กับการวิเคราะห์แบบพลวัต

การวิเคราะห์แบบสถิตินี้ (static analysis) เป็นวิธีที่ไม่ต้องให้โปรแกรมทำงานจริง ทดสอบแบบทบทวนส่วนประกอบต่างๆ ของโปรแกรม เช่น ใช้ภาษาเขียนโปรแกรมถูกต้องหรือไม่ ซึ่งเป็นวิธีที่เหมาะสมกับการทดสอบหน่วยย่อยของซอฟต์แวร์ ซึ่งจะทดสอบด้วยตนเองหรือใช้ซอฟต์แวร์ทดสอบก็ได้

การวิเคราะห์แบบพลวัต (dynamic analysis) เป็นการทดสอบโดยให้โปรแกรมทำงาน ทดสอบผลงานที่ได้เปรียบเทียบกับสิ่งที่ต้องการ อาจแบ่งการพิจารณาเป็นสองส่วน คือ ผลตามตรรกะ เช่น มีขั้นตอนการทำงานถูกต้องหรือไม่ การส่งผ่านข้อมูลให้ผลถูกต้องหรือไม่ และ ผลตามหน้าที่ เช่น ภายใต้เงื่อนไขต่างๆ โปรแกรมส่วนย่อยนั้นสามารถตอบสนองได้ถูกต้องตามหน้าที่หรือไม่²⁵

2.4.5 การส่งมอบและติดตั้งซอฟต์แวร์

ภายหลังจากที่มีการเขียนซอฟต์แวร์คอมพิวเตอร์แล้ว จะมีการส่งมอบซอฟต์แวร์ดังกล่าวให้แก่ผู้ใช้งาน (user) โดยก่อนที่ซอฟต์แวร์จะถูกนำไปใช้ให้เกิดประโยชน์ จะต้องมีการนำซอฟต์แวร์นั้นติดตั้งเข้าสู่ระบบคอมพิวเตอร์ด้วย ซึ่งซอฟต์แวร์บางประเภทสามารถใช้งานได้ทันทีหลังจากดำเนินการคัดลอก (copy) เข้าสู่ระบบคอมพิวเตอร์ แต่ก็มีซอฟต์แวร์อีกหลายประเภทที่จะต้องผ่านกระบวนการติดตั้ง (installation) และตั้งค่า (setup) ก่อนจึงจะสามารถนำไปใช้งานได้ ซึ่งอาจเป็นการการขยายไฟล์ (unpacking) จากที่ถูกย่อไว้ (compressed from) เพื่อความสะดวกและประหยัดพื้นที่สำหรับสื่อที่ใช้บรรจุซอฟต์แวร์ การคัดลอก (copy) ซอฟต์แวร์ดังกล่าวให้อยู่ในพื้นที่ที่เหมาะสม การปรับแต่งให้เหมาะสมกับระบบฮาร์ดแวร์คอมพิวเตอร์และความต้องการของผู้ใช้งาน การส่งข้อมูลเกี่ยวกับซอฟต์แวร์ไปยังระบบปฏิบัติการ เป็นต้น

กระบวนการติดตั้งหรือตั้งค่าซอฟต์แวร์คอมพิวเตอร์ก่อนใช้งานนั้นแตกต่างกันไปตามแต่ลักษณะของซอฟต์แวร์และเครื่องคอมพิวเตอร์ที่แตกต่างกัน ซึ่งมีซอฟต์แวร์จำนวนไม่น้อยที่มีโปรแกรมสำหรับการติดตั้งไว้ให้โดยอัตโนมัติ แต่ก็มีซอฟต์แวร์อีกส่วนหนึ่งที่มีความยุ่งยากในการติดตั้งจนต้องอาศัยผู้เชี่ยวชาญในการดำเนินการติดตั้ง

²⁵ เฟื่องอ้าง, น. 29-30.

2.4.6 การบำรุงรักษา (Software Maintenance)

หลังจากส่งมอบและติดตั้งตลอดจนทดสอบระบบเสร็จแล้ว ต่อไปจะเป็นเรื่องของการบำรุงรักษาซอฟต์แวร์ ซอฟต์แวร์ที่ดีควรจะต้องมีระบบการบำรุงรักษาที่ดีด้วย การวัดเกี่ยวกับเวลาเฉลี่ยในการแก้ไขความผิดพลาดที่เกิดขึ้นหรือเวลาเฉลี่ยในการแก้ไขข้อผิดพลาดก็เป็นปัจจัยประการหนึ่งในการวัดคุณภาพของซอฟต์แวร์คอมพิวเตอร์²⁶

เราอาจแบ่งประเภทของการบำรุงรักษาซอฟต์แวร์ออกได้ดังต่อไปนี้²⁷

2.4.6.1 Corrective maintenance

เป็นการดำเนินการด้านการบำรุงรักษาระบบให้สามารถทำงานในหน้าที่ประจำในแต่ละวันได้อย่างถูกต้อง (corrective maintenance)

2.4.6.2 Perfective Maintenance

เป็นการบำรุงรักษาให้ระบบสามารถปฏิบัติการได้ภายใต้สภาพแวดล้อมที่เปลี่ยนแปลงไปตามภาวะการณ์ของโลกที่เปลี่ยนไป รวมถึงการบำรุงรักษาหรือทำการปรับเปลี่ยนให้เป็นไปตามเทคโนโลยีที่เปลี่ยนแปลงไป

2.4.6.3 Adaptive Maintenance

เป็นการบำรุงรักษาระบบงานเดิม โดยเน้นเกี่ยวกับการปรับปรุงหรือแก้ไขระบบให้ดีขึ้น เช่น การปรับปรุงให้ระบบงานเดิมสามารถทำงานได้เร็วขึ้นหรือเพิ่มเติมหน้าที่การทำงานต่างๆ ให้มากขึ้นกว่าเดิม

2.4.6.4 Preventive Maintenance

เป็นการบำรุงรักษาระบบในเชิงป้องกันมิให้ระบบเกิดปัญหา การบำรุงรักษาในกรณีนี้อาจมีการสร้างเป็นโปรแกรมหรือโมดูลเพิ่มเติมก็ได้ เช่น โปรแกรมป้องกันผู้บุกรุก เป็นต้น

²⁶ วิทยา สุคตบวร, อ้างแล้ว เชิงอรรถที่ 23, น. 133.

²⁷ เฟื่องอ้าง, น. 134.

2.5 สถานภาพทางกฎหมายของซอฟต์แวร์คอมพิวเตอร์

วงการวิชาการด้านกฎหมายในประเทศไทยยังไม่มีกรณีเสนอ หรือ วิจารณ์ หรือ แสดงความเห็นเกี่ยวกับประเด็นดังกล่าวเหมือนในต่างประเทศ ซึ่งได้มีการแสดงทรรศนะเกี่ยวกับ สถานภาพของซอฟต์แวร์คอมพิวเตอร์ในทางกฎหมายกันอย่างกว้างขวาง โดยสรุปได้เป็นสาม แนวทาง ดังนี้

2.5.1 ความเห็นแรก

ซอฟต์แวร์เป็นวัตถุไม่มีรูปร่าง แต่อาจถือว่าเป็นส่วนหนึ่งของวัตถุมีรูปร่างที่บันทึก หรือบรรจุซอฟต์แวร์นั้นได้

ในคดี *St Albans City v DC International Computers Ltd*²⁸ ของประเทศอังกฤษ เป็นข้อพิพาทเกี่ยวกับการขายเครื่องคอมพิวเตอร์และซอฟต์แวร์คอมพิวเตอร์ โดยซอฟต์แวร์ที่ จำเลยขายให้กับโจทก์ซึ่งเป็นหน่วยงานราชการพร้อมเครื่องคอมพิวเตอร์นั้น มีความชำรุดบกพร่อง ส่งผลให้การจัดเก็บภาษีผิดพลาด ศาลวินิจฉัยว่าแม้ซอฟต์แวร์คอมพิวเตอร์จะเป็นวัตถุไม่มีรูปร่าง (intangible) ซึ่งไม่ตกอยู่ภายใต้อำนาจความคำว่า “สินค้า” ตาม Sale of Goods Act 1979 Section 61 แต่เมื่อซอฟต์แวร์คอมพิวเตอร์ถูกบรรจุหรือบันทึกอยู่ในทรัพย์สินที่มีรูปร่าง คือ เครื่อง คอมพิวเตอร์หรือดิสก์ (disk) ดังนั้น เมื่อซอฟต์แวร์ชำรุดบกพร่อง ย่อมถือว่าผู้ขายผิดข้อกำหนด ตามกฎหมาย (implied terms) ตาม Sale of Goods Act 1979

2.5.2 ความเห็นที่สอง

ความเห็นนี้มองว่าซอฟต์แวร์คอมพิวเตอร์เป็นข้อมูลที่ถูกเขียนขึ้น (coded information) โดยไม่ถือเป็นทรัพย์สิน (property) ในตัวของมันเอง²⁹ แต่เป็นสิ่งที่มีความสำคัญต่อ สิทธิและคุ้มครองสิทธิในการควบคุมการใช้ข้อมูลดังกล่าวโดยผู้มีสิทธิ (author or donor) นั้น ซอฟต์แวร์คอมพิวเตอร์ไม่ถือว่าเป็นวัตถุมีรูปร่างและไม่เป็นสินค้าตามความหมายที่ระบุไว้ใน Sale

²⁸ [1996] 4 All ER 481.

²⁹ *Oxford v. Moss* [1978] 68 Cr App R 183.

of Goods Act section 61 ซึ่งระบุว่า “‘Goods’ includes all personal chattels other than things in action and money ...”³⁰

อย่างไรก็ตาม ในกรณีที่ซอฟต์แวร์คอมพิวเตอร์ในสถานะที่เป็นข้อมูล (information) ดังกล่าว เมื่อถูกบันทึกไว้ในดิสก์หรือสื่ออย่างอื่นในทำนองเดียวกันแล้ว จะมีสถานะอย่างไรนั้น มีความเห็นแบ่งออกเป็นสองฝ่าย ฝ่ายแรกเห็นว่าซอฟต์แวร์คอมพิวเตอร์ในกรณีดังกล่าวเป็นวัตถุมีรูปร่าง (Physical)³¹ ฝ่ายที่สองเห็นว่ากรณีดังกล่าวนั้นซอฟต์แวร์คอมพิวเตอร์เป็น “งาน” หรือ “คำสั่ง” (functional) โดยซอฟต์แวร์เป็นสิ่งที่ถูกใช้เพื่อทำให้เครื่องคอมพิวเตอร์ทำงานไม่ใช่เป็นเพียงข้อมูล (information) เท่านั้น³²

2.5.3 ความเห็นที่สาม

ซอฟต์แวร์เป็นสิ่งที่ได้รับความคุ้มครองในเรื่องทรัพย์สินทางปัญญา (Intellectual Property) กล่าวคือ ซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์ไม่ใช่ลิขสิทธิ์ในตัวของมันเอง แต่เป็นสิ่งที่ถูกคุ้มครองโดยลิขสิทธิ์³³ หรือ เป็นงานอันมีลิขสิทธิ์ ซึ่งแนวคิดดังกล่าวได้ถูกรองรับโดยมาตรา 6 แห่ง พ.ร.บ. ลิขสิทธิ์ พ.ศ. 2537 ซึ่งบัญญัติให้งานสร้างสรรค์ประเภทวรรณกรรม ถือเป็นงานอันมีลิขสิทธิ์ โดยในมาตรา 4 ได้ระบุคำจำกัดความของ “วรรณกรรม” ว่าให้หมายความรวมถึงโปรแกรมคอมพิวเตอร์ด้วย

2.5.4 ความเห็นของผู้เขียน

ปัจจุบัน ด้วยลักษณะพิเศษของซอฟต์แวร์คอมพิวเตอร์ที่อาจบรรจุอยู่ในสื่อทางกายภาพได้ ทำให้เกิดประเด็นปัญหาในทางกฎหมายของประเทศในกลุ่มคอมมอนลอว์ว่าควรถือเป็นวัตถุมี

³⁰ Diane Rowland and Elizabeth Macdonald, *Information Technology Law*, 2nd ed. (London: Cavendish Publishing Limited, 2000), pp. 174-175.

³¹ ดูเพิ่มเติมในคดี *R v. Whiteley* (1991) The Times, 6 February. และ *South Central Bell Telephone Co v. Barthelemy* 643 So kd 1240 (La 1994).

³² Diane Rowland and Elizabeth Macdonald, *supra note* 30, pp. 177-178.

³³ *Ibid.*

รูปร่างหรือไม่ เนื่องจากหากเป็นวัตถุไม่มีรูปร่างจะไม่ถือว่าเป็นสินค้า (goods) แต่จะถือว่าเป็นเรื่องของการให้บริการ (services) ซึ่งจะมีหลักกฎหมายที่นำมาปรับใช้เพื่อความเป็นธรรมแก่คู่สัญญาที่แตกต่างกัน

ผู้เขียนเห็นว่า การที่ซอฟต์แวร์คอมพิวเตอร์มีลักษณะเป็นคำสั่ง โดยผู้พัฒนาหรือผู้เขียนโปรแกรมได้ทำการเขียนหรือลงรหัส (Coding) ขึ้นด้วยภาษาคอมพิวเตอร์ (Computer Language) สำคัญหรือมูลค่าของซอฟต์แวร์คอมพิวเตอร์จึงอยู่ที่การออกแบบ ตลอดจนภาษาคอมพิวเตอร์ที่ถูกเรียงร้อยเข้าด้วยกันจนเป็นคำสั่งที่สามารถสั่งการให้คอมพิวเตอร์สามารถทำงานได้ตามความประสงค์ของผู้พัฒนาหรือผู้เขียนโปรแกรมนั่นเอง ซึ่งไม่ว่าจะถูกพัฒนาขึ้นจากภาษาระดับล่าง สัญลักษณ์ หรือภาษาระดับสูง ย่อมอยู่ในรูปของวัตถุไม่มีรูปร่าง ซึ่งแม้จะถูกบรรจุอยู่ในสื่อกลางที่มีรูปร่างอย่างไรก็ตาม เนื้อหาสาระและมูลค่าแห่งซอฟต์แวร์คอมพิวเตอร์นั้นก็ยังอยู่ที่ตัวคำสั่งและแนวคิดในการออกแบบซอฟต์แวร์เช่นเดิม

หากพิจารณาตามกฎหมายไทย การมีรูปร่างหรือไม่มีรูปร่างของทรัพย์สินก็เคยส่งผลกระทบต่อสิทธิตามกฎหมายบางประการเช่นกัน เช่น กรณีของคำพิพากษาศาลฎีกาที่ 9544/2542 ซึ่งวินิจฉัยในกรณีของเครื่องหมายการค้าว่าเป็นทรัพย์สินทางปัญญาอย่างหนึ่ง ไม่ใช่อสังหาริมทรัพย์ หรือสังหาริมทรัพย์ ไม่มีรูปร่าง ทั้งนี้อาจยึดถือครอบครองได้อย่างทรัพย์สินทั่วไป ดังที่บัญญัติไว้ตามประมวลกฎหมายแพ่งและพาณิชย์ บรรพ 4 ว่าด้วยทรัพย์สิน การที่จำเลยนำเครื่องหมายการค้าของผู้อื่น (ของโจทก์) มาใช้กับสินค้าของตนโดยไม่ได้รับอนุญาต จากเจ้าของแม้เป็นระยะเวลาสั้นเพียงใด ก็ไม่ทำให้จำเลยมีสิทธิในเครื่องหมายการค้าดีกว่าโจทก์ได้ บทบัญญัติว่าด้วยการได้กรรมสิทธิ์ในทรัพย์สินโดยการครอบครองปรปักษ์ตามประมวลกฎหมายแพ่งและพาณิชย์ มาตรา 1382 หากอาจนำมาใช้บังคับแก่สิทธิในเครื่องหมายการค้าอันเป็นทรัพย์สินทางปัญญาได้ไม่ ซึ่งในกรณีของซอฟต์แวร์คอมพิวเตอร์เองก็น่าจะเทียบเคียงได้เช่นกัน กล่าวคือ แม้จะมีการครอบครองสื่อที่บรรจุ Source Code ไว้นานเพียงไร ก็ไม่อาจทำให้ผู้ครอบครองอ้างสิทธิในซอฟต์แวร์โดยการครอบครองปรปักษ์ได้ แม้สื่อที่ใช้บรรจุซอฟต์แวร์คอมพิวเตอร์จะเป็นวัตถุมีรูปร่างก็ตาม

ดังนั้น ผู้เขียนจึงเห็นว่าซอฟต์แวร์คอมพิวเตอร์นั้นเป็นวัตถุที่ไม่มีรูปร่างซึ่งได้รับการคุ้มครองตามกฎหมายลิขสิทธิ์ และแม้ว่าบางครั้งจะถูกบรรจุอยู่ในสื่อหรือวัตถุที่มีรูปร่าง แต่เราก็ยังสามารถแยกตัวสิทธิในซอฟต์แวร์คอมพิวเตอร์ออกจากสิทธิในวัตถุมีรูปร่างนั้นออกจากกันได้ และ

โดยที่ซอฟต์แวร์คอมพิวเตอร์เป็นงานอันมีลิขสิทธิ์อันได้รับความคุ้มครองตามกฎหมาย ซึ่งอาจมีราคา และอาจถือเอาได้ จึงถือว่าเป็นทรัพย์สินตามประมวลกฎหมายแพ่งและพาณิชย์ด้วย

2.6 ลักษณะและรูปแบบของการได้มาซึ่งสิทธิในการใช้ซอฟต์แวร์คอมพิวเตอร์

การที่บุคคลจะได้มาซึ่งสิทธิในการใช้ซอฟต์แวร์คอมพิวเตอร์นั้น อาจได้มาโดยหลายลักษณะด้วยกัน จึงขอยกตัวอย่างการได้มาซึ่งสิทธิในการใช้ซอฟต์แวร์คอมพิวเตอร์ที่สำคัญๆ ดังนี้

2.6.1 การได้มาซึ่งลิขสิทธิ์โดยการพัฒนาซอฟต์แวร์คอมพิวเตอร์ขึ้นเอง

เนื่องจากซอฟต์แวร์คอมพิวเตอร์หรือโปรแกรมคอมพิวเตอร์นั้น เป็นงานอันมีลิขสิทธิ์ ตามความในมาตรา 6 แห่งพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 ส่งผลให้เจ้าของลิขสิทธิ์มีสิทธิแต่ผู้เดียวตามที่ระบุไว้ในมาตรา 15 แห่งพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 ดังต่อไปนี้

มาตรา 15 ภายใต้บังคับมาตรา 9 มาตรา 10 และมาตรา 14 เจ้าของลิขสิทธิ์ย่อมมีสิทธิแต่เพียงผู้เดียวดังต่อไปนี้

- (1) ทำซ้ำหรือดัดแปลง
- (2) เผยแพร่ต่อสาธารณชน
- (3) ให้เช่าต้นฉบับหรือสำเนางานโปรแกรมคอมพิวเตอร์ โสตทัศนวัสดุ ภาพยนตร์ และสิ่งบันทึกเสียง
- (4) ให้ประโยชน์อันเกิดจากลิขสิทธิ์แก่ผู้อื่น
- (5) อนุญาตให้ผู้อื่นใช้สิทธิตาม (1) (2) หรือ (3) โดยจะกำหนดเงื่อนไขอย่างใดหรือไม่ก็ได้ แต่เงื่อนไขดังกล่าวจะกำหนดในลักษณะที่เป็นการจำกัดการแข่งขันโดยไม่เป็นธรรมไม่ได้

การพิจารณาว่าเงื่อนไขตามวรรคหนึ่ง (5) จะเน้นการจำกัดการแข่งขันโดยไม่เป็นธรรมหรือไม่ ให้เป็นไปตามหลักเกณฑ์ วิธีการและเงื่อนไขที่กำหนดในกฎกระทรวง

โดยตามมาตรา 8 แห่งพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 กำหนดให้ผู้สร้างสรรค์ (ผู้พัฒนาหรือผู้เขียนซอฟต์แวร์คอมพิวเตอร์) เป็นผู้มีลิขสิทธิ์ในงานที่ตนได้สร้างสรรค์ขึ้นภายใต้เงื่อนไขดังต่อไปนี้

(1) ในกรณีที่ยังไม่ได้มีการโฆษณางาน ผู้สร้างสรรค์ต้องเป็นผู้มีสัญชาติไทย หรืออยู่ในราชอาณาจักรหรือเป็นผู้มีสัญชาติหรืออยู่ในประเทศที่เป็นภาคีแห่งอนุสัญญาว่าด้วยการคุ้มครองลิขสิทธิ์ซึ่งประเทศไทยเป็นภาคีอยู่ด้วย ตลอดระยะเวลาหรือเป็นส่วนใหญ่ในการสร้างสรรค์งานนั้น

(2) ในกรณีที่ได้มีการโฆษณาแล้ว การโฆษณานั้นในครั้งแรกได้กระทำขึ้นในราชอาณาจักรหรือในประเทศที่เป็นภาคีแห่งอนุสัญญาว่าด้วยการคุ้มครองลิขสิทธิ์ซึ่งประเทศไทยเป็นภาคีอยู่ด้วย หรือในกรณีที่การโฆษณารั้งแรกได้กระทำนอกราชอาณาจักรหรือในประเทศอื่นที่ไม่เป็นภาคีแห่งอนุสัญญาว่าด้วยการคุ้มครองลิขสิทธิ์ซึ่งประเทศไทยเป็นภาคีอยู่ด้วย หากได้มีการโฆษณางานดังกล่าวในราชอาณาจักรหรือในประเทศที่เป็นภาคีแห่งอนุสัญญาว่าด้วยการคุ้มครองลิขสิทธิ์ซึ่งประเทศไทยเป็นภาคีอยู่ด้วยภายในสามสิบวันนับแต่วันที่ได้มีการโฆษณารั้งแรก หรือผู้สร้างสรรค์เป็นผู้มีลักษณะตามที่กำหนดไว้ใน (1) ในขณะที่มีการโฆษณางานครั้งแรก

ในกรณีที่ผู้สร้างสรรค์ต้องเป็นผู้มีสัญชาติไทย ถ้าผู้สร้างสรรค์เป็นนิติบุคคล นิติบุคคลนั้นต้องเป็นนิติบุคคลที่จัดตั้งขึ้นตามกฎหมายไทย

ทั้งนี้ หากผู้สร้างสรรค์ (ผู้พัฒนาหรือผู้เขียนซอฟต์แวร์คอมพิวเตอร์) ทำการพัฒนาหรือเขียนซอฟต์แวร์คอมพิวเตอร์ขึ้นในฐานะพนักงานหรือลูกจ้าง มาตรา 9 แห่งพระราชบัญญัติลิขสิทธิ์กำหนดว่า ถ้ามิได้ทำเป็นหนังสือตกลงกันไว้เป็นอย่างอื่น ให้ลิขสิทธิ์ในงานนั้นเป็นของผู้สร้างสรรค์ (ผู้พัฒนาหรือผู้เขียนซอฟต์แวร์คอมพิวเตอร์) แต่นายจ้างมีสิทธิในงานนั้นออกเผยแพร่ต่อสาธารณชนได้ตามที่เป็นวัตถุประสงค์แห่งการจ้างแรงงานนั้น

2.6.2 การได้มาซึ่งลิขสิทธิ์โดยการดัดแปลงงานอันมีลิขสิทธิ์ของผู้อื่นโดยได้รับอนุญาต

ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 มาตรา 11 บัญญัติไว้ดังนี้

งานใดมีลักษณะเป็นการดัดแปลงงานอันมีลิขสิทธิ์ตามพระราชบัญญัตินี้ โดยได้รับอนุญาตจากเจ้าของลิขสิทธิ์ ให้ผู้ที่ได้ดัดแปลงนั้น มีลิขสิทธิ์ในงานที่ได้ดัดแปลง

ตามพระราชบัญญัตินี้ แต่ทั้งนี้ ไม่กระทบกระเทือนสิทธิของเจ้าของลิขสิทธิ์ที่มีอยู่ในงานของผู้สร้างสรรค์เดิมที่ถูกดัดแปลง

โดยในกรณีของซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์ มาตรา 4 ได้ให้นิยามศัพท์ของคำว่า “ดัดแปลง” ไว้ว่า ในส่วนของโปรแกรมคอมพิวเตอร์ ให้หมายความรวมถึง ทำซ้ำ โดยเปลี่ยนรูปแบบ ปรับปรุง แก้ไขเพิ่มเติมโปรแกรมคอมพิวเตอร์ ในส่วนอันเป็นสาระสำคัญ โดยไม่มีลักษณะเป็นการจัดทำขึ้นใหม่

พึงสังเกตว่า การดัดแปลงอันจะทำให้ได้มาซึ่งลิขสิทธิ์นั้น จะต้องเป็นกรณีที่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ในงานเดิม ไม่เช่นนั้นแล้ว ผู้ดัดแปลงย่อมไม่ได้มาซึ่งลิขสิทธิ์ในงานที่ได้ดัดแปลงนั้น³⁴

2.6.3 การได้มาซึ่งลิขสิทธิ์จากการจ้างพัฒนาซอฟต์แวร์คอมพิวเตอร์ (Software Development Contract)

ตามมาตรา 10 แห่งพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 กำหนดให้งานที่ผู้สร้างสรรค์ (ผู้พัฒนาซอฟต์แวร์คอมพิวเตอร์) ได้สร้างสรรค์ (พัฒนาซอฟต์แวร์คอมพิวเตอร์) ขึ้นโดยการรับจ้างบุคคลอื่น ให้ผู้ว่าจ้างเป็นผู้มีลิขสิทธิ์ในงานนั้น เว้นแต่ผู้สร้างสรรค์และผู้ว่าจ้างจะได้ตกลงกันไว้เป็นอย่างอื่น

สัญญาจ้างพัฒนาซอฟต์แวร์เป็นสัญญาที่ผู้รับบริการเชื่อว่าจ้างบริษัทพัฒนาซอฟต์แวร์ (Software House) เขียนโปรแกรมเพื่อการใช้งาน (Application) ตามความประสงค์ (requirement) ของผู้ว่าจ้าง กรณีดังกล่าวย่อมเกิดนิติสัมพันธ์ระหว่างกันในลักษณะสัญญาโดยทั่วไปแล้ว สัญญาจ้างพัฒนาซอฟต์แวร์มักจะทำกันเป็นลายลักษณ์อักษรและมักจะมีข้อสัญญายกเว้นความรับผิดชอบของผู้พัฒนาซอฟต์แวร์³⁵

³⁴ ไชยยศ เหมะรัชตะ, คำอธิบายกฎหมายลิขสิทธิ์, พิมพ์ครั้งที่ 4, (กรุงเทพมหานคร: สำนักพิมพ์นิติธรรม, 2549), น. 97-98.

³⁵ Chris Reed and John Angel, Computer Law, 5th ed. (Oxford: Oxford University Press, 2003), p. 103.

ตามกฎหมายอังกฤษสัญญาดังกล่าวที่มีลักษณะเป็นการให้บริการซึ่งอยู่ภายใต้ Supply of Goods and Services Act 1982 โดยไม่ถือว่าเป็นการซื้อขายสินค้า (goods) ส่วนตามกฎหมายไทยอาจบังคับได้ตามประมวลกฎหมายแพ่งและพาณิชย์ลักษณะจำทำของเนื่องจากเป็นสัญญาจ้างที่มุ่งผลสำเร็จในการทำงาน³⁶

2.6.4 การซื้อแพคเกจซอฟต์แวร์ (Purchase of Software Packages)

ตามกฎหมายอังกฤษ การซื้อ Bespoke หรือ Customized Software มักถูกจัดประเภทเป็นการจ้างบริการ ส่วนซอฟต์แวร์มาตรฐาน (Package Software) โดยทั่วไปถือว่าเป็นสินค้าที่จำหน่ายในรูปของวัตถุทางกายภาพ เช่น CD – ROM อย่างไรก็ตาม ซอฟต์แวร์ที่ดาวน์โหลดจากเว็บไซต์หรือแหล่งอื่นที่คล้ายคลึงกันไม่อาจถือว่าเป็นสินค้าภายใต้คำจำกัดความใน Section 61 ของ Sale of Goods Act 1979 ของกฎหมายอังกฤษ เนื่องจากไม่อยู่ในรูปของวัตถุที่มีรูปร่าง (tangible) โดยจะกล่าวถึงต่อไปในบทที่ 4

ตามประมวลกฎหมายแพ่งและพาณิชย์ลักษณะซื้อขายของไทยมาตรา 453 ระบุว่าสัญญาซื้อขายคือสัญญาโอนกรรมสิทธิ์แห่งทรัพย์สิน ซึ่งตามมาตรา 138 ระบุว่า “ทรัพย์สิน” หมายความว่ารวมถึงวัตถุมีรูปร่างและไม่มีรูปร่างซึ่งอาจมีราคาและอาจถือเอาได้ ดังนั้น ทรัพย์สินไม่มีรูปร่าง จึงอาจซื้อขายได้ตามกฎหมายไทย³⁷

อย่างไรก็ตาม เมื่อพิจารณาการซื้อซอฟต์แวร์แล้ว วัตถุแห่งการซื้อขายที่คู่สัญญาประสงค์จะโอนกรรมสิทธิ์กันมีเพียงสื่อที่มีรูปร่างซึ่งใช้บรรจุซอฟต์แวร์คอมพิวเตอร์เท่านั้น มิได้มีความประสงค์จะโอนลิขสิทธิ์ในซอฟต์แวร์คอมพิวเตอร์ไปด้วย โดยในส่วนของซอฟต์แวร์คอมพิวเตอร์ที่บรรจุอยู่ในสื่อ นั้น เป็นเพียงการอนุญาตให้ใช้ซอฟต์แวร์³⁸ (License Agreement)

³⁶ ประมวลกฎหมายแพ่งและพาณิชย์ มาตรา 587 บัญญัติไว้ว่า “อันว่าจ้างทำของนั้น คือ สัญญาซึ่งบุคคลคนหนึ่งเรียกว่าผู้รับจ้าง ตกลงรับจะทำการงานสิ่งใดสิ่งหนึ่งจนสำเร็จให้แก่บุคคลอีกคนหนึ่งเรียกว่าผู้ว่าจ้าง และผู้ว่าจ้างตกลงจะให้สินจ้างเพื่อผลสำเร็จแห่งการที่ทำนั้น”

³⁷ ประพนธ์ ศาตะมาน และ ไพจิตร ปุณณพันธ์, คำอธิบายประมวลกฎหมายแพ่งและพาณิชย์ลักษณะซื้อขาย, แก้ไขเพิ่มเติม พิมพ์ครั้งที่ 11, (กรุงเทพมหานคร: นิติบรรณการ, 2539), น. 11.

³⁸ Diane Rowland and Elizabeth Macdonald, *supra* note 30, p. 178.

ระหว่างผู้ผลิตหรือผู้พัฒนาซอฟต์แวร์ (software house) กับ ผู้ใช้งาน (end user) ซึ่งจะได้กล่าวถึงต่อไป

2.6.5 สัญญาอนุญาตให้ใช้ซอฟต์แวร์คอมพิวเตอร์ (Contract on License to Use the Computer Software)

สัญญาอนุญาตให้ใช้ซอฟต์แวร์คอมพิวเตอร์นั้น หมายถึง สัญญาที่ฝ่ายหนึ่งเรียกว่า ผู้ให้อนุญาตอนุญาตให้บุคคลอีกฝ่ายหนึ่งเรียกว่าผู้รับอนุญาตมีสิทธิใช้ซอฟต์แวร์คอมพิวเตอร์ ภายใต้ข้อจำกัดและเงื่อนไขที่ระบุไว้ในสัญญา

โดยปกติแล้ว การนำสำเนาซอฟต์แวร์มาตรฐานออกจำหน่ายเป็นจำนวนมากนั้น จะไม่มีการลงนามสัญญาอนุญาตให้ใช้ระหว่างผู้อนุญาตกับผู้ซื้อแพ็คเกจซอฟต์แวร์ แต่เจ้าของลิขสิทธิ์หรือผู้มีสิทธิในการให้อนุญาตช่วงมักจะระบุเงื่อนไขการอนุญาตให้ใช้สิทธิไว้ที่แพ็คเกจซอฟต์แวร์ (Shrink-wrap Licensing) โดยระบุเงื่อนไขในลักษณะดังต่อไปนี้ไว้บนกล่องหรือซองบรรจุผลิตภัณฑ์โปรแกรมคอมพิวเตอร์ที่จำหน่าย

บริษัท จำกัด ซึ่งเป็นผู้อนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ตามสัญญาฉบับนี้ ขอเตือนให้ท่านอ่านข้อความในสัญญาฉบับนี้โดยตลอดก่อนที่จะท่านจะเข้าผูกพันตามสัญญาโดยการเปิดซองบรรจุแผ่นดิสก์นี้

เมื่อท่านอ่านข้อความในสัญญาฉบับนี้แล้ว หากไม่ตกลงตามข้อกำหนดและเงื่อนไขที่ระบุไว้ในสัญญา ขอให้ส่งแผ่นดิสก์พร้อมซองบรรจุที่ยังไม่ได้เปิด และสิ่งที่แนบมาด้วยทั้งหมด (รวมทั้งเอกสารและปกหรือที่ใส่เอกสารอื่นๆ) คืนไปยังสถานที่ที่ท่านได้รับสิ่งเหล่านี้มาภายใน 15 วัน (สิบห้าวัน) นับแต่วันที่ได้รับมอบโปรแกรมคอมพิวเตอร์ และบริษัทฯ จะคืนค่าผลิตภัณฑ์โปรแกรมคอมพิวเตอร์ให้แก่ท่านโดยเร็ว

เมื่อท่านเปิดช่องใส่แผ่นดิสก์ที่ปิดผนึกนี้ และ/หรือ ใช้โปรแกรมคอมพิวเตอร์นี้ ถือว่าท่านยอมผูกพันในฐานะผู้รับอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ตามสัญญาฉบับนี้ โดยผู้อนุญาตและผู้รับอนุญาตตกลงทำสัญญากันมีข้อความดังต่อไปนี้ ...³⁹

ซึ่งการดาวน์โหลดซอฟต์แวร์คอมพิวเตอร์จากอินเทอร์เน็ตส่วนใหญ่ก็จะระบุข้อความในทำนองเดียวกัน โดยให้ผู้ใช้งานแสดงเจตนายอมรับเงื่อนไขในการอนุญาตให้ใช้สิทธิก่อนเริ่มติดตั้งโปรแกรม อาจด้วยวิธีคลิกที่ปุ่มหรือวัตถุใดวัตถุหนึ่งบนหน้าจอคอมพิวเตอร์

กรณีดังกล่าวเป็นปัญหาที่น่าพิจารณาว่าเงื่อนไขดังกล่าวถือเป็นข้อสัญญาที่มีผลบังคับสมบูรณ์หรือไม่ และควรถือได้หรือไม่ว่าผู้ใช้งานได้อ่านและเข้าใจข้อความดังกล่าวและยินยอมผูกพันตามสัญญาหรือเงื่อนไขดังกล่าวแล้ว

2.6.6 สัญญาอนุญาตให้เข้าถึงข้อมูล (Access Contract)

ตาม Uniform Computer Information Transaction Act ของประเทศสหรัฐอเมริกา ได้ให้คำจำกัดความคำว่า “Access Contract” ไว้ว่า เป็นสัญญาที่ฝ่ายหนึ่งเรียกว่าผู้ให้อนุญาต (access provider) อนุญาตให้คู่สัญญาอีกฝ่ายหนึ่งเรียกว่าผู้รับอนุญาต (customer) ได้เข้าถึง (access) หรือได้รับข้อมูลจาก เครื่องมือหรืออุปกรณ์ทางอิเล็กทรอนิกส์ (electronic facility)⁴⁰ ซึ่งในคำอธิบาย (Comment) ทำยมาตราได้อธิบายเพิ่มเติมว่า เครื่องมือหรืออุปกรณ์ดังกล่าวนั้นรวมถึงเครื่องคอมพิวเตอร์หรือเว็บไซต์ ซึ่งลักษณะของสัญญาดังกล่าวไม่ได้หมายความว่าเพียงการอนุญาตเข้าไปยังสถานที่ตั้งของข้อมูลในทางกายภาพโดยไม่ได้มีการเข้าถึงข้อมูลอย่างแท้จริง และไม่ได้หมายถึงการซื้อเครื่องรับโทรทัศน์วิทยุ หรือสินค้าในทำนองเดียวกันที่มีความสามารถในการเข้าถึงข้อมูลได้ โดยมีได้มีข้อสัญญาในการอนุญาตให้เข้าถึงข้อมูล

สัญญาการเข้าถึงข้อมูล (access contract) เป็นลักษณะหนึ่งของการให้บริการแบบออนไลน์ (“on-line” services) ซึ่งรวมถึงการประมวลผลข้อมูลทางไกล (remote data processing) การเข้าถึงแอปพลิเคชันซอฟต์แวร์หรือข้อมูลบนเครื่องคอมพิวเตอร์หรือระบบจดหมาย

³⁹ “สัญญาตัวอย่างด้านลิขสิทธิ์”, วารสารลิขสิทธิ์สนเทศ, ปีที่ 5, ฉบับที่ 2, น. 11, (พฤษภาคม 2547).

⁴⁰ Uniform Computer Information Transaction Act, Section 102 (a) (1)

อิเล็กทรอนิกส์ของผู้อนุญาตหรือของบุคคลที่สาม และบริการอัปเดตข้อมูลที่ได้รับอนุญาตมีสิทธิครอบครองอยู่โดยอัตโนมัติผ่านทางอุปกรณ์อิเล็กทรอนิกส์ โดยการเข้าถึงดังกล่าวนั้นผู้ได้รับอนุญาตมักจะได้รับอนุญาตให้ใช้ซอฟต์แวร์บางตัวเพื่อประโยชน์ในการใช้งานบางประการด้วย ทั้งนี้ การที่ผู้เผยแพร่ซอฟต์แวร์ทำการดาวน์โหลดซอฟต์แวร์เข้าสู่ระบบหรือเครื่องคอมพิวเตอร์ของลูกค้า และให้สิทธิในการใช้ซอฟต์แวร์ดังกล่าวหลังจากการดาวน์โหลดเสร็จสิ้นนั้น ไม่ถือเป็นสัญญาการเข้าถึงข้อมูล (access contract)

ข้อสังเกตประการหนึ่งก็คือ ผู้ให้อนุญาตในการเข้าถึงนั้น อาจไม่ได้เป็นเจ้าของหรือผู้อนุญาตให้ใช้ข้อมูลก็ได้ กรณีดังกล่าว สัญญาจะประกอบไปด้วยสามฝ่าย คือ ลูกค้า (customer) ผู้อนุญาตให้เข้าถึง (access provider) และผู้จัดหาหรือผู้อนุญาตให้ใช้ข้อมูล (content provider)

ตัวอย่างของ Access Contract เช่น บัตรเอทีเอ็ม (ATM Card) สมาร์ทการ์ด บริการ e-banking ผ่านทางโทรศัพท์มือถือ เว็บไซต์ที่ให้บริการ home-banking software และเกมออนไลน์ ประเภทที่ต้องเชื่อมเข้าสู่ระบบของผู้ให้บริการในการเข้าเล่นทุกครั้ง เป็นต้น

2.7 ลักษณะเฉพาะของความชำรุดบกพร่องหรือความไม่ปลอดภัยของซอฟต์แวร์คอมพิวเตอร์ (The Nature of Software Defects)

เนื่องจากซอฟต์แวร์คอมพิวเตอร์เป็นวัตถุที่ไม่มีรูปร่าง (intangible products) ลักษณะของความชำรุดบกพร่องหรือความไม่ปลอดภัยจึงแตกต่างจากสินค้าหรือผลิตภัณฑ์ที่มีรูปร่างโดยทั่วไป

ในคำอธิบายเพิ่มเติม (Comment) ท้าย Section 403 ของ Uniform Computer Information Transaction Act ของประเทศสหรัฐอเมริกา ได้กล่าวถึงลักษณะของโปรแกรมคอมพิวเตอร์ว่า เป็นไปไม่ได้ที่จะให้โปรแกรมคอมพิวเตอร์มีความสมบูรณ์แบบ ความชำรุดบกพร่อง (defect) ของโปรแกรมคอมพิวเตอร์บางประการถือว่ายอมรับได้ ในช่วงปลายทศวรรษ 1990 ซอฟต์แวร์ปฏิบัติการ (operating systems) ซึ่งเป็นที่นิยมสำหรับเครื่องคอมพิวเตอร์ขนาดเล็ก ประกอบไปด้วยคำสั่ง (code or instruction) มากกว่าสิบล้านบรรทัด ในเครื่องคอมพิวเตอร์คำสั่งเหล่านี้ต้องทำงานร่วมกับคำสั่งของซอฟต์แวร์อื่นๆ จำนวนมาก ซึ่งเมื่อเทียบกับสินค้าอุปโภคบริโภคโดยทั่วไปจะมีส่วนประกอบไม่ถึงหนึ่งร้อยชิ้นส่วน และแม้แต่นั่งส้อมโดยทั่วไปจะมีจำนวนคำไม่ถึงหนึ่งแสนห้าหมื่นคำ นอกจากนี้ โปรแกรมคอมพิวเตอร์ส่วนใหญ่ไม่ได้มีส่วนประกอบเพียง

คำสั่งจำนวนมากเท่านั้น แต่ยังต้องทำงานสอดคล้องกันกับคำสั่งของโปรแกรมของบุคคลอื่น (third-party programs) จึงเป็นการไม่สมเหตุสมผลที่จะให้มีการรับประกันว่าซอฟต์แวร์คอมพิวเตอร์ปราศจากข้อผิดพลาด (errors) หรือที่เรียกกันว่า “bugs” การที่ซอฟต์แวร์คอมพิวเตอร์มีข้อผิดพลาดเล็กๆ น้อยๆ จึงเป็นที่ยอมรับได้ในทางการค้าทั่วไป

ปัจจุบัน จึงเป็นที่ยอมรับกันว่าในทุกส่วนของซอฟต์แวร์คอมพิวเตอร์ย่อมมีข้อผิดพลาดเกิดขึ้นได้เสมอ และเป็นไปไม่ได้ที่จะพัฒนาหรือเขียนโปรแกรมคอมพิวเตอร์ที่ปราศจากข้อผิดพลาด (bug) ซึ่งโดยทั่วไปภายหลังจากผู้ผลิตได้รับรายงานว่ามีข้อผิดพลาดจากการใช้งานจากผู้ใช้ในระยะเวลาแล้ว ผู้ผลิตจะทำการแก้ไขและพัฒนาซอฟต์แวร์ในเวอร์ชันใหม่⁴¹ หรืออาจนำซอฟต์แวร์ที่ใช้สำหรับแก้ไขข้อผิดพลาดดังกล่าว (หรือที่เรียกกันว่า patch) ให้ผู้ใช้งานได้ใช้ upgrade ซอฟต์แวร์ที่ตนได้ซื้อไปโดยไม่เสียค่าใช้จ่ายเพิ่มเติม

ข้อผิดพลาด (bug) ของซอฟต์แวร์คอมพิวเตอร์ส่วนใหญ่เกิดขึ้นจากตัวบุคคลในการเขียน source code หรือการออกแบบ และมีส่วนน้อยที่เกิดขึ้นจากโปรแกรมแปลภาษา (compilers) จาก source code ไปเป็น object code เมื่อโปรแกรมมีข้อผิดพลาดจะส่งผลได้แตกต่างกันหลายประเภท ข้อผิดพลาดบางประเภทส่งผลเฉพาะการทำงานของโปรแกรมในรายละเอียดบางประการโดยไม่กระทบต่อการทำงานอย่างมีนัยสำคัญ แต่ข้อผิดพลาดบางอย่างก็ส่งผลให้โปรแกรมหยุดทำงานได้ และหากเป็นกรณีของความผิดพลาดที่เกี่ยวกับระบบรักษาความปลอดภัยของตัวโปรแกรมคอมพิวเตอร์ (security bugs) อาจทำให้ผู้ไม่ประสงค์ดีสามารถผ่านระบบกลไกควบคุมการผ่านเข้าใช้งานโปรแกรมคอมพิวเตอร์หรือเข้าถึงข้อมูลที่หวงกันไว้ได้⁴²

บางกรณี ข้อผิดพลาดของซอฟต์แวร์คอมพิวเตอร์อาจก่อให้เกิดความเสียหายร้ายแรงต่อบุคคลได้ เช่น กรณีเครื่องบำบัดด้วยการฉายแสง Therac-25 ที่ซอฟต์แวร์ทำงานผิดพลาดทำให้ผู้ป่วยได้รับรังสีเกินกว่าขนาดที่ต้องการ ในปี 1985 – 1987 มีผู้ป่วยได้รับความเสียหาย 6 ราย โดย 3 รายได้รับอันตรายถึงขั้นเสียชีวิต โดยคดีทั้งหมดได้มีการตกลง

⁴¹ Ian J. Lloyd, *supra* note 8, p. 571.

⁴² Retrieved from http://en.wikipedia.org/wiki/Software_bugs, March 10, 2010.

ประนีประนอมกันนอกศาล⁴³ หรือในกรณีของเฮลิคอปเตอร์ Chinook ZD 576 ได้ประสบอุบัติเหตุชนกับ Mull of Kintyre ในปี ค.ศ. 1994 ซึ่ง House of Lords เชื่อว่าอุบัติเหตุเกิดขึ้นจากความผิดพลาดของซอฟต์แวร์คอมพิวเตอร์ส่วนที่ควบคุมเครื่องยนต์⁴⁴

ในปี ค.ศ. 2002 จากการศึกษาของ US Department of Commerce's National Institute of Standards and Technology พบว่า ความผิดพลาดของซอฟต์แวร์ (software bugs) นั้นส่งผลกระทบต่อต้นทุนทางเศรษฐกิจของสหรัฐอเมริกาเป็นมูลค่าประมาณ 59,000 ล้านดอลลาร์สหรัฐฯ และได้เรียกร้องให้ผู้ผลิตซอฟต์แวร์พัฒนาระบบการตรวจสอบซอฟต์แวร์ (software testing)⁴⁵

เนื่องจากซอฟต์แวร์คอมพิวเตอร์มีลักษณะเฉพาะ ซึ่งแตกต่างจากสินค้าอื่นดังที่ได้กล่าวมาข้างต้น ในทางทฤษฎีกฎหมายที่เกี่ยวข้องและทางปฏิบัติในทางการค้าเกี่ยวกับความรับผิดชอบของผู้พัฒนาซอฟต์แวร์คอมพิวเตอร์นั้น ไม่ควรพิจารณาความชำรุดบกพร่องหรือข้อผิดพลาดในทางภาวะวิสัยของตัวซอฟต์แวร์คอมพิวเตอร์เป็นหลัก แต่ควรมุ่งเน้นในเรื่องหน้าที่ของผู้พัฒนาซอฟต์แวร์หรือผู้ประกอบการในทางปฏิบัติว่าได้มีการทดสอบซอฟต์แวร์คอมพิวเตอร์ด้วยความพยายามตามสมควรก่อนนำออกสู่ท้องตลาด เท่าที่ความรู้ในอุตสาหกรรมซอฟต์แวร์ในขณะนั้นมีอยู่หรือไม่ และผู้พัฒนาหรือผู้ผลิตซอฟต์แวร์ดังกล่าวมีความรับผิดชอบภายหลังจากการที่ซอฟต์แวร์ออกสู่ท้องตลาดอย่างไร เช่น การติดตามผลการใช้งานของผู้ใช้ (users) ความพยายามในการแก้ไขข้อผิดพลาด “correctness” วิธีการในการลดระยะเวลาและความเสียหายภายหลังจากที่พบข้อผิดพลาดแล้ว เป็นต้น

⁴³ Retrieved from

http://computingcases.org/case_materials/therac/case_history/Case%20History.html,
March 10, 2010.

⁴⁴ Retrieved from <http://www.parliament.the-stationery-office.co.uk/pa/ld200102/ldselect/ldchin/25/1101606.htm>, March 10, 2010.

⁴⁵ Retrieved from http://www.nist.gov/public_affairs/releases/n02-10.htm,
November 12, 2009.