University of Nottingham
UK | CHINA | MALAYSIA

COMP2059
Developing Maintainable Software

LECTURE 09 – OPEN SOURCE

Boon Giin Lee (Bryan)

University of Nottingham
UK | CHINA | MALAYSIA

Libraries and Communal Software Development

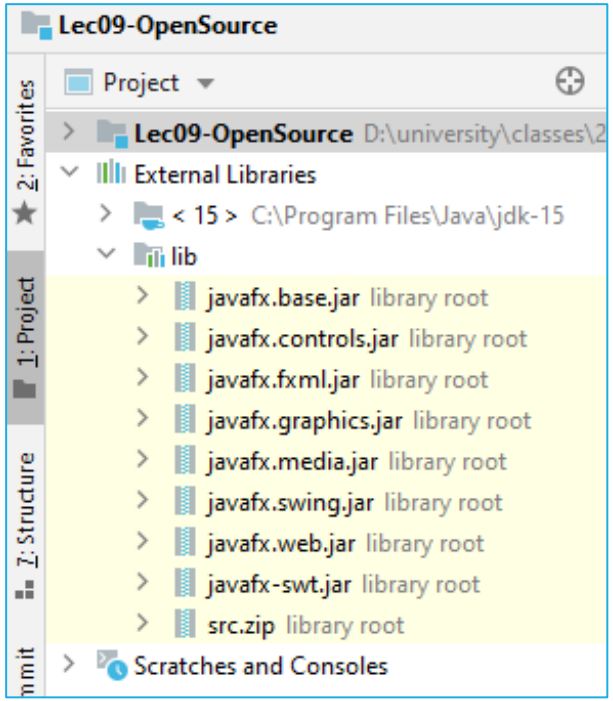
Different Philosophies of Software Development

- During life as a software maintainer, one will work on a number of different forms of projects.
- Some of theses may have open-source code or use open-source libraries.
- One may have to rework private code to be open source or vice versa.
- Will look at an overview of how to use third party code in the form of libraries and open-source projects.

Libraries

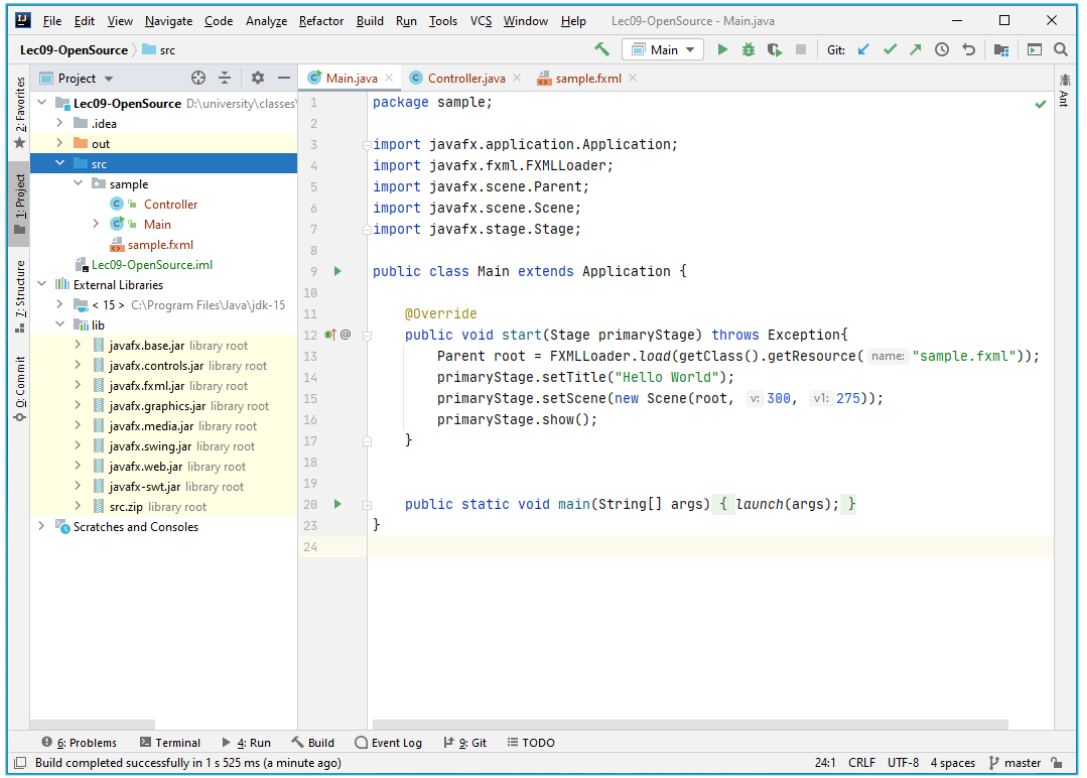
What Is A Library?

- What is a library?
  - Some 3rd party software packaged up (in binaries) and ready-to-use in own code.
  - It is a shared resource.
- Usually online documentation.
  - Supporting guides as well as Javadocs to show the API.
- You've already had experience of this.
  - E.g. using the JDK library in IDE.



What's In A Library

- Libraries in Java consists of
  - A Jar file.
    - Created via the jar tool, or via an IDE.
    - Basically, a zip file.
    - A way of packaging class files and resource files.
    - Note that a runnable JAR file is not a library!
  - An API.
    - Publicly accessible methods.
    - Interface important here; and interface stability.
  - Usually include a license.
    - How to distribute/change it – more later.



Make Use of Libraries in Java

- Need a library file – usually a `.jar` archive.
- Reference it in project.
  - E.g. adds the JAR to Java Build Path.
- Import relevant parts of the library into code.
- Make use of the methods.
  - May need to create an object, or static use access.
- Think about how to distribute the library with code.

Make Use of Libraries in Java

- Can package them up with deployed application, or perhaps include them in the project source distribution.
  - License permitting! (more later …)

- Can also use build files to help with collaborative development using libraries.
  - Script will copy a file from an external resource.
  - This way, do not need a license to re-distribute them as the only one downloading them.

Build Systems and Libraries

- Build system can pull required libraries from remote sources.
- Maven will place these dependencies in a folder locally.

  `(username).\m2`

  `(may be hidden)`

- Can request versions of libraries, so may see different versions of libraries in this folder.

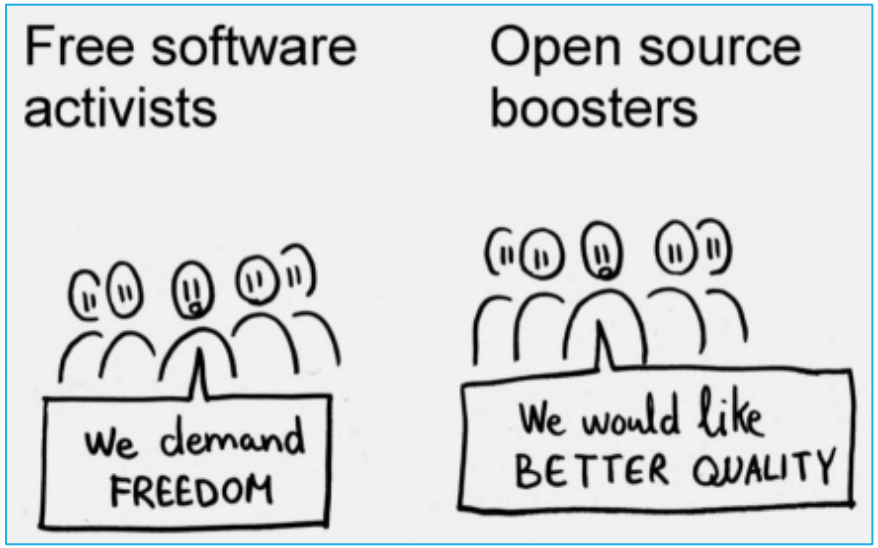
Open-Source Development And Maintenance

Open-Source Software

- What is Open-Source Software (OSS)?
  - OSS is (generally) free software that uses any license approved by the Open-Source Initiative (OSI) from their list of approved open-source licenses.
- What is Free OSS?
  - “Software that gives users rights to run, copy, distribute, change and improve it as they see it, without them asking permission from or make payments to any external group or person.”
- Open-source initiative.
  - https://opensource.org/

Free Software Foundation

- Richard Stallman: President and Founder of the Free Software Foundation.



Free software, free society: Richard Stallman at TEDxGeneva 2014
https://youtu.be/Ag1AKIl_2GM

Open-Source Software

- Why Go Open Source?
  - Customisable.
  - Improvable.
  - Collaborative bug finding/fixing.
  - Redistributable.
  - Transparency.
  - Free.

  - Freedoms 0: Run it
  - Freedoms 1: Change/modify.
  - Freedoms 2: Redistribute – free or sell.
  - Freedoms 3: Redistribute with charges.

Open-Source Software

- Starting to be recognised as a "Good Thing" by industry. Why?
  1. Good advertising.
  2. Attract talented developers.
     - More development possible.
     - Version-tracked contributions are good demonstration of potential employees' programming skill.
     - Develop outside of core skill set.
  - https://youtu.be/a8fHgx9mE5U

Open-Source Software

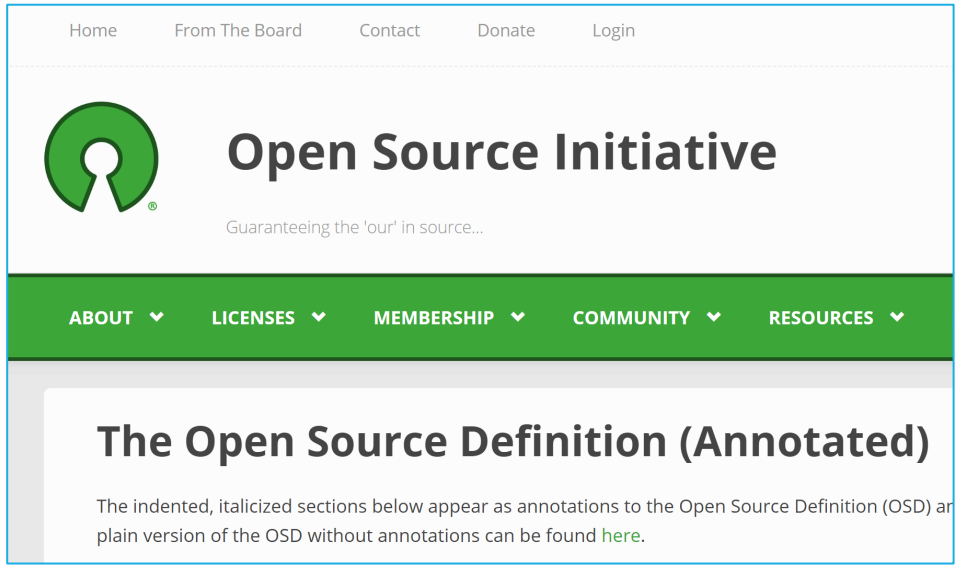
- Examples:
  - OpenJDK; Apache; LaTeX; Moodle; Firefox; Android; Linux; MySQL; OpenOffice; Blender; VLC; IntelliJ CE; Eclipse.
- Top "Open Source" companies.
  - Adobe; Docker; Facebook; GitHub; Google; Gradle; Huawei; IBM; Intel; LinkedIn; Microsoft; MongoDB; Oracle; Red Hat; Samsung Electronics; Twitter; …
  - https://www.datamation.com/open-source/35-top-open-source-companies-1.html
- Even the UK government supports OSS.
  - https://www.gov.uk/guidance/be-open-and-use-open-source
- China – one of the biggest consumers of open-source technology and increasingly one of the biggest contributors.
  - https://interconnected.blog/open-source-in-china-the-players/

Open-Source Software

- Questions that might ask yourself.
  - How can the technical work of dozens; or even hundreds, of developers be coordinated without the usual project management apparatus?
  - How do useful, well-designed products emerge from the individual actions of widely-distributed developers, with minimal central planning and relatively impoverished communication?

Open-Source Definition

- Open-Source Criteria:
  - https://opensource.org/osd-annotated



- Range of Code Adoption
  - Code adoption can happen at the level of a few lines of code, a method, a class, a library, a component, a tool, or a complete system.

What is Ethics?

- Ethics
  - Refers to what we believe to be “right” and “wrong” in terms of acceptable conduct.

- Research Ethics
  - Refers to ethical norms, codes, and regulation which govern our current research practice as part of an academic/scientific professional community.

The Ethics of Open Source

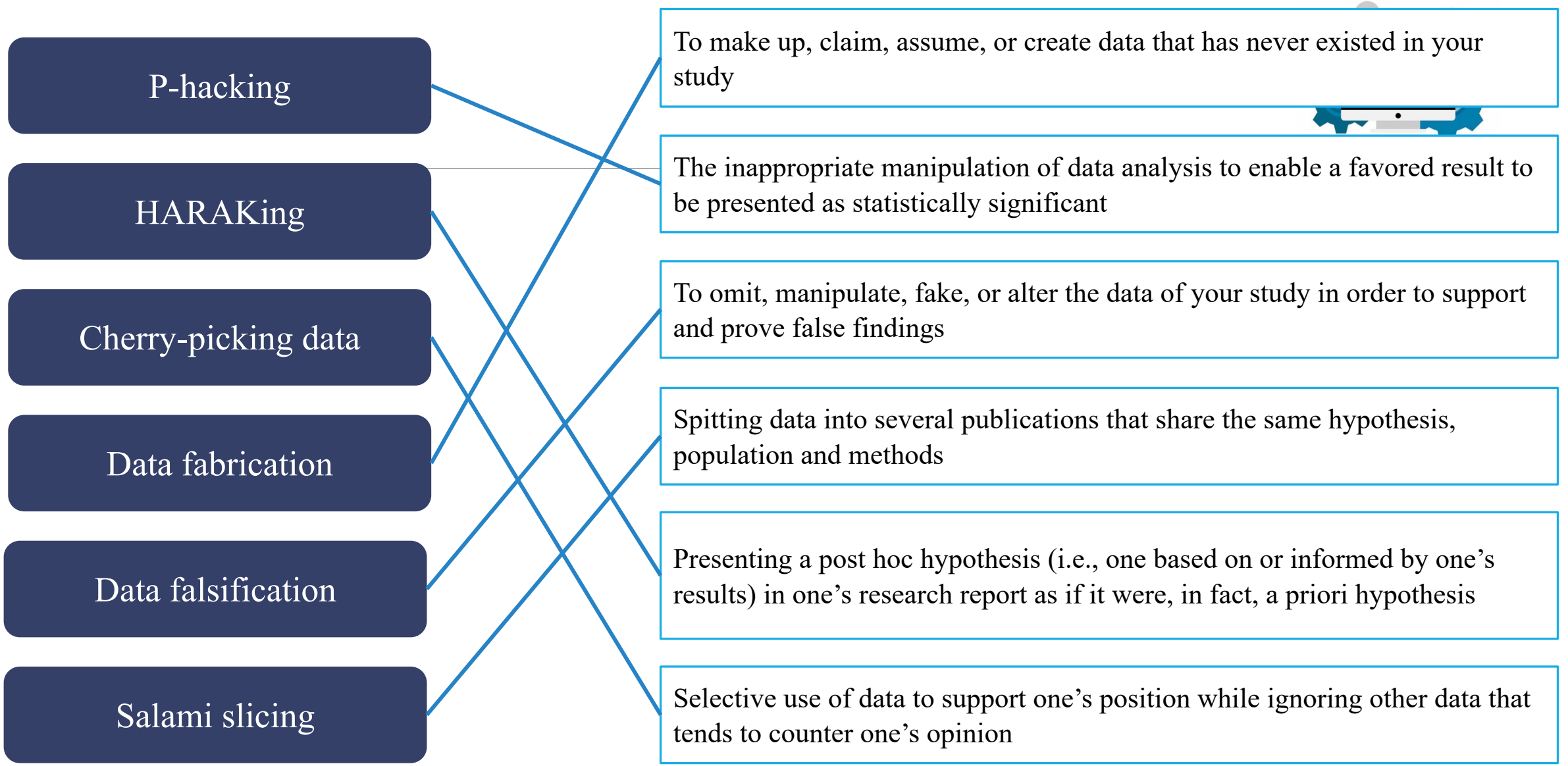
- Is there a significant moral or ethical reason why software should be open source?
- Are there ethical considerations for using OSS?
- Is having software available to anyone a security risk?

The Ethics of Open Source

- Different points of views.
  - The ethics of free software (Dr. Dobb's Journal 2000).
    - https://www.drdobbs.com/the-ethics-of-free-software/184414581
  - The ethics of open-source software (Erfanian's Blog 2013).
    - http://www.ericerfanian.com/the-ethics-of-open-source-software/
  - Why open-source software isn't as ethical as you think it is (Ethical Tech 2017)
    - https://words.werd.io/why-open-source-software-isnt-as-ethical-as-you-think-it-is-2e34d85c3b16

Can I Use Existing Online Code (Part) in My CW?

- Yes, you may, but provide proper citation(s) in your report.
- Be aware of the coding style that meets the design principles and patterns (refactoring).
  - Don’t copy blindly.


P-hacking
HARAKing
Cherry-picking data
Data fabrication
Data falsification
Salami slicing
To make up, claim, assume, or create data that has never existed in your study
The inappropriate manipulation of data analysis to enable a favored result to be presented as statistically significant
To omit, manipulate, fake, or alter the data of your study in order to support and prove false findings
Spitting data into several publications that share the same hypothesis, population and methods
Presenting a post hoc hypothesis (i.e., one based on or informed by one's results) in one's research report as if it were, in fact, a priori hypothesis
Selective use of data to support one's position while ignoring other data that tends to counter one's opinion

Licenses

Software Licenses

- "I just want to give my software away! **Why** do I care about licenses?"
- Scenario 1
  - Bob installs my disk optimisation software. Great! But Bobs hard drive catches fire and he loses an entire novel he is writing. Bad. Whose fault is it?

- Scenario 2
  - I write an awesome music sharing app. Great!
  - EvilCorp also like it, and they realise they can take it, close the source, and fill it with advertisement and sell it. Is this what I want?

Software Licenses

- Software licenses are there …
  - To protect you as well as your code.
  - To protect any future developers of the code.

- Need rules in order to secure certain freedoms.
  - What can be done with the code?
  - Who can change it?
  - Who can distribute it?
  - Is there any warranty or disclaimer?
  - …

Software Licenses

- Some common OSS license (for more, see https://opensource.org/licenses/ and https://en.wikipedia.org/wiki/Software_license)
  - Permissive license:
    - Subsequent users can produce 'closed source' versions and sell the software.
  - CopyLeft license:
    - Any subsequent versions are left with the same rights, e.g. source code must be supplied, and can be modified.
- Examples
  - Apache License 2.0 > Permissive license.
  - BSD 3-Clause "New" or "Revised" license > Permissive license.
  - GNU General Public License (GPL) > CopyLeft license.
  - MIT license > Permissive license.
  - Mozilla Public License 2.0 > Permissive license.

Copyleft Open-Source Licenses

- Copyleft licences necessitate that any derivative works or modifications to the software be released under the same licence as the original software.

- This implies that if you modify or build upon the original software, you must also release your work as open source under the same copyleft licence.

Copyleft: GPL

- The **GNU General Public License** (GPL) is an open-source licence widely employed by developers worldwide.
- Devised by the Free Software Foundation, this licence guarantees that any software developed under it remains freely accessible and modifiable by anyone.
- Numerous software projects: Web browsers to operating systems and stands as a cornerstone of the open-source movement.
- The GPL licence empowers developers to collaborate and innovate without concerns about proprietary restrictions or licensing fees.

Copyleft: AGPL

- As a derivative of the GPL, the **Affero General Public License** (AGPL) is explicitly tailored for software distributed over a network.
- Much like the GPL, the AGPL license guarantees that any alterations users make to the software are accessible to the public.
- Including online games and social media platforms, contributing to the preservation of the open and collaborative essence of these tools.

Copyleft: LGPL

- The **Lesser General Public License** (LGPL) is an additional open-source licence that permits the utilisation of open-source software in proprietary applications.
- The LGPL is a favoured option for developers seeking to employ open-source libraries and frameworks without the obligation to release their entire application under an open-source license.

Copyleft: EPL

- This permissive open-source licence is designed to be compatible with other open-source licenses.
- Developed by the Eclipse Foundation, the **Eclipse Public License** (EPL) has gained popularity among developers seeking to integrate code from various open-source projects.
- The license includes a patent license, guaranteeing that any linked patents are licensed for free and open use.

Copyleft: MPL

- Being a hybrid open-source license, the **Mozilla Public License** (MPL) integrates features from both the GPL and the BSD license.
- Originating from the Mozilla Foundation, the MPL permits the use of open-source software in proprietary applications, while stipulating that any modifications to the software must be accessible under the same licence.
- Consequently, the MPL is frequently chosen for open-source software incorporated into broader commercial products.
- In general, these licences have played a pivotal role in nurturing a culture of collaboration and innovation within the technology industry.

Permissive Open-Source Licenses

- Permissive licenses grant permission for anyone to use, modify, and distribute the software without any obligation to release derivative works as open source.
- They provide developers with greater freedom to employ the software in their projects, even if those projects are proprietary.
- The fundamental distinction between copyleft and permissive licenses lies in the extent of control they exert:
  - Copyleft licenses: Modifications or additions to the software must remain open source,
  - Permissive licenses: Permit developers to integrate the software into their projects, whether open source or proprietary.

Permissive: Apache

- The **Apache License**, the king of permissive licences, is designed for flexibility and user-friendliness.
- Users are empowered to modify, distribute, and use the software without encountering any constraints.
- Notably, it incorporates a patent license, offering users supplementary protection against potential claims of patent infringement.
- Frequently employed in web and cloud-based projects, Apache's adaptability renders it an immensely popular choice within the developer community.

Permissive: MIT

- The **MIT License** holds equal prominence within the open-source community.
- It stands as a simple and straightforward license, permitting users to utilise, modify, and distribute the software without encountering any constraints.
- As one might anticipate, the MIT license is commonly favoured for academic and research projects, as well as for smaller software endeavours.

Permissive: BSD

- The **BSD License** is another family of permissive open-source licenses that are commonly used in academic and research settings, and it does not require any derivative works to be released as open-source.
- The BSD license lends itself well to networking and server software, as well as operating systems and research projects.

Permissive: Unilicense

- **Unilicense**, designed for utmost simplicity and ease of use.
- It lacks any warranty or liability provisions, making it particularly suitable for smaller-scale personal or hobby projects.

License Compatibility

- Different licenses may have conflicting terms or requirement.
- License compatibility refers to the capacity of two or more open-source licenses to coexist and be amalgamated within a single software project.
  - E.g., the Apache license and the MIT license, both being permissive licenses, are considered compatible. This means that code released under the Apache license can be incorporated into a project using the MIT license, and vice versa.
- Not all licenses harmonize with each other.
  - E.g., the GPL is a copyleft license, signifying that if you integrate code released under the GPL into your project, your project must also be released under the GPL. This renders the GPL incompatible with certain permissive licenses, such as Apache or BSD.


snyk
Copyleft
Permissive
GPL
Free as in Freedom
AGPL
Free as in Freedom
LGPL
Free as in Freedom
EPL 1.0
MPL
Apache
MIT
BSD
Unlicense
Permissions in addition to commercial use, distribution, modification:
Patent use
Conditions
Disclose source
License & copyright notice
Source
Network use is distribution
Same license
Library
File
State changes
Some
Limitations/Disclaimers
Liability
Warranty
Trademark use
No explicit limitation

Which Open-Source License is Best?

- Copyleft licenses generally impose more restrictions — potentially offering less liability —compared to permissive licenses.
  - If to maximize code reusability and sharing: Permissive license
  - Developing software intended for use over a network: Affero General Public License (AGPL)
- The GNU General Public License (GPL) comes in two main versions: GPLv2 and GPLv3.
  - GPLv3 addresses issues not covered in GPLv2, such as patents, and enhances compatibility with other open-source licenses like the Apache License v2.
  - GPLv2 and GPLv3 are not compatible with each other.
- MIT licenses enjoy widespread usage, benefiting from their recognition and common understanding.
  - Software licensed under MIT entails no restrictions on redistribution or monetization.
  - Moreover, MIT licenses are compatible with many other open-source licenses.

License: Overview

- Ultimately, the selection of a license will hinge on the specific needs and requirements of your project.
- By comprehending the advantages of each license, you can make an informed decision that aligns with your goals and contributes to the open-source community.

Self-Test

- An open-source license grants users the rights:
  - The right to access and use the software's open-source code. (Yes / No)
  - The right to modify the software's source code. (Yes / No)
  - The right to distribute the software's source code and any modified version. (Yes / No)

These rights are often subject to certain conditions, such as giving credit to the original author or contributing any modifications back to the community.

Summary

License Types

Copyleft	permissive	Creative Commons	Public Domain
*GPL	MIT, BSD, ...	building blocks:	"Don't care"
viral	attribution	→ attribution → virality → no commercial use ⚡ → no derivatives ⚡	liability? ⚡
attribution	protection against liability		
protection against liability	patents, trademark rules in details		

Sample Scenario – 1

- You've developed a utility library for data processing, and you want others to freely use, modify, and integrate it into their own proprietary software without restrictions.
- **Suitable License**: **?**

Sample Scenario – 2

- You've developed a web framework and want to ensure that anyone who uses or modifies it must make their modified versions available under the same open-source license.
- **Suitable License**: **?**

Sample Scenario – 3

- You're developing an open-source software tool for deployment on servers, and you want to ensure that even if someone modifies and runs it on a server (without distributing it), they must release their modifications.
- **Suitable License**: **?**

Sample Scenario – 4

- You've created a widely-used open-source library (e.g., a machine learning framework), and you want businesses and developers to be able to use it freely in proprietary software while protecting your intellectual property (e.g., from patent claims).
- **Suitable License**: **?**

Sample Scenario – 5

- You've developed a mobile app that you want others to freely adapt and use, including for commercial purposes, but you don't mind if they close-source their changes.
- **Suitable License**: **?**

Sample Scenario – 6

- You've developed a library used for application development (e.g., a UI framework), and you want to allow proprietary software to link to it while keeping your library itself open-source.
- **Suitable License**: **?**

Sample Scenario – 7

- You've built a project with the intention of fostering an open-source community where contributions are shared openly, and you want to ensure that anyone modifying the project shares their changes.
- **Suitable License**: **?**

Sample Scenario – 8

- A company wants to release a small tool or utility as open-source to foster goodwill or contribute to the community but wants to avoid the risk of the tool's restrictions impacting their own closed-source software.
- **Suitable License**: **?**

Extra Studies

- An Empirical Study of License Violations in Open Source Projects
- An Analysis of Open-Source Software Licensing Questions in Stack Exchange Sites
- Open-Source Case Study

Commercializing Open-Source Software

TWO GENERAL MODELS

Model 1: Open Core

- "Open Core" businesses offer a free, open-source version of their software and a paid version with additional proprietary features that would be difficult to replicate (e.g., authentication support).
- Companies such as Elastic, D2iQ, GitLab, OmniSciDB etc.
- Target users are mostly software engineers.

Model 2: Systems Integration

- Most popular - Red Hat Enterprise Linux distribution.
- Does not invent the underlying, massively successful open-source project they are helping companies use.
- Invest heavily and influence its development, and generally the most important maintainers of the project.
- The maintainers are NOT fundamentally the OWNERS of the project, they are more the major stakeholders.
- Other companies such as CrunchyData

Model 2: Systems Integration

- Open Core Companies vs. Systems Integrators:
  - Open Core: Companies invest heavily in R&D for their core technology until it matures,
  - Systems Integrators: Build on open-source projects that have already achieved significant scale.

- Ownership and Scale:
  - Open Core: Often retain ownership of their projects,
  - Systems Integrators: Reach a massive scale and no longer have a single owner.

Model 2: Systems Integration

- R&D Budget Dynamics:
  - Open Core: Face the challenge of funding R&D until maturity,
  - Systems Integrators: Benefit from a vast R&D budget derived from the extensive open-source projects they leverage.

- Community Standing:
  - Both: Must maintain a positive standing in the engineering community using their open-source technology,
  - Systems Integrators: Enjoy a larger R&D budget, ensuring consistent product enhancements.

Model 2: Systems Integration

- Impact of Financial Performance:
  - Open Core: Significantly affects their R&D capabilities,
  - Systems Integrators: Often profitable with revenue from professional services, benefit from consistent product improvements and maintenance, irrespective of financial fluctuations.

Who Next?

Using and Developing Skills

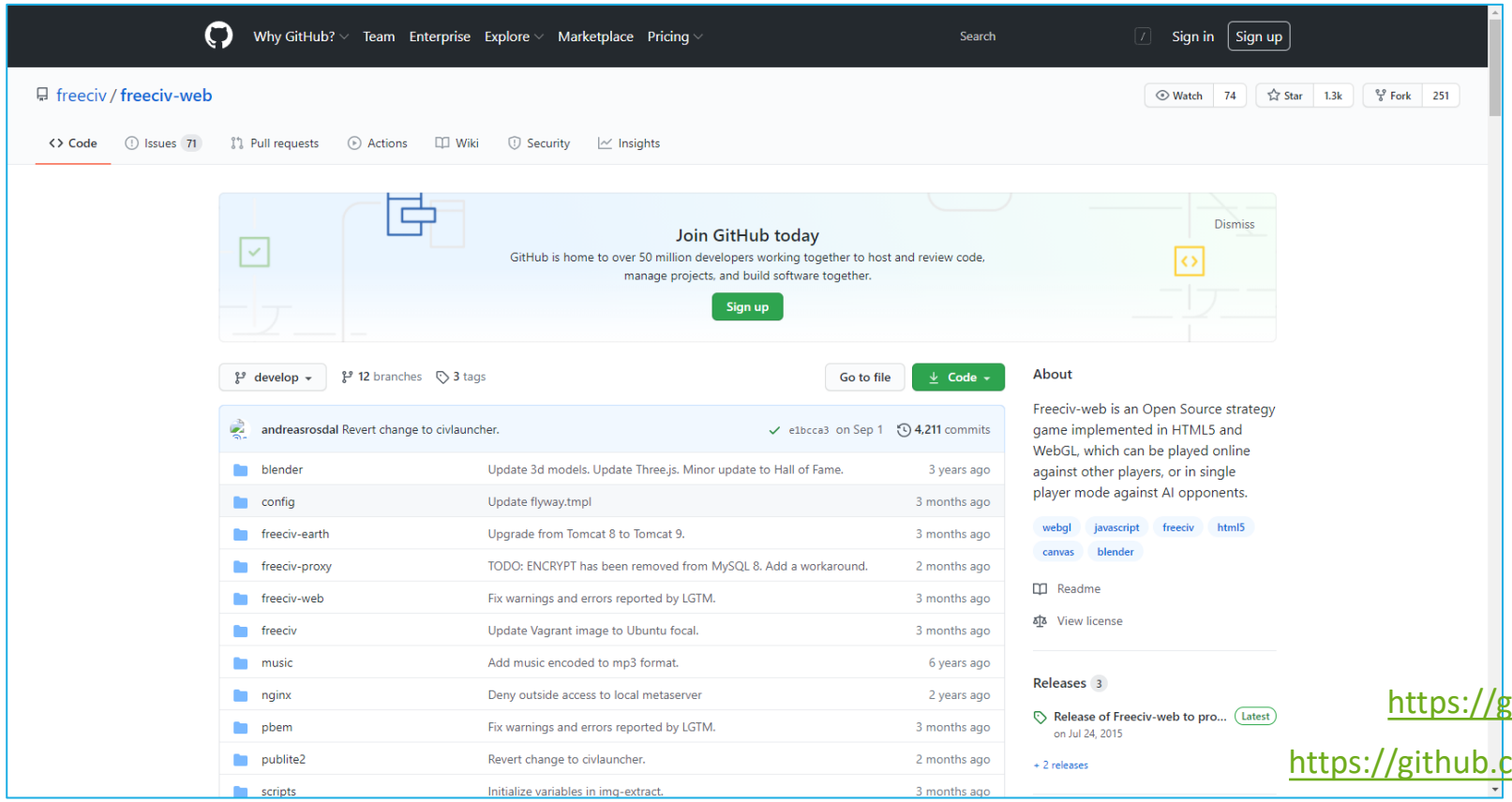
- What have you learnt during DMS module?
  - Much more experience with Java.
  - GUI programming.
  - Unit and regression testing.
  - Use of tools (e.g. Git/GitLab; Eclipse/IntelliJ; Ant/Maven/Gradle; …).
  - Modifying and adding to a sizeable existing project.
- And with GRP, you are learning to work as a team.
- What now? You might want to
  1. Initiate own open-source project.
  2. Contribute to an existing project.

Initiate Own Project

- How?
  - Discuss the “itch” with a few.
  - Code Development starts.
  - Create website and “announce”.
  - Community evolves.
  - Functionalities added.
  - Feedback and communication flows.
  - Reason of having Q&A discussion in Moodle – but little-to-none utilize it.

- Funding?
  - Donation, voluntary, crowd-funded.

Contribute To An Existing Project



https://github.com/freeciv/freeciv-web

https://github.com/freeciv/freeciv-web/issues

Contribute To An Existing Project

- How to get involved.
  - Look at the README file.
    - Should explain the purpose of the projects, direction of development etc.
    - Look out for sections on "How To Contribute".
  - Fork the project.
    - Create a branch?
  - Check for any rules on how contribution should work.
    - Respect the rules of the project. (some rules/instructions are not followed in DMS module)

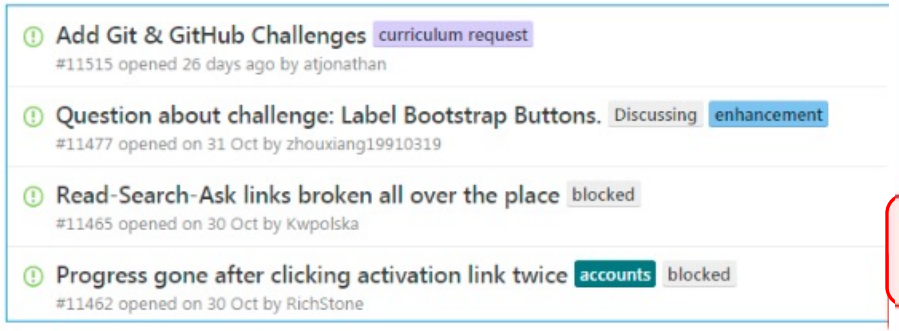
Contribute To An Existing Project

- How to get involved (continue).

  - Join a development chat or forum to see how development is organised.
  - Start to talk to the developers if you think you can help.
  - Learn how to use merge/pull requests.
  - Write a test to show you have fixed a bug.
  - How is the right attitude!?
    - Patient; patient.
    - **Polite communication and have respect to others**.

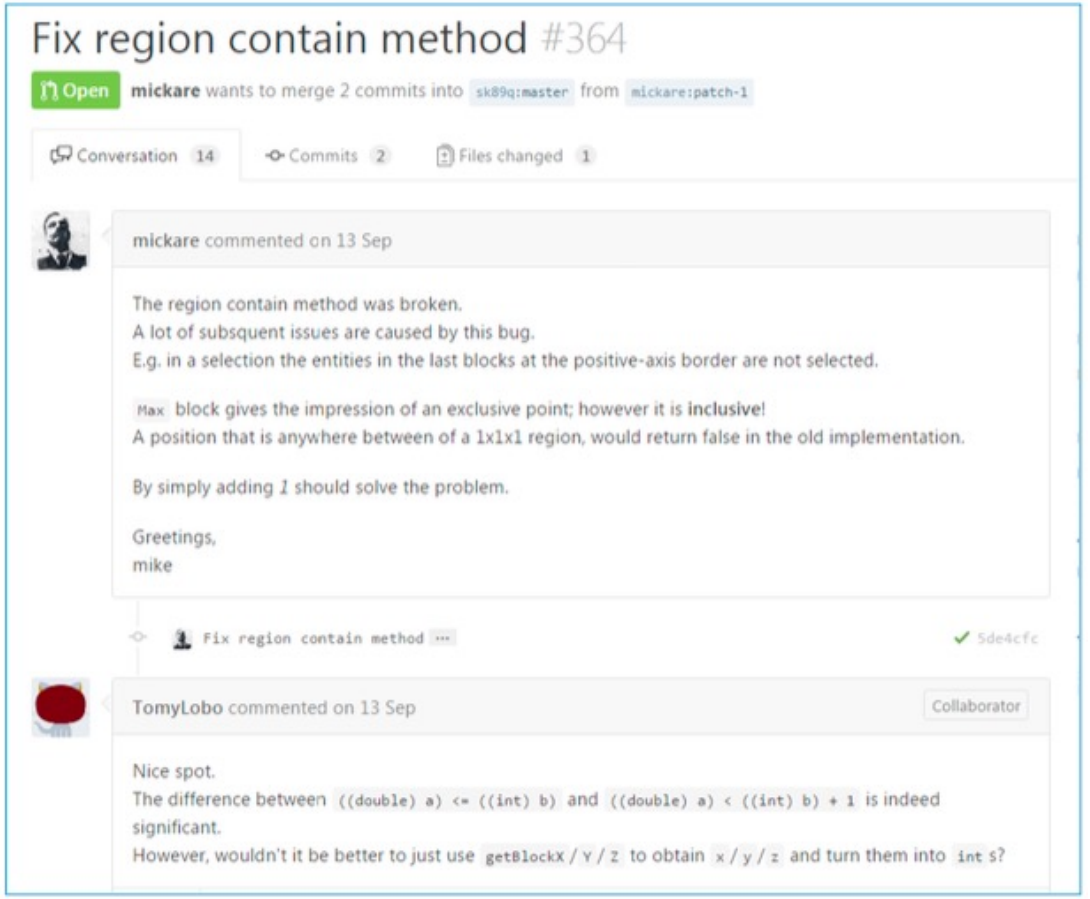
Contribute To An Existing Project

- Looking for open issues (*the tools are not fully utilised by students*).
  - Look under issues for a project.
  - Projects can assign labels.
  - Some target new developers.



Contribute To An Existing Project

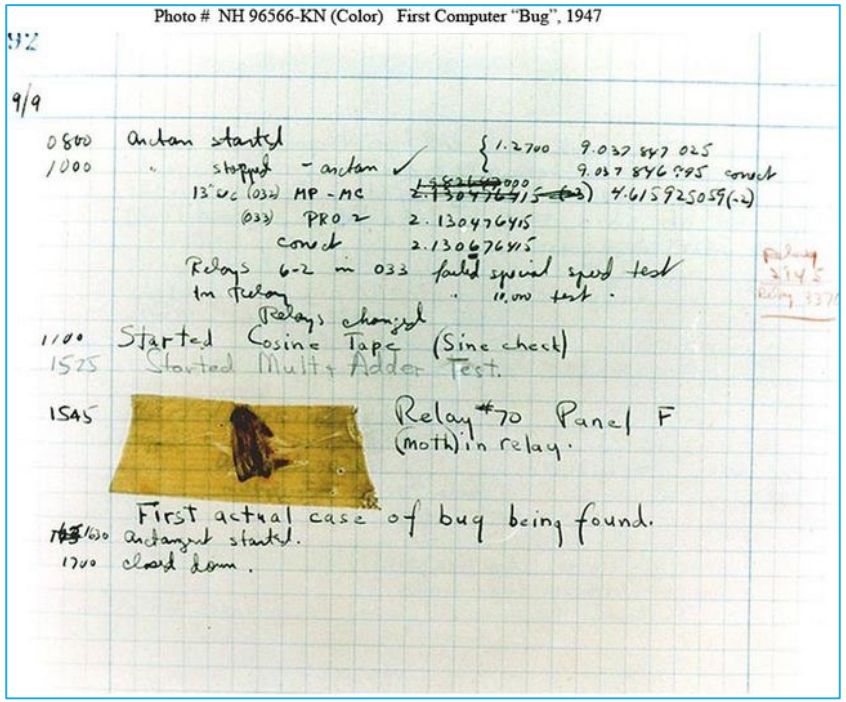
- Communication is informal.
  - Community communications.
  - Threaded discussion forums.
  - Email (list servers).
  - Newsgroups.
  - Messaging/chat.
  - Community digests.
  - Social networks.



Tracking Bugs Collaboratively

What About Bugs and Issues?

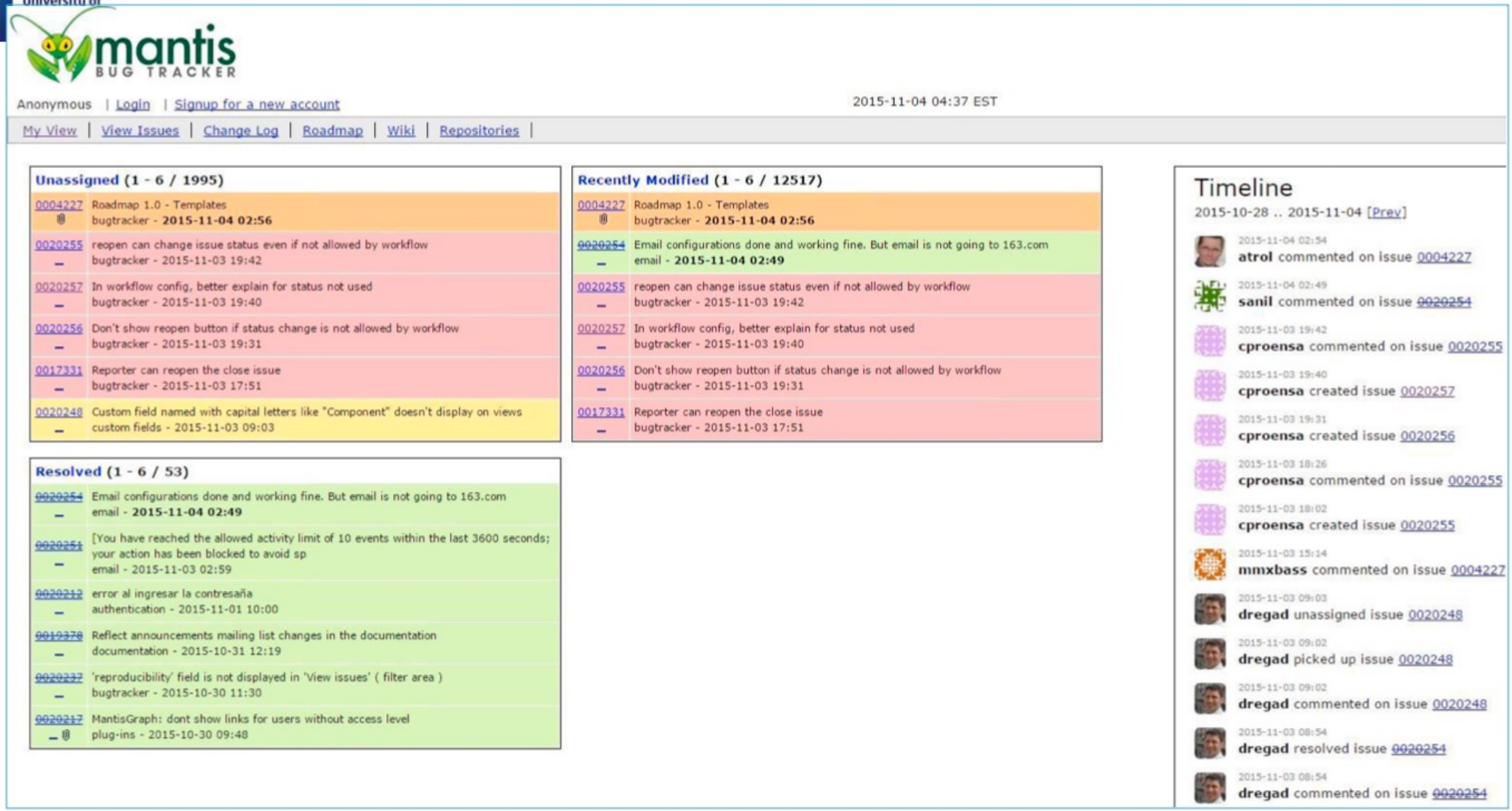
- A page from the Harvard Mark II electromechanical computer's log, featuring a dead moth that was removed from the device in 1947.

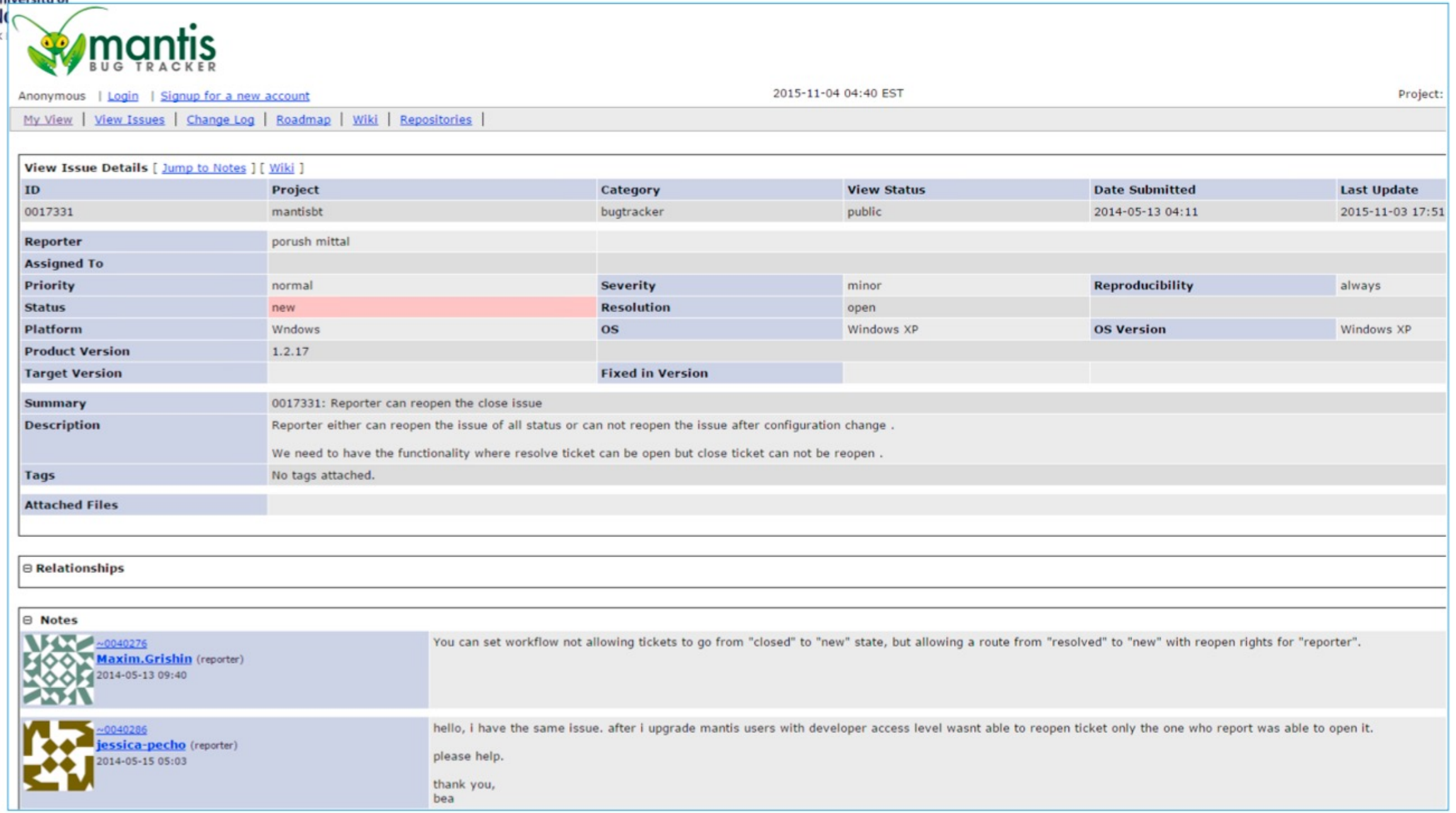


https://en.wikipedia.org/wiki/Software_bug

What About Bugs and Issues?

- Several tools exist to manage reporting bugs and assigning issues between multiple developers.
- Essentially, they are specialised databases.
  - Store and track individual bug issues.
    - Severity, repeatability, system, details.
  - Allow users to submit new and fix existing bugs.
  - Allow users to comment and bugs.
- Some common tools.
  - Mantis, web-based bug tracker: https://www.mantisbt.org/.
  - Bugzilla, from Mozilla Foundation: https://www.bugzilla.org/.


mantis
BUG TRACKER
Anonymous | Login | Signup for a new account
2015-11-04 04:37 EST
My View | View Issues | Change Log | Roadmap | Wiki | Repositories
Unassigned (1 - 6 / 1995)
0004227 Roadmap 1.0 - Templates
bugtracker - 2015-11-04 02:56
0020255 reopen can change issue status even if not allowed by workflow
bugtracker - 2015-11-03 19:42
0020257 In workflow config, better explain for status not used
bugtracker - 2015-11-03 19:40
0020256 Don't show reopen button if status change is not allowed by workflow
bugtracker - 2015-11-03 19:31
0017331 Reporter can reopen the close issue
bugtracker - 2015-11-03 17:51
0020248 Custom field named with capital letters like "Component" doesn't display on views
custom fields - 2015-11-03 09:03
Recently Modified (1 - 6 / 12517)
0004227 Roadmap 1.0 - Templates
bugtracker - 2015-11-04 02:56
0020254 Email configurations done and working fine. But email is not going to 163.com
email - 2015-11-04 02:49
0020255 reopen can change issue status even if not allowed by workflow
bugtracker - 2015-11-03 19:42
0020257 In workflow config, better explain for status not used
bugtracker - 2015-11-03 19:40
0020256 Don't show reopen button if status change is not allowed by workflow
bugtracker - 2015-11-03 19:31
0017331 Reporter can reopen the close issue
bugtracker - 2015-11-03 17:51
Resolved (1 - 6 / 53)
0020254 Email configurations done and working fine. But email is not going to 163.com
email - 2015-11-04 02:49
0020251 [You have reached the allowed activity limit of 10 events within the last 3600 seconds; your action has been blocked to avoid sp
email - 2015-11-03 02:59
0020212 error al ingresar la contresaña
authentication - 2015-11-01 10:00
0019378 Reflect announcements mailing list changes in the documentation
documentation - 2015-10-31 12:19
0020237 'reproducibility' field is not displayed in 'View issues' (filter area)
bugtracker - 2015-10-30 11:30
0020217 MantisGraph: dont show links for users without access level
plug-ins - 2015-10-30 09:48
Timeline
2015-10-28 .. 2015-11-04 [Prev]
2015-11-04 02:54 atrol commented on issue 0004227
2015-11-04 02:49 sanil commented on issue 0020254
2015-11-03 19:42 cproensa commented on issue 0020255
2015-11-03 19:40 cproensa created issue 0020257
2015-11-03 19:31 cproensa created issue 0020256
2015-11-03 18:26 cproensa commented on issue 0020255
2015-11-03 18:02 cproensa created issue 0020255
2015-11-03 15:14 mmxbass commented on issue 0004227
2015-11-03 09:03 dregad unassigned issue 0020248
2015-11-03 09:02 dregad picked up issue 0020248
2015-11-03 09:02 dregad commented on issue 0020248
2015-11-03 08:54 dregad resolved issue 0020254
2015-11-03 08:54 dregad commented on issue 0020254



mantis BUG TRACKER

Anonymous | Login | Signup for a new account

2015-11-04 04:40 EST

Project:

My View | View Issues | Change Log | Roadmap | Wiki | Repositories |

View Issue Details [Jump to Notes] [Wiki]

ID	Project	Category	View Status	Date Submitted	Last Update
0017331	mantisbt	bugtracker	public	2014-05-13 04:11	2015-11-03 17:51
Reporter	porush mittal				
Assigned To					
Priority	normal	**Severity**	minor	**Reproducibility**	always
Status	new	**Resolution**	open		
Platform	Wndows	**OS**	Windows XP	**OS Version**	Windows XP
Product Version	1.2.17				
Target Version		**Fixed in Version**			
Summary	0017331: Reporter can reopen the close issue				
Description	Reporter either can reopen the issue of all status or can not reopen the issue after configuration change . We need to have the functionality where resolve ticket can be open but close ticket can not be reopen .				
Tags	No tags attached.				
Attached Files					

Relationships

Notes

~0040276 **Maxim.Grishin** (reporter) 2014-05-13 09:40	You can set workflow not allowing tickets to go from "closed" to "new" state, but allowing a route from "resolved" to "new" with reopen rights for "reporter".
~0040286 **jessica-pecho** (reporter) 2014-05-15 05:03	hello, i have the same issue. after i upgrade mantis users with developer access level wasnt able to reopen ticket only the one who report was able to open it. please help. thank you, bea

University of Nottingham
UK | CHINA | MALAYSIA

ASF Bugzilla – Bug List

Home | New | Browse | Search | gui | Search | [?] | Reports | Help | New Account | Log In | Forgot Password

Wed Nov 4 2015 09:45:10 UTC

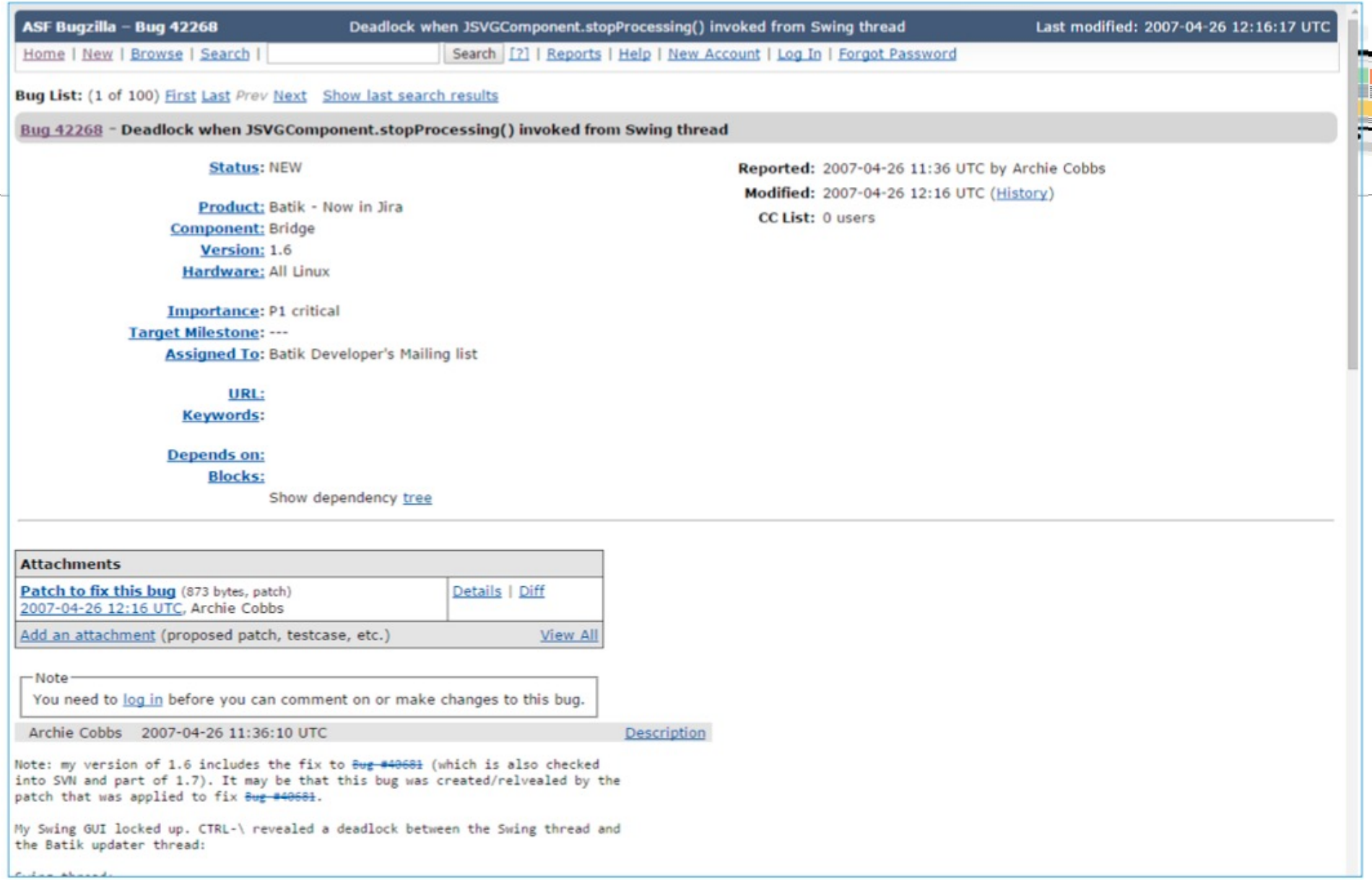
An idiot with a computer is a faster, better idiot. -- Rich Julius

Hide Search Description

Status: UNCONFIRMED, NEW, ASSIGNED, REOPENED, NEEDINFO **Product:** gui **Component:** gui **Alias:** gui **Summary:** gui **Whiteboard:** gui **Content:** "gui"

100 bugs found.

ID ▲	Product	Comp	Assignee ▲	Status ▲	Resolution	Summary	Changed
42268	Batik -	Bridge	batik-dev	NEW	---	Deadlock when JSVGComponent.stopProcessing() invoked from Swing thread	2007-04-26
44194	Batik -	Utilitie	batik-dev	NEW	---	RadialGradientPaintContext: hints can be null	2008-01-10
38480	Lenya	Miscella	dev	NEW	---	[PATCH] The error handler page should contain a link back to the authoring gui.	2007-11-02
33305	JMeter	Main	issues	NEW	---	Visual diff and merge functionality for JMeter scripts	2013-08-25
49742	Batik -	SVG DOM	batik-dev	NEW	---	NPE in EventListenerList (Concurrency)	2010-08-12
46074	Apache h	Runtime	bugs	NEW	---	Need of friendly and easy to use configuration graphic interface	2008-10-23
53704	Apache h	Runtime	bugs	NEW	---	True online configuration is needed - via LDAP?	2012-08-13
34088	Lenya	Build Sy	dev	NEW	---	Provide patch to upgrade publications from 1.2 to 2.0	2008-01-10
34813	Lenya	Access C	dev	NEW	---	Allow selection of user type on creation screen	2007-07-16
35010	Lenya	Schedule	dev	NEW	---	scheduler fails to restore jobs at tomcat start up and run schedued tasks	2007-04-23
35011	Lenya	Site Man	dev	NEW	---	page moves (up / down) within the site tree fail when lenya is root context application in tomcat	2007-04-23
39237	Lenya	Navigati	dev	NEW	---	Allow to edit sitetree href attributes in the GUI	2007-07-16
42457	Lenya	Miscella	dev	NEW	---	[PATCH] remove meta-stylesheet weirdness from src/webapp/lenya/config/sitemap/pipelines.xmap	2007-08-02
42468	Lenya	Access C	dev	NEW	---	Roles should be editable via gui	2007-07-16
42469	Lenya	Site Man	dev	NEW	---	Allow bulk import/export	2007-07-16



ASF Bugzilla – Bug 42268 Deadlock when JSVGComponent.stopProcessing() invoked from Swing thread Last modified: 2007-04-26 12:16:17 UTC

Home | New | Browse | Search | Search [?] | Reports | Help | New Account | Log In | Forgot Password

Bug List: (1 of 100) First Last Prev Next Show last search results

Bug 42268 - Deadlock when JSVGComponent.stopProcessing() invoked from Swing thread

Status: NEW

Reported: 2007-04-26 11:36 UTC by Archie Cobbs
Modified: 2007-04-26 12:16 UTC (History)
CC List: 0 users

Product: Batik - Now in Jira
Component: Bridge
Version: 1.6
Hardware: All Linux

Importance: P1 critical
Target Milestone: ---
Assigned To: Batik Developer's Mailing list

URL:
Keywords:

Depends on:
Blocks:
Show dependency tree

Attachments

Patch to fix this bug (873 bytes, patch) 2007-04-26 12:16 UTC, Archie Cobbs	Details | Diff
Add an attachment (proposed patch, testcase, etc.)	View All

Note
You need to log in before you can comment on or make changes to this bug.

Archie Cobbs 2007-04-26 11:36:10 UTC Description

Note: my version of 1.6 includes the fix to ~~Bug #40681~~ (which is also checked into SVN and part of 1.7). It may be that this bug was created/relvealed by the patch that was applied to fix ~~Bug #40681~~.

My Swing GUI locked up. CTRL-\ revealed a deadlock between the Swing thread and the Batik updater thread:

Challenges

Challenges of Collaborative OSS Development

- Product structure and comprehension.
  - Who understands the “whole system”?
- Effective ways of incorporating requirements of non-developer users?
- With larger scales, will coordination need force adoption of “commercial” development techniques?
- How to collaborate on “big” features?
- How to respond to unanticipated events? (Funding etc.)

Finally …

Final Advices

- Some warnings.
  - DO NOT dive into someone's project and start re-writing code so it is easier for you to understand.
    - Use existing code style.
  - DO NOT be afraid to ask for help.
  - As with any public space on the internet, there are good projects and bad projects.
    - Think about who you are interacting with.
    - Bear in mind public comments; contributions etc. will stay visible.
- Your GRP is the best place to start practicing this if you are not sure what you are doing. Make the most of your group project.

Final Advices

- Where to go now?
  - There are some sites targeted at teaching and starting to contribute.
  - Open-source community designed to help one to learn to code and contribute to projects.
    - Learn to develop collaboratively, get certification, start to contribute, and always open for comments/feedback.
- Further readings.
  - First Timers Only: https://www.firsttimersonly.com/.
  - Check GitHub for Beginner's Guides and Help: https://github.com/btford/participating-in-open-source.

Asking a Question

Before you ask, do some searching and reading. Check the docs, Google, GitHub, and StackOverflow. If your question is something that has been answered many times before, the project maintainers might be tired of repeating themselves.

Put Your Mind in Maintenance Mode