**1.ตอบ** ไม่เห็นด้วยอย่างยิ่ง

Agile ไม่มีอะไรที่เหมือนกับ Waterfall เลย Waterfall เป็นวิธีการพัฒนาระบบที่ใช้กันอยู่ค่อนข้างแพร่หลาย ซึ่งวิธีนี้จะมีลำดับขั้นตอนที่ตายตัว เริ่มตั้งแต่รวบรวมข้อมูล กำหนดความต้องการของผู้ใช้ วิเคราะห์ทางเลือก ออกแบบ เขียนโปรแกรม ทดสอบระบบ และสุดท้ายทำการติดตั้งระบบ โดยแต่ละส่วนของขั้นตอนดังกล่าวจะถือเป็นตัววัดความก้าวหน้าของงาน แต่ปัญหาสำคัญของ Waterfall คือขั้นตอนของการพัฒนาที่ไม่ยืดหยุ่น เพราะตัวงานจะแบ่งเป็นช่วงๆแบบตายตัว ทำให้มีข้อผูกมัดตั้งแต่เริ่มโครงงานและไม่สามารปรับเปลี่ยนความต้องการผู้ ใช้ได้ หมายความว่าการพัฒนาโดยใช้ Waterfall นั้น ไม่เหมาะกับงานที่ความต้องการของผู้ใช้เข้าใจยาก และมีแนวโน้มว่าจะเปลี่ยนแปลงตลอดเวลา

ในทางกลับกัน Agile จะแบ่งงานออกเป็นส่วนย่อยๆ แล้วค่อยๆ ทดสอบไปเรื่อยๆทุกสัปดาห์หรือทุกสองสัปดาห์ ทั้งนี้จะเน้นสร้างส่วนย่อยที่สุดของงานทั้งหมดที่สามารถใช้งานได้ทีละชิ้น เพื่อให้ส่งมอบได้รวดเร็วและทำการปรับปรุงเพิ่มเติมอย่างต่อเนื่องตลอดช่วงเวลาของโครงงาน Agile จะเน้นการติดต่อสื่อสาร ความง่าย ไม่ซับซ้อน feedback จากลูกค้า ความกลัาตัดสินใจ เน้นความเคารพกันและกัน

**2.ตอบ** เห็นด้วยอย่างยิ่ง

Git และ Github เป็น version control แบบ distributed คือ หลังจากที่ clone remote repo. มาแล้ว ก็เท่ากับว่าเรามี local repo. ที่เหมือนกันกับ remote repo. ด้วย จะ commit ก็สามารถทำได้แบบ offline เลยเพราะเป็นการ commit ไปที่ local repo. แต่ถ้าเราจะ push ไปไว้ที่ remote repo. แน่นอนว่าต้องเชื่อมต่อ network

ในขณะที่ CVS หรือ SVN ที่เป็น version control แบบ centralized การทำงานทุกอย่างจะถูกเก็บในตัวกลาง ซึ่งก็คือ remote repo. ทั้งหมด ทำใช้งานจำเป็นต้องเชื่อมต่อ network ไปยัง repo. ไม่เช่นนั้นก็จะ commit หรือ check revision ก่อนๆ ไม่ได้เลย changes ของ file ทั้งหมดถูกเก็บอยู่ที่ central repo หรือศูนย์กลางที่เดียว ถ้าสมมติว่า central repo เกิดพังขึ้นมาหรือผู้ใช้ไม่สามารถติดต่อกับ central repo ได้ก็จะทำให้ไม่สามารถดึงไฟล์ที่พึ่งเปลี่ยนแปลงมาทีตัวเองได้

ส่วน distributed version control ที่ใช้ใน Gid และ Github นั้น แต่ละคนจะมี copy ของไฟล์ตั้งแต่เริ่มแรกจนท้ายสุด ดังนั้นแต่ละทีมหรือแต่ละคนสามารถ maintain code ได้โดยที่ทุกคนมี copy ของไฟล์ทั้งหมดอยู่ที่ local computer และ change set ของแต่ละ version ที่เกิดขึ้นจากหลายๆ ทีมนั้นสามารถรวมกันได้ ไม่จำเป็นต้องรอ changes copy จาก central repo อีกต่อไป

ดังนั้น distributed version control จึงเร็วกว่า และผู้ใช้งานก็สามารถแก้ได้ทุกที่ โดยไม่จำเป็นจะต้องติดต่อกับ central repo ก็ได้

**3.ตอบ**

1. git checkout --orphan feature1

2. git add .

3. git commit -m " feature1"

4. git push origin feature1

**4.ตอบ** เห็นด้วยอย่างยิ่ง

Merge คือ การรวมไฟล์ จาก branch หนึ่งกับอีก branch หนึ่งเข้าด้วยกัน การ conflict เกิดจากจำนวน source code ที่ชนหรือขัดแย้งกันมากเหลือเกิน เหตุการณ์นี้จะเกิดขึ้นกับโปรเจ็กที่ร่วมกันทำหลายคนเท่านั้น ทำอยู่คนเดียวมันไม่เกิด

**6.ตอบ**  ไม่เห็นด้วยอย่างยิ่ง

แม้จุดประสงค์ของ Web Application กับ Software นั้น จะถูกสร้างมาเพื่อตอบสนองการใช้งานในรูปแบบต่างๆ บนแต่ละอุปกรณ์ แต่แอพและโปรแกรมก็มีความแตกต่างกันในหลายๆ ด้านดังนี้

**1. ในส่วนของการติดตั้ง**

ในอดีต เรามักจะคุ้นเคยกับการดาวน์โหลดโปรแกรม โดยอาจจะโหลดผ่านอินเทอร์เน็ตหรือติดตั้งผ่านแผ่นซีดี แต่การติดตั้ง App นั้นสามารถทำได้ง่ายดายกว่านั้น

หากคุณเป็นผู้ใช้งานมือถืออย่าง iPhone หรือแทบเล็ต iPad คุณสามารถดาวน์โหลดและติดตั้งแอพต่างๆ ได้อย่างง่ายดายผ่านทาง Appstore

ส่วนผู้ใช้งานมือถือหรือแทบเล็ตที่ใช้ระบบปฎิบัติการ Android คุณสามารถดาวน์โหลดและติดตั้งแอพต่างๆ ผ่านทาง Google Play Storeซึ่งจะเห็นได้ว่า การดาวน์โหลดและติดตั้งแอพนั้น ก็สามารถทำได้ง่ายดายกว่าเยอะ และหากคุณไม่ชอบใจแอพนั้น คุณก็สามารถถอนการติดตั้งได้ง่ายๆ ด้วยตัวเอง

**2. ในส่วนของการใช้งาน**

เนื่องจากอุปกรณ์ต่างๆ ที่ใช้ในมือถือหรือแทบเล็ตนั้นมีความสามารถด้อยกว่าอุปกรณ์ต่างๆ ที่ใช้ในคอมพิวเตอร์หรือโน้ตบุ้ค เนื่องจากต้องถูกจำกัดด้วยขนาด หรือการประหยัดพลังงาน หากใช้โปรแกรมที่มีความซับซ้อนหรือใช้ทรัพยากรเครื่องสูงอย่างคอมพิวเตอร์ ปกติแล้ว อาจจะทำให้มือถือหรือแทบเล็ตนั้นค้าง หรือไม่สามารถทำงานได้เลย หากทำงานได้ก็จะช้ามากหรือใช้แบตเตอรี่มากเกินไป

ทำให้แอพถูกเขียนขึ้นมามีความไม่ซับซ้อนมาก ใช้งานง่าย และตอบสนองความต้องการพื้นฐานของผู้ใช้งานเฉพาะด้าน และแอพจะถูกสร้างมาเฉพาะกับระบบปฎิบัติการบนมือถือหรือแทบเล็ตนั้นๆ เช่น Android หรือ iOS ครับดังนั้น เราจึงไม่สามารถเอาโปรแกรมที่ใช้บนคอมพิวเตอร์มาใช้งานบนแอพได้ครับ และกลับกัน แอพก็ไม่สามารถใช้งานบนคอมพิวเตอร์ได้

**3. ในส่วนของค่าใช้จ่าย**

ไม่ว่าจะเป็นทั้งแอพและโปรแกรม ก็จะเหมือนกันในส่วนของค่าใช้จ่าย มีทั้งแบบฟรี และแบบเสียเงิน แต่ที่แตกต่างกันในส่วนนี้ก็คือ ราคาของแอพนั้น ส่วนมากจะมีราคาถูกกว่าโปรแกรมหลายเท่า เนื่องจากแอพถูกสร้างขึ้นมากเพื่อการใช้งานที่ไม่ซับซ้อนเกินไป การพัฒนาของผู้ผลิตนั้นสามารถทำได้ง่ายกว่า ทำให้ราคานั้นถูกกว่า

**4. ในส่วนของการอัพเดทเป็นเวอร์ชั่นใหม่**

แอพส่วนใหญ่นั้นเมื่อมีการอัพเดทเวอร์ชั่น จะมีความถี่มากกว่าการอัพเดทโปรแกรม โดยส่วนมากการอัพเดทของแอพนั้นมักจะทำเพื่อปรับปรุงข้อผิดพลาด หรือเพิ่มเติมหรือแก้ไขจากเวอร์ชั่นก่อนหน้านี้ แต่โปรแกรมส่วนใหญ่มักจะเป็นการอัพเดทถี่น้อยกว่า แต่เวอร์ชั่นใหม่จะมีความแตกต่างจากเดิมมาก

**7.ตอบ**

1.Browser จะออกคำขอไปยัง /users

2. Browser ส่งต่อไป index ที่อยู่ใน Rails router ซึ่งเป็นไฟล์สำหรับโหลด Controller ให้ทำงาน

3. Controller ติดต่อกับ Model

4. Model ก็จะมีตัว Active Record สำหรับจัดการทุกอย่างที่เกียวข้องกับ Database

5.สุดท้ายก็จะส่งข้อมูลที่ได้จาก Model กลับไปให้ Controller

6. Controller ส่งผลลัพธ์กลับไปยังผู้ใช้ผ่าน View

7. View ส่งข้อมูลกลับมายัง browser เพื่อแสดงผลกับผู้ใช้งาน การแสดงผลหน้าเป็น HTML

8. Controller จะส่ง HTML กลับมายัง Browser

**8.ตอบ**

Grails เป็นเฟรมเวิร์กที่ได้รับแรงบันดาลใจจาก Rails แต่เกิดมาเพื่อ Groovy ที่ทำงานบนจาวาแพลตฟอร์ม แม้ Grails ยังใหม่มากเมื่อเทียบกับ Rails ทั้งในแง่ของอายุ และเรื่องความใส่ใจด้านสมรรถภาพการทำงาน แต่ทีมงาน Grails ก็หาญกล้าอยากวัดรอยเท้ากับผู้จุดประกายของตัวเอง

การทดสอบเป็นการวัดสมรรถภาพของการทำงาน CRUD ของเฟรมเวิร์ก ในแง่ของความเร็วและความสามารถในการให้บริการ โดยใช้การยิง request พร้อมๆ กันทีละ 50 ตัวเข้าไปจนครบ 1000 ตัวเพื่อดูความเร็วการตอบสนอง

Rails มีการทำ load balance ไปที่ Mongrels 10 ตัวผ่าน Pound ส่วน Grails ขอแค่ Tomcat ตัวเดียวที่มี Servlet Thread 10 ตัวพอ

ผลออกมาเดากันถูกอยู่แล้ว แม้ Rails จะทำได้ดีกว่าหากเทียบความเร็ว request เดี่ยวๆ แต่ถ้ามองประเด็นการให้บริการหนักๆ พร้อมๆ กัน Rails จะมีอาการเป๋พอสมควร ต่างจาก Grails ที่ดูแล้วสเกลได้ดีกว่าเยอะ

**9.ตอบ**

Heroku คือผู้ให้บริการ Platform as a Services (PaaS) เช่น แอพพลิเคชันเซิร์ฟเวอร์, ดาต้าเบสเซิร์ฟเวอร์ หรือมิดเดิลแวร์อื่นๆ ไม่ต้องเสียเวลาหา software ไม่ต้องหา server และลดความยุ่งยากในการ configuration เพราะเพียงแค่คลิกเลือกภาษาที่ต้องการสร้าง app ไม่ถึงนาทีเราก็มี environment พร้อมใช้งาน ที่สำคัญฟรี

บทบาทการพัฒนา web application สามารถสร้าง buildpack สำหรับภาษาอื่นๆได้ เช่น Lua ที่รันอยู่บน OpenResty ได้อีกด้วย สามารถ scale ให้รองรับผู้ใช้เยอะๆได้โดยง่าย นอกจากรองรับภาษาโปรแกรมที่หลากหลายแล้ว ตัว Heroku มี App Store ของมันด้วยเรียกว่า add-ons สำหรับเพิ่มเติมบริการอื่นๆเข้าไปในแอปของเรา เช่น PostgreSQL, MongoDB, Redis เป็นต้น ซึ่งก็มีทั้งฟรี และไม่ฟรีให้เลือกใช้งาน

**10.ตอบ**

เพราะเป็นการศึกษาเกี่ยวกับทฤษฎีสารสนเทศ ตั้งแต่การรับรู้, การทำความเข้าใจ, การวิเคราะห์, การสังเคราะห์, การเก็บ, การค้นคืน, การสื่อสาร สารสนเทศอย่างเป็นระบบ ในการศึกษาด้านวิทยาการสารสนเทศนั้น มีความจำเป็นต้องศึกษาวิชาในสาขาวิทยาการคอมพิวเตอร์ด้วยเนื่องจากใช้คอมพิวเตอร์เป็นเครื่องมือสำหรับประมวลผลสารสนเทศ

วิทยาการสารสนเทศนั้นยังสนใจกระบวนความคิดและสารสนเทศในเชิงประยุกต์ด้วย เช่น เรื่องของสารสนเทศในสิ่งมีชีวิตในประชานศาสตร์ (cognitive science) ซึ่งศึกษากระบวนความคิดและการทำงานของสมองของสิ่งมีชีวิต ตัวอย่างการประยุกต์วิทยาการสารสนเทศ อื่น ๆ เช่น ชีวสารสนเทศศาสตร์ ในกรณีนี้คือการศึกษาสารสนเทศทางชีววิทยา หรือ การศึกษาด้านวิทยาการสารสนเทศที่เกี่ยวกับการส่งสารสนเทศไปยังผู้รับสารด้วย ซึ่งคือบางส่วนของนิเทศศาสตร์