

## Agile

Agile software development เป็นแนวทางในการพัฒนาซอฟต์แวร์ที่เน้นเรื่อง

- การให้ความสำคัญกับตัวผู้ทำงานและการมีปฏิสัมพันธ์ระหว่างผู้ทำงาน มากกว่าขั้นตอนวิธีการหรือเครื่องมือ (Individuals and interactions over processes and tools)
- การสร้างซอฟต์แวร์ที่สามารถนำไปใช้งานได้จริง มากกว่าเอกสาร (Working software over comprehensive documentation)
- การทำงานร่วมกับลูกค้า มากกว่าการต่อรองสัญญากับลูกค้า (Customer collaboration over contract negotiation)
- การยอมรับปรับซอฟต์แวร์ตามความเปลี่ยนแปลงของความต้องการของลูกค้า มากกว่าการทำตามแผนการเพียงอย่างเดียว (Responding to change over following a plan)

ทั้งนี้ เรายังคงให้ความสำคัญต่อขั้นตอนวิธีการ เอกสาร สัญญา และแผนการดำเนินงานอยู่ เพียงแต่ ปฏิสัมพันธ์ระหว่างผู้ร่วมงาน ซอฟต์แวร์ที่ใช้ได้จริง การติดต่อกับลูกค้า และการปรับตามความเปลี่ยนแปลงนั้นมีความสำคัญยิ่งกว่า (Agile Alliance, 2001) [1] [2] [3]

## Agile Manifesto (แนวทางของ Agile)

จากหัวข้อที่แล้ว เราทราบว่าหัวใจหลักของ Agile นั้นเกิดจากแนวทาง 4 ข้อด้วยกัน ในหัวข้อนี้จึงจะกล่าวถึงความหมายของแนวทางแต่ละข้อนั้น

### Individuals and interactions over processes and tools

ในองค์กรหรือหน่วยงานส่วนใหญ่ นั้น มักสรรหากระบวนการหรือเครื่องมือราคาแพงที่จะคอยควบคุมคนให้ทำงานได้อย่างที่ต้องการ ซึ่งบ่อยครั้งเครื่องมือหรือกระบวนการเหล่านี้อาจไม่ได้มีความเหมาะสมต่อการนำมาใช้งานเลย เช่น หากต้องการแก้ไขโค้ดโปรแกรมชุดหนึ่งที่ทำอยู่กับนักพัฒนาอีกคน แทนที่จะใช้เครื่องมือสำหรับ review โค้ด โปรแกรมที่ถูกออกแบบมาให้สามารถดูได้ในทุกสถานที่ผ่านอินเทอร์เน็ต (ซึ่งไม่มีประโยชน์เลยหากนักพัฒนาที่ทำงานด้วยกันนั่งอยู่ในห้องเดียวกัน) เพียงแค่คุยกันหรือโทรศัพท์ติดต่อกัน (interact) แทนการใช้เครื่องมือยุ่งยากดังกล่าว (tool) ก็สามารถแก้ปัญหาได้อย่างรวดเร็วแล้ว

อีกหนึ่งปัจจัยที่จะทำให้ซอฟต์แวร์นั้นออกมาดี คือ แรงจูงใจในการทำงานของนักพัฒนา เมื่อใดที่นักพัฒนามีความสนุกกับงานที่ได้ทำ เมื่อนั้นผลงานก็จะออกมาดี แต่เมื่อใดที่นักพัฒนาไม่สามารถมีความสุขกับงานที่ได้ทำ คุณภาพของงานที่ออกมาก็มักจะลดลงไปด้วย ดังนั้น แทนที่จะใช้กระบวนการในการบังคับให้นักพัฒนาทำตามขั้นตอน ก็ควรจะหาวิธีให้นักพัฒนาได้ตระหนักถึงหน้าที่ของตน และอยากจะทำมันออกมาให้ดีด้วยตนเอง

### Working software over comprehensive documentation

ลูกค้าต้องการซอฟต์แวร์ที่สามารถใช้งานได้ มากกว่าเอกสารหรือคู่มือที่สมบูรณ์ (ที่ลูกค้ามักจะไม่ได้อ่าน) นั่นคือ ต่อให้สร้างเอกสารที่อ่านเข้าใจง่าย ครบถ้วนและสมบูรณ์แบบขึ้นมาก็ได้ แต่ซอฟต์แวร์กลับทำงานไม่ได้ตามที่ลูกค้าต้องการ เอกสารเหล่านี้ก็จะหมดค่าลงทันที

แต่เอกสารนั้นยังคงมีความจำเป็นอยู่ ทั้งสำหรับการพัฒนาและดูแลระบบต่อไปในอนาคต รวมไปถึงการใช้เป็นหลักฐานในการติดต่อกับลูกค้า ดังนั้น จึงต้องพัฒนาซอฟต์แวร์ที่สามารถตอบสนองความต้องการของลูกค้าได้ขึ้นมา ก่อน แล้วจึงค่อยสร้างเอกสารที่มีเนื้อหาเพียงพอต่อการพัฒนาระบบต่อไปในอนาคตขึ้นมา

### Customer collaboration over contract negotiation

ในการพัฒนาซอฟต์แวร์ดั้งเดิมนั้น ลูกค้าจะมีบทบาทอยู่สองช่วง ได้แก่ช่วงแรกที่ทำหน้าที่เป็นผู้ให้ requirement และช่วงสุดท้ายที่จะรับส่งมอบงาน โดยที่ในระหว่างขั้นตอนการพัฒนาซอฟต์แวร์นั้น ลูกค้าแทบจะไม่ได้มีบทบาทใดๆเลย และเนื่องจากความต้องการของลูกค้านั้นมีการเปลี่ยนแปลงอยู่เสมอ ประกอบกับการที่บางครั้งลูกค้าเองยังไม่ทราบด้วยซ้ำว่าตนเองต้องการอะไร ซึ่งหากเราดันทุรังทำตาม requirement ที่ได้รับมาในตอนต้น ผลลัพธ์ที่ได้คือ ซอฟต์แวร์ที่ไม่สามารถทำงานได้ตรงตามความต้องการของลูกค้า

การแก้ปัญหานี้คือ ให้ลูกค้ามีส่วนร่วมในทุกๆขั้นตอนของการพัฒนา พนักงานพัฒนาเริ่มพัฒนาโปรแกรมที่สามารถทำงานได้แล้วในระดับหนึ่ง อาจจะเพียง 5% หรือ 10% ของความสามารถทั้งหมด ก็ให้เริ่มนำเสนอต่อลูกค้าในทันที เพื่อให้ลูกค้าเกิดไอเดียว่านี่คือสิ่งที่ตนต้องการจริงหรือเปล่า หากไม่ใช่ ทางทีมพัฒนาจะได้เตรียมรับมือหาแนวทางแก้ไขกันต่อไป โดยในท้ายที่สุดแล้ว ซอฟต์แวร์ที่เกิดขึ้นมาจะทำงานได้ตรงตามความต้องการของลูกค้ามากที่สุด

### Responding to change over following a plan

สืบเนื่องจากข้อที่แล้ว คือ ความต้องการที่เปลี่ยนแปลงอยู่ตลอดเวลาของลูกค้า ที่บางครั้งอาจทำให้เกิดความขัดแย้งกับแผนงานที่ได้เตรียมไว้ การยอมให้เกิดความเปลี่ยนแปลงในแผนที่ได้วางไว้ถือเป็นอีกสิ่งหนึ่งที่จะช่วยให้ซอฟต์แวร์ที่ออกมา นั้น ไม่กลายเป็นชิ้นงานที่ไร้คุณค่า เนื่องจากลูกค้าไม่สามารถนำไปใช้งานได้

### Principles behind the Agile Manifesto

นอกจากแนวทางทั้ง 4 ข้อนี้แล้ว Agile ยังมีหลักในการปฏิบัติอีก 12 ข้อที่จะช่วยให้สามารถนำ Agile เข้าไปประยุกต์ใช้ได้อย่างเป็นรูปธรรม ได้แก่

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.  
หมายถึง สิ่งที่สำคัญที่สุดคือการทำให้ลูกค้าเกิดความพอใจด้วยการส่งมอบซอฟต์แวร์ที่สามารถใช้งานได้จริง ตั้งแต่เนิ่นๆ และสม่ำเสมอ
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.  
หมายถึง ยอมรับในการแก้ไข requirement แม้ว่าจะจะเป็นช่วงท้ายๆของการพัฒนา โดยการยอมรับการเปลี่ยนแปลงนี้ก็เพื่อให้ลูกค้ามีความได้เปรียบในการแข่งขัน
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

หมายถึง ส่งงานที่สามารถใช้ได้จริงให้บ่อยที่สุด อย่าเว้นระยะห่างนานจนเกินไป เน้นการทำงานที่อยู่ตรงหน้า ให้เสร็จ ไม่ต้องคิดเผื่อไปไกล

- Business people and developers must work together daily throughout the project.  
หมายถึง ต้องให้ลูกค้าหรือผู้ใช้งานมาทำงานร่วมกับนักพัฒนาให้ได้ เพื่อให้ นักพัฒนาสามารถสร้างซอฟต์แวร์ที่ตรงตามความต้องการของลูกค้าได้ ทั้งนี้ หากไม่สามารถทำงานกับลูกค้าได้ตลอด อาจต้องหาบุคคลที่เข้าใจในความต้องการของลูกค้ามาทำหน้าที่เป็นตัวแทนได้
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.  
หมายถึง project ควรจะถูกสร้างขึ้นโดยคนที่มีความกระตือรือร้นอยากจะทำให้งานออกมาดี จึงจำเป็นต้องสร้างสภาพแวดล้อมที่เหมาะสมให้แก่ นักพัฒนา ให้ความช่วยเหลือเท่าที่นักพัฒนาต้องการ และเชื่อมั่นในตัวนักพัฒนาว่า จะสามารถสร้างผลงานออกมาได้
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.  
หมายถึง วิธีการแลกเปลี่ยนข้อมูลภายในทีมที่ดีที่สุดคือ การพูดคุยกันตัวต่อตัว แทนการใช้เครื่องมือหรือเอกสารที่อาจทำให้เนื้อความที่ต้องการสื่อเกิดการตกหล่นและล่าช้าได้
- Working software is the primary measure of progress.  
หมายถึง การใช้ซอฟต์แวร์ที่ทำงานได้จริงเป็นตัวชี้วัดหลักเพื่อวัดความคืบหน้าของงาน
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.  
หมายถึง Agile สนับสนุนการพัฒนาแบบยั่งยืน นั่นคือ จะไม่มีการทำงานแบบสบายเกินไป (ในตอนต้น project) หรือเร่งรีบจนเกินไป (ในตอนท้าย project) แต่จะให้ทุกๆช่วงของการพัฒนานั้นสามารถรักษาระดับความเร็วไว้ได้คงที่ และการจะทำเช่นนี้ได้ ทุกฝ่ายที่เกี่ยวข้องจะต้องให้ความช่วยเหลือด้วย เช่น ถ้าลูกค้าไม่ให้ความร่วมมือในตอนเริ่ม project ท้ายที่สุดแล้ว งานส่วนใหญ่ก็จะถูกกระจุกไว้ตอนท้ายเช่นเดิม
- Continuous attention to technical excellence and good design enhances agility.  
หมายถึง ให้ปรับปรุงความรู้ด้านเทคนิคให้ดีเยี่ยมอยู่ตลอดเวลา และการใช้ความรู้นั้นร่วมกับการออกแบบที่ดี จะช่วยให้ซอฟต์แวร์มีความยืดหยุ่นมากยิ่งขึ้น ซึ่งเป็นการช่วยให้การแก้ไขซอฟต์แวร์เป็นไปได้อย่างง่าย
- Simplicity—the art of maximizing the amount of work not done—is essential.  
หรือ Simplicity is essential. หมายถึง ไม่จำเป็นต้องคิดอะไรให้ไกลหรือซับซ้อนมากจนเกินไป ทำงานที่อยู่ข้างตรงหน้าด้วยวิธีการที่ง่ายที่สุดก่อน แล้วถ้ามันยังทำงานได้ไม่ดีพอ งานนั้นจะถูกนำมาแก้ไขในภายหลังเอง
- The best architectures, requirements, and designs emerge from self-organizing teams.  
หมายถึง รูปแบบงานที่ดีที่สุดนั้น จะได้มาจากทีมที่สมาชิกทุกคนมีความรับผิดชอบในหน้าที่ของตนเอง รู้ว่า

ตนเองต้องทำอะไรโดยไม่จำเป็นต้องให้ใครคนใดคนหนึ่งบอก พยายามช่วยกันหาวิธีแก้ปัญหาที่เกิดขึ้นโดยไม่  
ต้องรอความช่วยเหลือจากใคร

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

หมายถึง มีการแลกเปลี่ยนความเห็นจากประสบการณ์ในการทำงานที่ผ่านมาว่า มีวิธีใดที่จะช่วยเพิ่ม  
ประสิทธิภาพในการทำงานได้ และปรับเปลี่ยนพฤติกรรมการทำงานไปตามนั้น และทำอย่างสม่ำเสมอ