What is GIT?

A distributed versioning control system with an emphasis on speed

What does it do?

- Git helps developers to collaborate on projects and keep track of changes in the source code
- Git allows you to revert back to a previous version of software if a contribution causes bugs
- You can also create branches to co-develop and test different aspects of the project which can be merged to the master at a later date

How does it do it?

- Git stores project data and information on repository servers within the company’s network or on the cloud
- Github is an example of such a service, providing cloud services for developers
- Changes to the git repository are most commonly done in the command line/terminal, although there are applications with simpler user interfaces available

Group Members:

- George Ritchie
- David Dahlgren
- Wes Paul
- Jeremy Moseley

Thank You!

What we learned from Norah

Rob and Anita

Situation Aware Systems

we are not quite there yet

- Situation Awareness = Perception & Comprehension of elements in the environment
- Creating a global context for individuals across the internet
- Allowing people to have a more significant control over their privacy
- A system should be aware of user context; any information that can affect the system operation or interaction with the system

Example

- Online Shopping
  - Understanding user preferences
    - Dynamic context: age, location, preferences, who not to share info with
    - Static context: name, birthday, gender
  - Rob likes to shop from AllPink.com
    - Dynamic context: 20-24, Victoria, prefers high heels to sneakers & paypal to pay for his transactions, doesn’t want Ebay to have his info.
    - Static Context: Rob; 14/02/89; Male

Version Control Systems

Geoff Gollmer, Michael Atavine, Adam Anderson

Problem Domain

- Concurrent development of projects by distributed teams present a series of progress impeding issues
  - Keeping track of revisions
  - Ensuring work is being performed on latest versions
  - Merging changes to files (Conflicts)
  - Keeping backups of projects
  - Branching

Centralized Solution

- Master copy of project stored in one location
- Users must update to latest version before submitting changes
- Users cannot submit changes on files with conflicts until manually resolved
- Revisions automatically stored as backup
  - Keep track of submitting user
  - Keep text summary of submission details
- Allow duplications (Branches) of project to be worked on concurrently

Decentralized Solution

- Master copy still exists
- Each user has their own local repository
    - May pull changes from master repository
    - May only push changes to master repository once up to date
- Why decentralized?
    - Users may keep personal revision history until pushing to master repository
    - Reverts can be performed on local changes
    - Commits can be performed offline

Source Control Examples

- Centralized
  - SVN
  - Perforce
  - TFS
- Decentralized
  - Git
  - Mercurial
  - Fossil

Version Control Systems

by Samuel Maskell, Mackenzie Marshall and Y Nguyen

Why do I need it?

- Helps with collaboration
  - Allows groups to work in parallel
  - Helps to resolve conflicts
  - Synchronization
    - Everyone has lastest copy
  - Allows for branching
    - Make variations of main project
- Easy backup system
  - Revert to previous versions
- Simple and reliable
- Maintained master copy

How does it work

- Main copy stored on server
- Each user has "working copy"
    - Snapshot of master copy
- Make changes to local, "working", copy
- Commit changes to server
    - Add comments to help track changes
    - If conflicts occur, merge them together
- Update to latest version
    - Get other users changes
- Two types
    - Centralized: e.g. svn
    - Decentralized: e.g. git

What We Learned

Vish Gahlot
Richard McKenzie
Nicolas Guillemot
Marcelo Gomes

Context Management

- Current Systems are lacking
  - Ebay: tries to sell you things you've already bought
  - Often don't take time into account
- Improvements:
  - Learn from things like plane tickets
  - Focus on current information
  - Use ability to change information

Personal Sphere

- Contains information about user
- Sphere can be shared with various sources
    - Sources can read information from sphere
    - Sources can write to sphere
- Information can be modified by user