

Software Modeling

- Software Development Process

Open Source Project with KUPE

Minwoo Kim

Hyuntaek Oh

Su-yong Choi

Outline

1. Open Source Project
2. Open Source Project with KUPE
3. Application to Project
4. Open Source Project with KUPE in Class

Open Source Project with KUPE

Open Source Project

Unix timeline

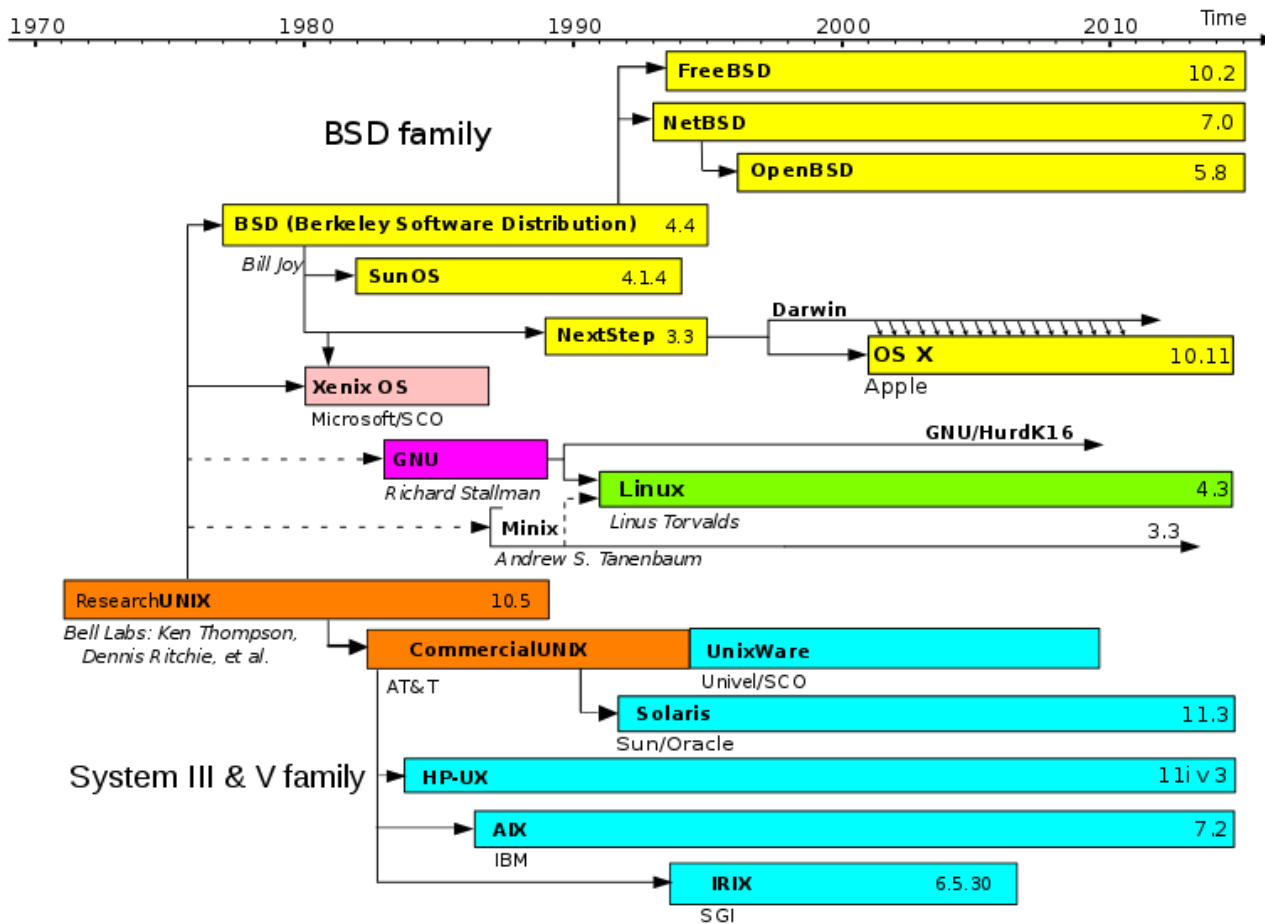


Image From: Wikipedia

Linux

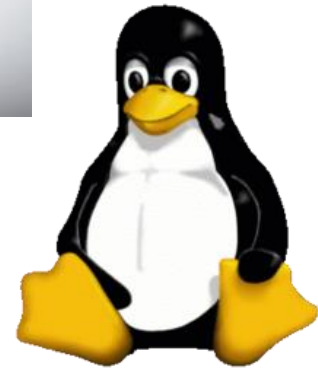
- 커널의 일종. 또한 그 커널을 사용 하는 운영체제를 가리키는 말
- 소스코드가 공개되어 있는 대표적인 오픈소스 소프트웨어
- ..어쩌다 보니 Open Source S/W 진영의 대표 (?)
 - 컴퓨터 역사상, 사람이 가장 많이 들어간, 가장 큰 프로젝트이다.
- 헬싱키 공대 대학생이었던 Linus Benedict Torvalds가 1991년 리눅스를 개발하고, "Linus의 MINIX"라는 뜻을 담아 "리눅스(Linux)"라고 명명
- Linus Benedict Torvalds
 - I absolutely love the GPLv2 -- because it embodies that "develop in the open" model.

Linux는 어떻게 만들어지는가?

- <http://youtu.be/yVpbFMhOAwE>

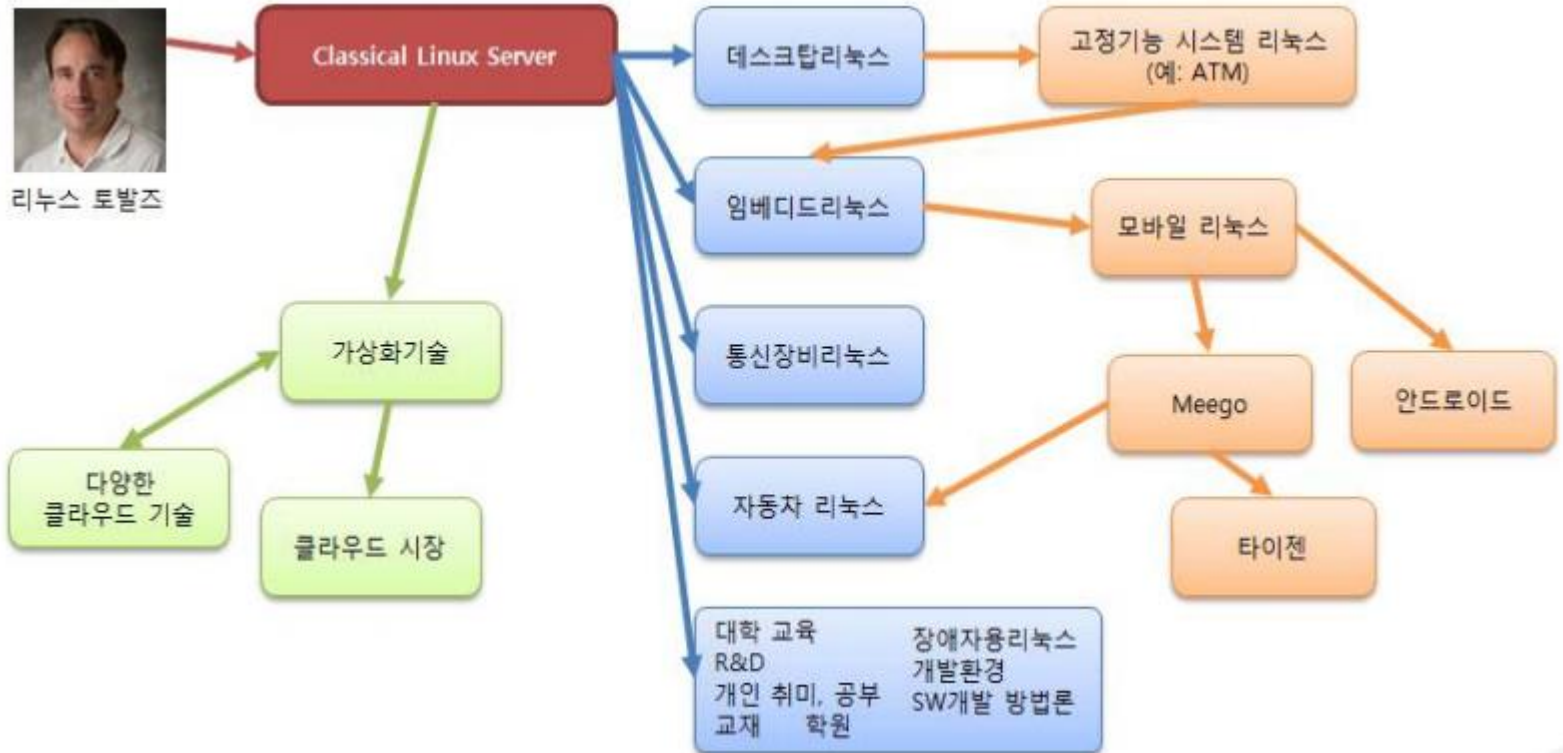


HOW LINUX IS BUILT



리눅스 커널의 마스코트인 Tux(턱스)

Linux 와 관련된 open project



Software and Open source software



- Richard Stallman
프린터 사용불편을 직접 해결하려고

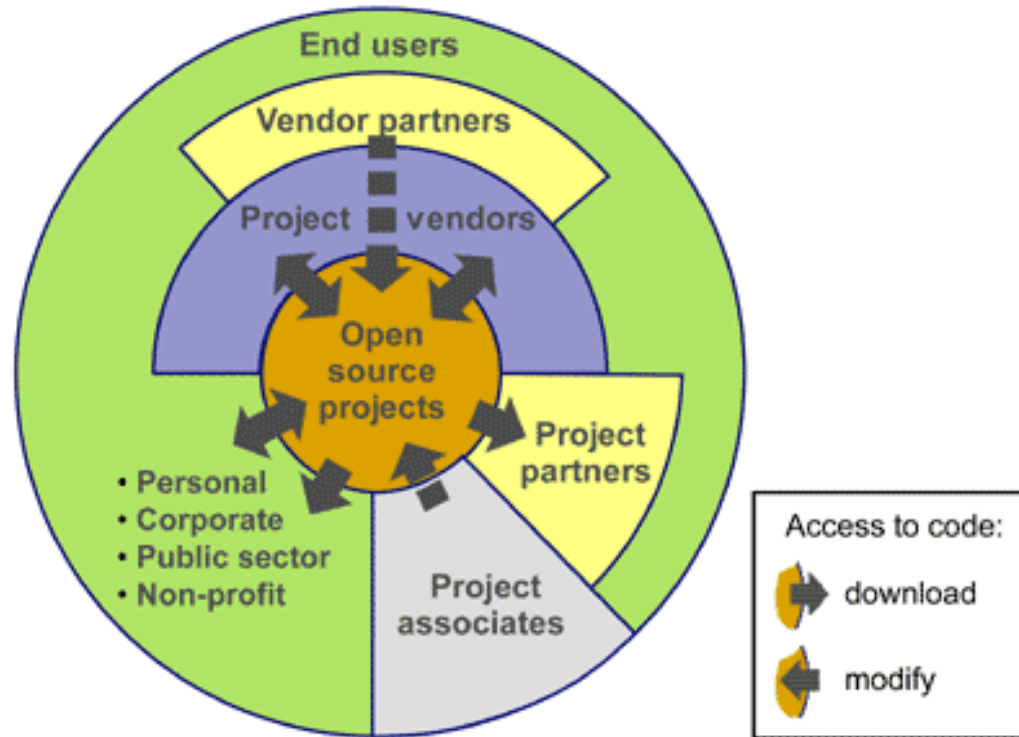
- Linus Torvalds
심심한 나머지(Just For Fun)



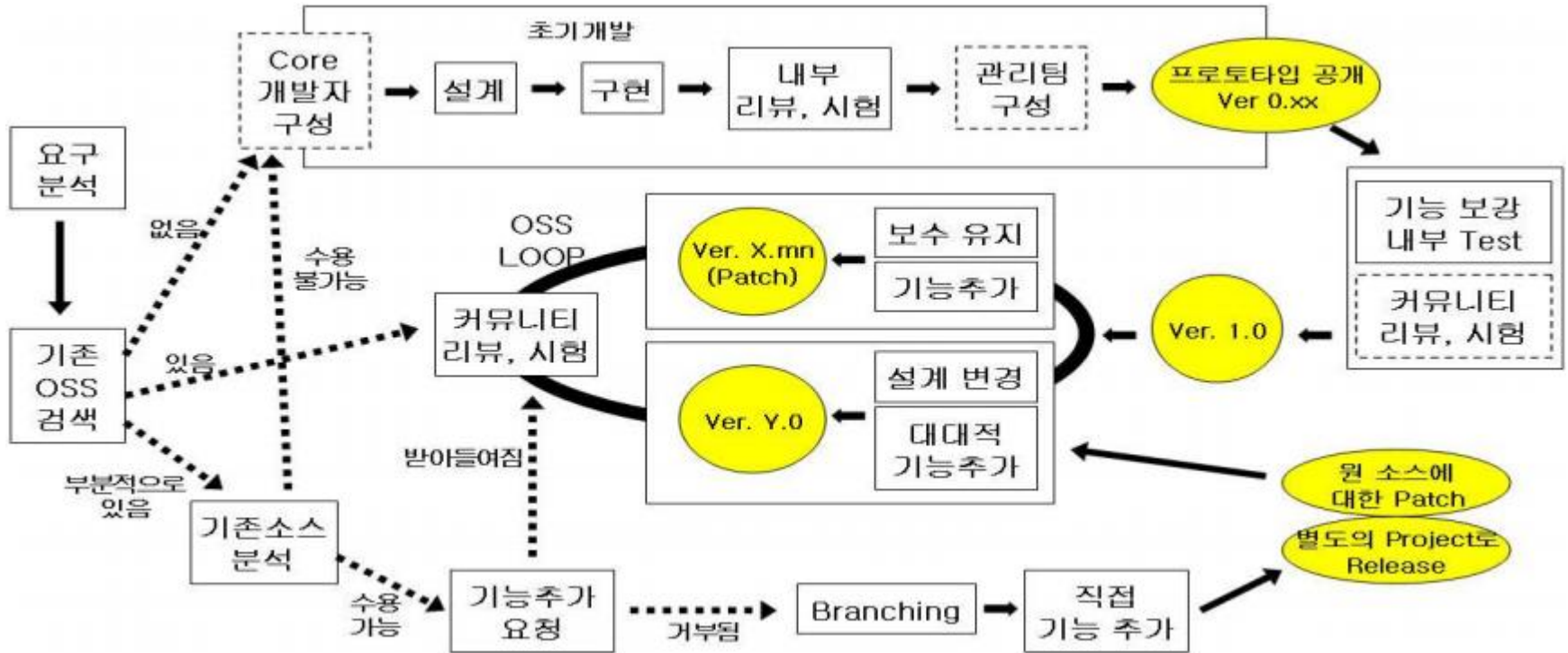
오픈 소스 소프트웨어 정의

- Freely access source code, run program for any purpose, modify the program, and redistribute on the basis of license
- Public collaboration

Open source project ecosystem



오픈소스 프로젝트 프로세스



- 표준화된 개발 절차와 환경에서 개발초기단계부터 검토
- 소스코드 관리도구, 버그관리 도구 등 자체적으로 구축 운영
- 메일링 리스트를 활용한 규정된 소통방식으로 기능제안, 패치제안, 검토의견, 시험결과 공유 및 문서 공유 등의

Producing OSS

- Karl Fogel, (2005-2016), Producing Open Source Software : How to Run a Successful Free Software Project

Table of Contents

Preface	IRC / Real-Time Chat Systems
Why Write This Book?	IRC Bots
Who Should Read This Book?	Archiving IRC
Sources	Wikis
Acknowledgments	Wikis and Spam
Disclaimer	Choosing a Wiki
1. Introduction	Q&A Forums
History	Translation Infrastructure
The Rise of Proprietary Software and Free Software	Social Networking Services
"Free" Versus "Open Source"	4. Social and Political Infrastructure
The Situation Today	Benevolent Dictators
2. Getting Started	Who Can Be a Good Benevolent Dictator?
Starting From What You Have	Consensus-based Democracy
Choose a Good Name	Version Control Means You Can Relax
Have a Clear Mission Statement	When Consensus Cannot Be Reached, Vote
State That the Project is Free	When To Vote
Features and Requirements List	Who Votes?
Development Status	Polls Versus Votes
Downloads	Vetoes
Version Control and Bug Tracker Access	Writing It All Down
Communications Channels	Joining or Creating a Non-Profit Organization
Developer Guidelines	5. Participating as a Business or a Government Agency
Documentation	The Economics of Open Source
Demos, Screenshots, Videos, and Example Output	Types of Corporate Involvement
Hosting	Governments and Open Source
Choosing a License and Applying It	Being Open Source From Day One is Especially Im
The "Do Anything" Licenses	Hire for the Long Term
The GPL	Case study
How to Apply a License to Your Software	Appear as Many, Not as One
Setting the Tone	Be Open About Your Motivations
Avoid Private Discussions	Money Can't Buy You Love
Nip Rudeness in the Bud	Contracting
Codes of Conduct	Review and Acceptance of Changes
Practice Conspicuous Code Review	Update Your RFI, RFP and Contract Language
Be Open From Day One	Open Source IV&V: Use Third-Party Review Thro
Opening a Formerly Closed Project	Don't Surprise Your Lawyers
Announcing	Funding Non-Programming Activities
3. Technical Infrastructure	Quality Assurance (i.e., Professional Testing)
What a Project Needs	Legal Advice and Protection
Web Site	Documentation and Usability
Canned Hosting	Providing Hosting/Bandwidth
Mailing Lists / Message Forums	Providing Build Farms and Development Servers
Choosing the Right Forum Management Software	Sponsoring Conferences, Hackathons, and other Developer Meetings
Version Control	Marketing
Version Control Vocabulary	Open Source and Freedom from Vendor Lock-In
Choosing a Version Control System	Remember That You Are Being Watched
Using the Version Control System	Don't Bash Competing Open Source Products
Receiving and reviewing contributions	Don't Bash Competing Vendors' Developers
Bug Tracker	"Commercial" vs "Proprietary"
Interaction with Email	Open Source and the Organization
Pre-Filtering the Bug Tracker	Dispel Myths Within Your Organization
	Foster Pools of Expertise in Multiple Places
	Don't Let Publicity Events Drive Project Schedule
	The Key Role of Middle Management

+ 📖	Chapter 1. Introduction
+ 📖	Chapter 2. Getting Started
+ 📖	Chapter 3. Technical Infrastructure
+ 📖	Chapter 4. Social and Political Infrastructure
+ 📖	Chapter 5. Participating as a Business or a Government Agency
+ 📖	Chapter 6. Communications
+ 📖	Chapter 7. Packaging, Releasing, and Daily Development
+ 📖	Chapter 8. Managing Participants
+ 📖	Chapter 9. Legal Matters: Licenses, Copyrights, Trademarks and Patents
📖	Appendix A. Canned Hosting Sites
📖	Appendix B. Obsolete Appendix (was: Free Version Control Systems)
📖	Appendix C. Obsolete Appendix (was: Free Bug Trackers)
📖	Appendix D. Obsolete Appendix: (was: Why Should I Care What Color the Bikeshed Is?)
📖	Appendix E. Obsolete Appendix (was: Example Instructions for Reporting Bugs)
📖	Appendix F. Copyright

Open Source Project with KUPE

Open Source Project with KUPE

Open Source Project with KUPE

Main Idea

- KUPE 를 open source project 에 사용할 수 있도록 적합하게 변형 혹은 추가.
- Open source project 의 개발 방법론은 프로젝트에 따라 다름.
 - 개발 프로세스로 KUPE 를 이용
- Open source project 의 주요점은 communication & collaboration.
 - 개발 프로세스 보다 개발 환경에 대한 제안을 제시.
 - KUPE에 없는 communication 과 collaboration 을 위한 방법을 추가.

What is KUPE?

- KUPE (Konkuk Unified Process for Education)
 - A software process based on RUP
 - Revision of OSP (by Tailored to SE classes in universities) and OOPT
- Characteristics of KUPE
 - 3 Stages
 1. Iterative : Multiple development cycles
 2. Incremental : System grows incrementally as each cycle is completed
 3. Architecture : Stage > Cycle > Phase > Activity

```
graph LR; A[1000 Plan and Elaboration] --> B[2000 Build]; B --> C[3000 Deployment]
```

1000 Plan and Elaboration → 2000 Build → 3000 Deployment

DEPENDABLE SOFTWARE LABORATORY

KU KONKUK UNIVERSITY

2

Open Source Project with KUPE

communication & collaboration

- KUPE의 전신인 OSP(OOAD & CM) 의 경험을 바탕으로 추가.

팀 프로젝트 주제 : OOO 엘리베이터 컨트롤러 시뮬레이터

- JAVA / GUI / MS Windows / Eclipse
- 도구 사용 (UML, SVN, JUnit, Mantis, Redmine 등 --> [소프트웨어뱅크](#) 자료 참고하세요.)

2014년도 소프트웨어 모델링 학부 수업.

- 4학년 "소프트웨어 검증" 수업과 연동해서 진행합니다.
- 4학년 수업에서 수행한 시스템 테스트 및 정적분석 결과를 반영하여 다음 cycle을 진행합니다. (시스템테스트대응서 및 정적분석대응서 제출)
- CTIP 환경을 전수받아 설치한 후, 4학년 학생팀과, SVN을 사용하여 코드를 공유하고, Mantis를 이용하여 이슈를 관리합니다.
- CTIP 환경에서 JUnit을 사용하여 단위시험을 수행한 후 단위시험보고서(단위시험)를 제출 및 발표합니다.
- 4학년 수업에서 수행한 정적분석보고서를 결과를 반영하여 정적분석대응서(커버리지 분석, 모듈간 의존성 분석)를 제출 및 발표합니다.

팀명	팀원	보고서	팀발표 #1	팀발표 #2	팀발표 #3	팀발표 #4	팀발표 #5	팀발표 #6	팀발표 #7
		Introduction to UML	OSP Stage 1000	OSP Stage 2030	OSP Stage 2040 Stage 2050	SVN, Mantis, JUnit 사용법	OOP 1 st Cycle	OSP 2 nd Cycle	OSP 3 rd Cycle
T2	김민우 201111389 김재엽 201111344 최하나 201211386	보고서	발표자료 보고서 v2 v3 wmv	발표자료 보고서 v2 v3 wmv	발표자료 보고서 v2 v3 wmv	발표자료 wmv	발표자료 v2 프로그램 v2 단위시험보고서 wmv	발표자료 시스템테스트대응서 단위시험보고서 프로그램 wmv	발표자료 시스템테스트대응서 단위시험보고서 정적분석대응서 프로그램 wmv

- **JAVA Program Testing (System Testing)**
- 수업 시간에 학습한 이론을 기반으로 Pairwise Testing 또는 Category-Partition Testing을 수행합니다.
- **CTIP for JAVA 환경 구축**
- Mantis, SVN, Ant, CruiseControl, Eclipse, JUnit, Clover, JDepend 사용 (Redmine도 사용해 보세요.) System Testing (+Requirement Management)
- **Static Analysis**
- Prevent, Dart, QAC, PolySpace, Eclipse TPTP, cppcheck, Sonar 등 3개 이상 사용하여 유용한 분석을 수행합니다.

- 3학년 "소프트웨어 모델링 및 분석" 수업과 연동해서 진행합니다.
- 3학년 수업의 개발 결과물에 대해서 시스템 테스트를 수행한 후 결과를 Mantis를 사용하여 공유합니다.
- 3학년 수업에서 단위 테스트 및 관련 분석 (5,6,7번 발표)을 수행할 수 있도록 CTIP 환경을 제공합니다.
- 테스트 결과에 대한 의견교환은 Mantis 등을 사용합니다. 다른 도구를 사용하셔도 됩니다.
- 테스트를 위한 소스코드는 SVN을 통해 공유합니다.
- NIPA SW 벤처에서 제공하는 다양한 도구를 사용할 수 있습니다.

2015년도 소프트웨어 검증 학부 수업.

팀명	팀원	검증대상	팀발표 #1	팀발표 #2	팀발표 #3	팀발표 #4	팀발표 #5	팀발표 #6
			JUnit, Eclipse, 및 빌드환경	Mantis, SVN & CTIP	정적분석 도구 & 시스템테스트 도구	1 st Testing - System Test	2 nd Testing - System Test - Static Analysis	Final Presentation
T4	201211389 함진아 201010768 최정환 200911393 박현규 201111339 김민우	SMA T4 T6	발표자료 wmv 2	발표자료 wmv	발표자료 wmv	발표자료 시스템테스트보고서4 6 wmv	발표자료 시스템테스트보고서4 6 정적분석보고서4 6 wmv	발표자료 wmv

Open Source Project with KUPE

communication & collaboration

- KUPE 에서 사용하기 위한 개발 환경 구축을 위한 가이드라인 혹은 개발 환경을 제공
→communication 과 collaboration 을 위한 방법
- 제안할 방법
 1. 프로젝트 통합 관리를 위한 페이지
 2. 프로젝트 설명을 위한 문서 관리 툴 or Space
 3. 프로젝트 개발을 진행 사항 이슈를 확인할 수 있는 페이지

Open Source Project with KUPE

Application to Project

Open Source Project – Application to Project

소스 코드 유사도 검사 프로그램 개발 및 개발 커뮤니티 구축 (Source Code Similarity Checker develop & develop community build)

- 2016년도 1학기 종합 설계(졸업 작품) 프로젝트
- 팀원: 김민우, 오현택, 최수용

- 개발한 프로그램: TwoDoX

Open Source Project – Application to Project

Redmine

- 웹 기반 프로젝트 관리 프로그램
- 프로젝트에서 진행되는 이슈 사항들 관리, 소스 관리, 일정 관리 등 전반적인 진행 상황을 모니터링 하고 관리 할 수 있는 도구

OpenSourceProject Redmine

초기화면

Open Source Project 관리를 위한 Redmine 입니다.

상단의 프로젝트 탭을 통해 프로젝트 구성을 확인할 수 있습니다.

개발을 참여하기 위해서는 TwoDox 프로젝트의 게시판을 이용해주세요!

레드마인 가입은

ID ==> 학번으로 가입하시고, 이름과 성에 실명을 사용해주세요 바랍니니다.

Open Source Project – Application to Project

Redmine

OpenSourceProject Redmine

프로젝트

새 프로젝트 | 모든 일감 보기 | 총 소요시간 | 전체 작업내역

OSS Project

TwoDoX

TwoDoX-Client

WorkingSet-cloneCheck Func1

TwoDoX-DistributedModule

★ 내 프로젝트

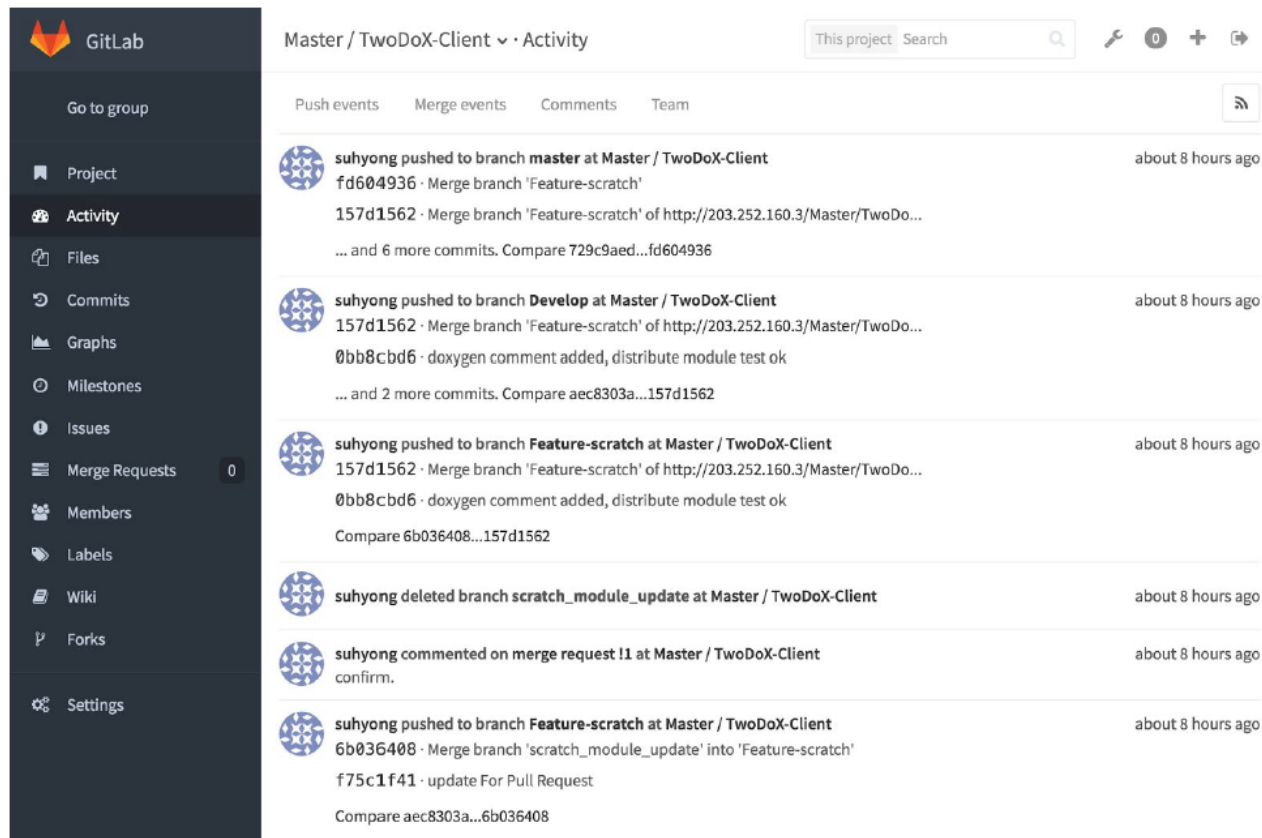
내보내기 Atom

게시판	주제	글	마지막 글
개발 참가 요청 TwoDox 개발에 참가 신청 게시판	0	0	
Q&A 질문게시판	0	0	
개발환경 개발환경에 대한 Q&A	0	0	
기능 기능에 대한 Q&A	0	0	
아이디어 제안 TwoDox 아이디어 제안	0	0	
기능 추가 기능 추가에 대한 아이디어	0	0	
개선 및 수정 사항 TwoDox 개선 사항, 수정 사항	0	0	

Open Source Project – Application to Project

버전 관리 프로그램

- 하나의 프로그램이 여러 고객의 필요를 맞추기 위해, 다양한 플랫폼을 대응하기 위해 다양한 버전의 프로그램으로 변화될 수 있다.
- 효과적으로 소스를 관리 할 수 있게 도와주는 버전 관리 프로그램 도구

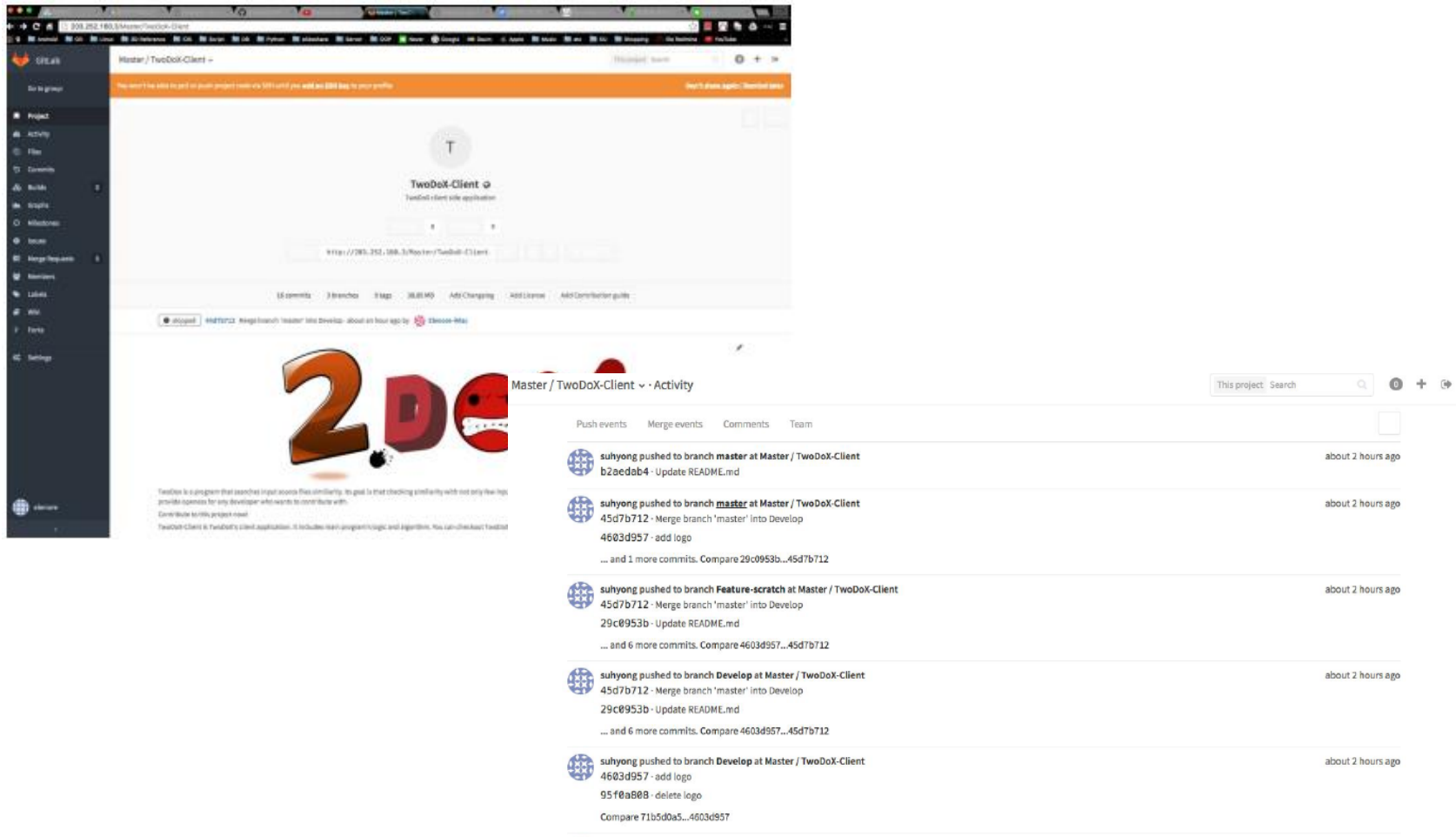


The screenshot shows the GitLab interface for the 'TwoDoX-Client' project. The left sidebar contains navigation options: Go to group, Project, Activity (selected), Files, Commits, Graphs, Milestones, Issues, Merge Requests (0), Members, Labels, Wiki, Forks, and Settings. The main content area displays the 'Activity' for the 'Master' branch. It lists several events:

- suhyong pushed to branch master at Master / TwoDoX-Client** (about 8 hours ago): fd604936 - Merge branch 'Feature-scratch' (157d1562 - Merge branch 'Feature-scratch' of http://203.252.160.3/Master/TwoDo...). Includes 6 more commits. Compare 729c9aed...fd604936.
- suhyong pushed to branch Develop at Master / TwoDoX-Client** (about 8 hours ago): 157d1562 - Merge branch 'Feature-scratch' of http://203.252.160.3/Master/TwoDo... (0bb8cbdb6 - doxygen comment added, distribute module test ok). Includes 2 more commits. Compare aec8303a...157d1562.
- suhyong pushed to branch Feature-scratch at Master / TwoDoX-Client** (about 8 hours ago): 157d1562 - Merge branch 'Feature-scratch' of http://203.252.160.3/Master/TwoDo... (0bb8cbdb6 - doxygen comment added, distribute module test ok). Compare 6b036408...157d1562.
- suhyong deleted branch scratch_module_update at Master / TwoDoX-Client** (about 8 hours ago).
- suhyong commented on merge request !1 at Master / TwoDoX-Client** (about 8 hours ago): confirm.
- suhyong pushed to branch Feature-scratch at Master / TwoDoX-Client** (about 8 hours ago): 6b036408 - Merge branch 'scratch_module_update' into 'Feature-scratch' (f75c1f41 - update For Pull Request). Compare aec8303a...6b036408.

Open Source Project – Application to Project

Gitlab



The screenshot shows the GitLab web interface for a project named 'TwoDoX-Client'. The page is divided into a header, a main content area, and an activity feed.

Header: The top navigation bar includes 'Master / TwoDoX-Client' and a search bar. Below the navigation bar, there is a message: 'You weren't able to add this project because you don't have the necessary permissions to add it to your profile. Search for an app to import this project.' The project name 'TwoDoX-Client' is displayed with a 'T' icon, and a description: 'TwoDoX-Client is a client application'. Below this, there are statistics: '11 files / 285,252,108 bytes / TwoDoX-Client'. At the bottom of the header, there are links for '15 commits', '3 branches', '3 tags', '38.0k MB', 'Add Changelog', 'Add License', and 'Add Contributor guide'.

Main Content Area: The main content area features a large, stylized logo for '2DOX' in orange and red. Below the logo, there is a description: 'TwoDoX is a program that searches input source files similarity. Its goal is that checking similarity with not only file logs provide licenses for any developer who wants to contribute with. Core code for this project here! TwoDoX-Client is TwoDoX's client application. It includes main program logic and algorithms. You can check here TwoDoX'.

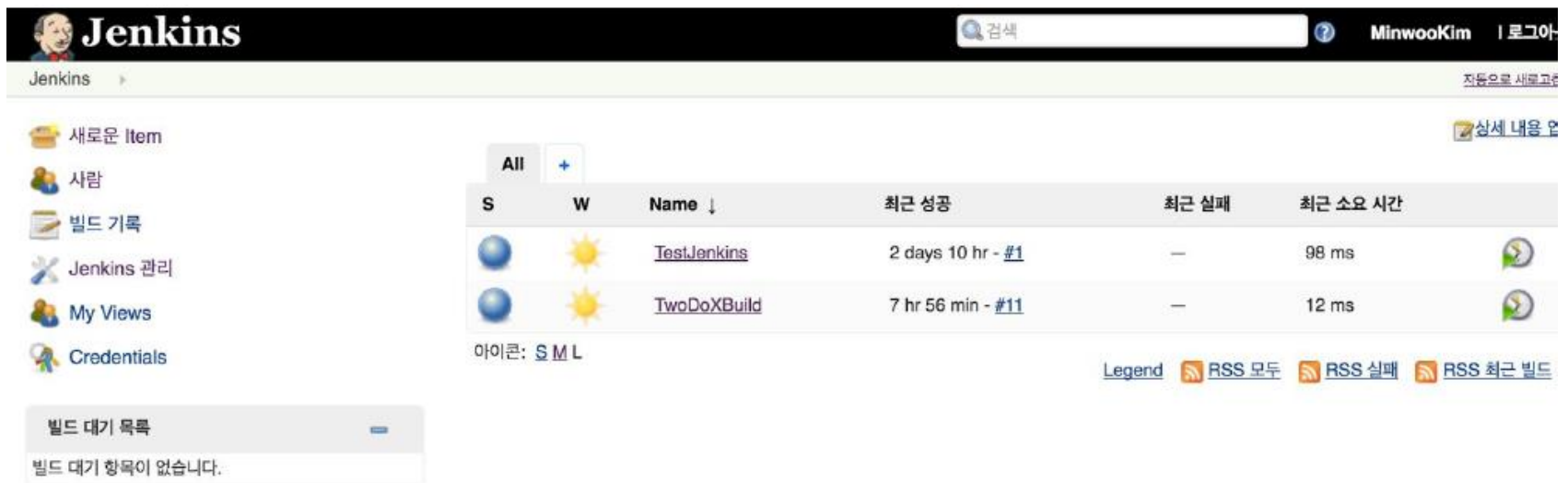
Activity Feed: The activity feed is titled 'Master / TwoDoX-Client · Activity' and includes a search bar. It lists several push events:

- suhyong pushed to branch **master** at **Master / TwoDoX-Client** about 2 hours ago
b2aedab4 · Update README.md
- suhyong pushed to branch **master** at **Master / TwoDoX-Client** about 2 hours ago
45d7b712 · Merge branch 'master' into Develop
4603d957 · add logo
... and 1 more commits. Compare 29c0953b...45d7b712
- suhyong pushed to branch **Feature-scratch** at **Master / TwoDoX-Client** about 2 hours ago
45d7b712 · Merge branch 'master' into Develop
29c0953b · Update README.md
... and 6 more commits. Compare 4603d957...45d7b712
- suhyong pushed to branch **Develop** at **Master / TwoDoX-Client** about 2 hours ago
45d7b712 · Merge branch 'master' into Develop
29c0953b · Update README.md
... and 6 more commits. Compare 4603d957...45d7b712
- suhyong pushed to branch **Develop** at **Master / TwoDoX-Client** about 2 hours ago
4603d957 · add logo
95f0a808 · delete logo
Compare 71b5d0a5...4603d957

Open Source Project – Application to Project

자동 빌드 서버

- 환경 파일의 설정에 따라 전체 프로젝트를 쉽게 빌드 할 수 있는 환경을 제공해주는 도구



The screenshot shows the Jenkins web interface. At the top, there is a search bar and the user name 'MinwooKim'. The main content area displays a table of build jobs. The table has columns for 'S' (Success), 'W' (Warning), 'Name', '최근 성공' (Last Success), '최근 실패' (Last Failure), and '최근 소요 시간' (Last Duration). Two jobs are listed: 'TestJenkins' and 'TwoDoXBuild'. Below the table, there are links for 'Legend', 'RSS 모두', 'RSS 실패', and 'RSS 최근 빌드'. On the left side, there is a sidebar with navigation links like '새로운 Item', '사람', '빌드 기록', 'Jenkins 관리', 'My Views', and 'Credentials'. At the bottom left, there is a '빌드 대기 목록' (Build Queue) section which is currently empty.

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
		TestJenkins	2 days 10 hr - #1	—	98 ms
		TwoDoXBuild	7 hr 56 min - #11	—	12 ms

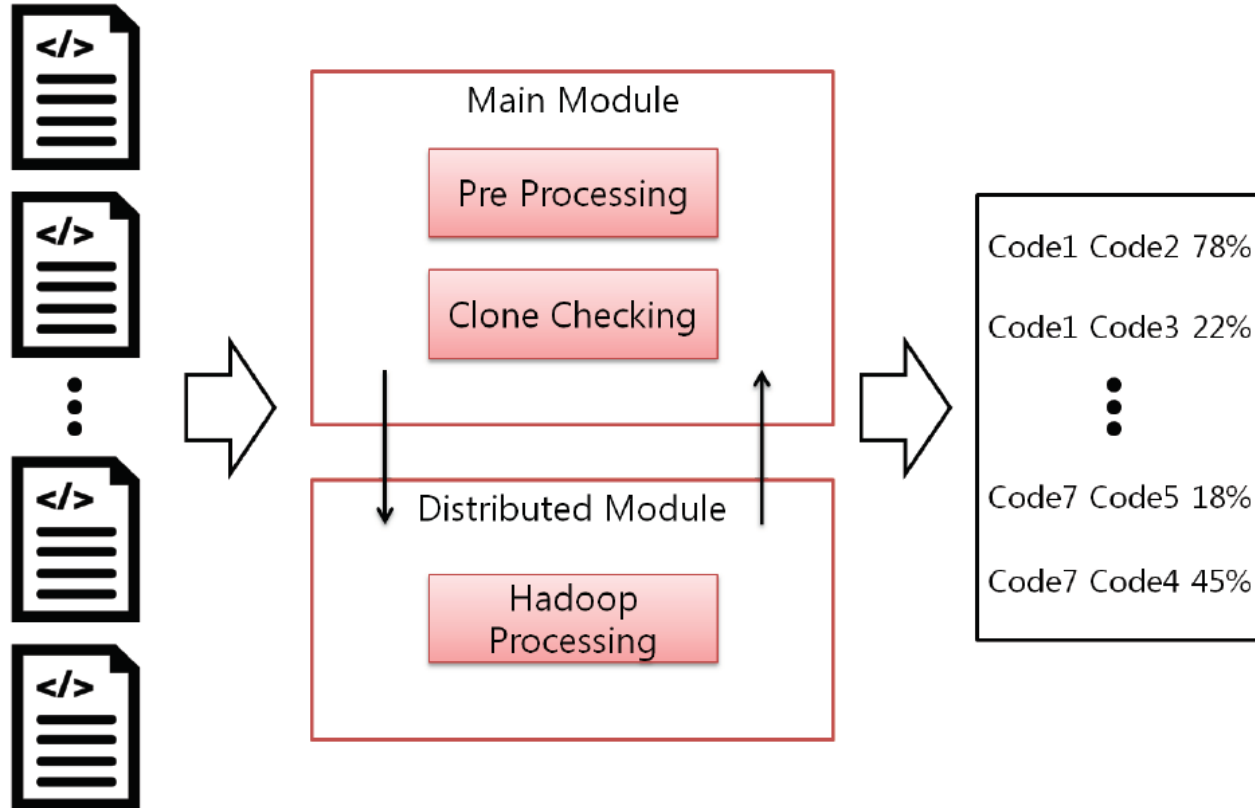
Open Source Project – Application to Project

TwodoX Main Homepage

The screenshot shows the TwodoX main homepage. The browser address bar displays '203.252.160.3:9000'. The navigation menu includes 'GETTING STARTED', 'TUTORIAL', 'DOWNLOAD', 'COMMUNITY', and 'ABOUT'. The main heading is 'Getting Started' with the subtitle 'An overview of TwoDoX, how to download and install TwoDoX, and more.' Below this, there is a sidebar with links: 'Introduction of TwoDoX', 'What's included?', 'Download', 'News', and 'License'. The main content area is titled 'Introduction of TwoDoX' and features a section 'What is Clone checker?' with a Korean description: 'TwoDoX는 소스 코드의 유사도를 검사하는 프로그램입니다. 적은 수의 소스 파일에 대한 유사도 검사 뿐만 아니라 1,500개 이상의 소스 파일에 대한 빠르고 정확한 유사도 검사를 목표로 합니다.' Below the text is a flow diagram illustrating the clone checker process. It starts with a stack of code files on the left, which feed into a 'Main Module' containing 'Pre Processing' and 'Clone Checking' steps. The 'Main Module' is connected to a 'Distributed Module' containing 'Hadoop Processing'. The output of the process is a list of code files and their similarity percentages: 'Code1 Code2 78%', 'Code1 Code3 22%', 'Code7 Code5 18%', and 'Code7 Code4 45%'.

Open Source Project – Application to Project

프로그램 개요



Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Redmine

- 프로젝트의 진행 상황을 확인할 수 있는 dashboard 형태



Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Redmine

- 새 일감 만들기 새로운 일감 등록 / 담당자, 중요도, 날짜 등을 지정할 수 있다.

새 일감만들기

유형 * 새기능

제목 * 새로운 기능 추가

설명
기능 추가에 대한 설명

상태 * 신규

우선순위 * 보통

담당자 jns Admin

시작시간 2015-03-31

완료기한 2015-04-03

추정시간 6 시간

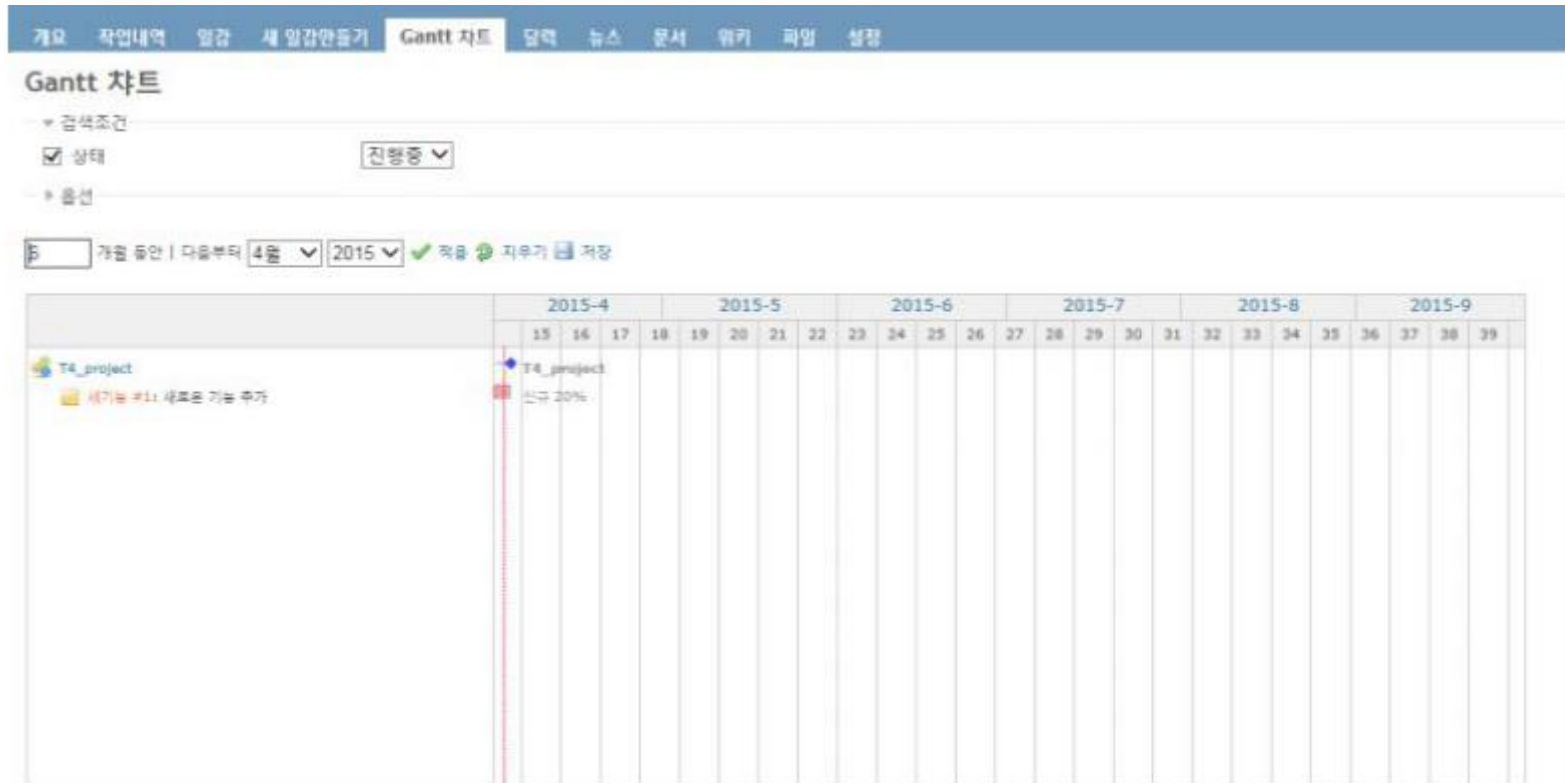
진척도 10%

만들기 취소 계속하기 다시보기

Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

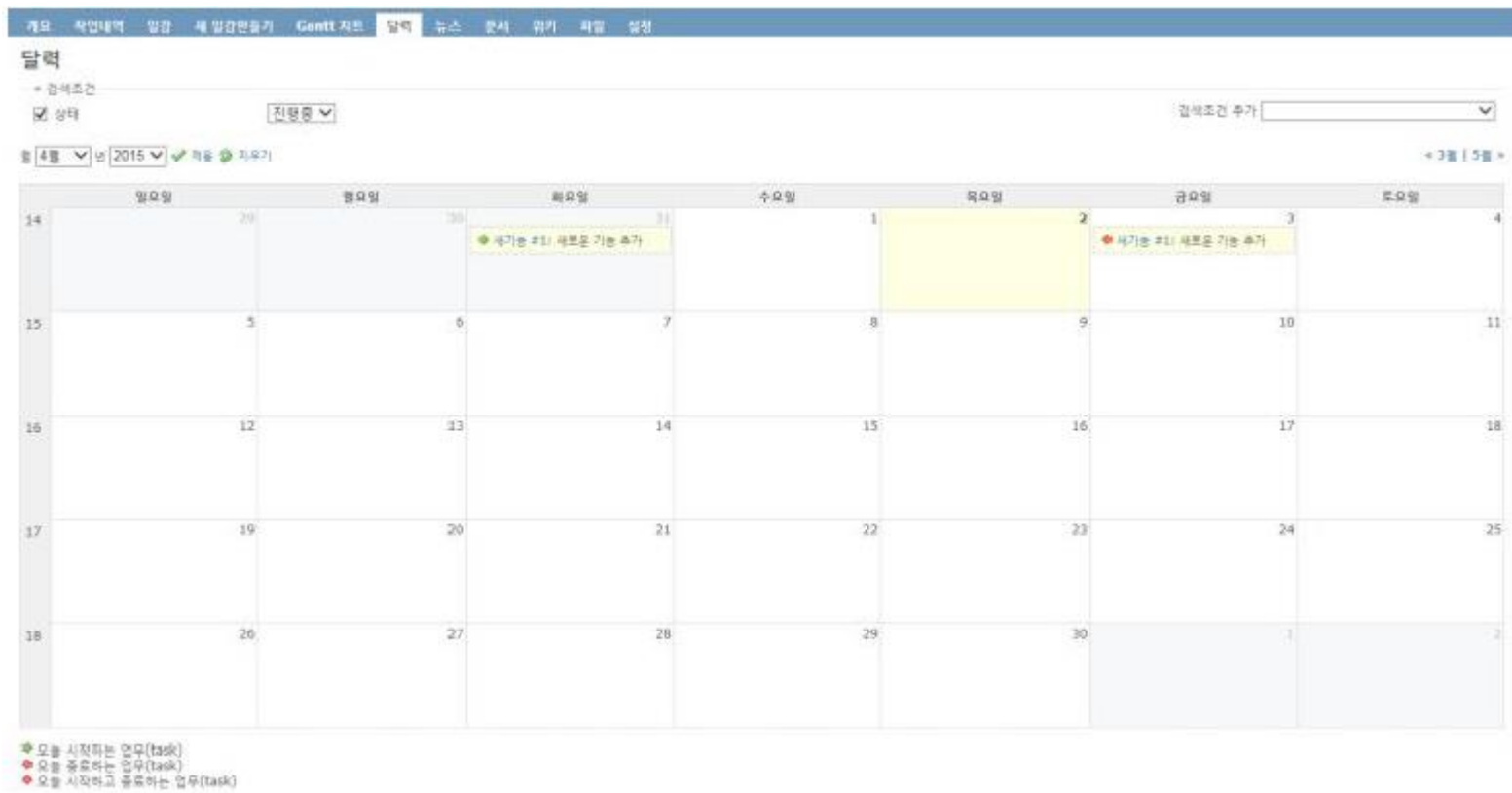
- Redmine
 - Gantt 차트 차트/ 달력 형태로 진행 상황 파악 가능



Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

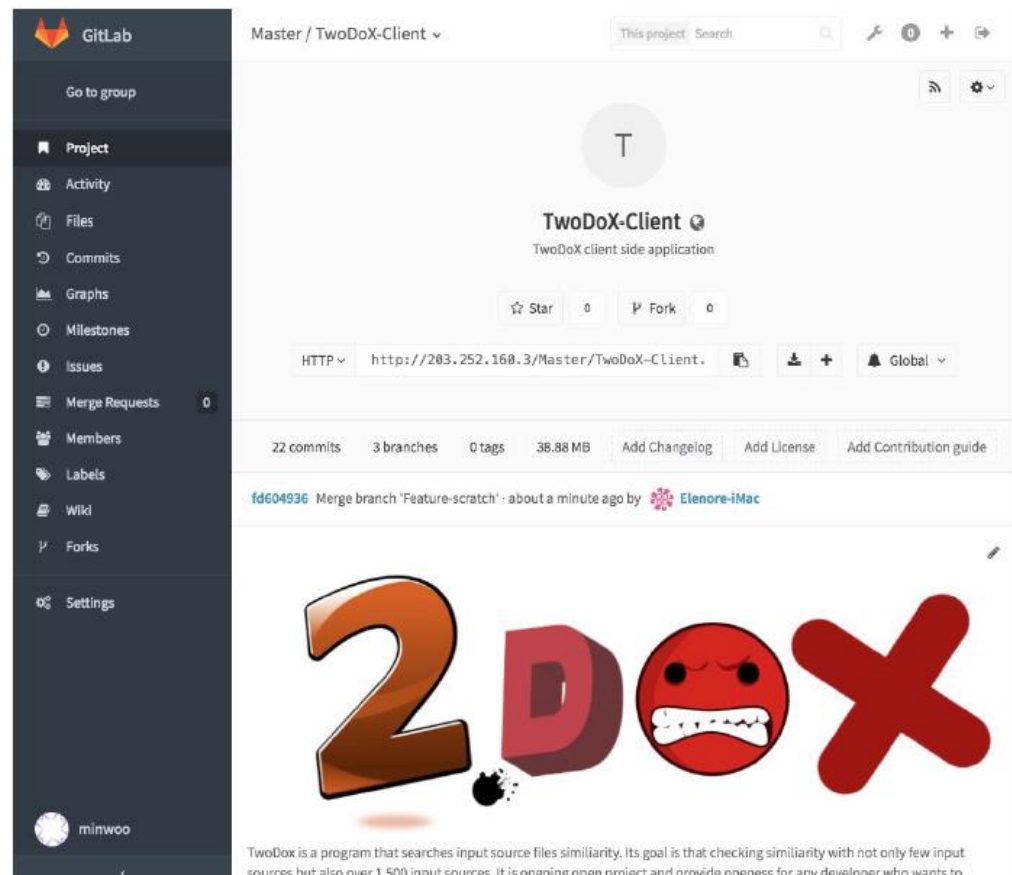
- Redmine
 - 달력 차트/ 달력 형태로 진행 상황 파악 가능



Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Gitlab
 - GitLab 페이지에서 소스코드를 받아온다.

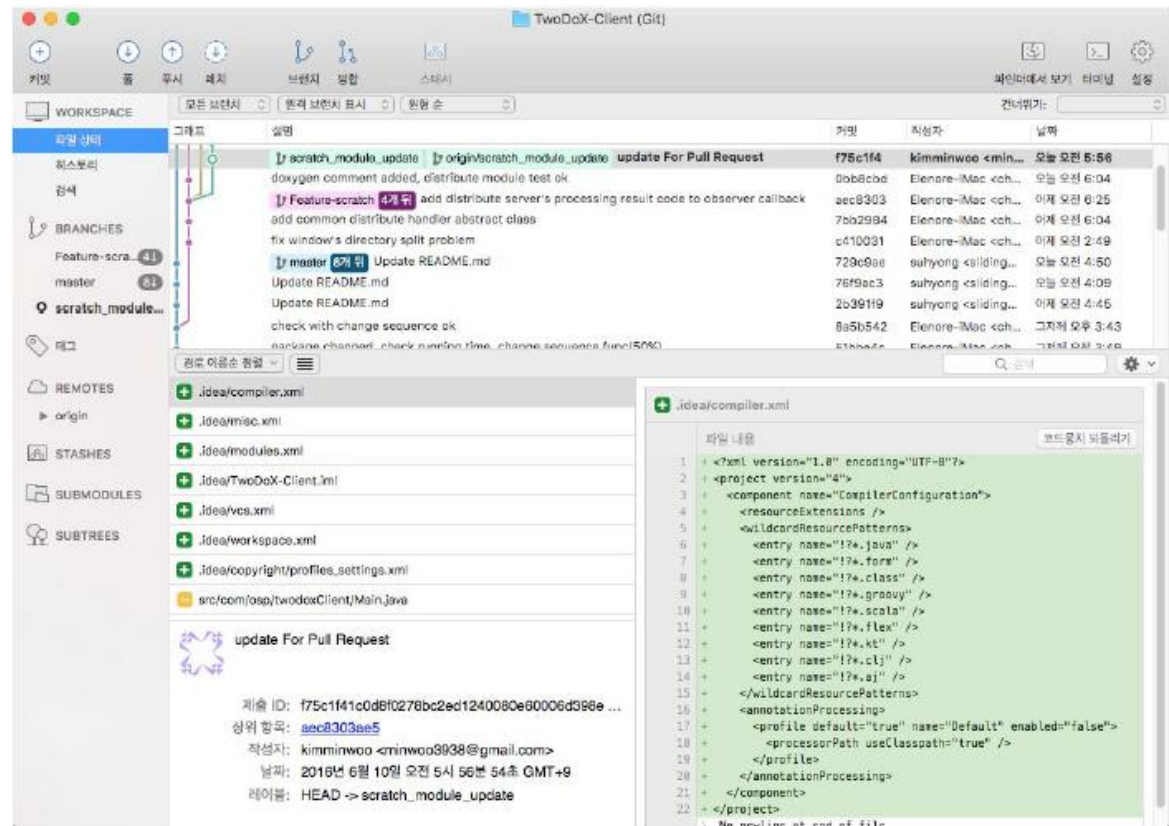


Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Gitlab

- Git 을 사용하기 위한 Desktop application (e.g. Source Tree, github 등)을 사용.
- Source Tree 사용



Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Gitlab
 - 개발을 완료한 후 branch를 push한다.

TwoDoX-Client (Git)

다음 저장소에 푸시: origin http://203.252.160.3/Master/TwoDoX-Client.git

푸시할 브랜치들	원래 브랜치	추적?
<input type="checkbox"/> Feature-scratch	Feature-scratch	↕ -
<input type="checkbox"/> master	master	↕ -
<input checked="" type="checkbox"/> scratch_module_update	scratch_module_update	↕ -

모두선택

모든 태그 푸시

취소 확인

update For Pull Request

제출 ID: f75c1f41cd8f0278bc2ed1240080e60006d398e ...
상위 항목: [a6c8303ae5](#)
작성자: kimminwoo <minwoo3938@gmail.com>
날짜: 2016년 6월 10일 오전 5시 56분 54초 GMT+9
레이블: HEAD -> scratch_module_update

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="CompilerConfiguration">
    <resourceExtensions />
    <wildcardResourcePatterns>
      <entry name="!*.*.java" />
      <entry name="!*.*.form" />
      <entry name="!*.*.class" />
      <entry name="!*.*.groovy" />
      <entry name="!*.*.scala" />
      <entry name="!*.*.flex" />
      <entry name="!*.*.kt" />
      <entry name="!*.*.clj" />
      <entry name="!*.*.aj" />
    </wildcardResourcePatterns>
    <annotationProcessing>
      <profile default="true" name="Default" enabled="false">
        <processorPath useClasspath="true" />
      </profile>
    </annotationProcessing>
  </component>
</project>
No newline at end of file
```


Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Gitlab

- GitLab에서 개발을 위해 새로 만든 branch와 기존 branch와의 merge를 위해 merge request(pull request)를 한다.

The screenshot displays the GitLab Merge Requests interface. On the left is a dark sidebar with navigation links: Go to group, Project, Activity, Files, Commits, Graphs, Milestones, Issues, Merge Requests (0), Members, Labels, Wiki, and Forks. The main content area is titled 'New Merge Request' and shows two columns: 'Source branch' and 'Target branch'. The Source branch is set to 'Master/TwoDoX-Client' and 'Feature-scratch'. The Target branch is set to 'Master/TwoDoX-Client' and 'Develop'. Both columns show a merge request for 'Merge branch 'Feature-scratch' of http://203.252.160.3/Master/TwoDoX-Client into Feature-scratch' by Elenore-IMac about an hour ago. A green button at the bottom says 'Compare branches and continue'.

Open Source Project – Application to Project

Redmine 과 Gitlab 의 사용법 – 개발자

- Gitlab

- 관리자의 merge 승인이 나면 기존 branch와 merge 된다.

The screenshot displays a GitLab Merge Request interface. At the top, it shows the project path 'Master / TwoDoX-Client' and the title 'Merge Requests'. A search bar and utility icons are visible. The main content area features a 'Merged' status badge, a title 'Merge Request #11', and a description 'update For Pull Request' with the comment 'plz merge'. It indicates the request was opened by 'minwoo' and is now merged by 'suhyong'. A summary box states: 'Merged by suhyong about an hour ago. The changes were merged into Feature-scratch. The source branch has been removed.' Below this, there are buttons for 'Revert' and 'Cherry-pick'. The interface also shows a 'Discussion' section with 1 comment, 1 commit, and 8 changes. A comment from 'suhyong @elenore' confirms the merge. The right sidebar contains metadata: '1 of 1' items, assignee 'suhyong @elenore', no milestone, and 2 participants. A notification box indicates the user is subscribed to the thread.

Open Source Project – Application to Project

오픈소스 프로젝트 기여 시나리오

1. TwoDoX redmine 프로젝트에서 개발하려는 기능을 선택하고 프로젝트에 대한 정보를 확인한다.
 - 해당 기능이 Client 인 경우 Client redmine
 - distributed moduel 인 경우 distributed module redmine

TwoDoX

개요 작업내역 일감 새 일감만들기 Gantt 차트 달력 뉴스 문서 위키 게시판 파일 설정

Wiki

TwoDoX를 진행하기 위해 참조해야할 WIKI 입니다.

분류

1. Contribution 방법
 1. Working Set 에 대한 분류
 2. 각 Working Set에 대한 설명
2. Client code 에 대한 Documents
3. Distributed moduel에 대한 Documents

기타 문의는 minwoo3938@gmail.com 으로 해주시기 바랍니다.

Open Source Project – Application to Project

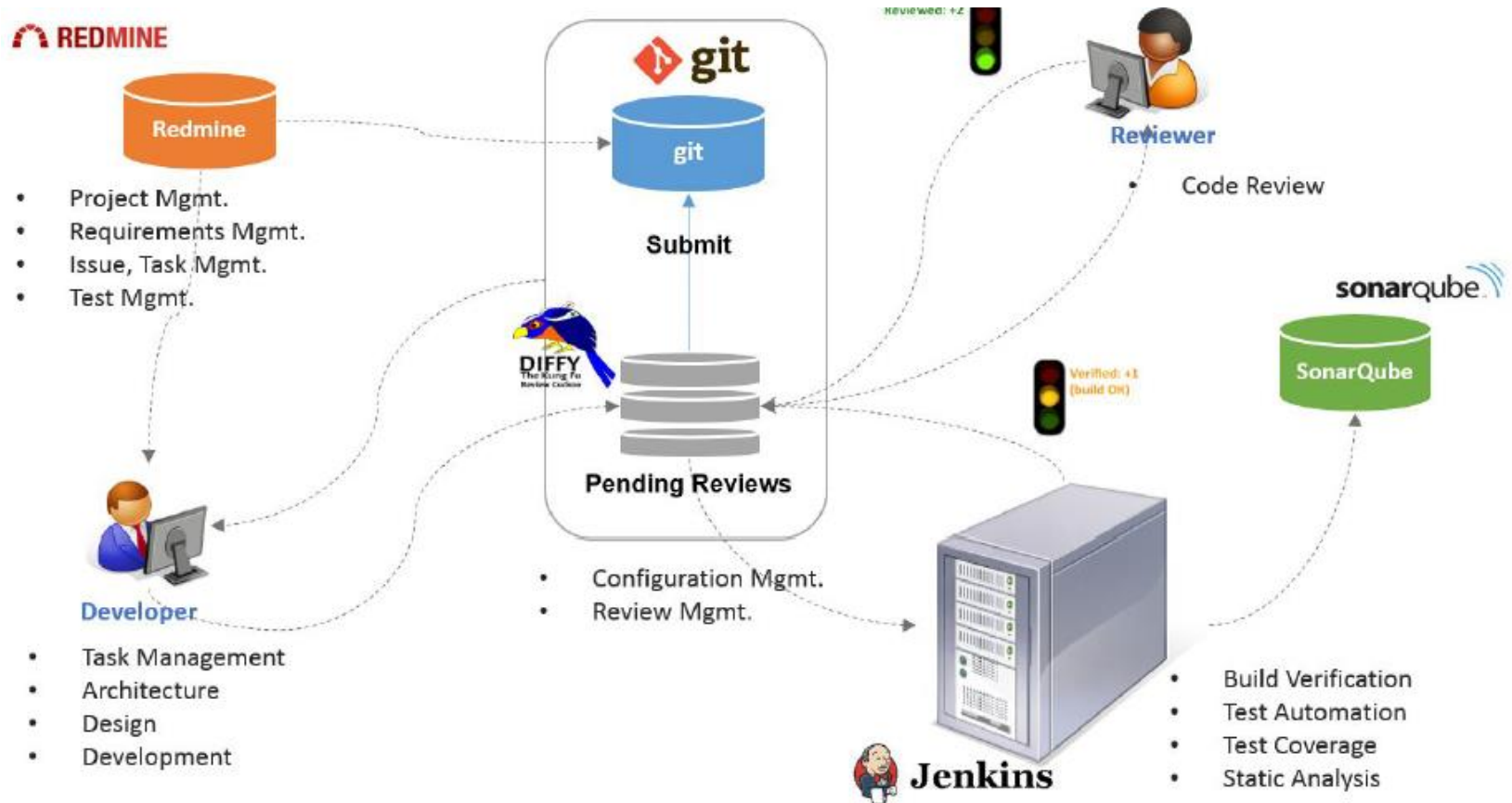
오픈소스 프로젝트 기여 시나리오

2. 해당 프로젝트의 Gitlab에서 코드를 확인한 뒤 git 을 통하여 코드를 받아 기능 개발을 진행 한다.
3. 기능 개발이 완료 된 후, gitlab에서 merge request를 하고, project manager의 승인을 기다리면 된다.

The image displays two overlapping screenshots of the GitLab web interface. The background screenshot shows the 'Files' view for a repository named 'TwoDoX-Client'. The file browser lists directories like '.idea', 'images', 'lib', 'sample_input/scratch/input', and 'src/com/osp/twodoxClient', along with a 'README.md' file. The foreground screenshot shows a 'Merge Requests' page for the same repository. It displays a merged merge request titled 'update For Pull Request' by user 'minwoo', which was merged into the 'Feature-scratch' branch. The merge request details include the assignee 'suhyong @elenore', a milestone of 'None', and 2 participants. The merge status is 'Merged by suhyong about 8 hours ago'. The interface also shows navigation options like 'Revert' and 'Cherry-pick', and a discussion section with 1 comment, 1 commit, and 8 changes.

Open Source Project – Application to Project

오픈소스 프로젝트 관리 시나리오



Open Source Project with KUPE

Open Source Project – Class

Open Source Project with KUPE in Class

목표

1. 소프트웨어 모델링 + 소프트웨어 검증 수업에서 open source project 를 대상으로 수행 시 수행 방법
2. KUPE 방법론을 통해 Open source project 를 수행하는 수업을 위한 가이드 라인 제시
 - 가이드 라인에 포함되는 내용: 1) 개발 환경, 2) 개발 프로세스 (KUPE), 3) 개발 방법 (agile, lean 등의 방법론)
 - 오픈소스 프로젝트를 실제로 수행하여 오픈 소스 프로젝트에 대한 이해와 참여 방법을 익힐 수 있도록 수업 과정을 제시.

Q&A

Reference – relate open source project

- Karl Fogel, (2005-2016), **Producing Open Source Software : How to Run a Successful Free Software Project**
- **Software Release Practice HOWTO by Eric Raymond**
- Sharma, S., Sugumaran, V. and Rajagopalan, B. (2002), A framework for creating hybrid-open source software communities. *Information Systems Journal*, 12: 7–25.
- Robbins, J. E. (2003). *Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools*. *Making Sense of the Bazaar: Perspectives on Open Source and Free Software*, Fall 2003.
- Fuggetta, A. (2003). *Open source software – an evaluation*, *Journal of Systems and Software*, 66, 77 – 90.
- Mockus, A., Fielding, R. & Herbsleb, J. (2002). *Two case studies of open source software development: Apache and mozilla*, *ACM Transactions on Software Engineering and Methodology* 11 (3), 1 – 38.
- Free Software Movement (gnu.org)
- GNU Project. "What is Free Software". Free Software Foundation.
- Stallman, Richard (2013-05-14). "The advantages of free software". Free Software Foundation. Retrieved 2013-08-12.
- "Open Source". Standish Newsroom. Standishgroup.com. 2008-04-16. Retrieved 2010-08-22.

Reference – relate open source project

- "Downloads". Creative Commons. 2015-12-16. Retrieved 2015-12-24. [urce.com](#) by Richard Fontana (on 13 Aug 2013).
- tinyspeck (2013-11-18). "Glitch is Dead, Long Live Glitch! - Art & Code from the Game Released into Public Domain". [glitchthegame.com](#). Retrieved 2013-12-11.
- "Public domain software". Categories of free and nonfree software. Free Software Foundation. Retrieved 2013-07-31
- Puckette, Miller. "Who Owns our Software?: A first-person case study." eContact (September 2009). Montréal: CEC
- Philosophy of the GNU Project ([gnu.org](#))
- What is free software? ([gnu.org](#))
- What is free software? ([fsf.org](#))
- Free as in Freedom (originally the oggcast of the Software Freedom Law Center, it includes a focus on law issues, and other topics)
- Source code snippets and frameworks placed in the public domain on [openwall.info](#).
- Post open source software, licensing and GitHub on [opensourceoftware](#)

Reference – relate RUP, UP

- Ivar Jacobson, Grady Booch, and James Rumbaugh (1999). The Unified Software Development Process
- Gary Pollice, Liz Augustine, Chris Lowe, and Jas Madhur (2003). Software Development for Small Teams: A RUP-Centric Approach
- Per Kroll, Bruce Mac Isaac (2006). Agility and Discipline Made Easy: Practices from OpenUP and RUP
- Walker Royce, Software Project Management, A Unified Framework
- Ahmad Shuja, Jochen Krebs (2007). RUP Reference and Certification Guide
- Ambler, Scott (2002). Agile Modeling: Effective Practices for EXtreme Programming and the Unified Process. J. Wiley.
- Kruchten, Philippe (2004). The Rational Unified Process: An Introduction (3rd Ed.).

Reference – Two DO X

- Open source software 관련 레퍼런스
 - Introduction to CTIP15SV.pdf
 - <http://www.jenkins.or.kr/boards/6/topics/20urlhttps>
- Git
 - <https://git scm.com/about>
- GitLab
 - <https://about.gitlab.com/>
- Redmine
 - <http://www.redmine.org/>
- Open source project
 - <https://opensource.org/>
 - <https://opensource.com/resources/whatopensource>
- DevOps
 - <https://www.sumologic.com/2015/10/20/opensourcedevops/>
- Producing Open Source Software How to Run a Successful Free Software Project
 - <http://producingoss.com/>
- Ebook 오픈 소스 : 오픈 소스 혁명의 목소리 Vol 2
 - http://www.hanbit.co.kr/store/books/look.php?p_code=E8610612218
- 오픈 소스 : 오픈 소스 혁명의 목소리 Vol 1
 - http://www.hanbit.co.kr/store/books/look.php?p_code=E5329835060