🡺1.จงวิจารณ์คำกล่าวอ้างต่อไปนี้

Waterfall

* เป็นแนวทางการพัฒนาซอฟต์แวร์ที่มีความเป็นระบบสูงและเป็นลำดับชัดเจน โดยจะเริ่มจากความต้องการซอฟต์แวร์ของลูกค้า ต่อเนื่องไปถึงการวางแผน การออกแบบ การสร้าง และการนำไปใช้ซึ่งเหมาะกับสถานการณ์ที่มีการเพิ่มความสามารถให้กับระบบที่มีอยู่แล้ว
* หรือถ้าเป็นการสร้างฟังก์ชั่นการทำงานใหม่ก็จะเป็นการพัฒนาเพิ่มแบบไม่มากนัก มีความต้องการที่ชัดเจนและไม่เปลี่ยนแปลง

Agile

* สำหรับแนวทางการพัฒนาแบบอาไจล์จะเน้นการวิเคราะห์และออกแบบให้พอเหมาะเพื่อให้ซอฟต์แวร์สามารถส่งมอบได้ทันเวลาและมีการสื่อสารกันอย่างต่อเนื่องระหว่างลูกค้าและนักพัฒนา

🡺2.จงวิจารณ์คำกล่าวอ้างต่อไปนี้

Centralized

* Centralized Version Control Systems (CVCSs) หรือระบบ Version Control Systems แบบรวมศูนย์
* ทุกคนสามารถรู้ได้ว่าคนอื่นในโปรเจคกำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง
* คนสามารถต่อเข้ามาเพื่อดึงไฟล์จากศูนย์กลางนี้ไปแก้ไขได้
* จุดอ่อน : ตรงที่การรวมศูนย์ทำให้มันเป็นจุดอ่อนจุดเดียวที่จะล่มได้เหมือนกันเพราะทุกอย่างรวมกันอยู่ที่เซิร์ฟเวอร์ที่เดียว ถ้าเซิร์ฟเวอร์นั้นล่มซักชั่วโมงหนึ่ง หมายความว่าในชั่วโมงนั้นไม่มีใครสามารถทำงานร่วมกันหรือบันทึกการเปลี่ยนแปลงงานที่กำลังทำอยู่ไปที่เซิร์ฟเวอร์ได้เลย

Distributed

* Distributed Version Control Systems (DVCSs) หรือระบบ VCS แบบกระจายศูนย์
* แต่ละคนไม่เพียงได้ก๊อปปี้ล่าสุดของไฟล์เท่านั้น แต่ได้ทั้งก๊อปปี้ของ repository
* หมายความว่าถึงแม้ว่าเซิร์ฟเวอร์จะเสีย client ก็ยังสามารถทำงานร่วมกันได้ต่อไป และ repository เหล่านี้ของ client ยังสามารถถูกก๊อปปี้กลับไปที่เซิร์ฟเวอร์เพื่อกูข้อมูลกลับคืนก็ได้
* ระบบนี้ยังทำงานกับหลาย ๆ repository ได้อย่างดี ทำให้สามารถทำงานกับคนหลายกลุ่มซึ่งทำงานในรูปแบบต่างกันในโปรเจคเดียวกันได้อย่างง่ายดาย

🡺3. - git init

- git add -A

- git commit -m "feature1"

- git push origin master

🡺4. การแก้ไข conflict

* เปิดไฟล์ที่มีปัญหาขึ้นมา ลบโค๊ดส่วนเกินออก แล้วแก้ไขใหม่ให้เรียบร้อย จากนั้นลองเช็คสถานะเมื่อมั่นใจว่าโปรแกรมทำงานถูกต้องแล้ว ก็สั่ง commit และ push เป็นอันเสร็จเรียบร้อย

🡺5. abcde

🡺6. ด้วยเทคโนโลยีปัจจุบันยังสามารถประยุกต์เพิ่มเติมได้ไปถึงการตั้ง web server ใช้ภายในหน่วยงาน และให้ภายนอกเรียกใช้งานเว็บแอพพลิเคชั่นผ่านทาง Internet ได้อีกด้วย ทำให้ไม่ว่าจะเรียกใช้งานจากช่องทางไหนข้อมูลจะถูกบันทึกหรือนำเสนอจากที่ที่เดียวกัน การ Update ข้อมูลจะรวดเร็ว ซึ่งการทำระบบแบบนี้มีค่าใช้จ่ายไม่มากเลย เมื่อเทียบกับความต้องการทางธุรกิจ ที่มีการแข่งขันสูง

🡺7. Ruby on rails เป็น framework ที่ใช้สำหรับสร้า่ง web application

มีโครงสร้างแบบ MVC มาจาก

- Model -> โปรแกรมที่ทำหน้าที่คำนวณงานของ เว็บ app

- View -> โปรแกรมที่ทำหน้าที่ติดต่อสื่อสารกับผู้ใช้

- Controller ->โปรแกรมที่ควบคุมการทำงานระหว่าง model กับ view

โดยขั้นตอนการทำงานมีดังนี้คือ

* เริ่มจาก client ส่ง request ไปที่ web app โดยส่งต่อมาที่ controller
* ก่อนหน้านั้นจะต้องถูก ActionPack(ActionView->module สร้า่ง view + ActionController->module สร้างcontroller)
* ตีความของ request path ที่ user ส่งเข้ามาแล้วทำการโยนไปให้ action ในcontroller ซึ่งส่วนใหญ่แล้ว action ใน controller จะเรียก ไปยัง Model
* ซึ่งมีไว้สำหรับการคำนวณหรือติดต่อไปยัง database query ข้อมูลออกมา
* เมื่อโมเดลทำเสร็จจะส่้งผลกลับมาที่controller แล้วทำการ render view template(เป็นรูปแบบของViewซึ่งเป็น code ruby ที่จะใช้สำหรับสร้าง view (html page)เพื่อส่งกลับมาให้user)

🡺8. Framework

กรอบของการทำงาน แปลตรงตัวเลยครับ Frame + work แต่ถ้ามองในแง่โปรแกรม ก็จะเป็น ข้อกำหนดหรือขอบเขตในการเขียนโปรแกรมให้เป็นไปในรูปแบบเดียวกัน และสามารถนำไปเขียนเพิ่มเพื่อตอบสนองความต้องการอย่างใดอย่างหนึ่งได้ โดยที่เราไม่ต้องไปเสียเวลาเขียนขึ้นมาเอง ซึ่งจะทำให้เกิดความรวดเร็วในการพัฒนา Web Application

Rails

เป็นอันดับหนึ่งของการเผยแพร่ ผ่าน Ruby Gems (เป็นตัวจัดการ package ในภาษา

Ruby ที่จะดูในเรื่องของ รูปแบบของ packaging และมี Libraries ไว้ให้ใช้มากมาย

ข้อดีของ Framework

* เป็น MVC Framework
* มีระบบ template
* ลดการเขียน code ที่ซ้ำไปซ้ำมา เพราะว่ามีการเขียนเป็น class , function
* มีการวางโครงสร้างไฟล์ ไว้อย่างเป็นระบบ ระเบียบที่แน่นอน

ข้อดีของRails

* Rails เป็น Web Framework มีลักษณะเป็น MVC (Model-View-Controller)
* Rails ถูกออกแบบมาให้มีการใช้งานที่ง่ายและรวดเร็ว ลดปัญหางานที่ต้องทำซ้ำๆ ทำให้ได้ productivityที่สูงขึ้น สอดคล้องกับ Methodology แบบ Agile

ข้อเปรียบเทียบ

การทำงาน **Framework** โดยเราไม่ต้องเขียนโค้ดเองทั้งหมด ทั้งการสร้าง Application กับ ทำงานกับ Database ภาษาในการเขียนก็ใช้ง่ายอ่านง่าย หรือ View ที่มีการซับพอรต์ไว้ค่อนข้างมาก เรียกได้ว่า เหมาะกับการทำเชิงธุรกิจสุดๆ เพราะทุกวันนี้บ้างงาน บ้าง Application ที่เราสร้างมาใหม่ก็ล้วนเเต่มีสิ่งที่เหมือนกัน ซ้ำกัน แต่ **Rails** ได้จัดการทำสิ่งซ้ำในในการพัฒนาเบื้องต้นให้หมด (CRUD)

🡺9. Heroku คือ

เป็น Platform as a Service (Paas) ที่ให้เราใช้งานได้ฟรี (มีแบบเสียเงินด้วย) โดยรองรับภาษาโปรแกรมที่หลากหลาย เช่น Ruby, PHP, Node.js, Python, Java, Clojure, Scala และยังสามารถสร้าง buildpack สำหรับภาษาอื่นๆได้ เช่น Lua ที่รันอยู่บน OpenResty ได้อีกด้วย

บทบาทการพัฒนา web Application

- การใช้งาน PHP บน Heroku มีได้สองทางเลือกคือใช้ตัวรันไทม์หรือ VM ของ PHP รุ่นปกติ หรือใช้ HipHop VM ของ Facebook ที่ช่วยให้ประสิทธิภาพของ PHP ดีขึ้นมาก (ในขณะที่ยังคงความเข้ากันได้กับโค้ด PHP เดิม) ซึ่งนักพัฒนาสามารถกำหนดได้เองว่าจะเลือก VM ตัวไหน

- Heroku ยังมีแผนจะรองรับระบบจัดการแพ็กเกจ Composer เพื่อให้การจัดการ dependency ของโมดูลต่างๆ ของ PHP ง่ายกว่าเดิม

🡺10. เพื่อช่วยให้การพัฒนาเป็นไปอย่างมีระบบ มีเป้าหมายของการพัฒนา ไม่ใช่ทำตามโมเดลหรือทำตามกำหนดการที่กำหนดไว้แต่แรก ต้อง เข้าใจอย่างถ่องแท้ว่าเป้าหมายในการพัฒนาซอฟแวร์คืออะไร แล้วพยายามสร้างซอฟแวร์ที่มีประสิทธิผลมากที่สุดภายใต้ทรัพยากรที่มีอยู่ จึงต้องยืดหยุ่น สามารถปรับเปลี่ยนทุกอย่างได้ตามความจำเป็นในการทำงานจริง จะมีโมเดลที่ใช้ประจำ