

Software Development on Linux Systems

4002-XXX-XX

By

Cody Van De Mark

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Today

- Introduction to Course
- Syllabus
- Introduction to Project
- Brainstorming Process
- Getting Involved in the Open Source Community

Introduction

- This course is about
 - Open Source Software Development
 - Software Development on Linux
 - Open Source/Closed Source Legalities
 - Software Design, Release and Maintenance

Syllabus

- The syllabus is not set in stone
- It is subject to expansion, change or modification at my discretion
- Syllabus should only be used as a general guide for the course

Texts

- There are no required texts for the course.
- There are, however, recommended texts for open source software concepts, Linux development and Python
- You will also rely directly on open source documentation available online in many forms
- You will be developing your own documentation for others to use in the future

Course

- Course requires Introduction to Database and Programming for IT III
- This is an advanced elective course
- This means
 - you are here by choice
 - you are expected to put in extra effort
 - the course will be more rigorous than a typical course
 - course concepts will leverage previous courses
 - course concepts will be more advanced

Goals

- Be introduced to the basic concepts of Linux and Unix
- Be familiar with the history of open source
- Understand the fundamentals of licensing, trademarks, software ownership and rights
- Understand the basics of open source economics and the relationship between open source and businesses
- Understand open source communities, societies and politics
- Examine how to use Linux for development and services

Goals

- Explore how to get involved in the open source community
- Explore high level process of open source software development
- Examine Linux package types and package managers
- Explore how to use open source version control software and compare various packages
- Explore open source software development tools and potential benefits of each

Goals

- Examine the publishing process and release cycle using hosting facilities
- Examine the bug tracking process, maintenance and patching
- Explore future development and project forking

Grade Distribution

- Lab Exercises - 25%
- Lab Practical - 25%
- Homeworks - 10%
- Project - 40%
- Peer Evaluation - 10%

Academic Dishonesty

- Follows RIT's standard policies
- Follows IST's standard policies
- This must be strictly enforced due to the legal implications of dealing with open source licenses and intellectual property

Introduction to the Project

- This course requires you work on an open source project
- This will be the main assignment for the semester, outside of lab
- Project requirements document is available on MyCourses
- You may work in groups or alone
- All group projects **MUST** be open source licensed

Introduction to the Project

- You or your group may choose your project idea
- Idea must be approved by the professor
- You may work on new code or an existing open source project
- You may use any existing open source code, projects, frameworks, libraries, SDKs or utilities
- You may use open source GUI designing/building tools

Introduction to the Project

- Project may be done in a variety of languages, or even multiple languages
- Project must not already exist as an open source package, or if it is it must be justified
- Project may be porting or packaging of a non-Linux application
- Project should be something you care about or want to improve

Introduction to the Project

- Creators will choose which open source license to use
- Creators may choose to license under multiple licenses
- All projects leveraging existing code must provide the licenses for that code
- You will need to brainstorm ideas and research technologies

Brainstorming Process

- The first lab is a group brainstorming session
- All students are expected not to criticize any idea even the slightest
- Absurd ideas and failing ideas tend to drive technology forward
- You are expected to focus on quantity and not quality
- It is most important to have a great number of ideas than a few good ideas

Brainstorming Rules

- You are now allowed to criticize or judge another student's ideas
- You should encourage unusual and unrealistic ideas as they provide completely new ways of solving problems
- You need to generate as many ideas as you can with minimal detail
- Ideas should merge, fork and evolve to create further ideas
- You should look at many domains to be creative
- You should consider new or unrealistic domains

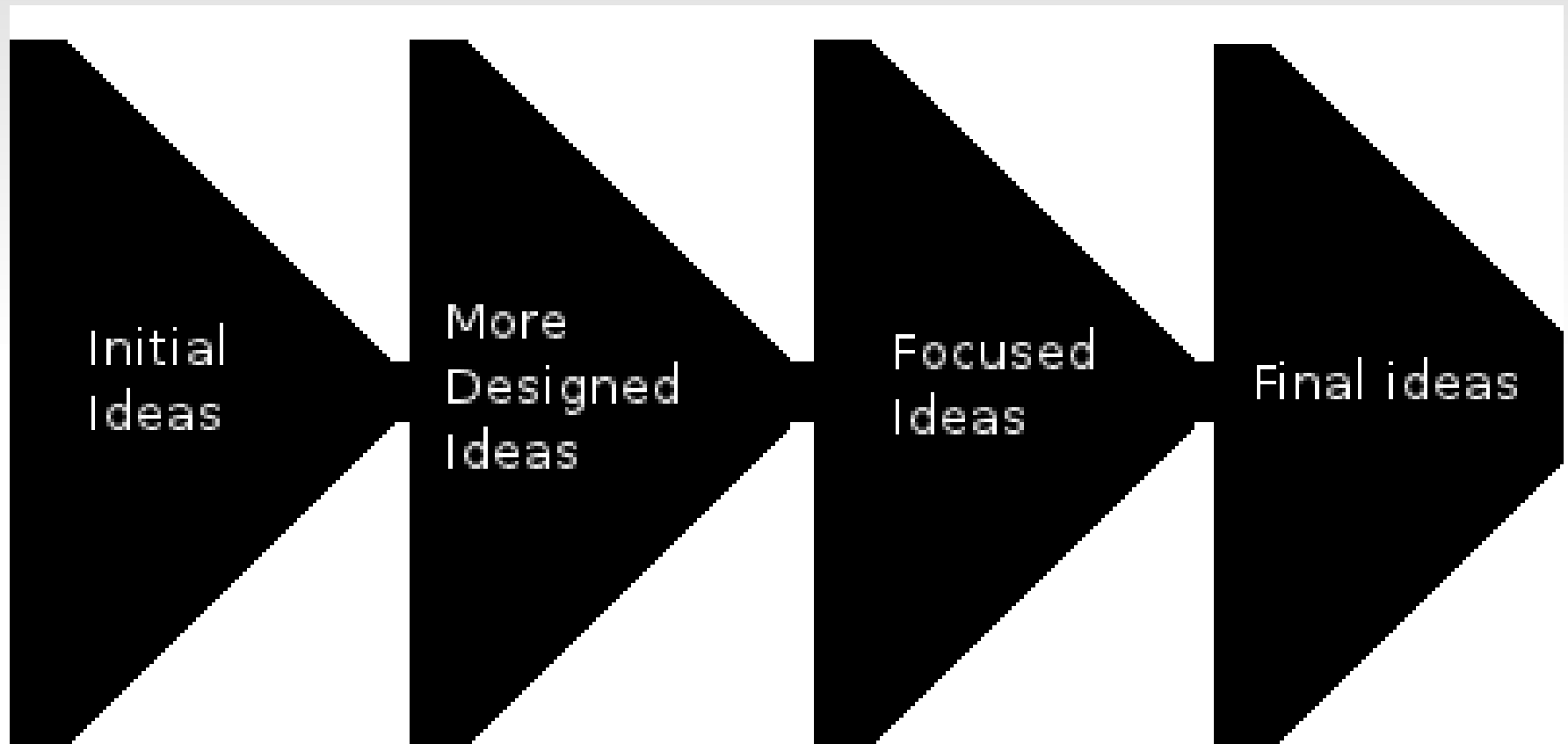
Brainstorming Suggestions

- Use sticky notes or index cards to write idea down
- Do not put much detail into any single idea
- Write new ideas down quickly
- Take inspiration from your group members' ideas

Process

- Develop a lot of ideas initially (the first brainstorm session)
- Narrow those ideas down and do another brainstorm session to create many more ideas
- Then narrow those down into categories (or focused ideas) and do the exercise again; This time you should do it with a little more detail on each idea
- Lastly, narrow your ideas down to your best and choose from them

Process



Final Ideas

- When determining your final ideas, you do not need to exclude absurd or unrealistic ideas;

By only completing half of an absurd or unrealistic idea, you have likely already exceeded the reach of current technology

- Final ideas should be the ones the group considers to be the best based on how important it is to you;

This differs from commercial/proprietary environments where your ideas are based on society, marketing, implementation, cost and competition

Final Ideas

- You should narrow your final ideas down to about five and choose one from them

This does not mean you should discard other good ideas as you may want to develop them in the future as a new project or add-on

- Your final ideas should be the ones you have the desire to make as you will be more likely to complete them and do them well
- You do not need to consider commercial value as these are open source projects

Open Source Ideas

- Open Source software is developed because:
 - Developers wanted to
 - or*
 - There was a need for it
 - or*
 - There was social benefit (IE: One Laptop Per Child)
 - or*
 - Developers wanted to innovate or experiment (IE: Scientific)
 - or*
 - It made life easier (IE: Automation software)

Open Source Ideas

- It is important to note that these ideas are only some of the reasons open source software is made
- Open source projects are developed for many reason

Example: Developers have made projects to help their own children learn and gave them away in hopes it would help others

- You may choose ideas for your own reasons

Many projects are made for personal use only and then given away

Getting Involved

- Lab 2 will require you to get involved in the open source community
- The open source community communicates in many ways

Often times these ways are specific to a subset of the community
(IE: Ubuntu community, MySQL community, etc)

- These ways tend to include mailing lists, IRC, blogs, planets, forums, distribution websites, repository websites, bug reports, etc

Getting Involved

- Lab 2 will require you to get involved in the open source community
- The open source community communicates in many ways

Often times these ways are specific to a subset of the community
(IE: Ubuntu community, MySQL community, etc)

- These ways tend to include mailing lists, IRC, blogs, planets, forums, distribution websites, repository websites, bug reports, etc

Getting Involved

- The easiest way to get involved is to become a member of a forum, join an IRC channel or start a blog
- In the future, you may consider joining a mailing list, joining a planet, joining a distribution site, joining a repository site or discussing bug reports
- Even if you are not active within a community, you can still be involved

It is encouraged that you are active, but commenting occasionally or solving a problem every so often is tremendously helpful

IRC (Internet Relay Chat)

- IRC is one of the most popular forms of communication in the open source community as communication is almost immediate

You can find an IRC channel for most things

- IRC channels provide a feed of informative discussion, problem solving and future direction very quickly
- They can be a form of support for more advanced problems, features and discussions

Less advanced support can usually also be found in dedicated channels

Blogs

- Blogging about your experiences and how you has solved problems is a tremendous help to the community

Blogging provides feedback and helps other users find solutions to a problem you once faced

- You will often find guides and problem solutions in blogs from other users
- You will be required to keep a blog for this course that documents your work and your experiences

Forums

- Forums are a large portion of the open source community and are often used for problem solving
- Most new users will turn to forums for support from others

The open source community has thousands of forums and is extremely well documented through them

You will find that many of the answers to your problems have been solved on forums

- Many experienced users will help newer users solve their problems thus helping them become more experienced and keeping the forum active

Lab 2 - Getting Involved

- In lab 2, you will need to join an IRC channel and learn how to use them
- You will also need to start an on-going blog for this course that will be your project documentation, lab experiences and general course experiences
- You will also be required to gather data about your project that you will combine with your other group members' data for discussion

Personal Blog

- You may use various blogging software for your blog (wordpress, tumblr, blogger, etc)
- The class must be able to read your blog as it may help other groups;

Commenting on others' blogs is encouraged as you may find they have solved a problem you were facing

- You may also want to post about your struggle with a problem even if you have not come up with a solution;

Another student or group may be able to help

Personal Blog

- Each student must keep an individual blog even though there will be overlap between certain posts
- You are encouraged to post to your blog whenever you:
 - decide on a design direction for your project
 - Solve/find a problem in your project or a bug in software
 - find other open source projects of use
 - have a related experience you would like to share or would be helpful to others
- If you solve a problem or want to share an experience, provide the details and a guide on how you did it; This will be helpful to others in the future