นางสาว ประภาพร ศรีสัมฤทธิ์ 57160584 กลุ่ม 2

11.โปรแกรม version control มีประโยชน์อย่างไร

ตอบ

1. ย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจคกลับไปเป็นเวอร์ชั่นเก่าได้
2. กู้คืนไฟล์ที่คุณลบหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย
3. จัดการ backup source code ได้

12. ข้อได้เปรียบของ distributed version control เมื่อเทียบกับ centralized version control คืออะไร

ตอบ

DVC ลดภาระในหลาย ๆ action ให้ไม่ต้องเชื่อมต่อกับ remote repository ทำให้มีความเร็วมากกว่า

DVC แบ่งเฟสการทำงานได้ละเอียดคือมี local repository ไว้ในเครื่องเรา ถึงแม้จะ commit ผิดเราก็สามารถแก้ไขได้โดยไม่ต้องเชื่อมต่อไปที่ remote repository ถ้าแก้ไขเสร็จเรียบร้อยก็ค่อย push ทั้งหมดไปทีเดียว

DVC ไม่ต้องพึ่งอินเตอร์เน็ต ทำไว้เท่าไหร่ก็ได้ มีเน็ตใช้ก็ค่อย push

การมี working repository ทำให้ DVC สามารถแชร์การเปลี่ยนแปลงให้คนอื่นดูได้โดยที่ไม่ต้อง push ขึ้น remote repository

ระบบ Version Control ที่ไม่มี Server หลักเป็นศูนย์กลาง แต่ละเครื่องของแต่ละคนก็จะเก็บข้อมูลของโปรเจคทั้งหมดไว้ หมายความว่าข้อมูล History ของโปรเจคก็จะอยู่ในทุกเครื่อง และก็มีการ Share Changes กัน คล้ายๆ ระบบ Peer-to-Peer ครับ แนวคิดคือเวลาข้อมูลของเครื่องหนึ่งมีการเปลี่ยนแปลงก็จะมีการติดต่อไปยังเครื่องๆ อื่นๆ เพื่ออัพเดทให้ทุกคนเหมือนกัน ข้อดีคือมันกระจายความเสี่ยงของ Project History จะหายได้ครับ เพราะพังไปเครื่องหนึ่งเครื่องอื่นๆ ก็ยังมีอยู่ไม่หายไปไหน

13.ข้อได้เปรียบของ centralized version control เมื่อเทียบกับ distributed version control คือออะไร

ตอบ

Server หลักเป็นศูนย์กลางในการเก็บข้อมูลของโปรเจคทั้งหมด Client แต่ละเครื่องจะเป็นแค่ Working Copy เท่านั้น วิธีทำงานคร่าวๆ เริ่มต้นเราก็สร้างโปรเจคขึ้นมาบน Server หลังจากนั้นแต่ละคนก็มาดาวน์โหลดลงเครื่องของตัวเอง เวลามีการเปลี่ยนแปลงอะไรก็อัพโหลดกลับขึ้นไปบน Server แล้วคนอื่นๆ ก็จะสามารถดาวน์โหลด Update นั้นมายังเครื่องของตัวเองได้ ข้อเสียคือถ้า Server เดี้ยงไป Project History ทั้งหมดก็จะหายไปหมด

หลักการง่าย ๆ คือมี repository คอย track กับเก็บประวัติการแก้ไขอยู่ตรงกลาง ผู้ใช้งานจะทำการ pull มาจาก repository แก้ไขจนเสร็จและแน่ใจว่าทำงานได้ จึงค่อย commit กับมาที่ repository

14.บอกแนวทางในการแก้ไข conflict ที่เกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

ตอบ

Resolving a removed file conflict

Removed file conflicts occur when one person edits a file, and another person deletes that file in their branch. Git does not know if you want to keep the file with the new edits, or delete the file and forget about those edits. This example will show how to resolve this both ways.

1. Keeping the edited file

First, you'll resolve the conflict by keeping the new changes. Suppose that you added a line to your README.md in branch-b, but someone else has deleted the file entirely in branch-c. Git declares the following conflict:

2.Resolving by removing the file

Now, you'll resolve that conflict the opposite way: by leaving the file deleted. Again, with a merge conflict, Git declares:

15. บอกแนวทางในการลด conflict ที่จะเกิดขึ้นเมื่อมีการ merge โปรแกรมของผู้พัฒนาหลายๆคนเข้าด้วยกัน

ตอบ

16. Git คืออะไร แตกต่างจาก Github อย่างไร

Git คือ Version Control ตัวหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็คเรา มีการ backup code ให้เรา สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของโปรเจ็คที่ใด เวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็สามารถทำได้ ฉะนั้น Version Control ก็เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวโดยเฉพาะอย่างยิ่งจะมีประสิทธิภาพมากหากเป็นการพัฒนาเป็นทีม

git แตกต่าง GitHub GitHubคือผู้ให้บริการ hosting สำหรับ git

17.จุดประสงค์หลักในการ branch คืออะไร

ใช้เพื่อแยกการพัฒนาความสามารถใหม่ๆ ออกมาจากส่วนหลัก โดยที่ git มี master branch มาให้อยู่แล้ว หลังจากทำการสร้าง git repository ขึ้นมา

ในการใช้ branch เพื่อแยกการพัฒนานั้น จะต้องทำการ merge การเปลี่ยนแปลงเข้ามายัง master branch หลังจากที่พัฒนาเสร็จแล้วเสมอ

18.Fast forward merge คืออะไรและทำไมการ Push ไปที่ remote repo จึงควรจะต้อง merge แบบนี้

"Fast forward" ใน merge นั้น เพราะ commit ที่ถูกชี้โดย branch ที่คุณ merge มันเป็น upstream ของ commit ที่คุณอยู่โดยตรง Git ก็เลยขยับ pointer ไปข้างหน้า พูดอีกนัยหนึ่งก็คือ เวลาที่คุณพยายามจะ merge commit ซักอันเข้ากับ commit ที่สามารถไปถึงได้โดยการตาม history ของ commit อันแรก Git จะทำให้ทุกอย่างง่ายขึ้นโดยการขยับ pointer ไปข้างหน้าเพราะมันไม่มีงานที่ถูกแยกออกไปให้ merge สิ่งนี้เรียกว่า "fast forward".

19.หน้าที่หลักของคำสั่ง git pull คืออะไร

ใช้ดึงความเปลี่ยนแปลงจาก remote มายัง local และรวมเข้าด้วยกัน (มีค่าเท่ากับ fetch+merge)

20.แผนภาพด้านล่างนี้ต้องการสื่อความหมายอะไร



ทำการ Develop

Our git workflow looks like it is shown in the picture. There is a master branch for the releases. Development takes place in the develop branch respectively in feature branches. I connected my IntelliJ (version 13.4.1) to our ticket system (redmine). So when I open a task, IntelliJ automatically creates a feature branch for me.

When I switch back to the Default Task, IntelliJ checks out the master branch. However in most cases I want to have the develop branch checked out, and I have to switch manually. How do I configure IngelliJ (or git) to check out the develop brach each time

Once develop has acquired enough features for a release (or a predetermined release date is approaching), you fork a release branch off of develop. Creating this branch starts the next release cycle, so no new features can be added after this point.Only bug fixes, documentation generation, and other release-oriented tasks should go in this branch(which includes testing also). Once it's ready to ship, the release gets merged into master and tagged with a version number. In addition, it should be merged back into develop, which may have progressed since the release was initiated.

Using a dedicated branch to prepare releases makes it possible for one team to polish the current release while another team continues working on features for the next release. It also creates well-defined phases of development (e.g., it's easy to say, “this week we're preparing for version 4.0” and to actually see it in the structure of the repository).