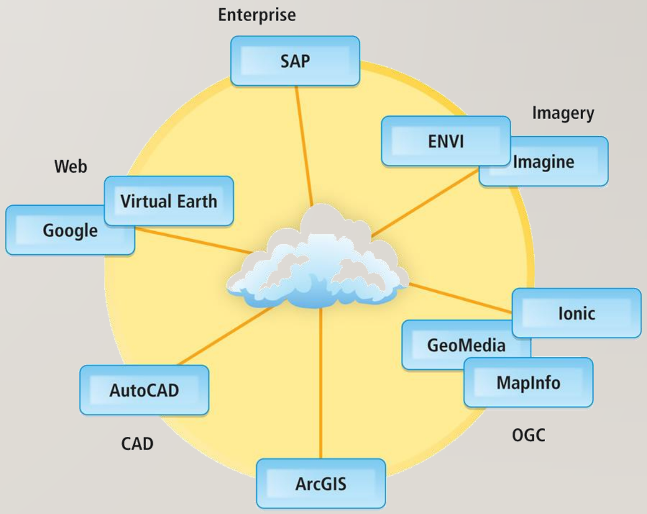
ENTERPRISE SYSTEMS



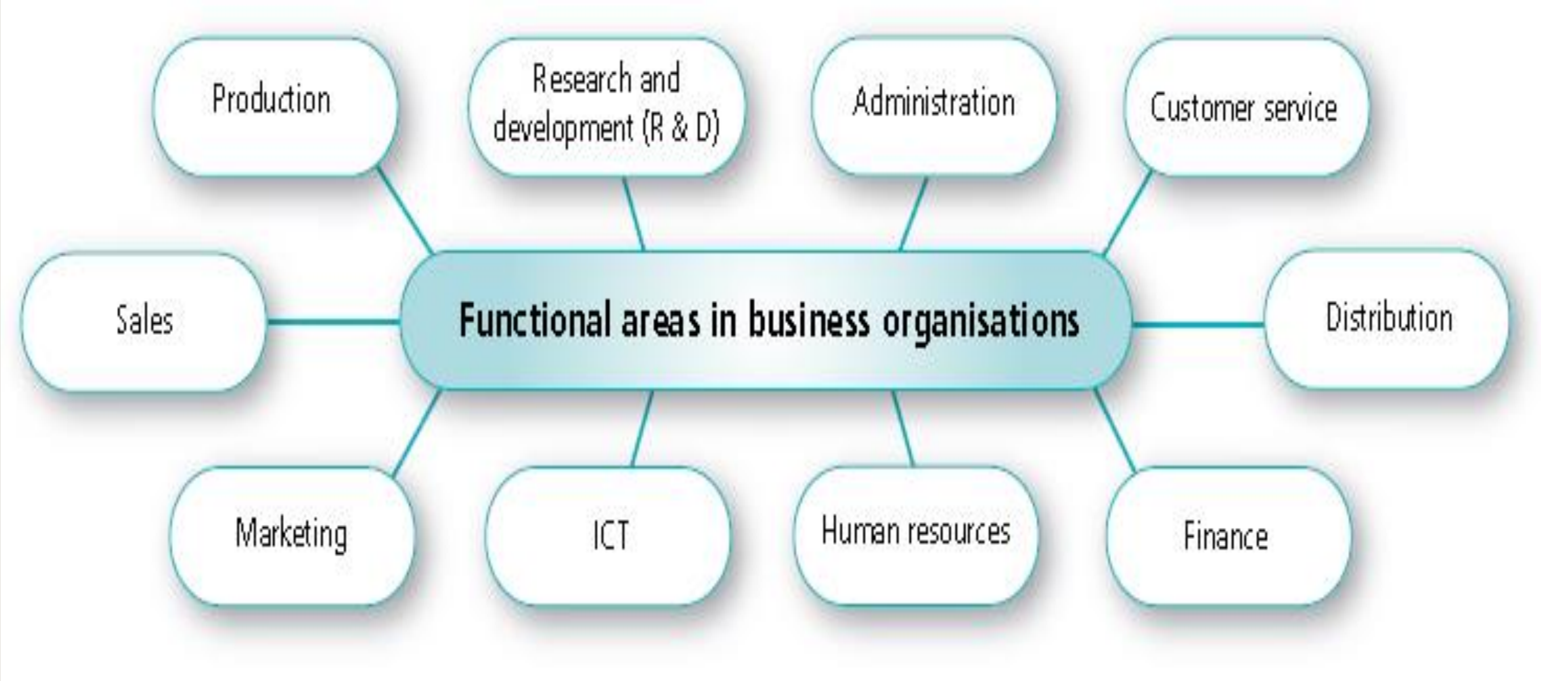
Josephine Andrews
Email: Josephine.Andrews@ncirl.ie

LEARNING OUTCOMES

- To gain an understanding of the following:
- Enterprise Systems and their different functions
- ERP Systems
  - Benefits
  - Implementation
  - Vendors
  - Pitfalls
- Customer Relationship Management
- Github

ENTERPRISE SYSTEMS

- What do you think of the different functional areas in a business organisation?

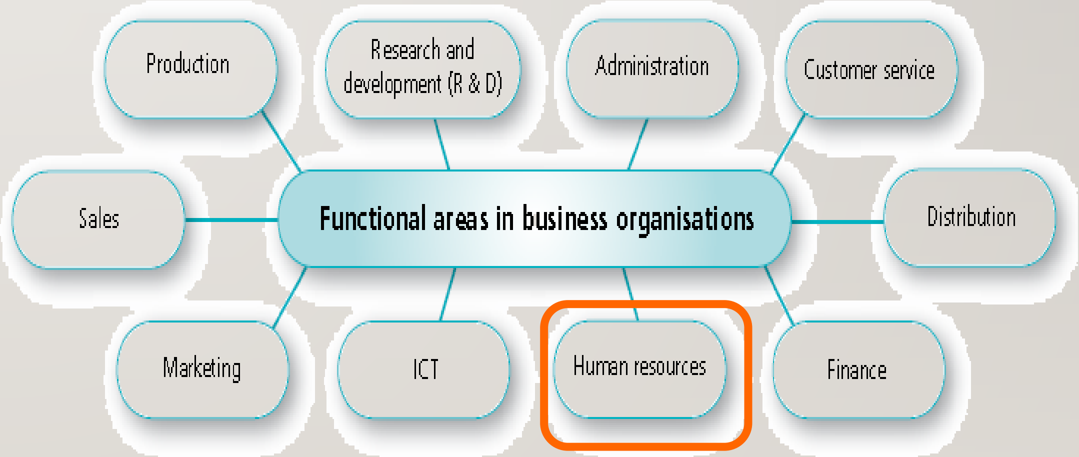


ENTERPRISE SYSTEMS

- Is the overall combination of computer hardware and software that a business uses to organise and run its operations
- For example, an integrated enterprise system will generally handle more than one operation for a company to facilitate its business and management reporting needs

ENTERPRISE SYSTEMS

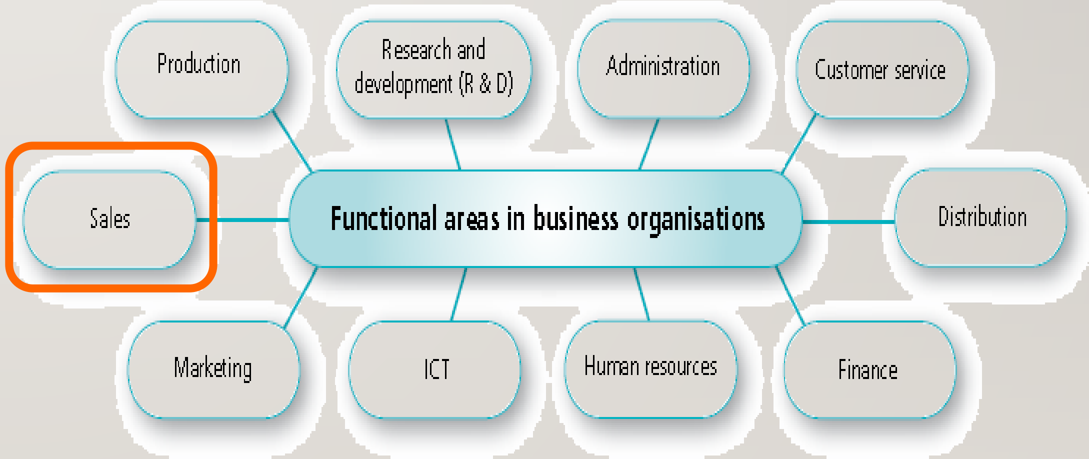
- What types of information do you think these different functional areas need to maintain?
- Human Resources:
  - Contracts
  - Payroll
  - Time-sheets
  - Holidays



Functional areas in business organisations

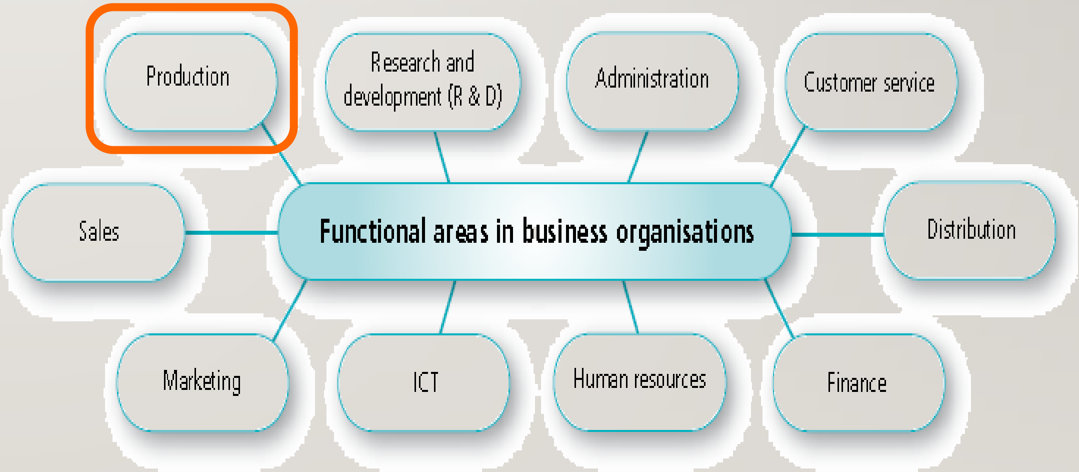
ENTERPRISE SYSTEMS

- What types of information do you think these different functional areas need to maintain?
- Sales:
  - Sales Reps
  - Customer Details
  - Order Details
    - On Order
    - History
  - Agreements/Contracts



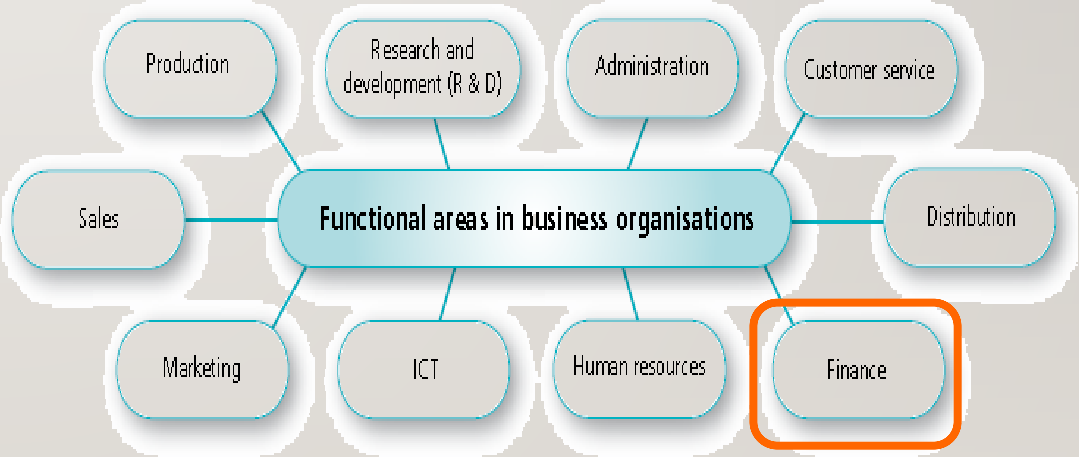
ENTERPRISE SYSTEMS

- What types of information do you think these different functional areas need to maintain?
- Production
  - Stock levels
  - Demands (new orders)
  - Capacity
  - Work Schedule



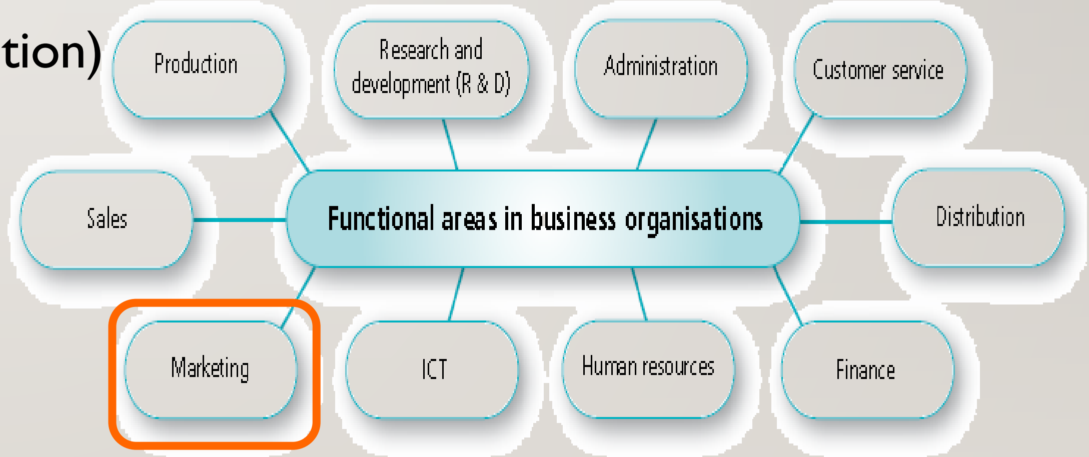
ENTERPRISE SYSTEMS

- What types of information do you think these different functional areas need to maintain?
- Finance
  - Accounts Payable
  - Accounts Receivable
  - Cash on hand
  - Fixed Assets
  - Salaries



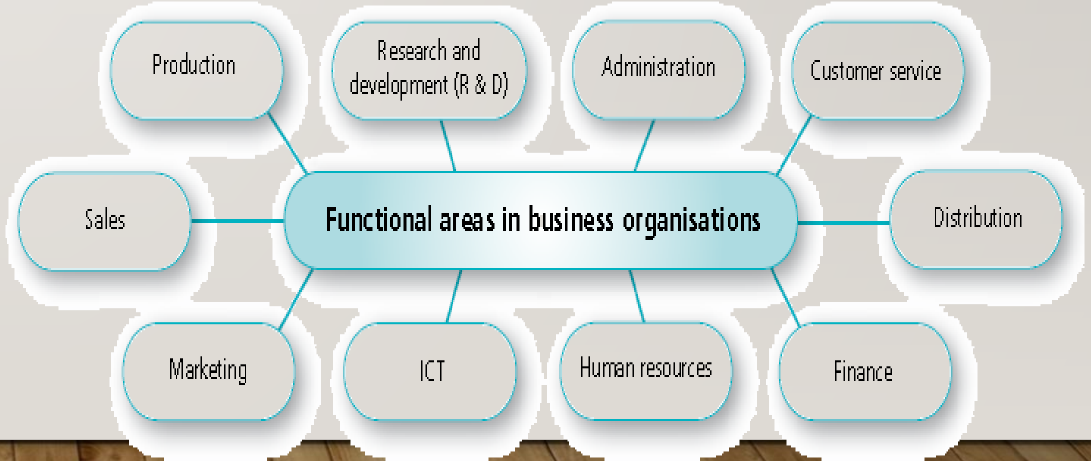
ENTERPRISE SYSTEMS

- What types of information do you think these different functional areas need to maintain?
- Marketing
  - Product offering
    - 4 Ps (product, price, place, promotion)
  - Customer profile
  - Market share
  - Market trends



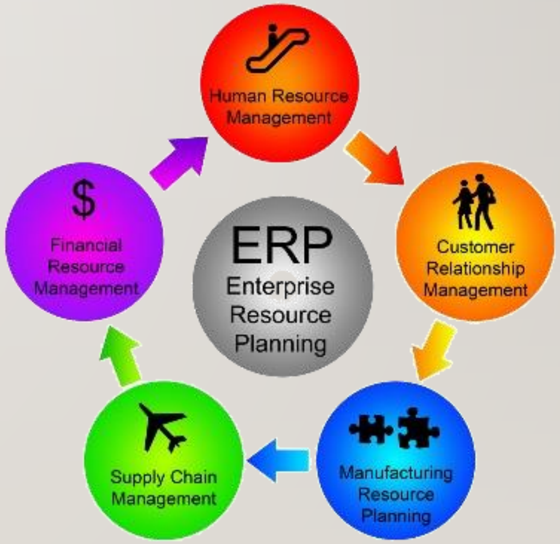
ENTERPRISE SYSTEMS

- Is there overlap within the functional areas in terms of data needs?
- How might this data be shared?
- Multiple systems/applications versus the one stop shop (ERP)



ERP SYSTEM - DEFINITION

- ERP system is a set of integrated business applications, or modules which carry out common business functions such as general ledger, accounting, or order management.


Human Resource Management
Customer Relationship Management
Manufacturing Resource Planning
Supply Chain Management
Financial Resource Management
ERP
Enterprise Resource Planning

WHAT MAKES ERP DIFFERENT

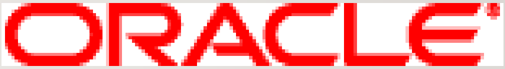
- Integrated modules
- Common definitions
- Common database
- Update one module, automatically updates others
- ERP systems reflect a specific way of doing business
- Must look at your value chains, rather than functions

BENEFITS OF ERP

- Common set of data
- Help in integrating application's for decision making and planning
- Allow departments to talk to each other
- Easy way to solve Y2K problem (remember?)
- Easy to integrate by using processes built into the ERP software
- A way to force BPR (reengineering)

VENDORS

SAP


ORACLE

PEOPLE
Soft

IFS

SIEBEL
IT'S ALL ABOUT THE CUSTOMER™

ssa global™
forward faster

JDEdwards
Alliance Partner

Microsoft Dynamics

ERP IMPLEMENTATION

Do you think implementing an ERP system into an organisation would be an easy or difficult project?

DIFFICULTY IN IMPLEMENTATION

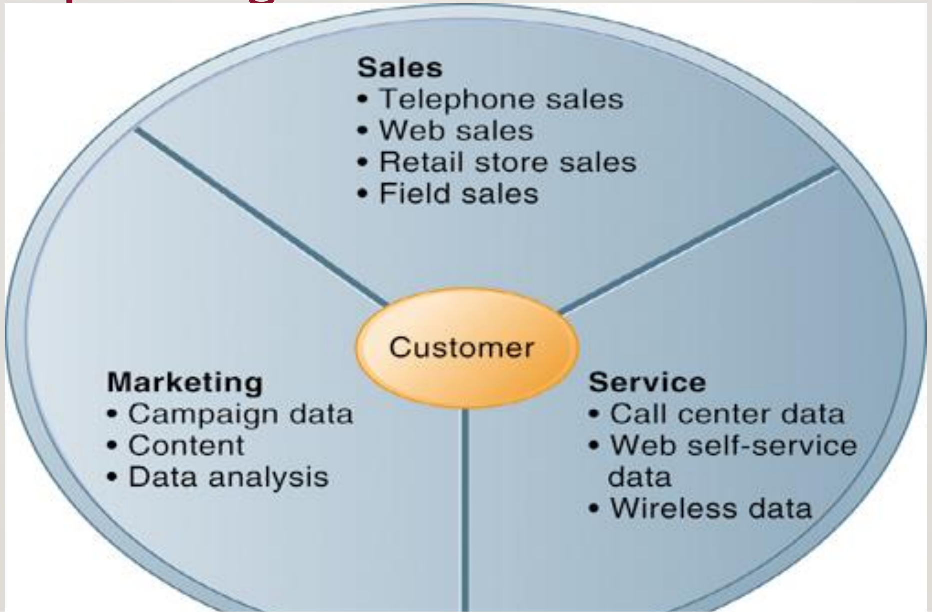
- Very difficult
- Extremely costly and time intensive
- Typical: over $10,000,000 and over a year to implement
- Company may implement only certain modules of entire ERP system
- You will need an outside consultant

COMMON PITFALLS

- Do not adequately benchmark current state
- Did not plan for major transformation
- Did not have executive sponsorship
- Did not adequately map out goals and objectives
- Highly customized systems to look like old MRP systems

CRM

- Customer relationship management



CRM systems examine customers from a multifaceted perspective. These systems use a set of integrated applications to address all aspects of the customer relationship, including customer service, sales, and marketing

GETTING APPLICATION SOFTWARE

- Organisations have two options when acquiring software:

Ready made application "Package"

Developed from scratch "Custom Made"

READY MADE

- Ready to use software that has been developed by another company, usually a commercial software concern
- Only works when it is of a standard variety (common system required by a number of company's such as accounting systems)

CUSTOM BUILT

- Developed from scratch to meet the specific requirements of the organisation
- A system that is uniquely required by a single organisation when not be available and development of a custom built system is the only option

RECAP

- To gain an understanding of the following:
- Enterprise Systems and their different functions
- ERP System
  - Benefits
  - Implementation
  - Vendors
  - Pitfalls
- Supply Chain Management
- Customer Relationship Management

BREAK TIME

10 Minute Break

GITHUB

Josephine Andrews
Email: Josephine.Andrews@ncirl.ie

Revision Control Allows you to track and manage changes you make to your code as you go.

WHAT IS GIT?

- Created in 2005 by Linus Torvalds (same guy who created Linux)
- Version Control System (VCS) for tracking changes in Computer files
- No set programming language – can be used with HTML, C# etc
- Distributed version control
- Coordinates work between multiple developers
- Who made what changes and when
- Revert back at any time
- Local and remote repository (repos)
- Once you make a commit to a remote repo other developers can pull this

CONCEPTS OF GIT

- Keeps track of code history
- Takes “snapshots” of your files
- You decide when to take a snapshot by making a “commit”
- You can visit any snapshot at any time
- You can stage files before committing

BASIC COMMANDS

$ git init	// Initialise Local Git Repository
$ git add <file>	// Add File(s) to Index
$ git status	// Check Status of Working Tree
$ git commit	// Commit Changes in Index

REMOTE REPO - BASIC COMMANDS (GITHUB)

$ git push	// Push to Remote Repository
$ git pull	// Pull Latest From Remote Repository
$ git clone	// Clone Repository Into a New Directory

INSTALLING GIT

- Linux (Debian)

```
$ sudo apt-get install git
```

- Linux (Fedora)

```
$ sudo yum install git
```

- Mac

http://git-scm.com/download/mac

- Windows

http://git-scm.com/download/win

GITHUB – HOW TO SET IT UP

- We will be using the command line in this series as this is the most powerful way to use Git
- Git: base version control system – don’t need internet
- Github: Cloud, hosting service, instead of backing up to own computer, can backup online.

STEPS TO GETTING SET UP IN GITHUB

Step 1: Setting up a local repository

Step 2: Staging and Committing

Step 3: Better logging and checkouts

Step 4: GitHub Setup – Creating and Pushing

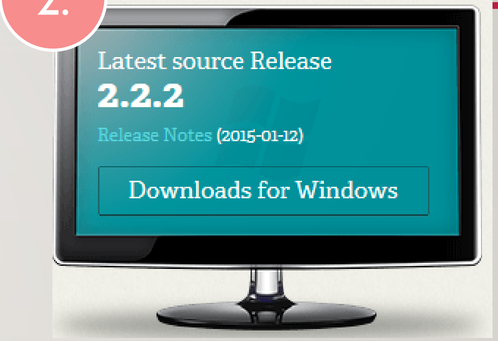
Step 5: Adding Collaborators and pulling from GitHub

Step 6: Conflict and Resolution

STEP 1: SETTING UP A LOCAL REPO

1. Go to: http://git-scm.com/
2. Click download for windows
3. Click save first
4. Click run until you see the following
5. Select two components
   1. Git Bash here
   2. Git GUI here
6. Untick view release notes

2.



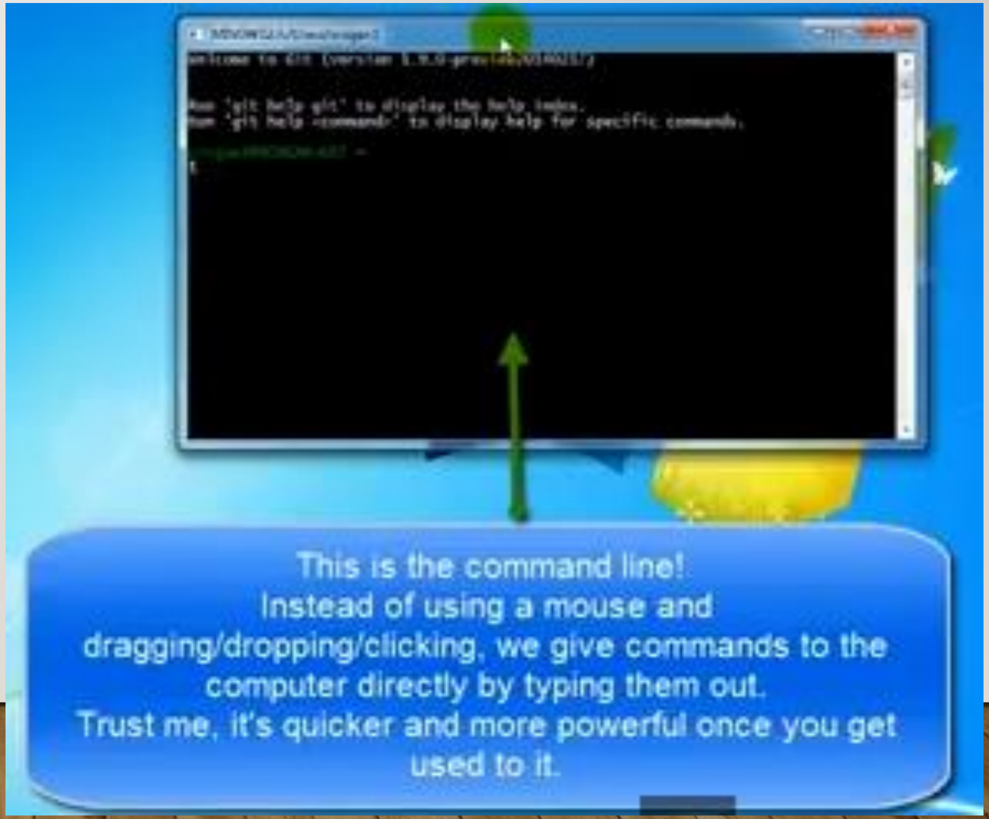
Latest source Release
2.2.2
Release Notes (2015-01-12)

STEP 1: SETTING UP A LOCAL REPO

7. Create a folder on your desktop with a html file inside
8. Right click on folder and click GitBash here
9. Type ls and click return

ls

Unit style command that lists information about the files in the current folder/directory



```
Welcome to Git (version [illegible])

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

[illegible] ~
$
```

This is the command line!
Instead of using a mouse and dragging/dropping/clicking, we give commands to the computer directly by typing them out.
Trust me, it's quicker and more powerful once you get used to it.

STEP 1: SETTING UP A LOCAL REPO

10. Type in "git status"

Check the status of git in this folder. Is it a repository yet?

11. Type in "git init"

Turns the folder into a git repository

We are on the main branch and have untracked files

STEP 2: STAGING AND COMMITTING

- **Typing: clear –** clears the history of the command line

You can commit in two ways: one file at a time or a number of files.

STEP 2: STAGING AND COMMITTING

1.To add a new file, type:

```
git add index.html
```

2.To commit a new fileInsert:

```
git commit  – m ”Created basic template”
```

3.To check the status:

```
git status
```

4.To check all the logs:

```
git log
```

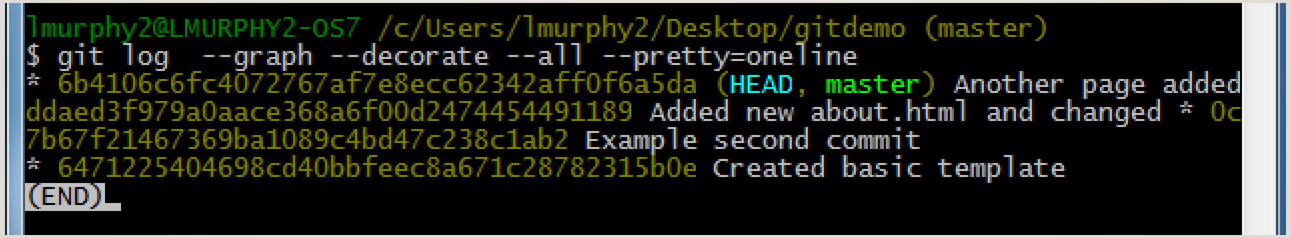
5.To add a number of files:

```
git add *
```

STEP 3: BETTER LOGGING AND CHECKOUTS

New way of logging:

git log --graph --decorate --all--pretty=oneline



```
lmurphy2@LMURPHY2-OS7 /c/Users/lmurphy2/Desktop/gitdemo (master)
$ git log  --graph --decorate --all --pretty=oneline
* 6b4106c6fc4072767af7e8ecc62342aff0f6a5da (HEAD, master) Another page added
ddaed3f979a0aace368a6f00d2474454491189 Added new about.html and changed * 0c
7b67f21467369ba1089c4bd47c238c1ab2 Example second commit
* 6471225404698cd40bbfeec8a671c28782315b0e Created basic template
(END)
```

master = says this is the main up to date part of the branch
head = this is the bit we are working with

STEP 3: BETTER LOGGING AND CHECKOUTS

Going back and looking at a commit

1. Copy the id you wish to investigate by clicking the logo at the top of the window, click edit, mark and highlight the id.
2. Click **q** on the keyboard to get out of the git log graph
3. Type git checkout followed by the id code. Paste it in

Now it should go back to the original layout

You can simply put it back by typing in git checkout master

STEP 4: GITHUB SETUP

GitHub is a web based system

1. Go to GitHub.com and create a new account
2. Select free plan
3. Create a new repository and select public
4. Copy the piece of code from the 2nd column down:

```
git remote add origin https://github.com/yourusername/yourrepo.git
```

Paste it in GitBash

Pushing the code up to GitHub

STEP 4: GITHUB SETUP

6. Copy and paste the next line of code:

```
git push -u origin master
```

7. Enter username and password

Please note you will not see your password being typed

8. Refresh the page to see your changes

Pushing the code up to GitHub

STEP 4: GITHUB SETUP

Pushing files to GitHub

1. Type in git add*
2. Type in git commit -m”new file”
3. Git push origin master

STEP 5:ADDING COLLABORATORS

- Make sure a project is open
- Click Settings on the right hand side
- Click Collaborators on left hand side, add a person.
  - Here you can pull down the code and push it back etc

Git pull

  - This command pulls down the changes. Type it into git Bash

STEP 6: CONFLICT RESOLUTION

Conflict pops up when two people are working on the code.

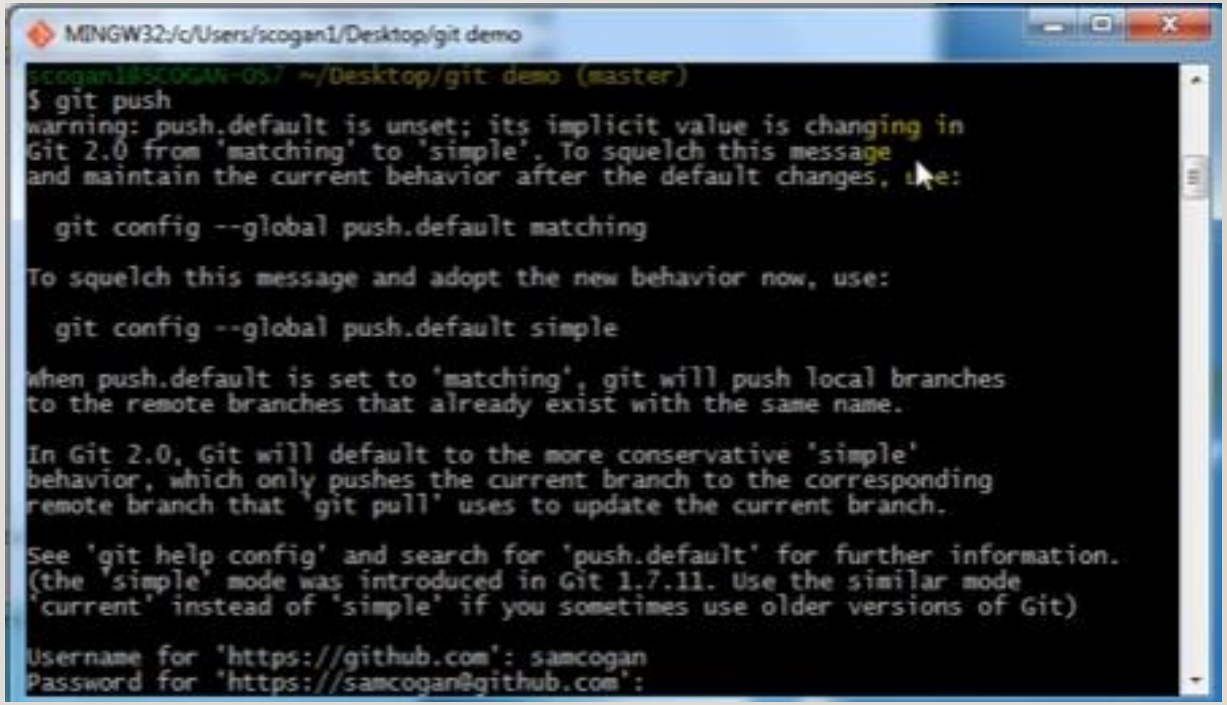
Git pull (Pull it down)
Open the file
Check the differences

1. **Git status**
2. **Git add***
3. **Git commit** (no message needed as it will merge with previous file) VIM message will appear

4. Type the following and click return

:wq

5. Type **Git push**



MINGW32:/c/Users/scogan1/Desktop/git demo

```
scogan1@SCOGAN-OS7 ~/Desktop/git demo (master)
$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': samcogan
Password for 'https://samcogan@github.com':
```

TO RUN A REPORT

- In order to see who has done what, it’s a good idea to print a report/log
- To do this – type git log
- This will give a detailed list of who has done what

HOW TO FIX YOUR ERRORS

Not a repository

- This means you have not set up your repository
- To do this type: git init

Tell me who you are

- This problem with the configuration. To fix this, you need to input your username and password again. Type the first line:
  - Git config user.name "BillyEve1111"
  - Press enter and type the following line
  - Git config user.password "BillyEveryteen@ncirl.ie"

ANY QUESTIONS

Any Questions ?