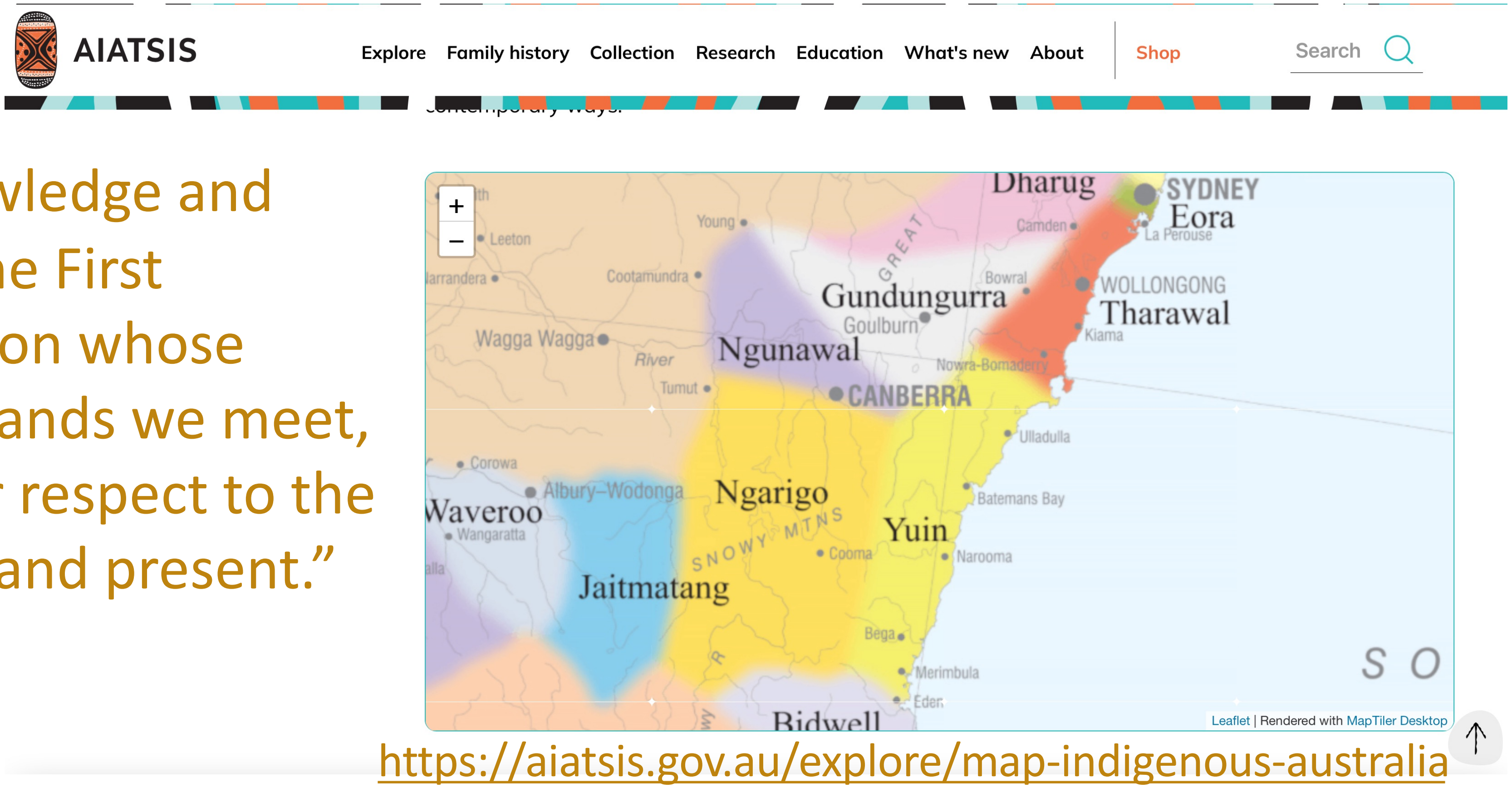
Week:
12 of 12

COMP 2120 / COMP 6120

OPEN SOURCE

A/Prof Alex Potanin

ANU Acknowledgment of Country

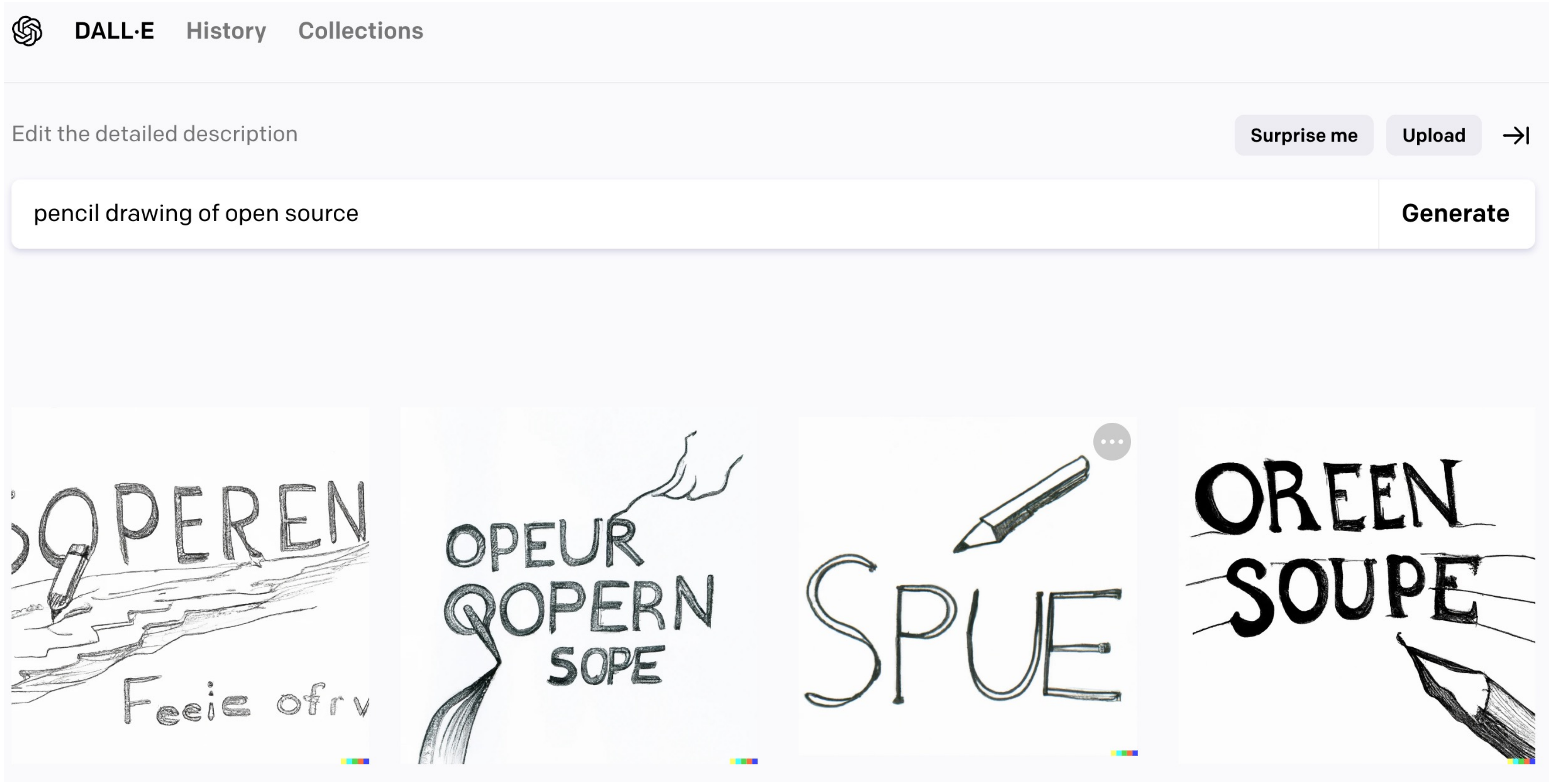


"We acknowledge and celebrate the First Australians on whose traditional lands we meet, and pay our respect to the elders past and present."

https://aiatsis.gov.au/explore/map-indigenous-australia

Today

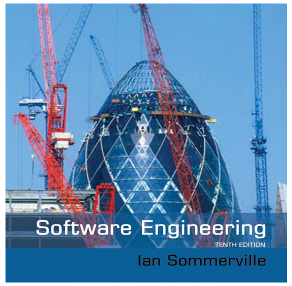
- Open Source
- Licenses
- Dependency Management
- Ethics



pencil drawing of open source

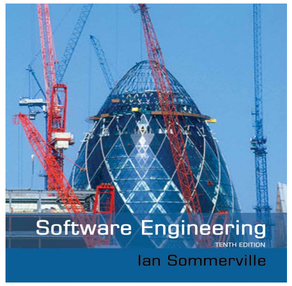
Open Source

Open source development



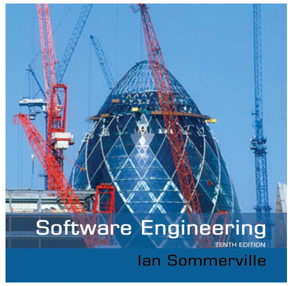
- Open source development is an approach to software development in which the source code of a software system is published and volunteers are invited to participate in the development process
- Its roots are in the Free Software Foundation (www.fsf.org), which advocates that source code should not be proprietary but rather should always be available for users to examine and modify as they wish.
- Open source software extended this idea by using the Internet to recruit a much larger population of volunteer developers. Many of them are also users of the code.

Open source systems



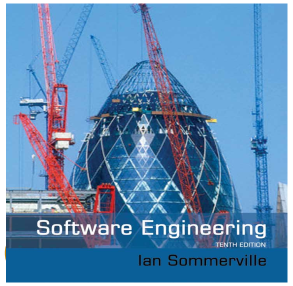
- The best-known open source product is, of course, the Linux operating system which is widely used as a server system and, increasingly, as a desktop environment.
- Other important open source products are Java, the Apache web server and the mySQL database management system.

Open source issues



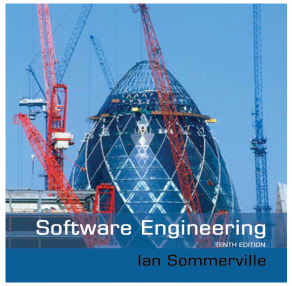
- Should the product that is being developed make use of open source components?
- Should an open source approach be used for the software's development?

Open source business



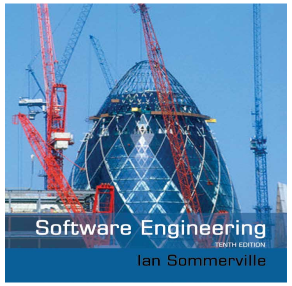
- More and more product companies are using an open source approach to development.
- Their business model is not reliant on selling a software product but on selling support for that product.
- They believe that involving the open source community will allow software to be developed more cheaply, more quickly and will create a community of users for the software.

Open source licensing



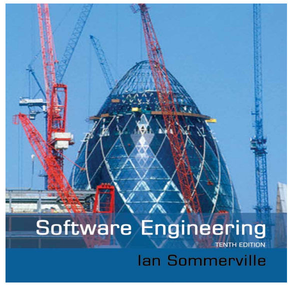
- A fundamental principle of open-source development is that source code should be freely available, this does not mean that anyone can do as they wish with that code.
  - Legally, the developer of the code (either a company or an individual) still owns the code. They can place restrictions on how it is used by including legally binding conditions in an open source software license.
  - Some open source developers believe that if an open source component is used to develop a new system, then that system should also be open source.
  - Others are willing to allow their code to be used without this restriction. The developed systems may be proprietary and sold as closed source systems.

License models



- The GNU General Public License (GPL). This is a so-called ‘reciprocal’ license that means that if you use open source software that is licensed under the GPL license, then you must make that software open source.
- The GNU Lesser General Public License (LGPL) is a variant of the GPL license where you can write components that link to open source code without having to publish the source of these components.
- The Berkley Standard Distribution (BSD) License. This is a non-reciprocal license, which means you are not obliged to re-publish any changes or modifications made to open source code. You can include the code in proprietary systems that are sold.

License management



- Establish a system for maintaining information about open-source components that are downloaded and used.
- Be aware of the different types of licenses and understand how a component is licensed before it is used.
- Be aware of evolution pathways for components.
- Educate people about open source.
- Have auditing systems in place.
- Participate in the open source community.

"Free as in free speech."

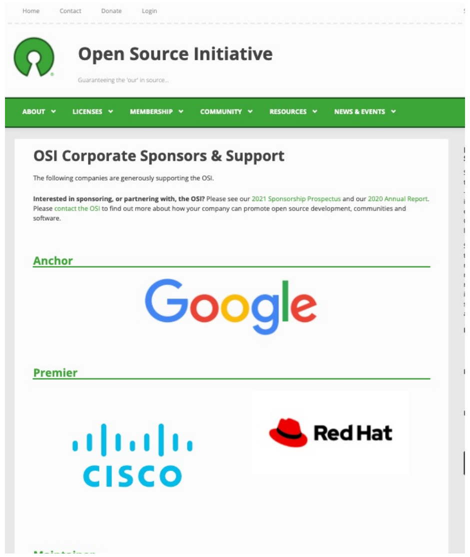
Open Source

aka Free Software

aka Free and Open Source Software

Open Source

17-313 Foundations of SOFTWARE ENGINEERING



Open Source Initiative

Guaranteeing the 'our' in source...

OSI Corporate Sponsors & Support

The following companies are generously supporting the OSI.

Interested in sponsoring, or partnering with, the OSI? Please see our 2021 Sponsorship Prospectus and our 2020 Annual Report. Please contact the OSI to find out more about how your company can promote open source development, communities and software.

Anchor

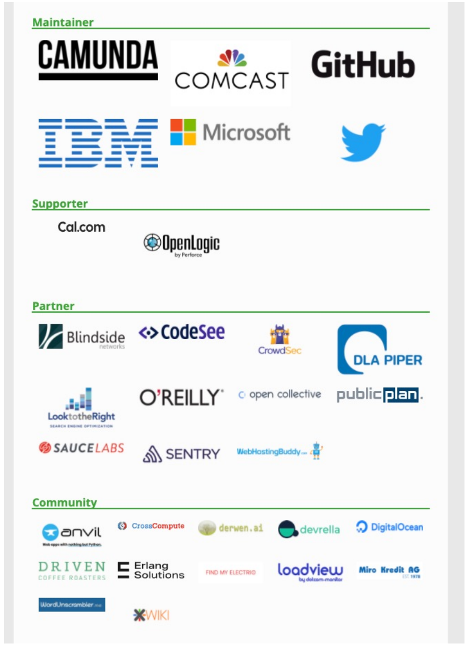
Google

Premier

Cisco

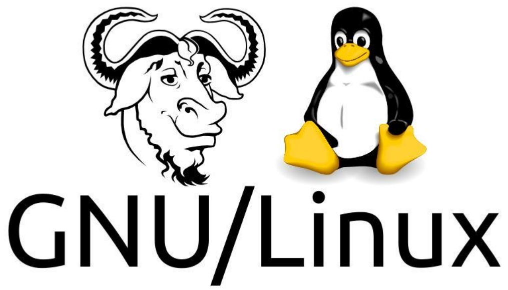
Red Hat

open source initiative®



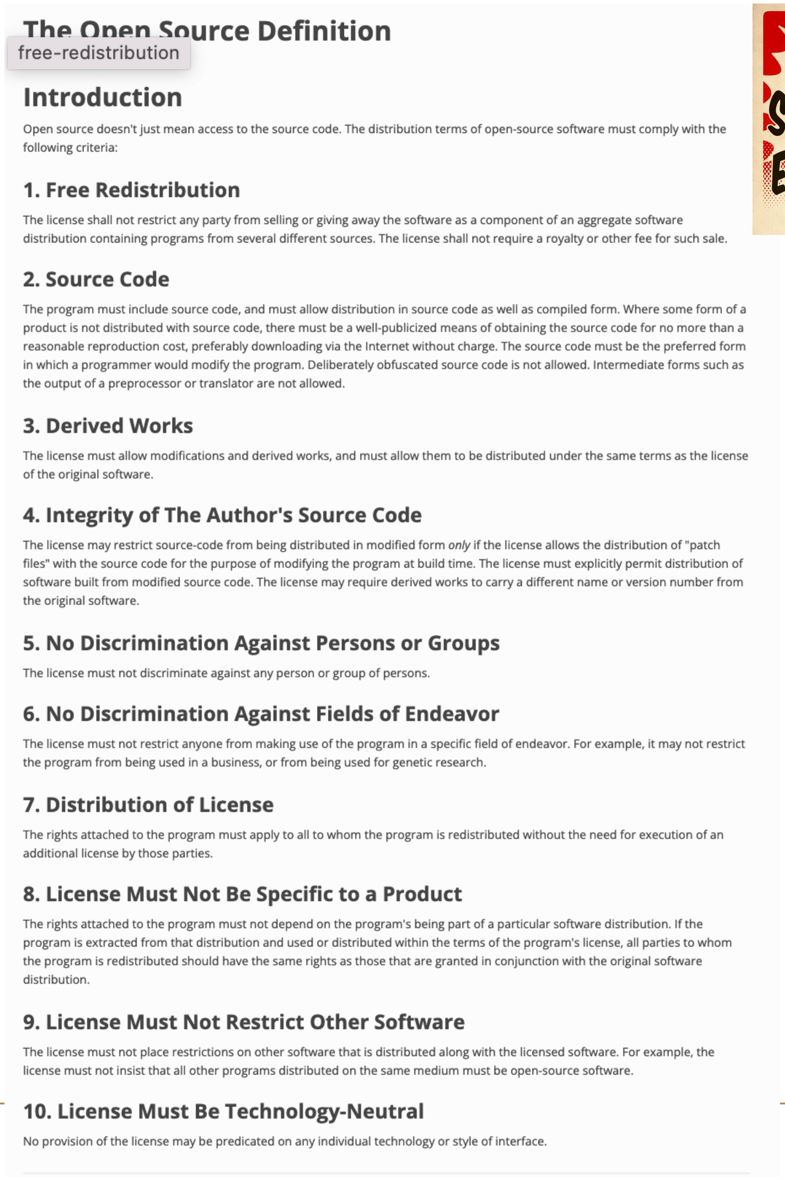
Free Software vs Open Source

- Free software origins (70-80s ~Stallman)
  - ~~Cultish~~ Political goal
  - Software part of free speech
    - free exchange, free modification
    - proprietary software is unethical
    - security, trust
  - GNU project, Linux, GPL license
- Open source (1998 ~ O'Reilly)
  - Rebranding without political legacy
  - Emphasis on internet and large dev./user involvement
  - Openness toward proprietary software/coexist
  - (Think: Netscape becoming Mozilla)



Free Software vs Open Source

- The freedom to run the program as you wish, for any purpose **(freedom 0).**
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.



The Open Source Definition

free-redistribution

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

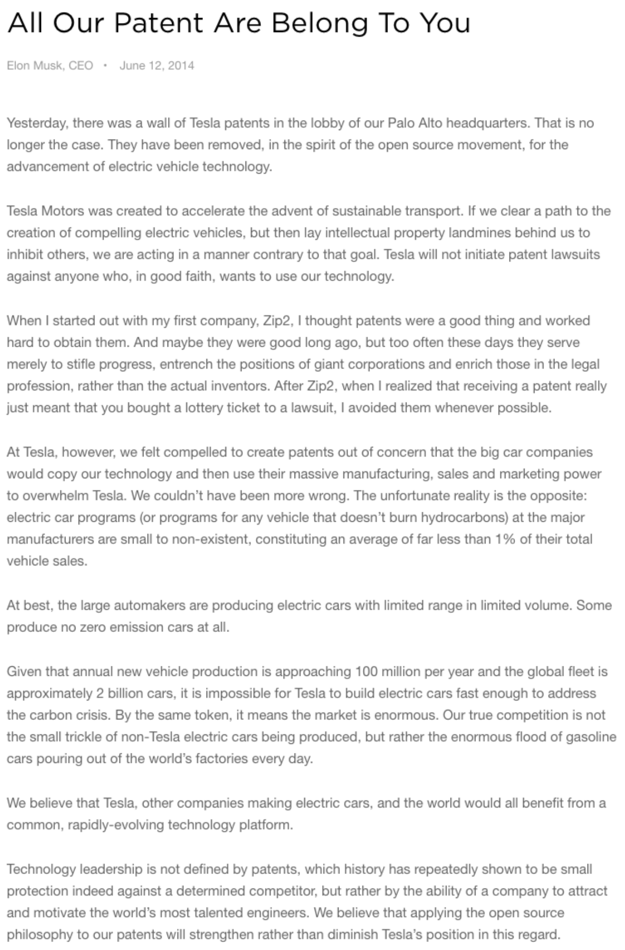
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

The Cathedral and the Bazaar

https://www.tesla.com/blog/all-our-patent-are-belong-you



All Our Patent Are Belong To You

Elon Musk, CEO • June 12, 2014

Yesterday, there was a wall of Tesla patents in the lobby of our Palo Alto headquarters. That is no longer the case. They have been removed, in the spirit of the open source movement, for the advancement of electric vehicle technology.

Tesla Motors was created to accelerate the advent of sustainable transport. If we clear a path to the creation of compelling electric vehicles, but then lay intellectual property landmines behind us to inhibit others, we are acting in a manner contrary to that goal. Tesla will not initiate patent lawsuits against anyone who, in good faith, wants to use our technology.

When I started out with my first company, Zip2, I thought patents were a good thing and worked hard to obtain them. And maybe they were good long ago, but too often these days they serve merely to stifle progress, entrench the positions of giant corporations and enrich those in the legal profession, rather than the actual inventors. After Zip2, when I realized that receiving a patent really just meant that you bought a lottery ticket to a lawsuit, I avoided them whenever possible.

At Tesla, however, we felt compelled to create patents out of concern that the big car companies would copy our technology and then use their massive manufacturing, sales and marketing power to overwhelm Tesla. We couldn't have been more wrong. The unfortunate reality is the opposite: electric car programs (or programs for any vehicle that doesn't burn hydrocarbons) at the major manufacturers are small to non-existent, constituting an average of far less than 1% of their total vehicle sales.

At best, the large automakers are producing electric cars with limited range in limited volume. Some produce no zero emission cars at all.

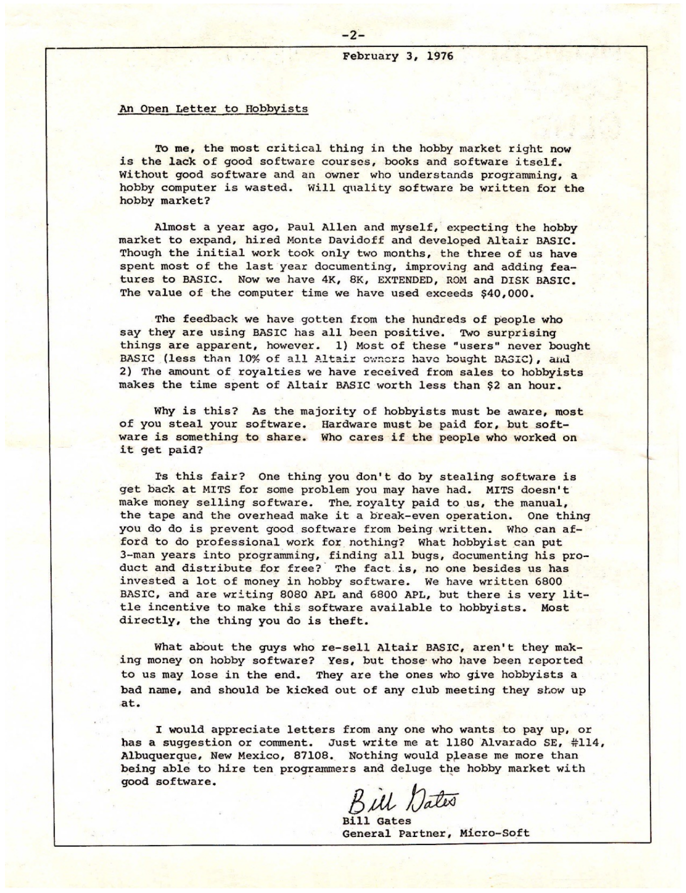
Given that annual new vehicle production is approaching 100 million per year and the global fleet is approximately 2 billion cars, it is impossible for Tesla to build electric cars fast enough to address the carbon crisis. By the same token, it means the market is enormous. Our true competition is not the small trickle of non-Tesla electric cars being produced, but rather the enormous flood of gasoline cars pouring out of the world's factories every day.

We believe that Tesla, other companies making electric cars, and the world would all benefit from a common, rapidly-evolving technology platform.

Technology leadership is not defined by patents, which history has repeatedly shown to be small protection indeed against a determined competitor, but rather by the ability of a company to attract and motivate the world's most talented engineers. We believe that applying the open source philosophy to our patents will strengthen rather than diminish Tesla's position in this regard.

Tables have turned

17-313 Foundations of SOFTWARE ENGINEERING



-2-

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds $40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than $2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

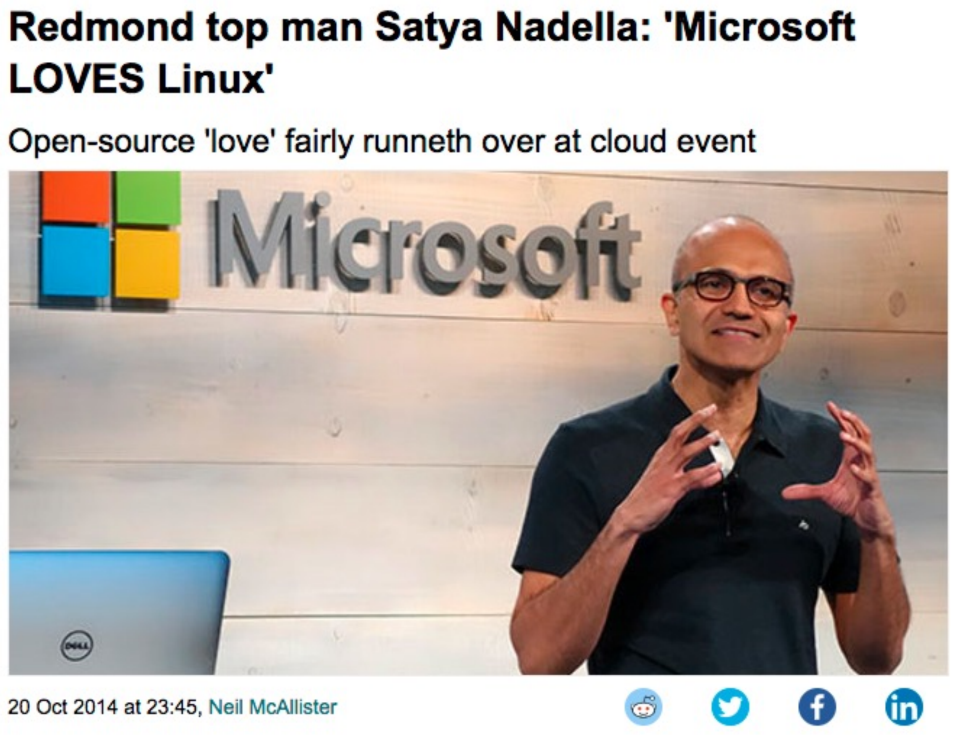
Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates

Bill Gates
General Partner, Micro-Soft

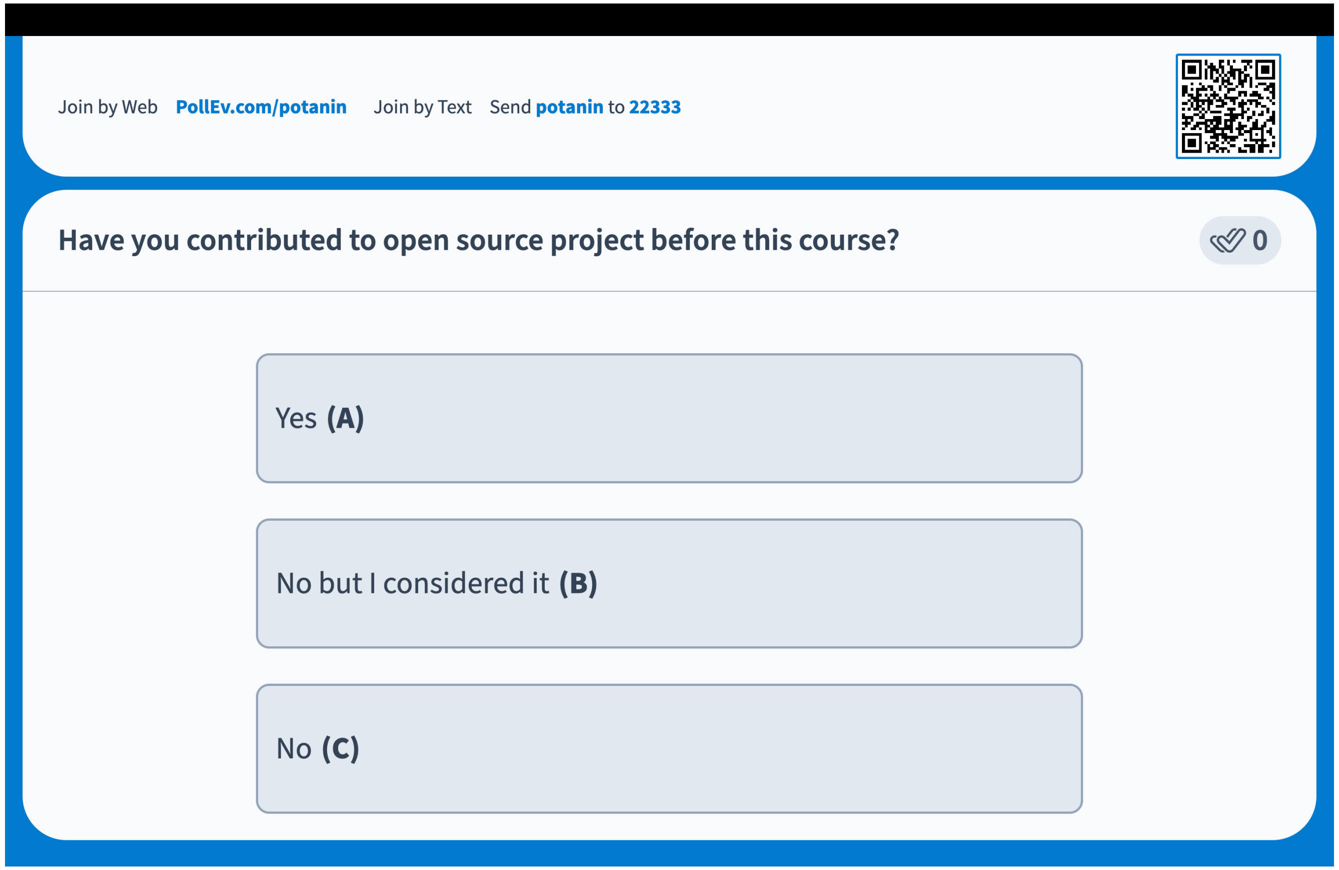


Redmond top man Satya Nadella: 'Microsoft LOVES Linux'

Open-source 'love' fairly runneth over at cloud event

20 Oct 2014 at 23:45, Neil McAllister

Poll Everywhere Time!



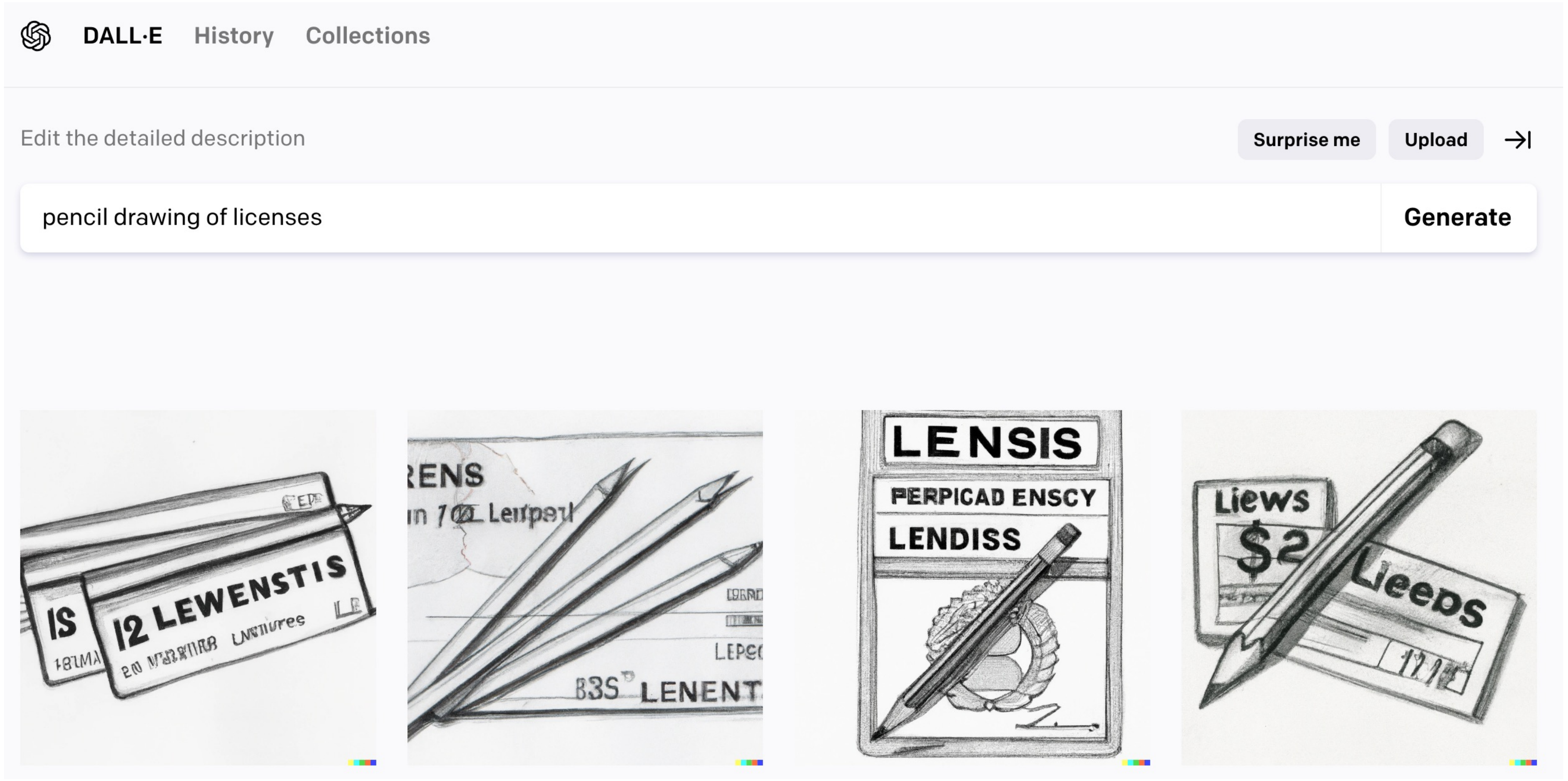
Join by Web **PollEv.com/potanin** Join by Text Send **potanin** to **22333**

Have you contributed to open source project before this course?

Yes **(A)**

No but I considered it **(B)**

No **(C)**



Edit the detailed description

pencil drawing of licenses

Licenses

Why learn about licenses?

- Companies will avoid certain licenses – commonly the copyleft licenses
- Specific licenses may provide competitive advantages
- You may eventually want to release open source software or become more involved in an open source project

Open Source Licenses

Software	Percentage
MIT License	24%
GNU General Public License (GPL) 2.0	23%
Apache License 2.0	16%
GNU General Public License (GPL) 3.0	9%
BSD License 2.0 (3-clause, New or Revised) License	6%
GNU Lessor General Public License (LGPL) 2.1	5%
Artistic License (Perl)	4%
GNU Lesser General Public License (LGPL) 3.0	2%
Microsoft Public License	2%
Eclipse Public License	2%

List from: https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses

GNU General Public License: The Copyleft License

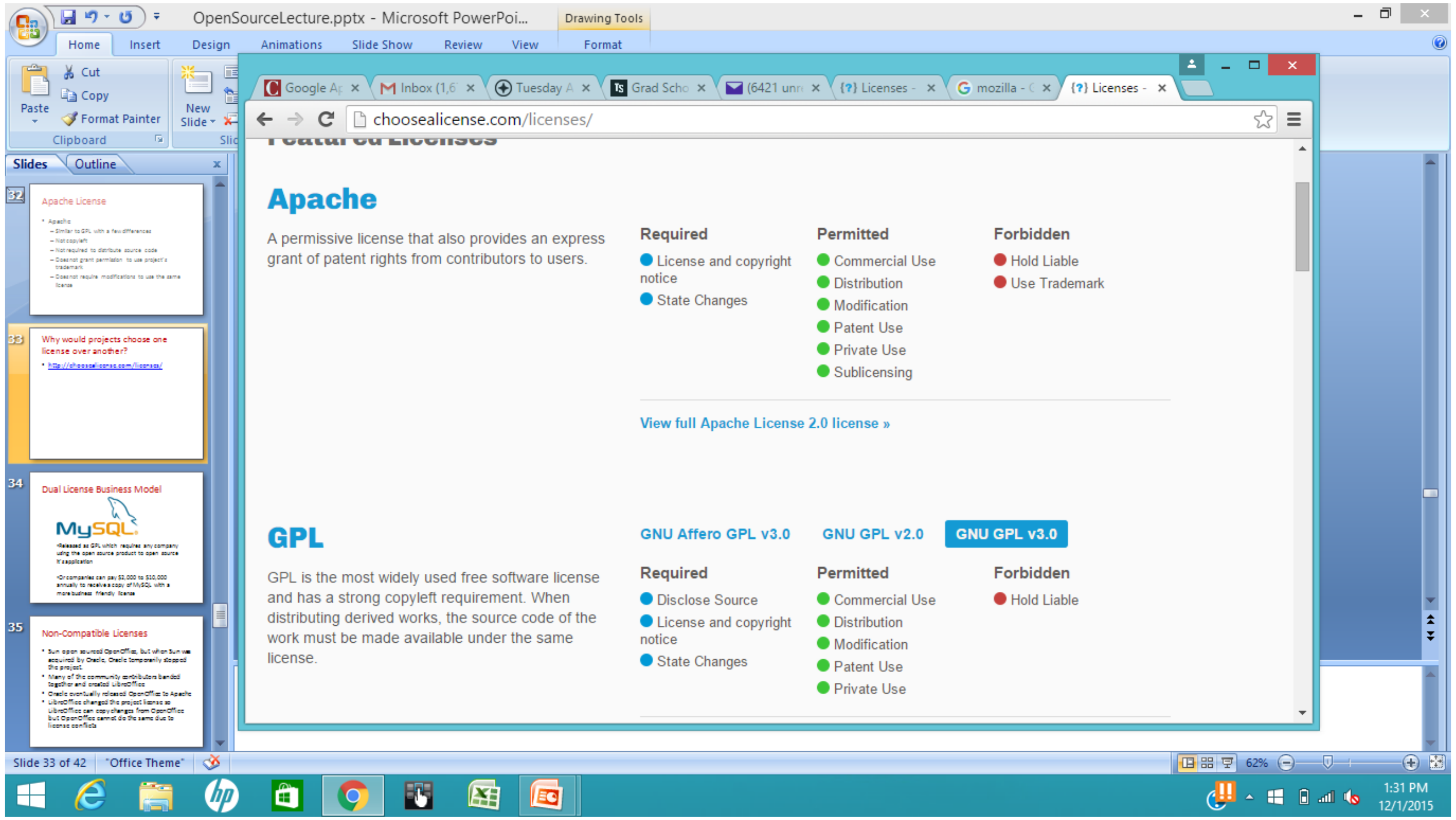
- Nobody should be restricted by the software they use. There are four freedoms that every user should have:
  - the freedom to use the software for any purpose,
  - the freedom to change the software to suit your needs,
  - the freedom to share the software with your friends and neighbors, and
  - the freedom to share the changes you make.
- Code must be made available
- Any modifications must be relicensed under the same license (copyleft)

GPL 2.0 and 3.0 – Addresses free software problems

- 2.0 - Court ruling cannot nullify the license and if a court decision and this license contradict in distribution requirements, then the software cannot be distributed
- 3.0 – patent grant and prevent Tivoization
- Not compatible with each other; Can’t copyleft both at the same time – phrase: “GLP Version 3 or any later version”

Why would projects choose one license over another?

17-313 Foundations of SOFTWARE ENGINEERING



OpenSourceLecture.pptx - Microsoft PowerPoi...
Drawing Tools
Home Insert Design Animations Slide Show Review View Format
Cut
Copy
Format Painter
Paste
Clipboard
New Slide
Slides Outline
choosealicense.com/licenses/
Slide 33 of 42 "Office Theme"
1:31 PM 12/1/2015

Apache

A permissive license that also provides an express grant of patent rights from contributors to users.

Required	Permitted	Forbidden
● License and copyright notice	● Commercial Use	● Hold Liable
● State Changes	● Distribution	● Use Trademark
	● Modification	
	● Patent Use	
	● Private Use	
	● Sublicensing	

View full Apache License 2.0 license »

GPL

GNU Affero GPL v3.0 GNU GPL v2.0 GNU GPL v3.0

GPL is the most widely used free software license and has a strong copyleft requirement. When distributing derived works, the source code of the work must be made available under the same license.

Required	Permitted	Forbidden
● Disclose Source	● Commercial Use	● Hold Liable
● License and copyright notice	● Distribution	
● State Changes	● Modification	
	● Patent Use	
	● Private Use	

[From http://choosealicense.com/licenses/]

Dual License Business Model

- Released as GPL which requires a company using the open source product to open source it's application

- Or companies can pay $2,000 to $10,000 annually to receive a copy of MySQL with a more business friendly license

Risk: Incompatible Licenses

- Sun open sourced OpenOffice, but when Sun was acquired by Oracle, Oracle temporarily stopped the project.
- Many of the community contributors banded together and created LibreOffice
- Oracle eventually released OpenOffice to Apache
- LibreOffice changed the project license so LibreOffice can copy changes from OpenOffice but OpenOffice cannot do the same due to license conflicts

MIT License

- Must retain copyright credit
- Software is provided as is
- Authors are not liable for software
- No other restrictions

LGPL

- Software must be a library
- Similar to GPL but no copyleft requirement

BSD License

- No liability and provided as is.
- Copyright statement must be included in source and binary
- The copyright holder does not endorse any extensions without explicit written consent

Apache License

- Apache
  - Similar to GPL with a few differences
  - Not copyleft
  - Not required to distribute source code
  - Does not grant permission to use project's trademark
  - Does not require modifications to use the same license

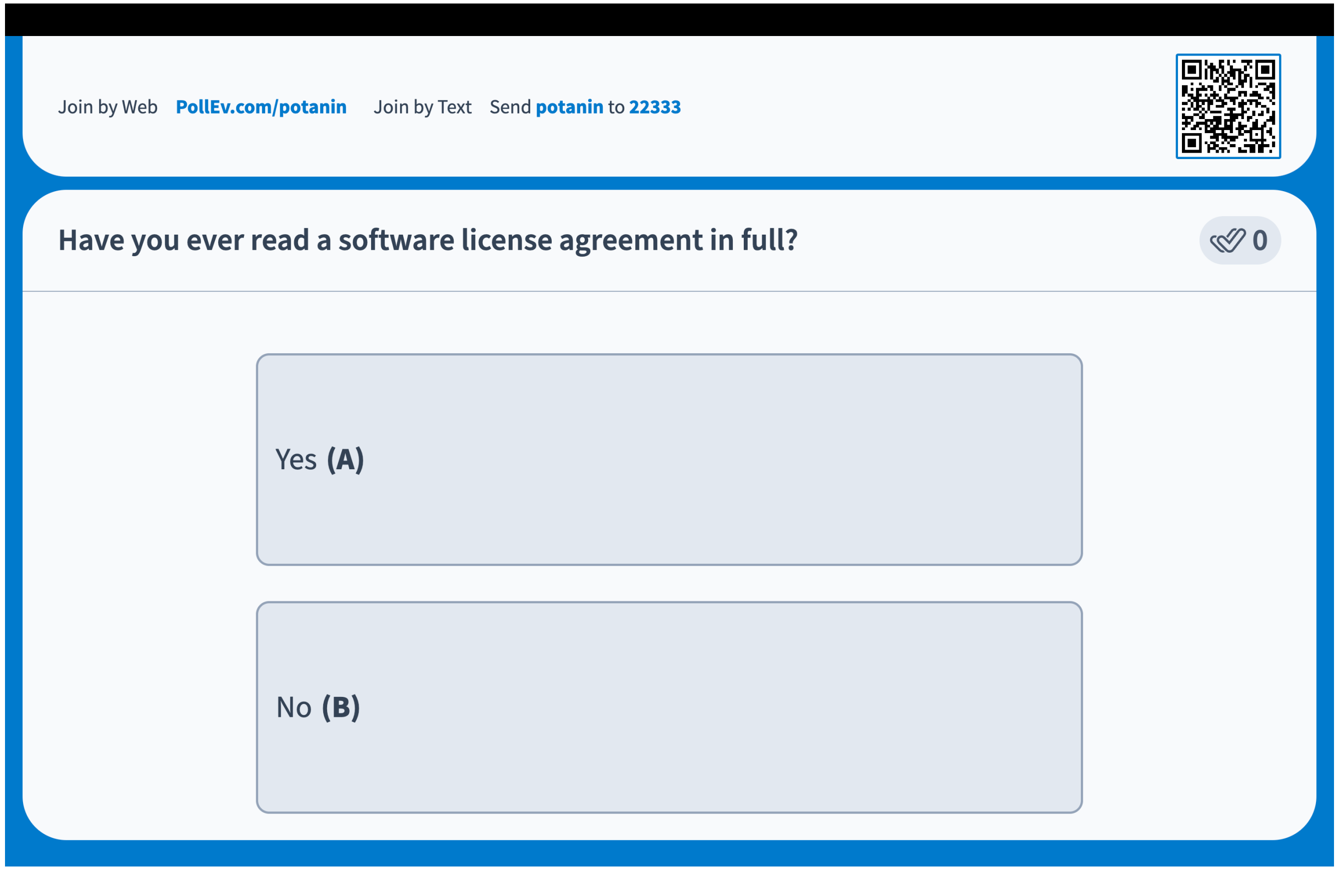
Perception:

- Anarchy
- Demagoguery
- Ideology
- Altruism
- Many eyes

WHEN YOU PROGRAM OPEN SOURCE, YOU'RE PROGRAMMING
COMMUNISM
A REMINDER from YOUR FRIENDS AT MICROSOFT

17-313 Foundations of SOFTWARE ENGINEERING

Poll Everywhere Time!

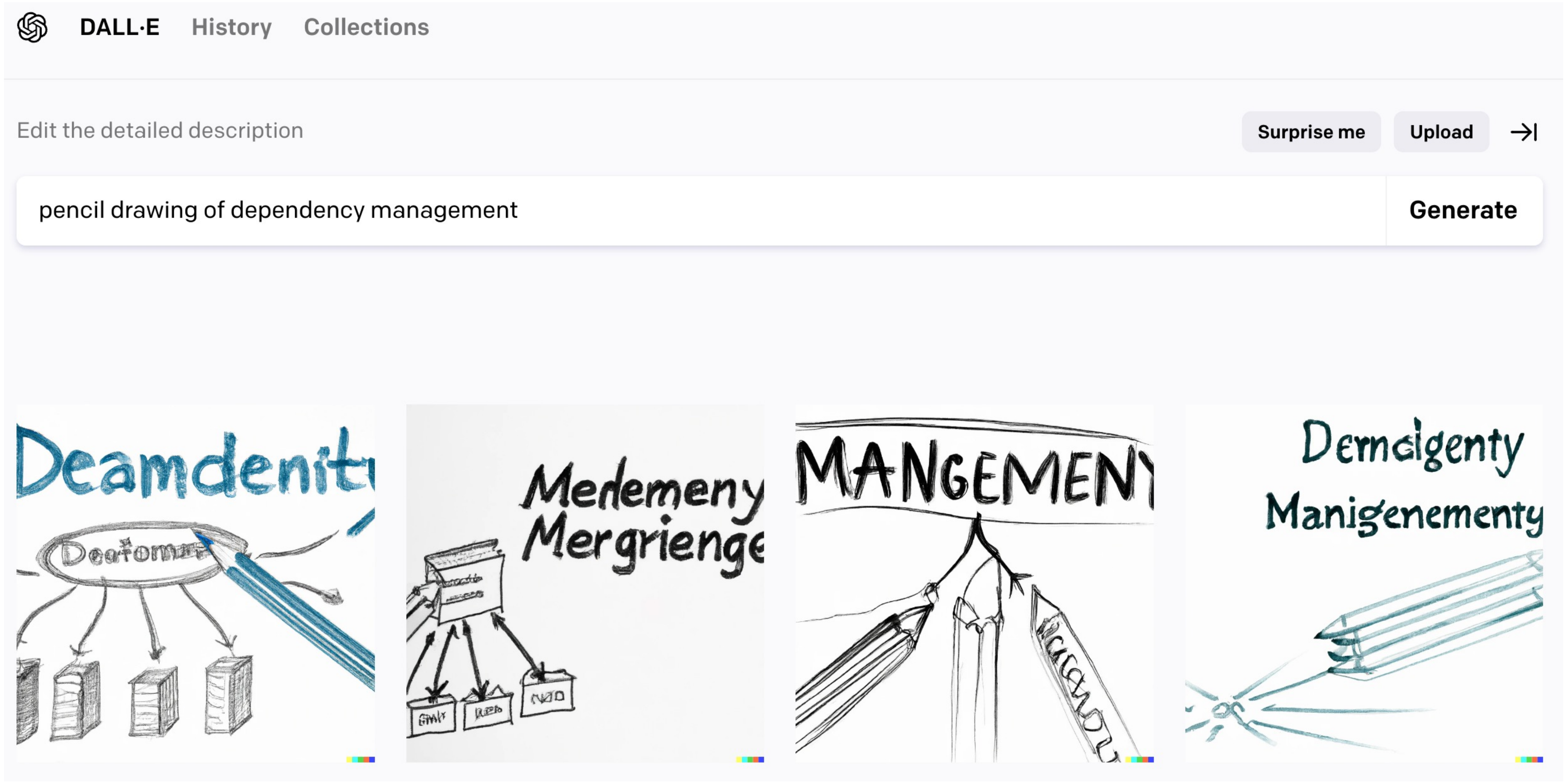


Join by Web **PollEv.com/potanin** Join by Text Send **potanin** to **22333**

Have you ever read a software license agreement in full?

Yes **(A)**

No **(B)**

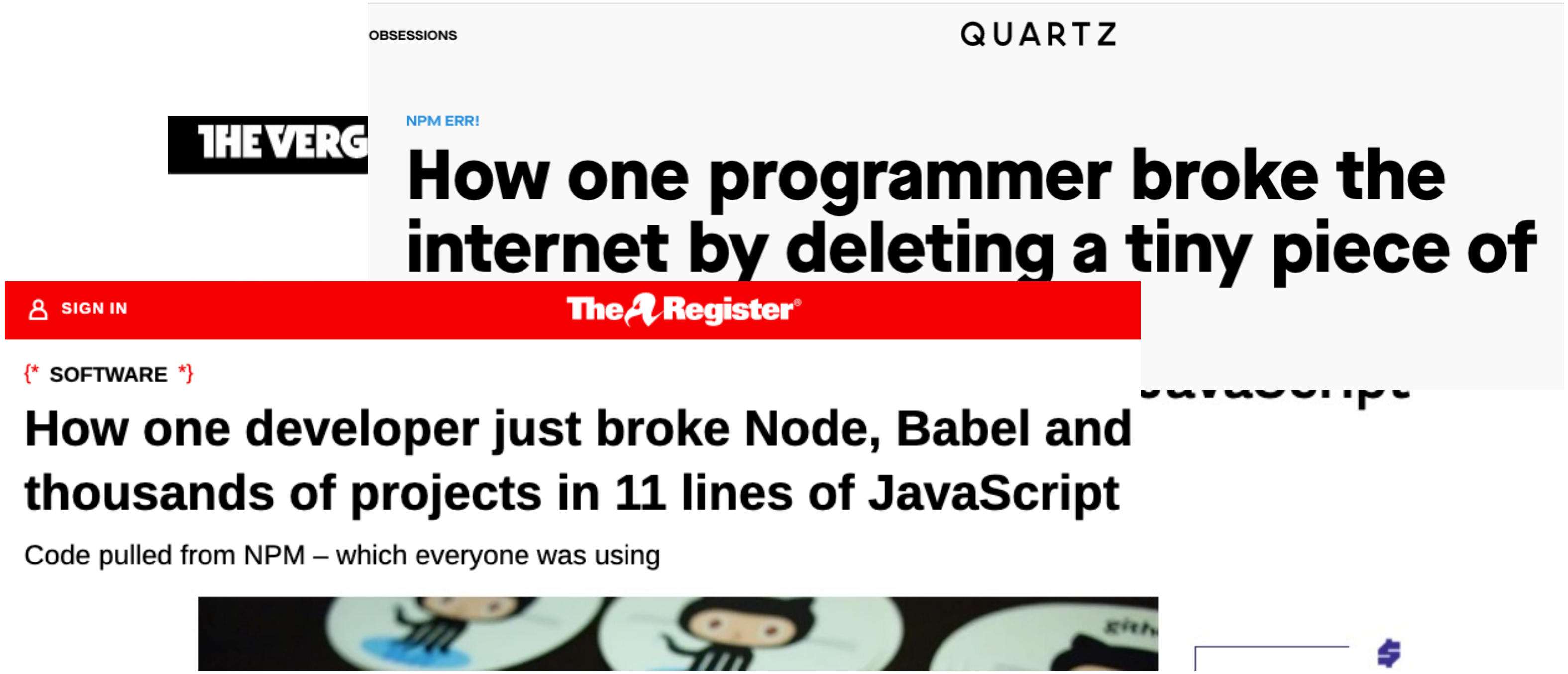


pencil drawing of dependency management

Dependency Management

Left-pad (March 22, 2016)

17-313 Foundations of SOFTWARE ENGINEERING



OBSESSIONS QUARTZ

THE VERG

NPM ERR!

How one programmer broke the internet by deleting a tiny piece of

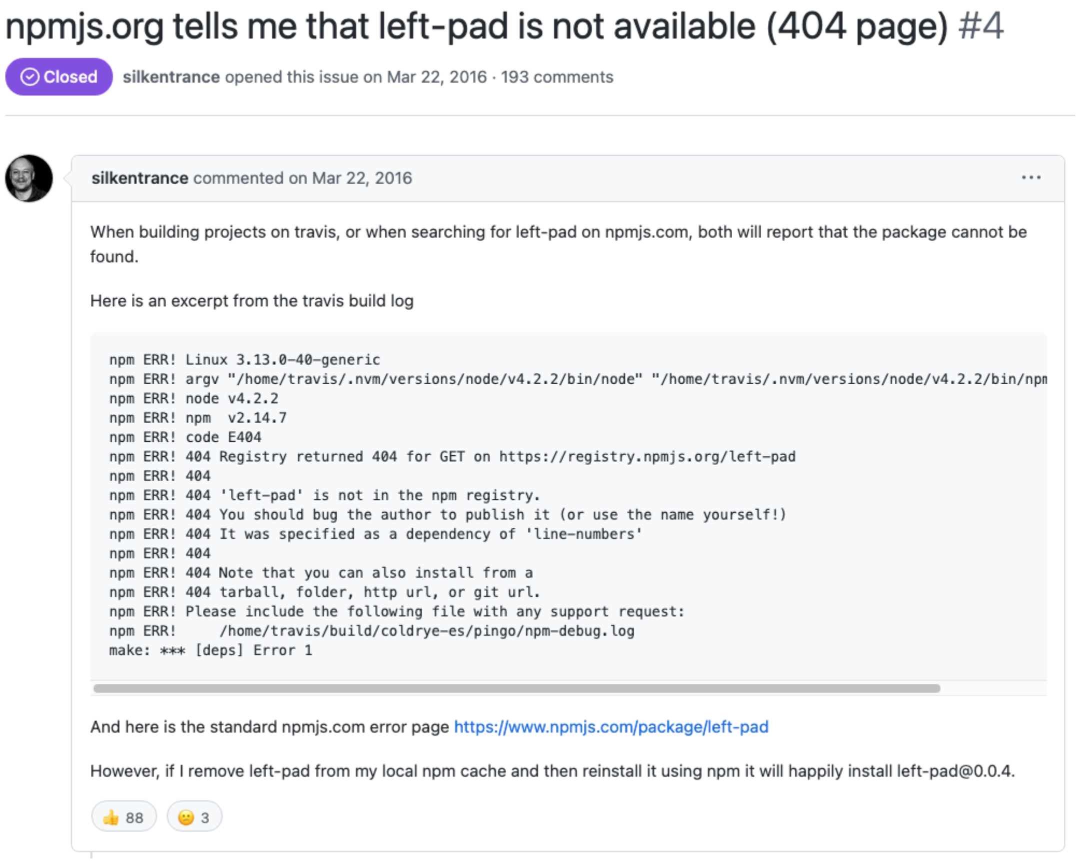
SIGN IN The Register

{* SOFTWARE *}

How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

Code pulled from NPM – which everyone was using

Left-pad (March 22, 2016)



npmjs.org tells me that left-pad is not available (404 page) #4

Closed silkentrance opened this issue on Mar 22, 2016 · 193 comments

silkentrance commented on Mar 22, 2016

When building projects on travis, or when searching for left-pad on npmjs.com, both will report that the package cannot be found.

Here is an excerpt from the travis build log

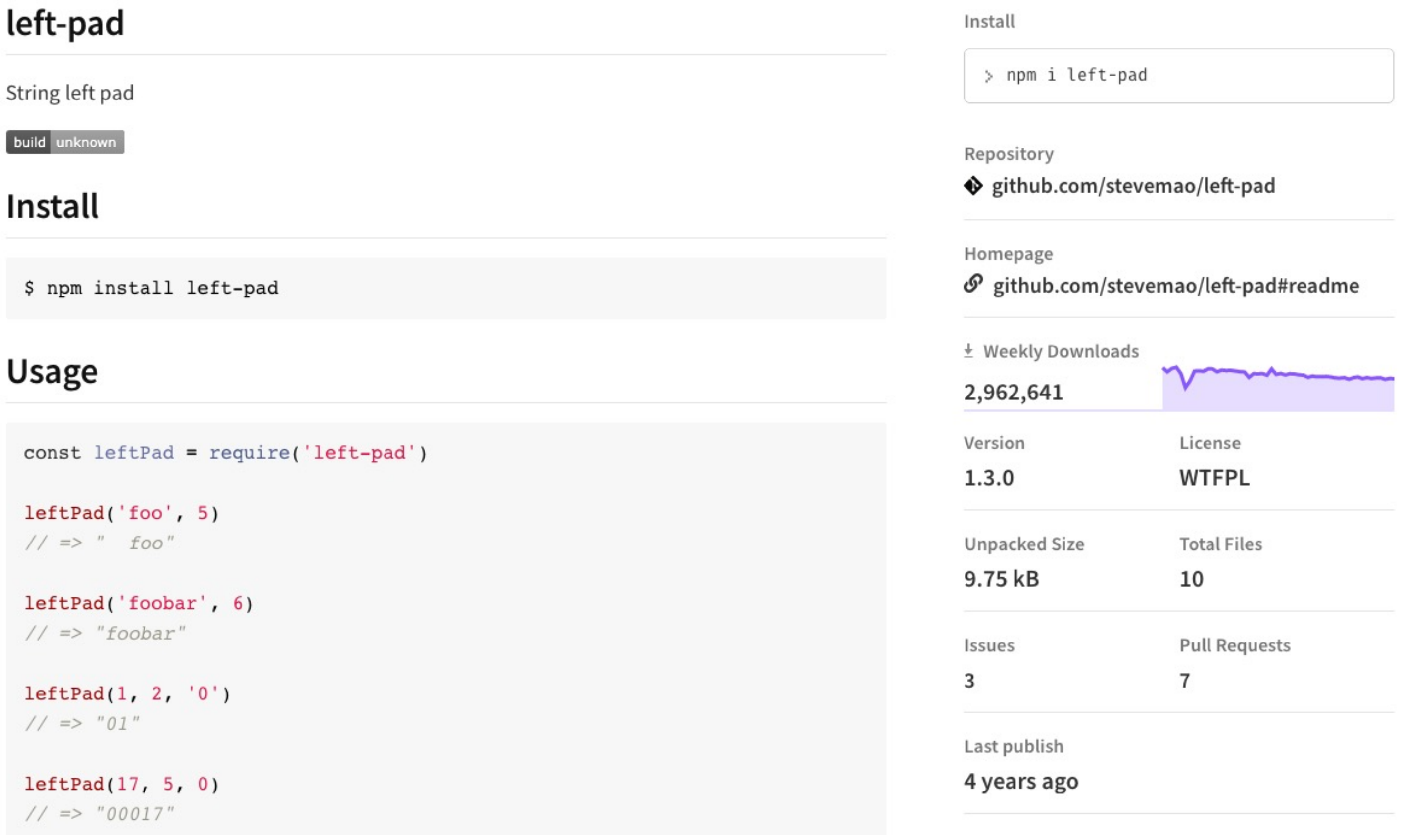
```
npm ERR! Linux 3.13.0-40-generic
npm ERR! argv "/home/travis/.nvm/versions/node/v4.2.2/bin/node" "/home/travis/.nvm/versions/node/v4.2.2/bin/npm
npm ERR! node v4.2.2
npm ERR! npm  v2.14.7
npm ERR! code E404
npm ERR! 404 Registry returned 404 for GET on https://registry.npmjs.org/left-pad
npm ERR! 404
npm ERR! 404 'left-pad' is not in the npm registry.
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)
npm ERR! 404 It was specified as a dependency of 'line-numbers'
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.
npm ERR! Please include the following file with any support request:
npm ERR!     /home/travis/build/coldrye-es/pingo/npm-debug.log
make: *** [deps] Error 1
```

And here is the standard npmjs.com error page https://www.npmjs.com/package/left-pad

However, if I remove left-pad from my local npm cache and then reinstall it using npm it will happily install left-pad@0.0.4.

👍 88 😕 3

Left-pad (Docs)



left-pad

String left pad

build unknown

Install

```
$ npm install left-pad
```

Usage

```
const leftPad = require('left-pad')

leftPad('foo', 5)
// => "  foo"

leftPad('foobar', 6)
// => "foobar"

leftPad(1, 2, '0')
// => "01"

leftPad(17, 5, 0)
// => "00017"
```

Install

```
> npm i left-pad
```

Repository
github.com/stevemao/left-pad

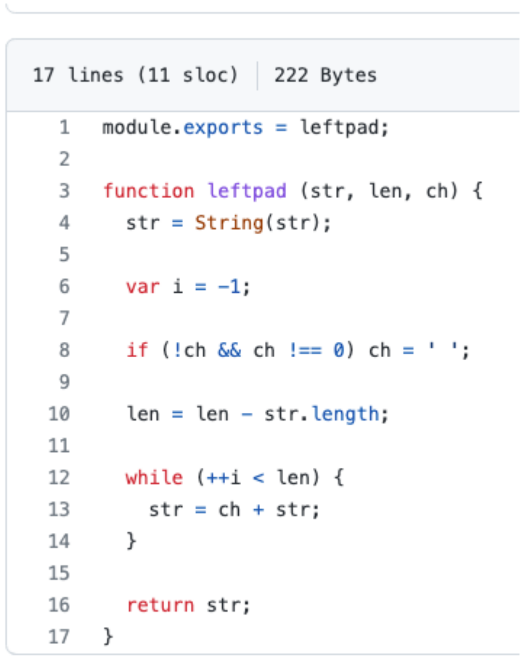
Homepage
github.com/stevemao/left-pad#readme

Weekly Downloads
2,962,641

Version	License
1.3.0	WTFPL
Unpacked Size	Total Files
9.75 kB	10
Issues	Pull Requests
3	7

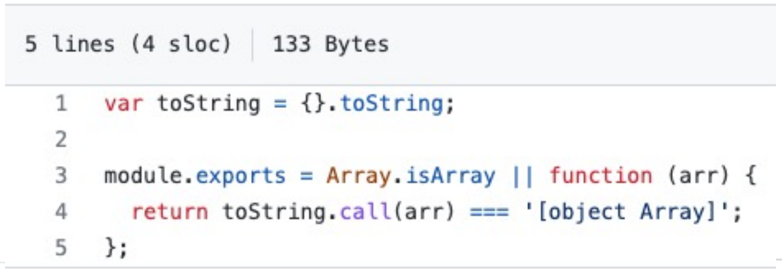
Last publish
4 years ago

Left-pad (Source Code)


17 lines (11 sloc) | 222 Bytes

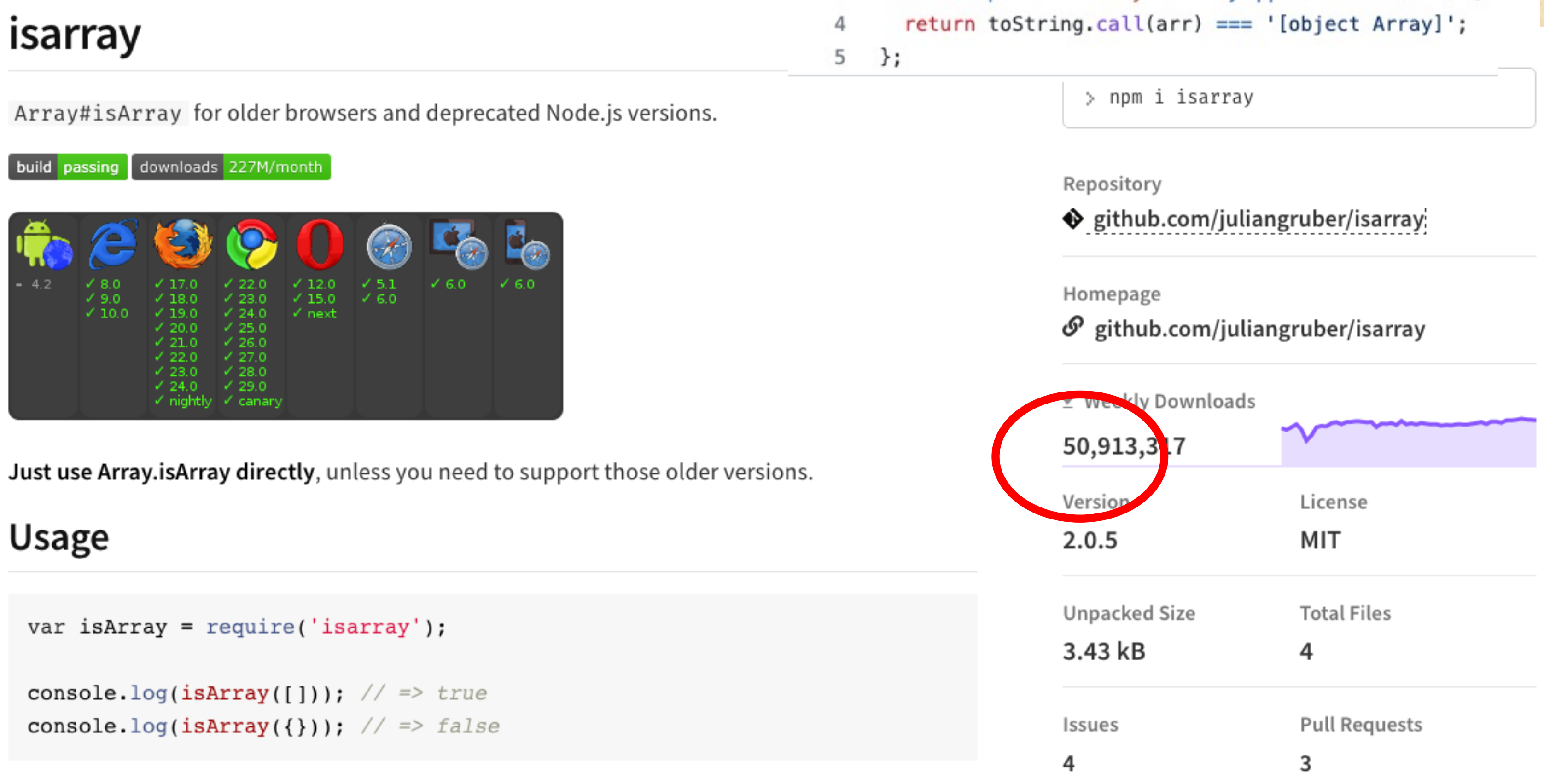
```javascript
module.exports = leftpad;

function leftpad (str, len, ch) {
  str = String(str);

  var i = -1;

  if (!ch && ch !== 0) ch = ' ';

  len = len - str.length;

  while (++i < len) {
    str = ch + str;
  }

  return str;
}
```

See also: isArray



```
5 lines (4 sloc) | 133 Bytes
```

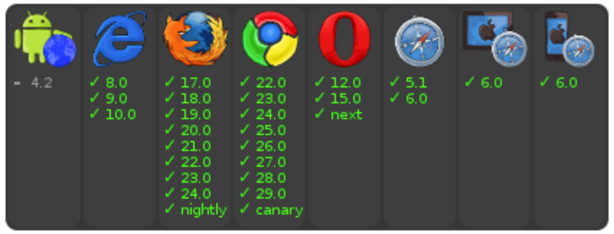
```
var toString = {}.toString;

module.exports = Array.isArray || function (arr) {
  return toString.call(arr) === '[object Array]';
};
```



isarray

`Array#isArray` for older browsers and deprecated Node.js versions.

build passing | downloads 227M/month



- 4.2	✓ 8.0	✓ 17.0	✓ 22.0	✓ 12.0	✓ 5.1	✓ 6.0	✓ 6.0
	✓ 9.0	✓ 18.0	✓ 23.0	✓ 15.0	✓ 6.0		
	✓ 10.0	✓ 19.0	✓ 24.0	✓ next			
		✓ 20.0	✓ 25.0				
		✓ 21.0	✓ 26.0				
		✓ 22.0	✓ 27.0				
		✓ 23.0	✓ 28.0				
		✓ 24.0	✓ 29.0				
		✓ nightly	✓ canary				

Just use Array.isArray directly, unless you need to support those older versions.

Usage

```
var isArray = require('isarray');

console.log(isArray([])); // => true
console.log(isArray({})); // => false
```

```
> npm i isarray
```

Repository
github.com/juliangruber/isarray

Homepage
github.com/juliangruber/isarray

Weekly Downloads
50,913,317

Version	License
2.0.5	MIT

Unpacked Size	Total Files
3.43 kB	4

Issues	Pull Requests
4	3

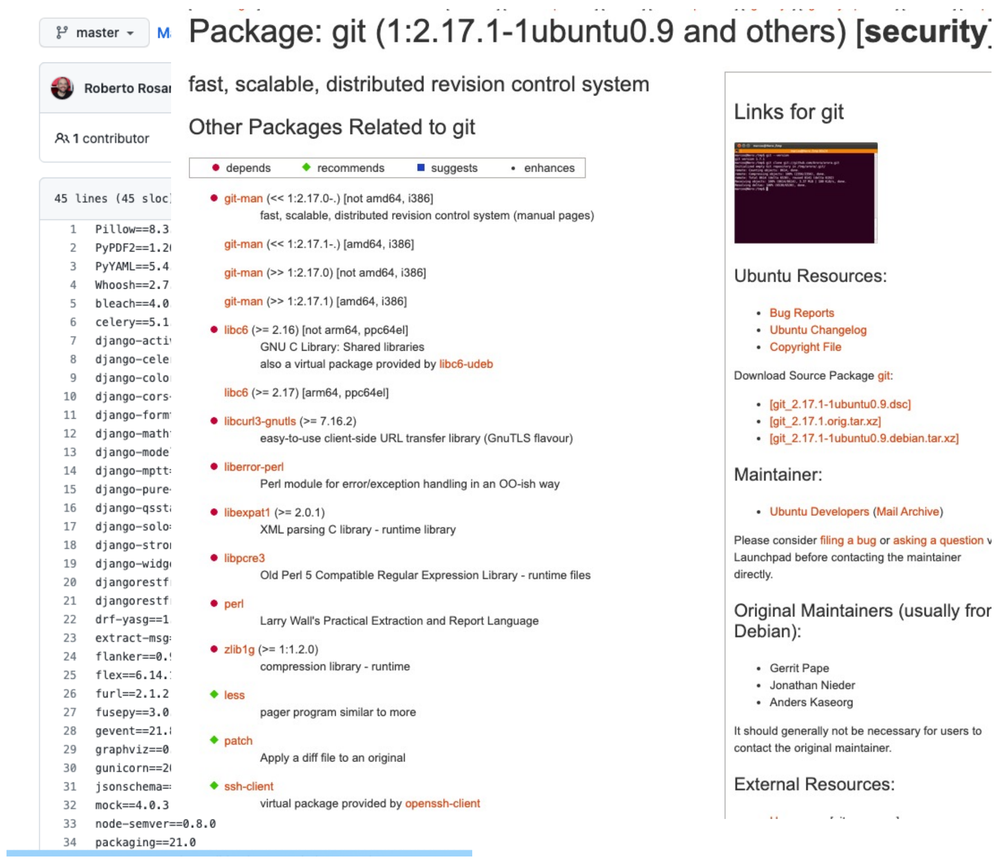
Dependency Management

- It’s hard
- It’s mostly a mess (everywhere)
- But it’s critical to modern software development

What is a Dependency?

- Core of what most build systems do
  - “Compile” and “Run Tests” is just a fraction of their job
- Examples: Maven, Gradle, NPM, Bazel, …
- **Foo->Bar**: To build Foo, you may need to have a built version of Bar
- Dependency Scopes:
  - **Compile**: Foo uses classes, functions, etc. defined by Bar
  - **Runtime**: Foo uses an abstract API whose implementation is provided by Bar (e.g. logging, database, network or other I/O)
  - **Test**: Foo needs Bar only for tests (e.g. JUnit, mocks)
- Internal vs. External Dependencies
  - Is Bar also built/maintained by your org or is it pulled from elsewhere using a package manager?

Dependencies: Example


master ▾ M

Roberto Rosa

1 contributor

```
45 lines (45 sloc
Pillow==8.3
PyPDF2==1.2
PyYAML==5.4
Whoosh==2.7
bleach==4.0
celery==5.1
django-acti
django-cele
django-colo
django-cors
django-form
django-math
django-mode
django-mptt
django-pure
django-qsst
django-solo
django-stro
django-widg
djangorestf
djangorestf
drf-yasg==1
extract-msg
flanker==0.
flex==6.14.
furl==2.1.2
fusepy==3.0
gevent==21.
graphviz==0
gunicorn==2
jsonschema=
mock==4.0.3
node-semver==0.8.0
packaging==21.0
```

Package: git (1:2.17.1-1ubuntu0.9 and others) **[security**

fast, scalable, distributed revision control system

Other Packages Related to git

● depends ◆ recommends ■ suggests • enhances

- ● git-man (<< 1:2.17.0-.) [not amd64, i386]
  fast, scalable, distributed revision control system (manual pages)
  git-man (<< 1:2.17.1-.) [amd64, i386]
  git-man (>> 1:2.17.0) [not amd64, i386]
  git-man (>> 1:2.17.1) [amd64, i386]
- ● libc6 (>= 2.16) [not arm64, ppc64el]
  GNU C Library: Shared libraries
  also a virtual package provided by libc6-udeb
  libc6 (>= 2.17) [arm64, ppc64el]
- ● libcurl3-gnutls (>= 7.16.2)
  easy-to-use client-side URL transfer library (GnuTLS flavour)
- ● liberror-perl
  Perl module for error/exception handling in an OO-ish way
- ● libexpat1 (>= 2.0.1)
  XML parsing C library - runtime library
- ● libpcre3
  Old Perl 5 Compatible Regular Expression Library - runtime files
- ● perl
  Larry Wall's Practical Extraction and Report Language
- ● zlib1g (>= 1:1.2.0)
  compression library - runtime
- ◆ less
  pager program similar to more
- ◆ patch
  Apply a diff file to an original
- ◆ ssh-client
  virtual package provided by openssh-client

Links for git

Ubuntu Resources:

- Bug Reports
- Ubuntu Changelog
- Copyright File

Download Source Package git:

- [git_2.17.1-1ubuntu0.9.dsc]
- [git_2.17.1.orig.tar.xz]
- [git_2.17.1-1ubuntu0.9.debian.tar.xz]

Maintainer:

- Ubuntu Developers (Mail Archive)

Please consider filing a bug or asking a question v Launchpad before contacting the maintainer directly.

Original Maintainers (usually fror Debian):

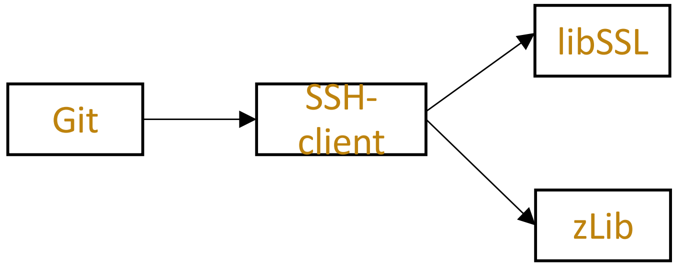
- Gerrit Pape
- Jonathan Nieder
- Anders Kaseorg

It should generally not be necessary for users to contact the original maintainer.

External Resources:

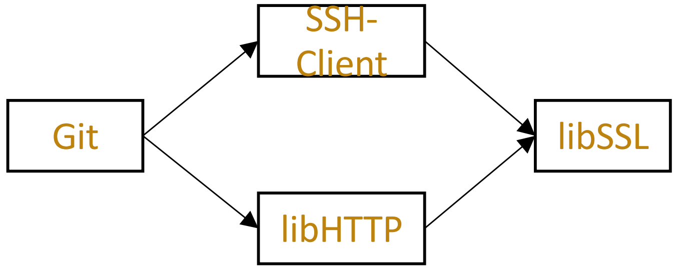
Transitive Dependencies

- Should Git be able to use exports of libSSL (e.g. certificate management) or zLib (e.g. gzip compression)?

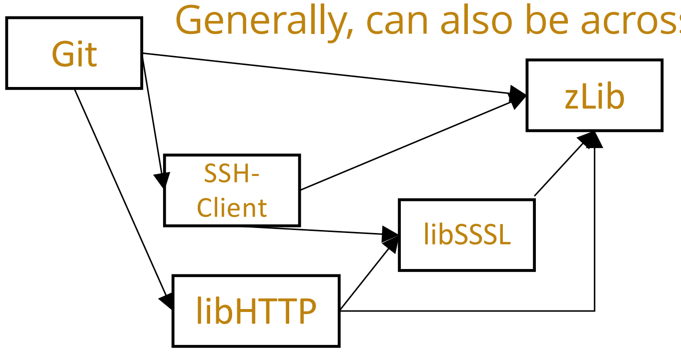


Diamond Dependencies

- What are some problems when multiple intermediate dependencies have the same transitive dependency?

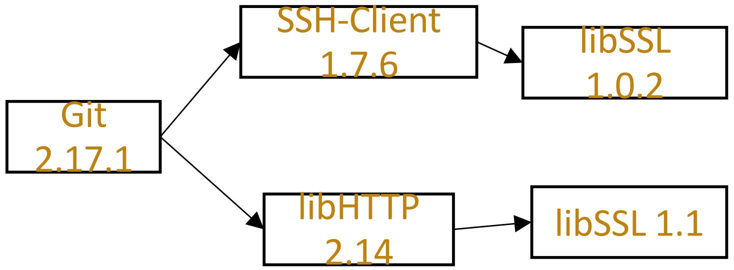


Generally, can also be across levels



Diamond Dependencies

- What are some problems when multiple intermediate dependencies have the same transitive dependency?



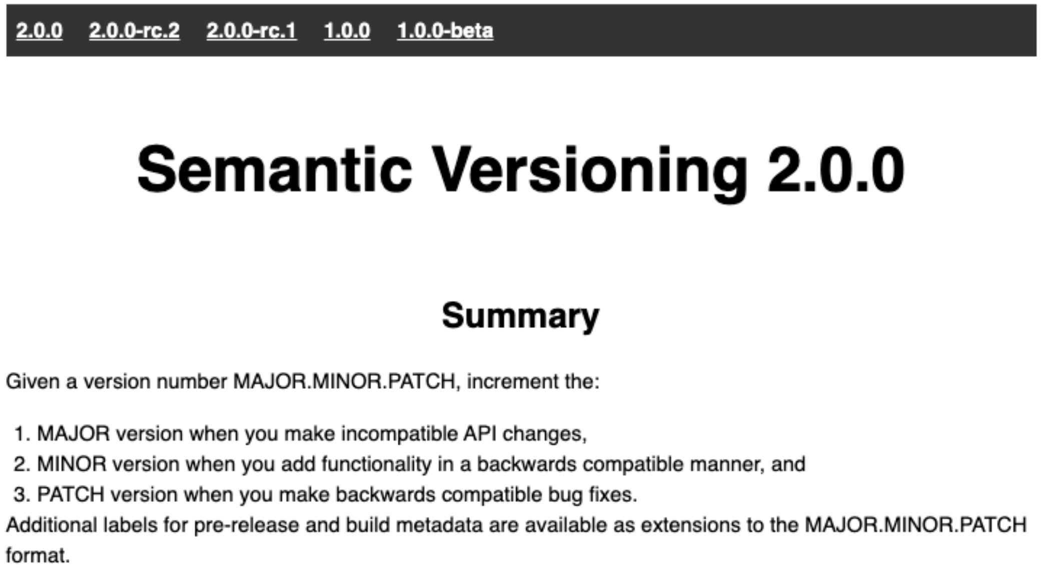
Resolutions to the Diamond Problem

1. Duplicate it!
   - Doesn't work with static linking (e.g. C/C++), but may be doable with Java (e.g. using ClassLoader hacking or package renaming)
   - Values of types defined by duplicated libraries cannot be exchanged across
2. Ban transitive dependencies; just use a global list with one version for each
   - Challenge: Keeping things in sync with latest
   - Challenge: Deciding which version of transitive deps to keep
3. Newest version (keep everything at latest)
   - Requires ordering semantics
   - Intermediate dependency may break with update to transitive
4. Oldest version (lowest denominator)
   - Also requires ordering semantics
   - Sacrifices new functionality
5. Oldest non-breaking version / Newest non-breaking version
   - Requires faith in tests or semantic versioning contract

Semantic Versioning

- Widely used convention for versioning releases
  - E.g. 1.2.1, 3.1.0-alpha-1, 3.1.0-alpha-2, 3.1.0-beta-1, 3.1.0-rc1
- Format: {MAJOR} . {MINOR} . {PATCH}
- Each component is ordered (numerically, then lexicographically; release-aware)
  - 1.2.1 < 1.10.1
  - 3.1.0-alpha-1 < 3.1.0-alpha-2 < 3.1.0-beta-1 < 3.1.0-rc1 < 3.1.0
- Contracts:
  - MAJOR updated to indicate breaking changes
    - Same MAJOR version => backward compatibility
  - MINOR updated for additive changes
    - Same MINOR version => API compatibility (important for linking)
  - PATCH updates functionality without new API
    - Ninja edit; usually for bug fixes

https://semver.org/



Semantic Versioning 2.0.0

Summary

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

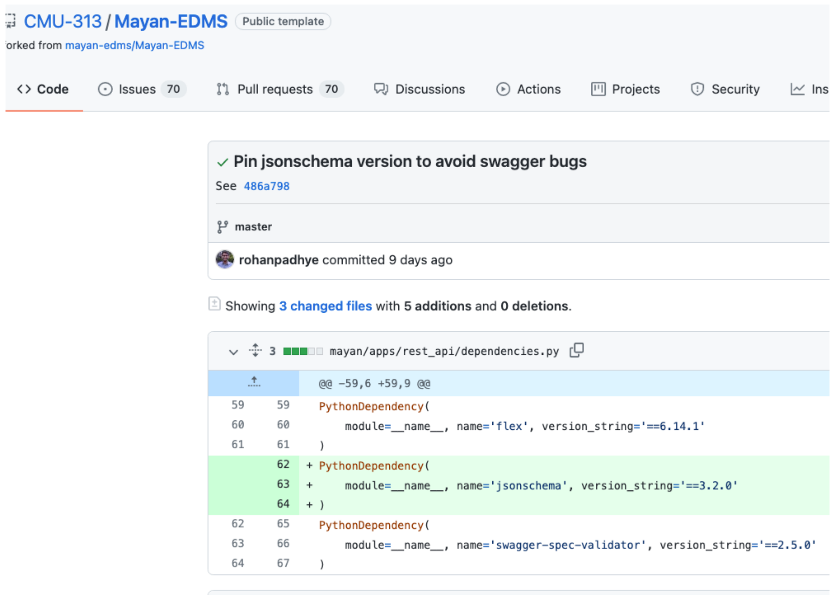
Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Dependency Constraints

- E.g. Declare dependency on ”Bar > 2.1”
  - Bar 2.1.0, 2.1.1, 2.2.0, 2.9.0, etc. all match
  - 2.0.x does NOT match
  - 3.0.x does NOT match
- Diamond dependency problem can be resolved using SAT solvers
  - E.g. Foo 1.0.0 depends on “Bar >= 2.1” and “Baz 1.8.x”
    - Bar 2.1.0 depends on “Qux [1.6, 1.7]”
    - Bar 2.1.1 depends on “Qux 1.7.0”
    - Baz 1.8.0 depends on “Qux 1.5.x”
    - Baz 1.8.1 depends on “Qux 1.6.x”
  - Find an assignment such that all dependencies are satisfied
    - Solution: Use Bar 2.1.0, Baz 1.8.1, and Qux 1.6.{latest}

Semantic Versioning Contracts

- Largely trusting developers to maintain them
- Constrained/range dependencies can cause unexpected build failures
- Automatic validation of SemVer is hard


CMU-313 / Mayan-EDMS Public template
orked from mayan-edms/Mayan-EDMS

Pin jsonschema version to avoid swagger bugs
See 486a798

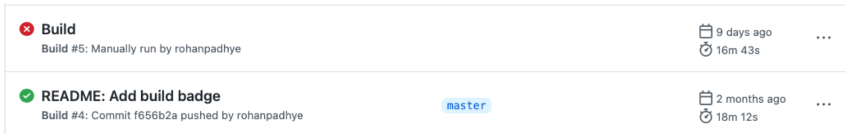
master

rohanpadhye committed 9 days ago

Showing 3 changed files with 5 additions and 0 deletions.

3 mayan/apps/rest_api/dependencies.py

```
@@ -59,6 +59,9 @@
  PythonDependency(
      module=__name__, name='flex', version_string='==6.14.1'
  )
+ PythonDependency(
+     module=__name__, name='jsonschema', version_string='==3.2.0'
+ )
  PythonDependency(
      module=__name__, name='swagger-spec-validator', version_string='==2.5.0'
  )
```


Build
Build #5: Manually run by rohanpadhye
9 days ago
16m 43s

README: Add build badge
Build #4: Commit f656b2a pushed by rohanpadhye
master
2 months ago
18m 12s

Cyclic Dependencies

- A very bad thing
- Avoid at all costs
- Sometimes unavoidable or intentional
  - E.g. GCC is written in C (needs a C compiler)
  - E.g. Apache Maven uses the Maven build system
  - E.g. JDK tested using JUnit, which requires the JDK to compile

$$\begin{tikzcd}\boxed{A} \arrow[r, bend left] & \boxed{B} \arrow[l, bend left]\end{tikzcd}$$

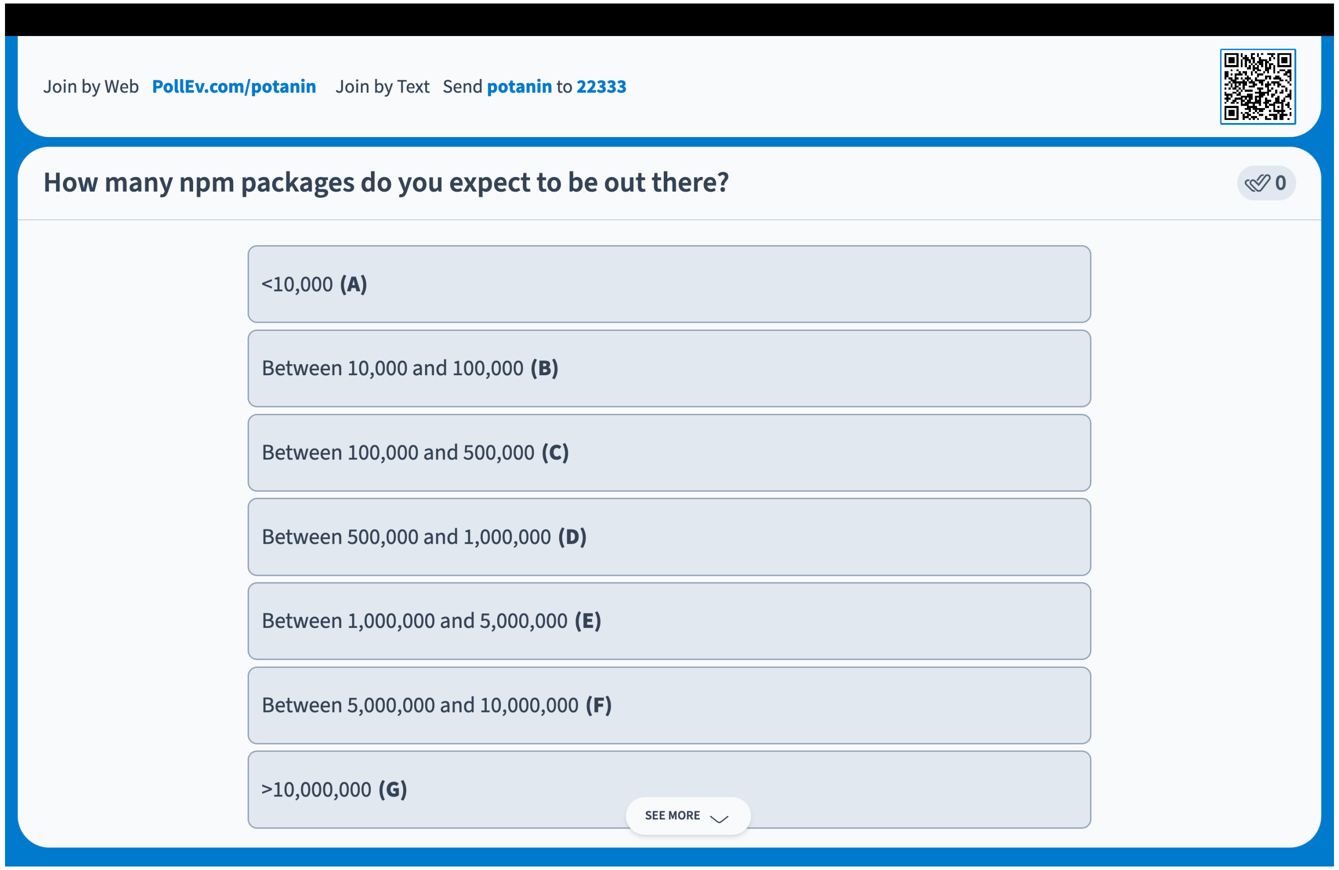
Cyclic Dependencies

- Bootstrapping: Break cycles over time
- Assume older version exists in binary (pre-built form)
- Step 1: Build A using an older version of B
- Step 2: Build B using new (just built) version of A
- Step 3: Rebuild A using new (just built) version of B
- Now, both A and B have been built with new versions of their dependencies
- Doesn’t work if both A and B need new features of each other at the same time (otherwise Step 1 won’t work)
  - Assumes incremental dependence on new features
- How was the old version built in the first place? (it’s turtles all the way down)
  - Assumption: cycles did not exist in the past
  - Successfully applied in compilers (e.g. GCC is written in C)

Dependency Reliability

- Availability
  - Remember left-pad?
  - Many orgs will mirror package repositories
- Security
  - Will you let strangers execute arbitrary code on your laptop?
  - Think about this every time you do “pip install” or “npm install” or “apt-get upgrade” or “brew upgrade” or whatever (esp. with sudo)
    - Scary, right? Who are you trusting? Why?
  - Typo squatting
    - “pip install numpi”

Poll Everywhere Time!



Join by Web **PollEv.com/potanin** Join by Text Send **potanin** to **22333**

How many npm packages do you expect to be out there?

0

<10,000 **(A)**

Between 10,000 and 100,000 **(B)**

Between 100,000 and 500,000 **(C)**

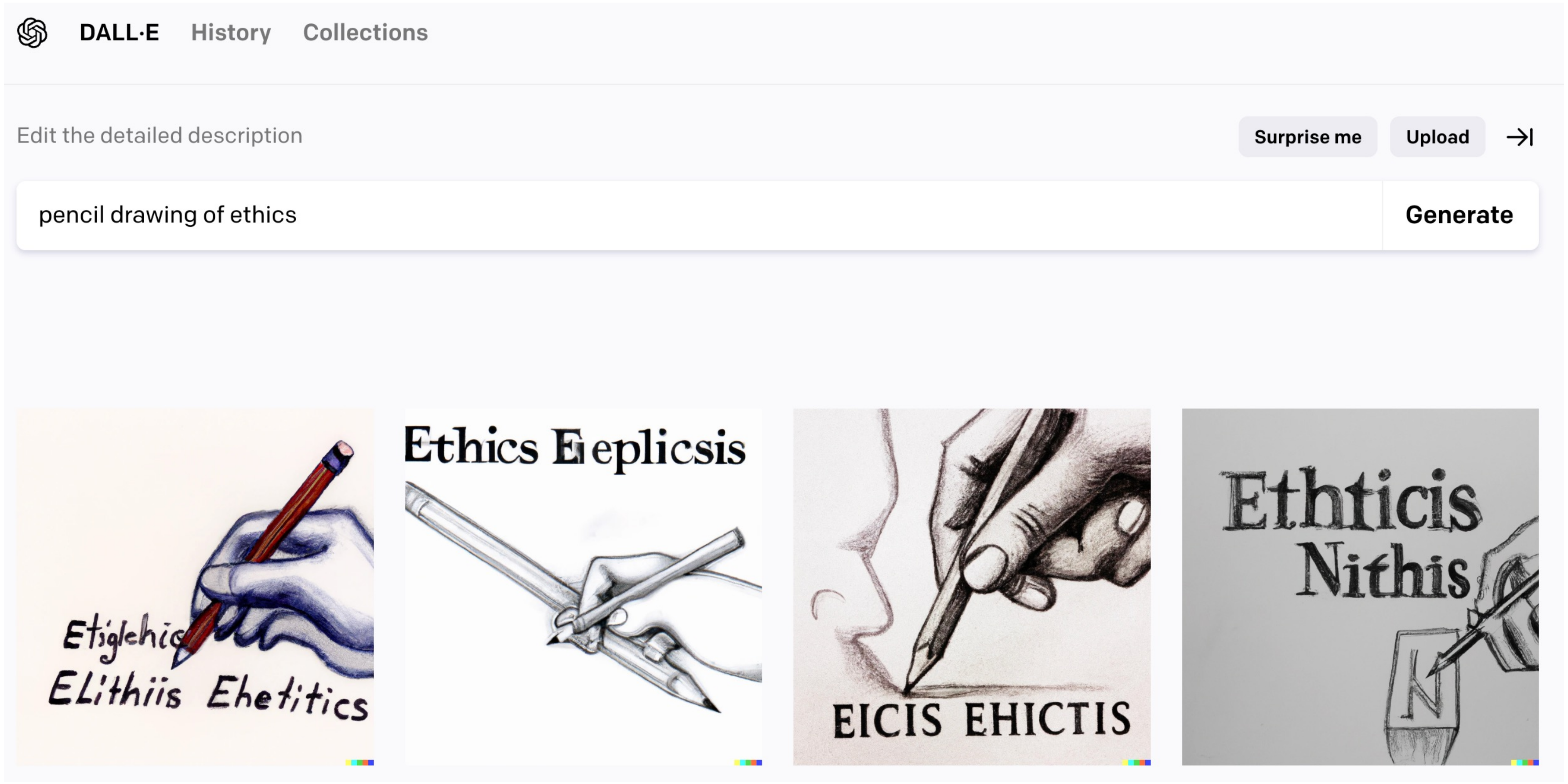
Between 500,000 and 1,000,000 **(D)**

Between 1,000,000 and 5,000,000 **(E)**

Between 5,000,000 and 10,000,000 **(F)**

>10,000,000 **(G)**

SEE MORE



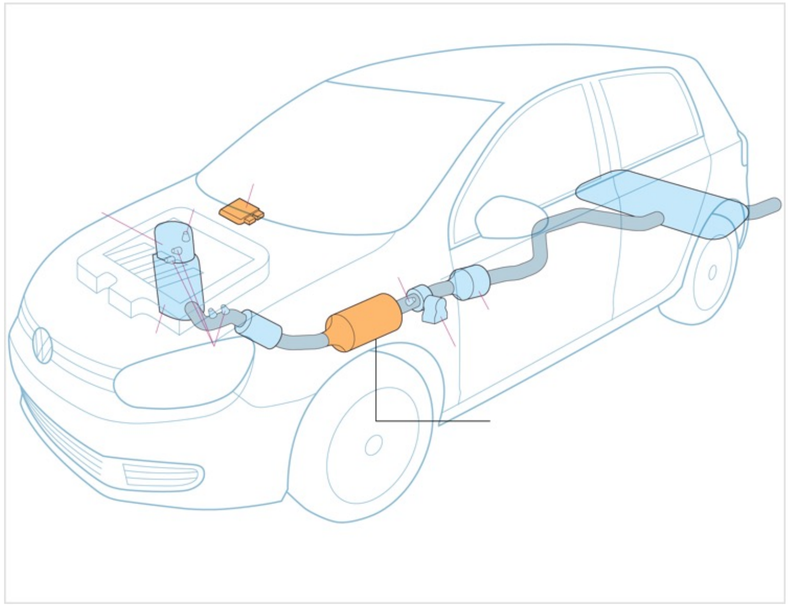
Edit the detailed description

pencil drawing of ethics

Ethics

Volkswagen Scandal

VW was caught cheating on emissions for Diesel engines



https://www.nytimes.com/interactive/2015/business/international/vw-diesel-emissions-scandal-explained.html?mtrref=www.google.com&assetType=REGIWALL

What is Human Flourishing?

According to Harvard's Human flourishing program: Human flourishing is composed of five central domains: **happiness and life satisfaction, mental and physical health, meaning and purpose, character and virtue,** and **close social relationships.**

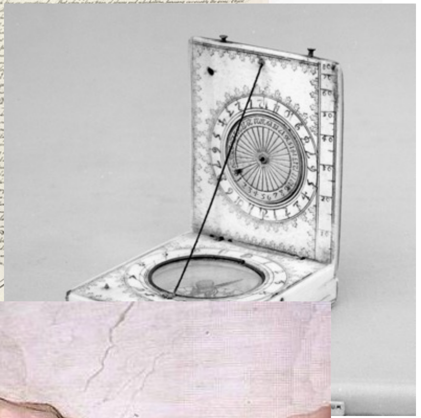
Why Human Flourishing?

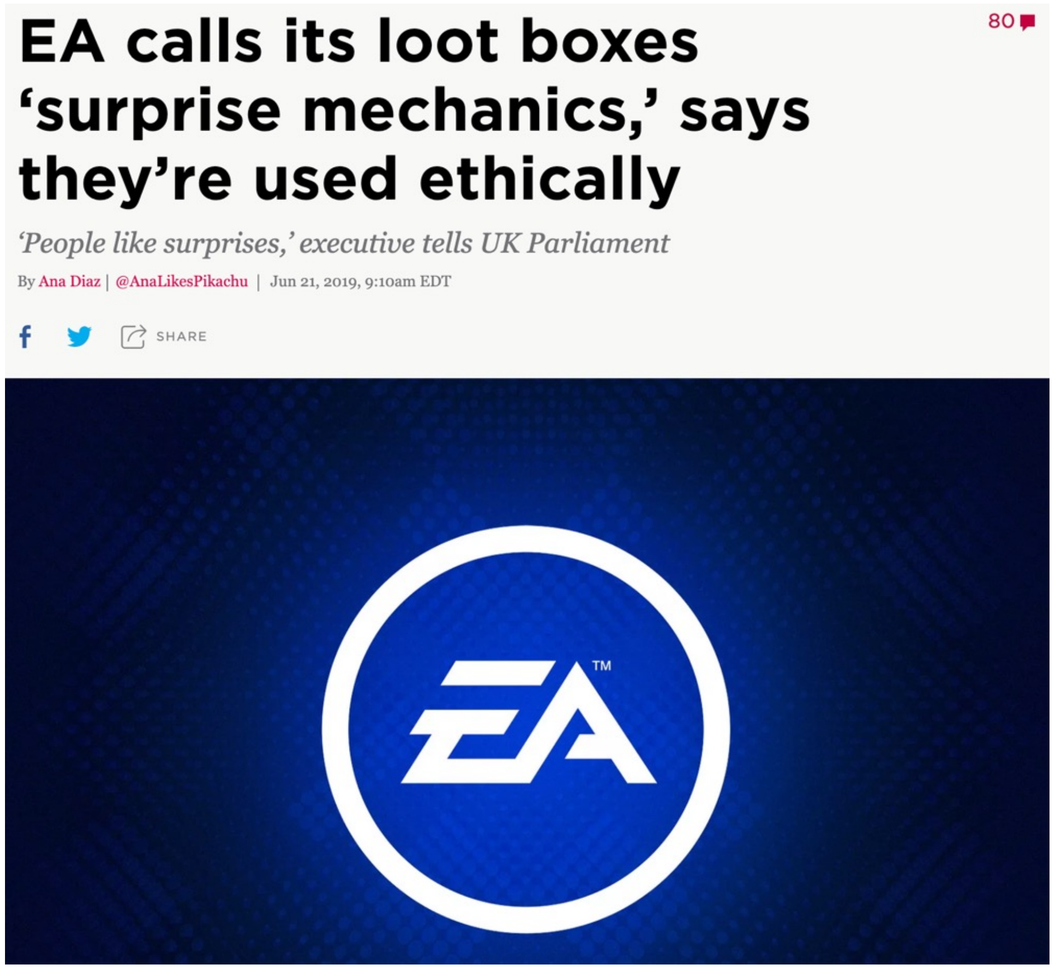
- Universal Declaration of Human Rights: “All human beings are born free and equal in dignity and rights.”
- Declaration of Independence: “We hold these truths to be self-evident…”
- Internal Compass
- Faith

THE UNIVERSAL DECLARATION OF Human Rights

IN CONGRESS, JULY 4, 1776.

The unanimous Declaration of the thirteen united States of America.





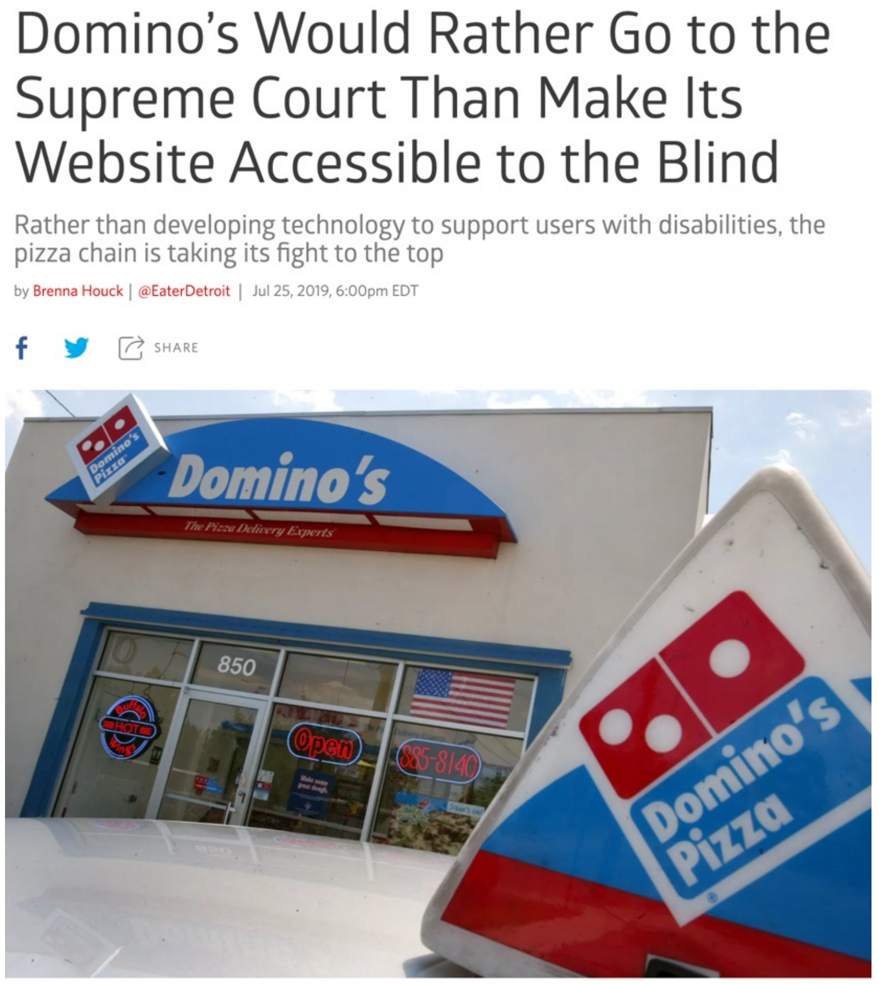
80

EA calls its loot boxes 'surprise mechanics,' says they're used ethically

'People like surprises,' executive tells UK Parliament

By Ana Diaz | @AnaLikesPikachu | Jun 21, 2019, 9:10am EDT

SHARE



Domino's Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind

Rather than developing technology to support users with disabilities, the pizza chain is taking its fight to the top

by Brenna Houck | @EaterDetroit | Jul 25, 2019, 6:00pm EDT

SHARE

Some airlines may be using algorithms to split up families during flights

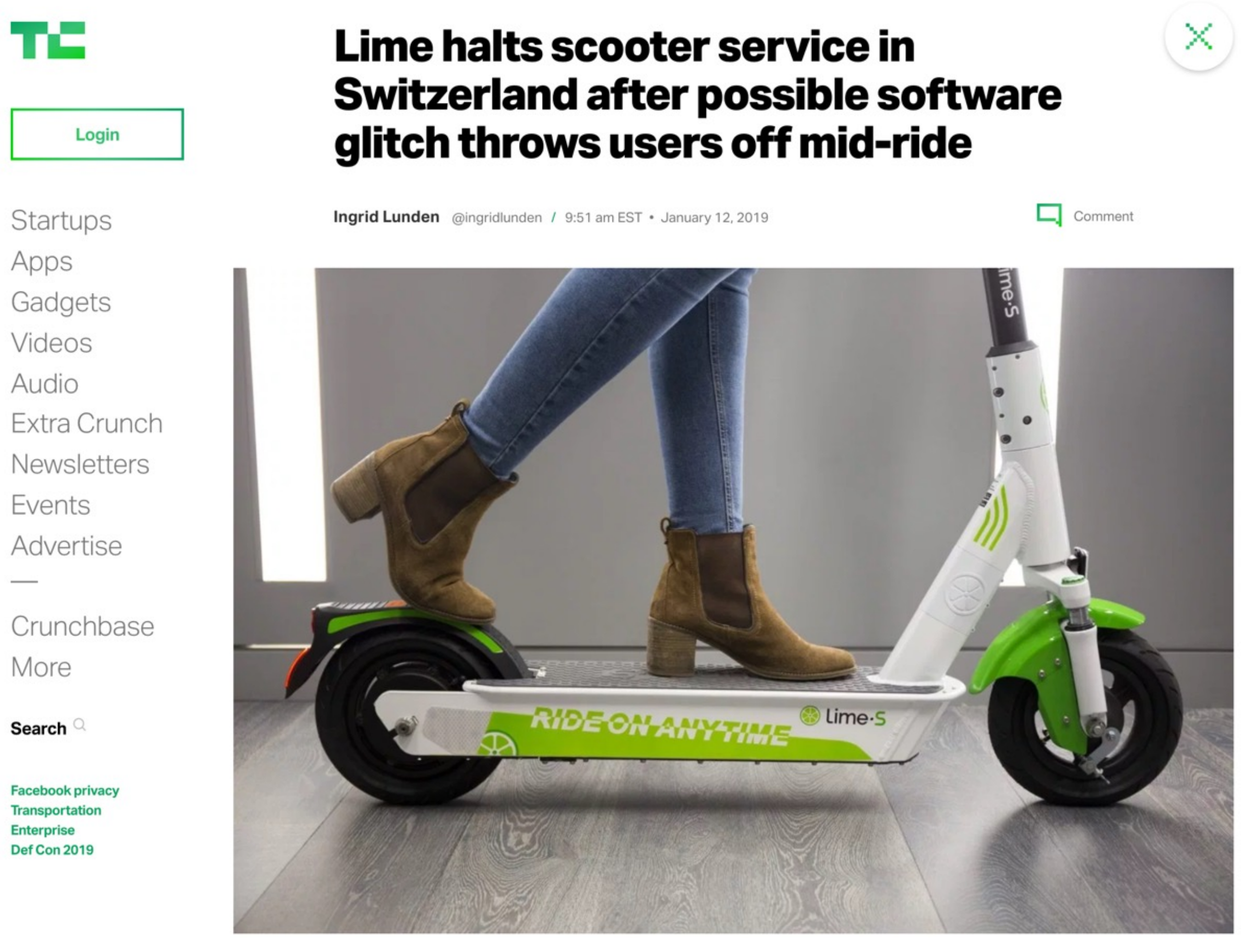
Your random airplane seat assignment might not be random at all.

By Aditi Shrikant | aditi@vox.com | Nov 27, 2018, 6:10pm EST

SHARE

Passengers boarding a Boeing aircraft of the low cost airline carrier Ryanair in Thessaloniki Macedonia Airport, Greece. | Nicolas Economou/NurPhoto/Getty Images

17-313 Foundations of Software Engineering



Login

Startups
Apps
Gadgets
Videos
Audio
Extra Crunch
Newsletters
Events
Advertise
—
Crunchbase
More

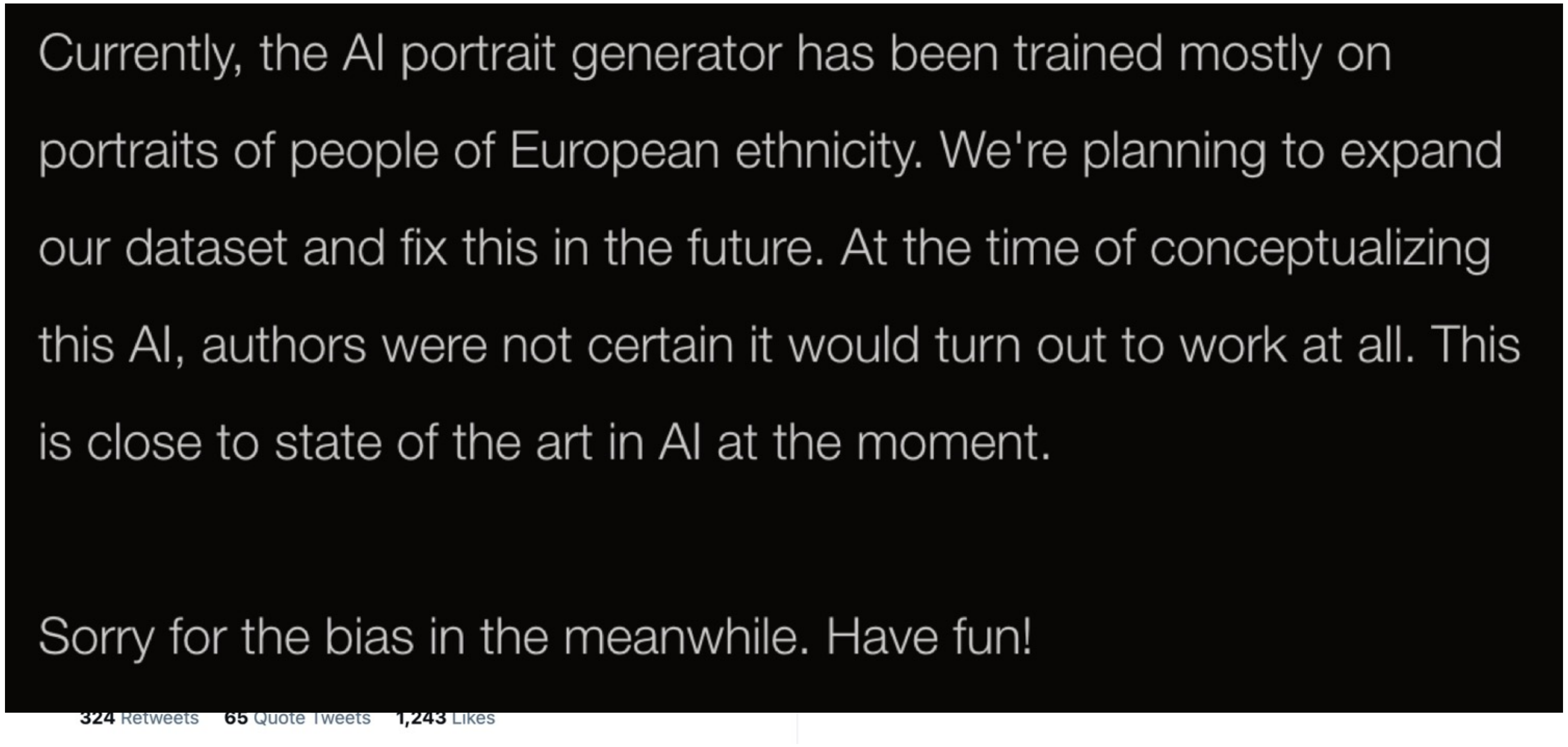
Search

Facebook privacy
Transportation
Enterprise
Def Con 2019

Lime halts scooter service in Switzerland after possible software glitch throws users off mid-ride

Ingrid Lunden @ingridlunden / 9:51 am EST • January 12, 2019

Comment


Currently, the AI portrait generator has been trained mostly on portraits of people of European ethnicity. We're planning to expand our dataset and fix this in the future. At the time of conceptualizing this AI, authors were not certain it would turn out to work at all. This is close to state of the art in AI at the moment.

Sorry for the bias in the meanwhile. Have fun!

324 Retweets 65 Quote Tweets 1,243 Likes

17-313 Foundations of SOFTWARE ENGINEERING

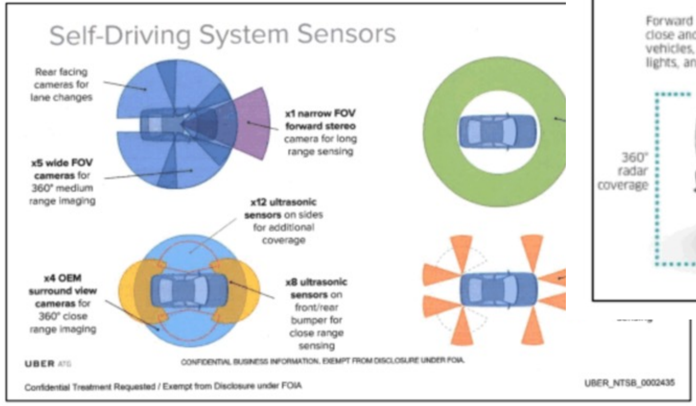
Uber self-driving car involved in fatal crash couldn't detect jaywalkers

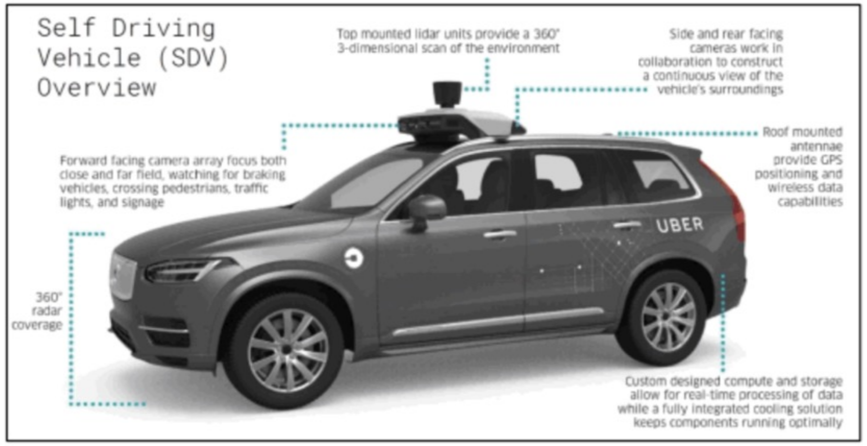
The system had several serious software flaws, the NTSB said.

Steve Dent, @stevetdent
11.06.19 in Transportation

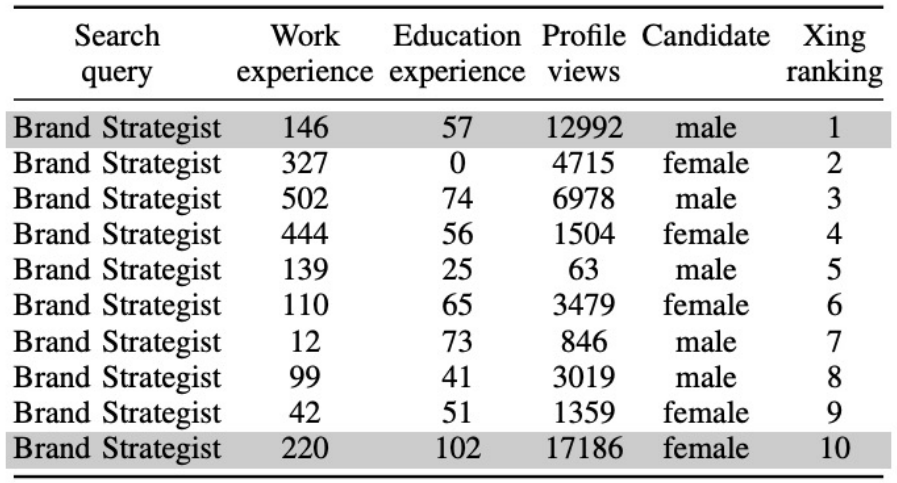
25
Comments

1131
Shares


Self-Driving System Sensors
Rear facing cameras for lane changes
x1 narrow FOV forward stereo camera for long range sensing
x5 wide FOV cameras for 360° medium range imaging
x12 ultrasonic sensors on sides for additional coverage
x4 OEM surround view cameras for 360° close range imaging
x8 ultrasonic sensors on front/rear bumper for close range sensing
UBER ATG
CONFIDENTIAL BUSINESS INFORMATION. EXEMPT FROM DISCLOSURE UNDER FOIA.
Confidential Treatment Requested / Exempt from Disclosure under FOIA
UBER_NTSB_0002435


Self Driving Vehicle (SDV) Overview
Top mounted lidar units provide a 360° 3-dimensional scan of the environment
Side and rear facing cameras work in collaboration to construct a continuous view of the vehicle's surroundings
Roof mounted antennae provide GPS positioning and wireless data capabilities
Forward facing camera array focus both close and far field, watching for braking vehicles, crossing pedestrians, traffic lights, and signage
360° radar coverage
UBER
Custom designed compute and storage allow for real-time processing of data while a fully integrated cooling solution keeps components running optimally

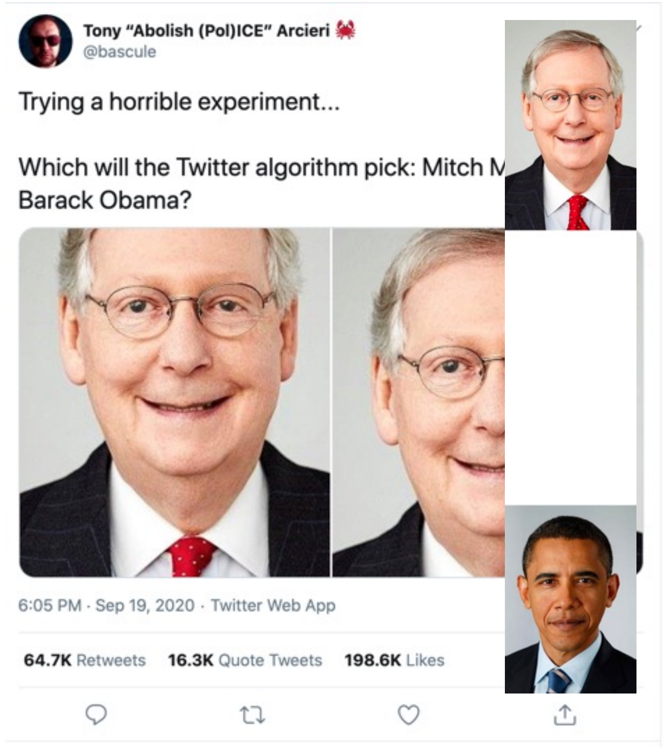
xing.com search for “Brand Strategist”



Search query	Work experience	Education experience	Profile views	Candidate	Xing ranking
Brand Strategist	146	57	12992	male	1
Brand Strategist	327	0	4715	female	2
Brand Strategist	502	74	6978	male	3
Brand Strategist	444	56	1504	female	4
Brand Strategist	139	25	63	male	5
Brand Strategist	110	65	3479	female	6
Brand Strategist	12	73	846	male	7
Brand Strategist	99	41	3019	male	8
Brand Strategist	42	51	1359	female	9
Brand Strategist	220	102	17186	female	10

Lahoti, Preethi, Krishna P. Gummadi, and Gerhard Weikum. “iFair: Learning Individually Fair Data Representations for Algorithmic Decision Making.” 2019 IEEE 35th International Conference on Data Engineering (ICDE) (2019

Twitter cropping photos



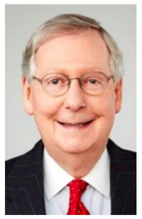
Tony "Abolish (Pol)ICE" Arcieri
@bascule

Trying a horrible experiment...

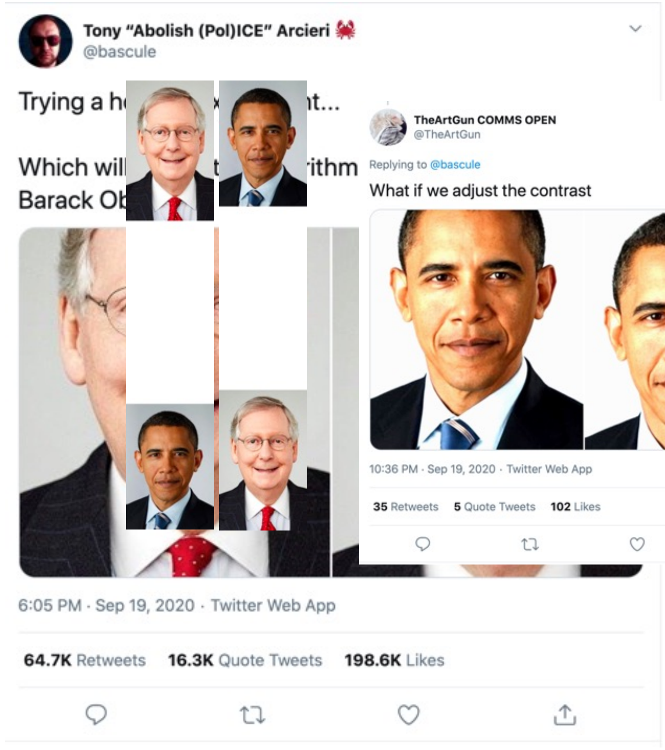
Which will the Twitter algorithm pick: Mitch M
Barack Obama?

6:05 PM · Sep 19, 2020 · Twitter Web App

64.7K Retweets **16.3K** Quote Tweets **198.6K** Likes



Twitter cropping photos

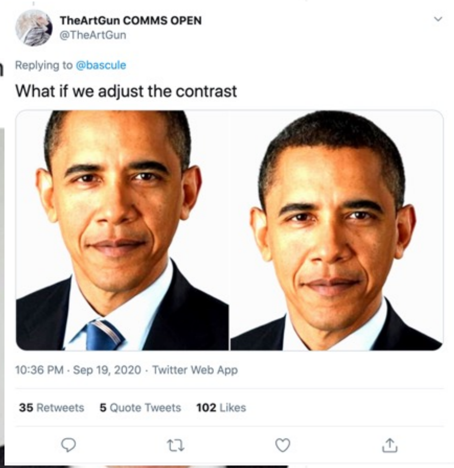

Tony "Abolish (Pol)ICE" Arcieri
@bascule

Trying a h x t...

Which wil t rithm
Barack Ob

6:05 PM · Sep 19, 2020 · Twitter Web App

64.7K Retweets 16.3K Quote Tweets 198.6K Likes


TheArtGun COMMS OPEN
@TheArtGun

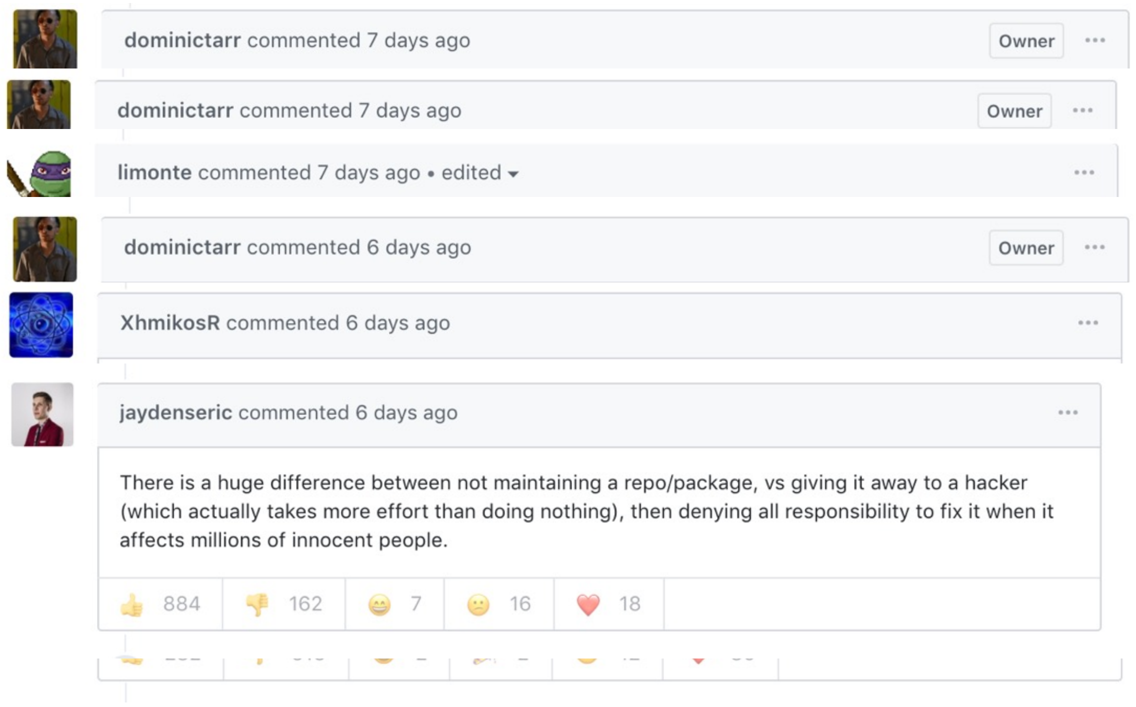
Replying to @bascule

What if we adjust the contrast

10:36 PM · Sep 19, 2020 · Twitter Web App

35 Retweets 5 Quote Tweets 102 Likes

Open Source Maintainers


dominictarr commented 7 days ago Owner

dominictarr commented 7 days ago Owner

limonte commented 7 days ago • edited

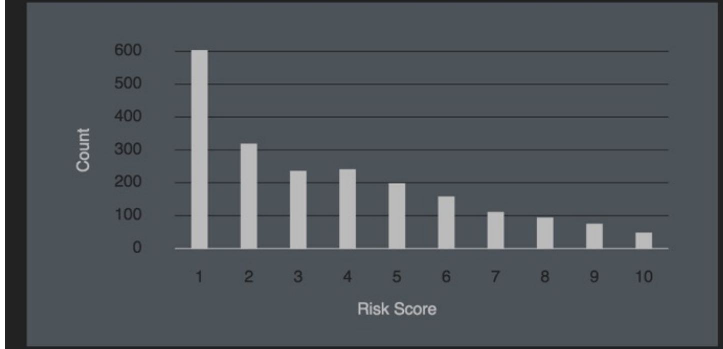
dominictarr commented 6 days ago Owner

XhmikosR commented 6 days ago

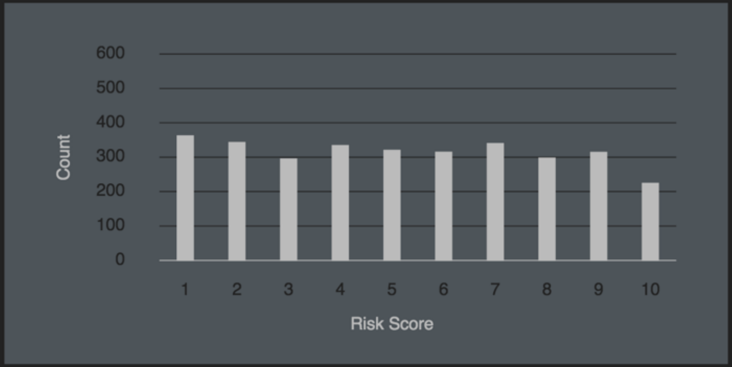
jaydenseric commented 6 days ago

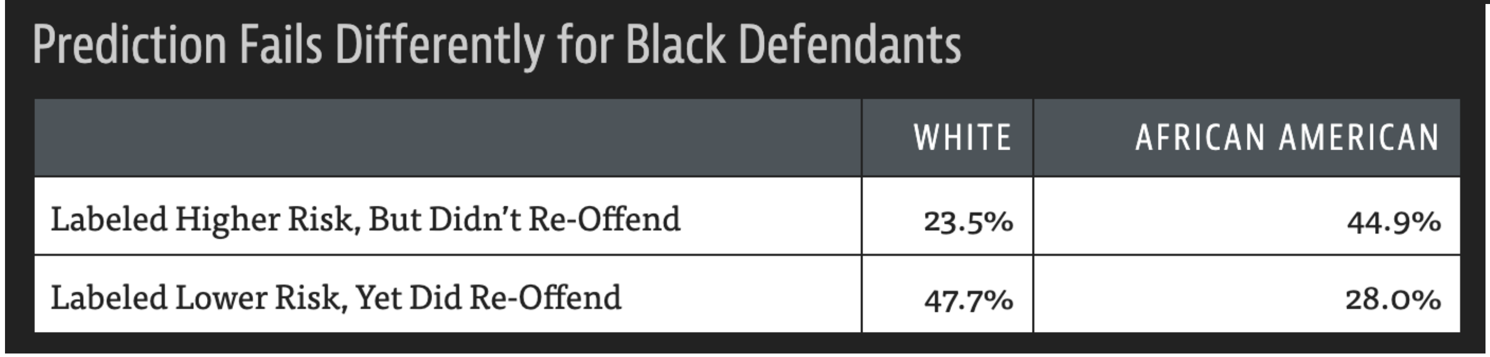
There is a huge difference between not maintaining a repo/package, vs giving it away to a hacker (which actually takes more effort than doing nothing), then denying all responsibility to fix it when it affects millions of innocent people.

👍 884 👎 162 😄 7 😕 16 ❤️ 18



These charts show that scores for white defendants were skewed toward lower-risk categories. Scores for black defendants were not. (Source: ProPublica analysis of data from Broward County, Fla.)





Prediction Fails Differently for Black Defendants

	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Algorithmic Bias

Algorithms affect:

Where we go to school

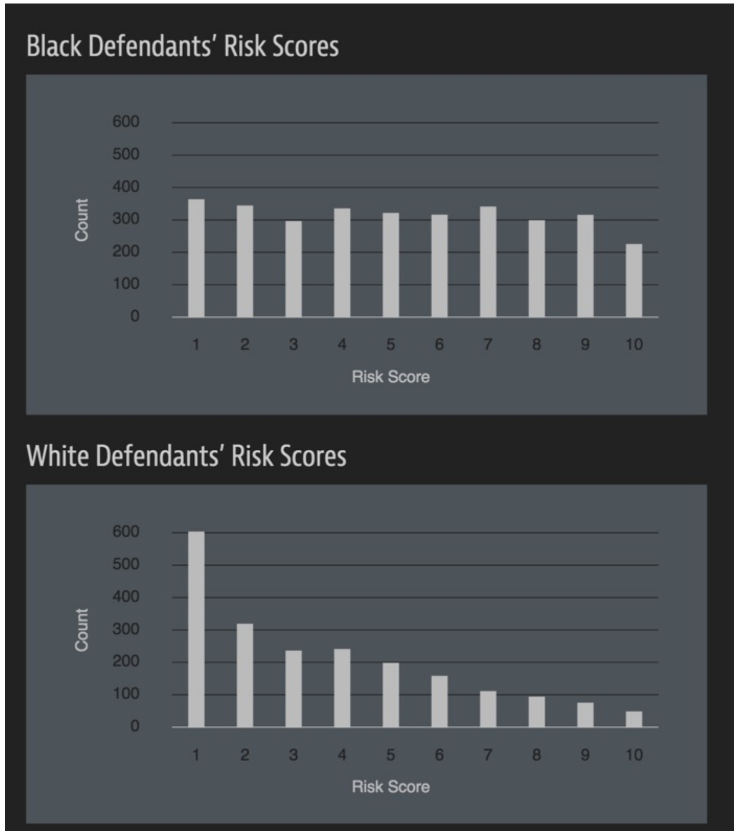
Access to money

Access to health care

Receiving parole

Possibility of Bail

Risk Scores



These charts show that scores for white defendants were skewed toward lower-risk categories. Scores for black defendants were not. *(Source: ProPublica analysis of data from Broward County, Fla.)*

Therac-25

Bug (race-condition) in software lead to at least 6 deaths

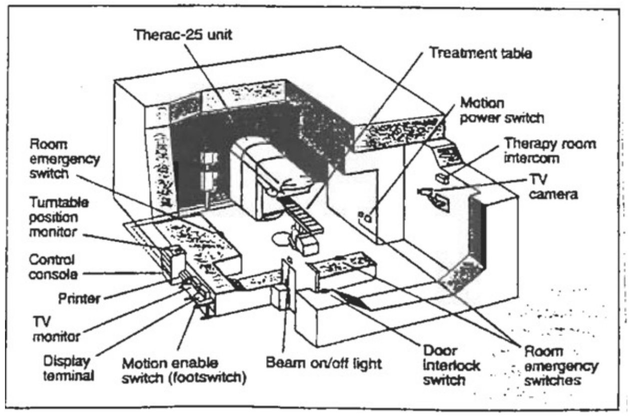
Traced to:

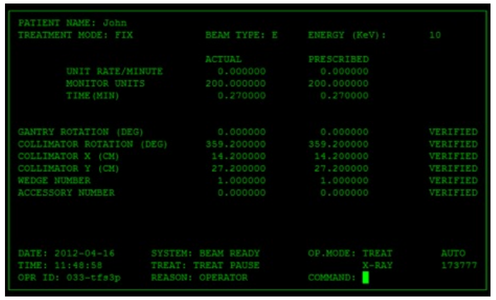
Lack of reporting bugs

Lack of proper due diligence

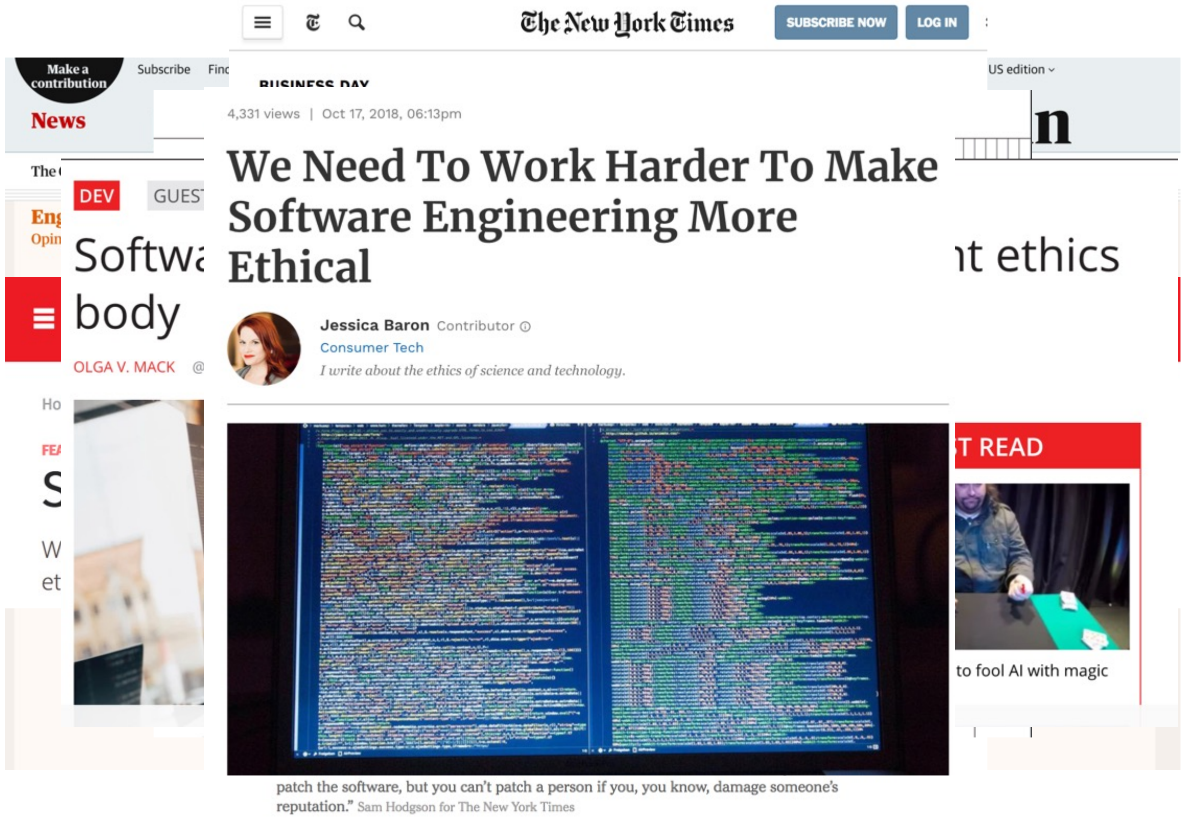
Engineers were overconfident, removed hardware locks

Race condition of 8 seconds could lead to problems





```
PATIENT NAME: John
TREATMENT MODE: FIX                BEAM TYPE: E     ENERGY (KeV):       10

                                   ACTUAL           PRESCRIBED
        UNIT RATE/MINUTE             0.000000         0.000000
        MONITOR UNITS              200.000000       200.000000
        TIME (MIN)                   0.270000         0.270000


GANTRY ROTATION (DEG)                0.000000         0.000000          VERIFIED
COLLIMATOR ROTATION (DEG)          359.200000       359.200000          VERIFIED
COLLIMATOR X (CM)                   14.200000        14.200000          VERIFIED
COLLIMATOR Y (CM)                   27.200000        27.200000          VERIFIED
WEDGE NUMBER                         1.000000         1.000000          VERIFIED
ACCESSORY NUMBER                     0.000000         0.000000          VERIFIED



DATE: 2012-04-16      SYSTEM: BEAM READY        OP.MODE: TREAT       AUTO
TIME: 11:48:58        TREAT: TREAT PAUSE                 X-RAY       173777
OPR ID: 033-tfs3p     REASON: OPERATOR          COMMAND: █
```



Make a contribution Subscribe Find

News

The

Eng
Opin

DEV GUEST

Softwa
body

OLGA V. MACK @

Ho

FEA

S

W
et

The New York Times SUBSCRIBE NOW LOG IN

BUSINESS DAY

4,331 views | Oct 17, 2018, 06:13pm

We Need To Work Harder To Make Software Engineering More Ethical

Jessica Baron Contributor
Consumer Tech
I write about the ethics of science and technology.

patch the software, but you can't patch a person if you, you know, damage someone's reputation." Sam Hodgson for The New York Times

US edition

n

nt ethics

T READ

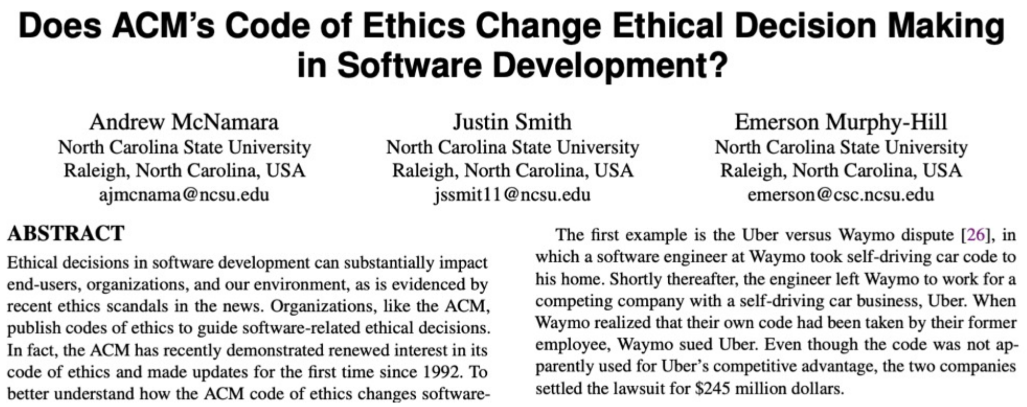
to fool AI with magic

ACM Code of Ethics

As an ACM member I will ....
Contribute to society and human well-being.
Avoid harm to others.
Be honest and trustworthy.
Be fair and take action not to discriminate.
Honor property rights including copyrights and patent.
Give proper credit for intellectual property.
Respect the privacy of others.
Honor confidentiality.

Code of Ethics

Research shows that the code of ethics does not appear to affect the decisions made by software developers.



Does ACM's Code of Ethics Change Ethical Decision Making in Software Development?

Andrew McNamara
North Carolina State University
Raleigh, North Carolina, USA
ajmcnama@ncsu.edu

Justin Smith
North Carolina State University
Raleigh, North Carolina, USA
jssmit11@ncsu.edu

Emerson Murphy-Hill
North Carolina State University
Raleigh, North Carolina, USA
emerson@csc.ncsu.edu

ABSTRACT

Ethical decisions in software development can substantially impact end-users, organizations, and our environment, as is evidenced by recent ethics scandals in the news. Organizations, like the ACM, publish codes of ethics to guide software-related ethical decisions. In fact, the ACM has recently demonstrated renewed interest in its code of ethics and made updates for the first time since 1992. To better understand how the ACM code of ethics changes software-

The first example is the Uber versus Waymo dispute [26], in which a software engineer at Waymo took self-driving car code to his home. Shortly thereafter, the engineer left Waymo to work for a competing company with a self-driving car business, Uber. When Waymo realized that their own code had been taken by their former employee, Waymo sued Uber. Even though the code was not apparently used for Uber's competitive advantage, the two companies settled the lawsuit for $245 million dollars.

Challenge:

How do we apply ethics to a field (Software Engineering) that is changes so often?

Remember the Dominos case? The ADA law was written before the first website (1990)

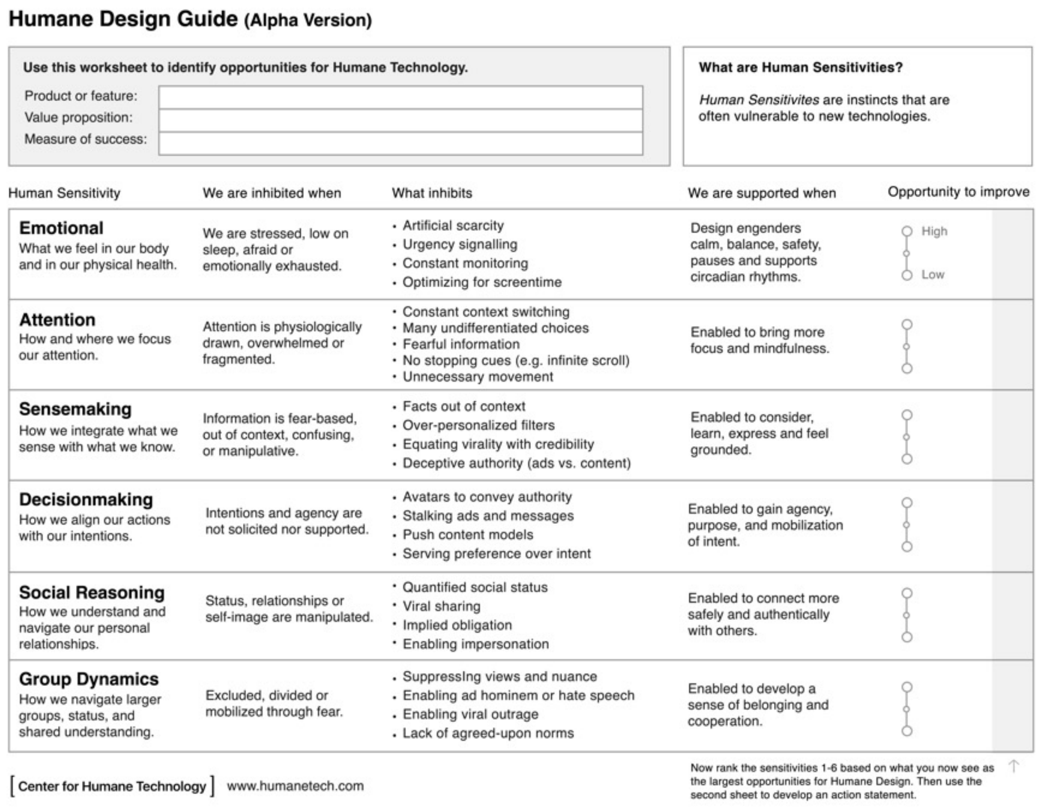
To handle this uncertainty about the future, let’s focus on three questions we can ask to remind ourselves to focus on promoting human flourishing.

Three questions to promote human flourishing

1. Does my software respect the **humanity** of the **users**?
2. Does my software **amplify positive** behavior, or **negative** behavior for users and society at large?
3. Will my software's **quality** impact the **humanity** of others?

1. Does my software respect the **humanity** of the **users**?

Humane Design Guide
http://humanetech.com



Humane Design Guide (Alpha Version)

Use this worksheet to identify opportunities for Humane Technology.

Product or feature:	
Value proposition:	
Measure of success:	

What are Human Sensitivities?

Human Sensitivites are instincts that are often vulnerable to new technologies.

Human Sensitivity	We are inhibited when	What inhibits	We are supported when	Opportunity to improve
Emotional What we feel in our body and in our physical health.	We are stressed, low on sleep, afraid or emotionally exhausted.	• Artificial scarcity • Urgency signalling • Constant monitoring • Optimizing for screentime	Design engenders calm, balance, safety, pauses and supports circadian rhythms.	High Low
Attention How and where we focus our attention.	Attention is physiologically drawn, overwhelmed or fragmented.	• Constant context switching • Many undifferentiated choices • Fearful information • No stopping cues (e.g. infinite scroll) • Unnecessary movement	Enabled to bring more focus and mindfulness.	
Sensemaking How we integrate what we sense with what we know.	Information is fear-based, out of context, confusing, or manipulative.	• Facts out of context • Over-personalized filters • Equating virality with credibility • Deceptive authority (ads vs. content)	Enabled to consider, learn, express and feel grounded.	
Decisionmaking How we align our actions with our intentions.	Intentions and agency are not solicited nor supported.	• Avatars to convey authority • Stalking ads and messages • Push content models • Serving preference over intent	Enabled to gain agency, purpose, and mobilization of intent.	
Social Reasoning How we understand and navigate our personal relationships.	Status, relationships or self-image are manipulated.	• Quantified social status • Viral sharing • Implied obligation • Enabling impersonation	Enabled to connect more safely and authentically with others.	
Group Dynamics How we navigate larger groups, status, and shared understanding.	Excluded, divided or mobilized through fear.	• SuppressIng views and nuance • Enabling ad hominem or hate speech • Enabling viral outrage • Lack of agreed-upon norms	Enabled to develop a sense of belonging and cooperation.	

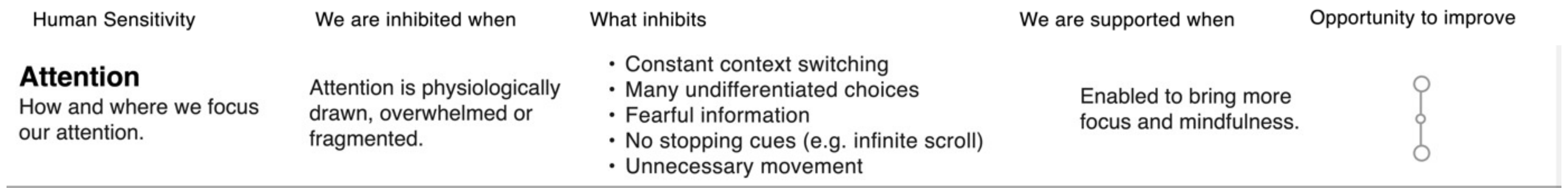
[Center for Humane Technology] www.humanetech.com

Now rank the sensitivities 1-6 based on what you now see as the largest opportunities for Humane Design. Then use the second sheet to develop an action statement.

Humane Design Guide
http://humanetech.com

Provides a template for considering a piece of software, and asking questions to help us arrive at a "humane design"

Consider 6 human sensitivities: Emotional, Attention, Sense making, Decision making, Social Reasoning, and Group Dynamics



Human Sensitivity	We are inhibited when	What inhibits	We are supported when	Opportunity to improve
Attention How and where we focus our attention.	Attention is physiologically drawn, overwhelmed or fragmented.	• Constant context switching • Many undifferentiated choices • Fearful information • No stopping cues (e.g. infinite scroll) • Unnecessary movement	Enabled to bring more focus and mindfulness.	

Identify Opportunities to improve

Humane Design Guide
http://humanetech.com

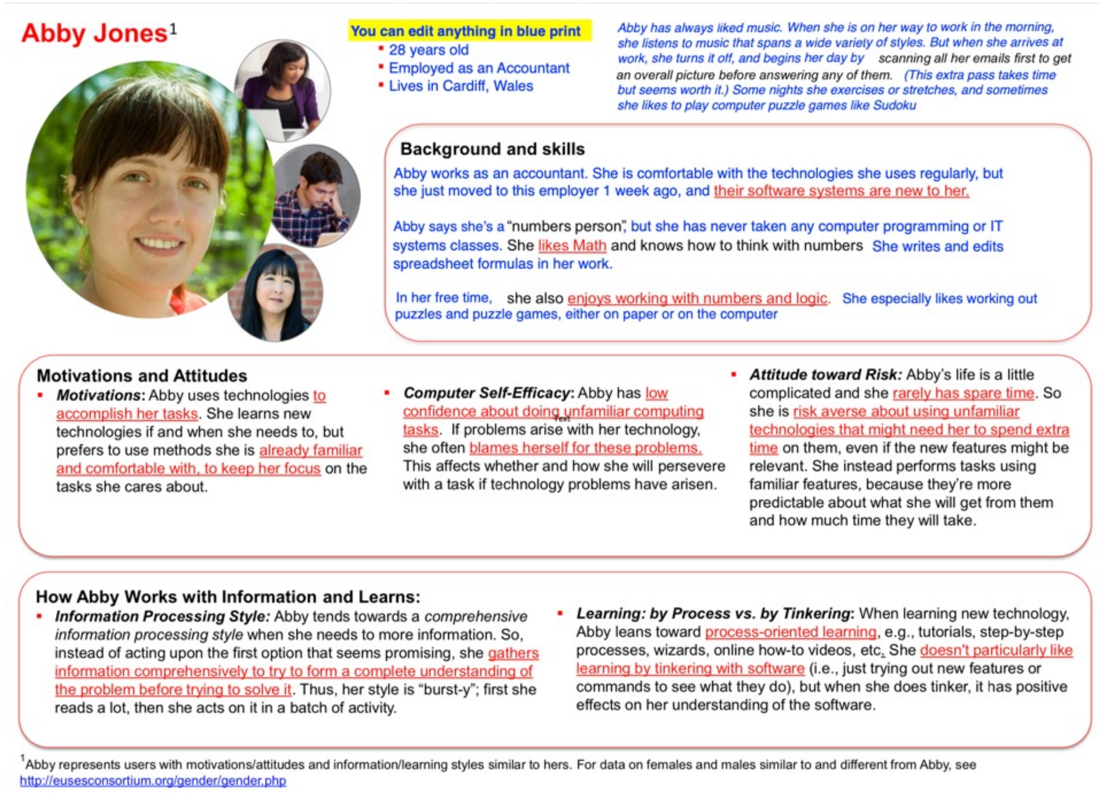
After analysis step, develop plan of action:

1. In what ways does your product/feature currently engage Human Sensitivities?

2. How might your product/feature support or elevate human sensitivities?

3. Action Statement

GenderMag
https://gendermag.org



Abby Jones[1]

You can edit anything in blue print
- 28 years old
- Employed as an Accountant
- Lives in Cardiff, Wales

Abby has always liked music. When she is on her way to work in the morning, she listens to music that spans a wide variety of styles. But when she arrives at work, she turns it off, and begins her day by scanning all her emails first to get an overall picture before answering any of them. (This extra pass takes time but seems worth it.) Some nights she exercises or stretches, and sometimes she likes to play computer puzzle games like Sudoku

Background and skills

Abby works as an accountant. She is comfortable with the technologies she uses regularly, but she just moved to this employer 1 week ago, and their software systems are new to her.

Abby says she's a "numbers person", but she has never taken any computer programming or IT systems classes. She likes Math and knows how to think with numbers. She writes and edits spreadsheet formulas in her work.

In her free time, she also enjoys working with numbers and logic. She especially likes working out puzzles and puzzle games, either on paper or on the computer

Motivations and Attitudes

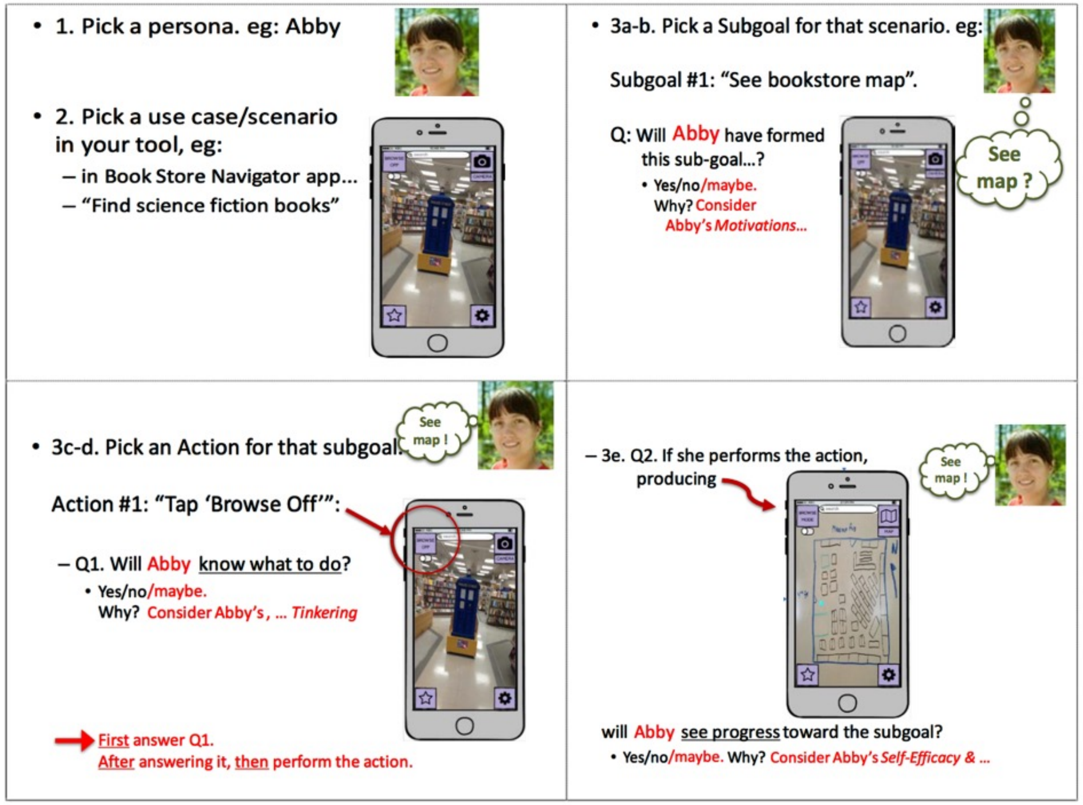
- ***Motivations***: Abby uses technologies to accomplish her tasks. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.
- ***Computer Self-Efficacy***: Abby has low confidence about doing unfamiliar computing tasks. If problems arise with her technology, she often blames herself for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.
- ***Attitude toward Risk:*** Abby's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they're more predictable about what she will get from them and how much time they will take.

How Abby Works with Information and Learns:

- ***Information Processing Style:*** Abby tends towards a *comprehensive information processing style* when she needs to more information. So, instead of acting upon the first option that seems promising, she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, her style is "burst-y"; first she reads a lot, then she acts on it in a batch of activity.
- ***Learning: by Process vs. by Tinkering***: When learning new technology, Abby leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

[1]Abby represents users with motivations/attitudes and information/learning styles similar to hers. For data on females and males similar to and different from Abby, see http://eusesconsortium.org/gender/gender.php

GenderMag
https://gendermag.org



- 1. Pick a persona. eg: Abby
- 2. Pick a use case/scenario in your tool, eg:
  - in Book Store Navigator app...
  - "Find science fiction books"

- 3a-b. Pick a Subgoal for that scenario. eg:

  Subgoal #1: "See bookstore map".

  Q: Will Abby have formed this sub-goal...?
  - Yes/no/maybe.
    Why? Consider Abby's Motivations...

- 3c-d. Pick an Action for that subgoal:

  Action #1: "Tap 'Browse Off'":

  – Q1. Will Abby know what to do?
  - Yes/no/maybe.
    Why? Consider Abby's, ... Tinkering

  First answer Q1.
  After answering it, then perform the action.

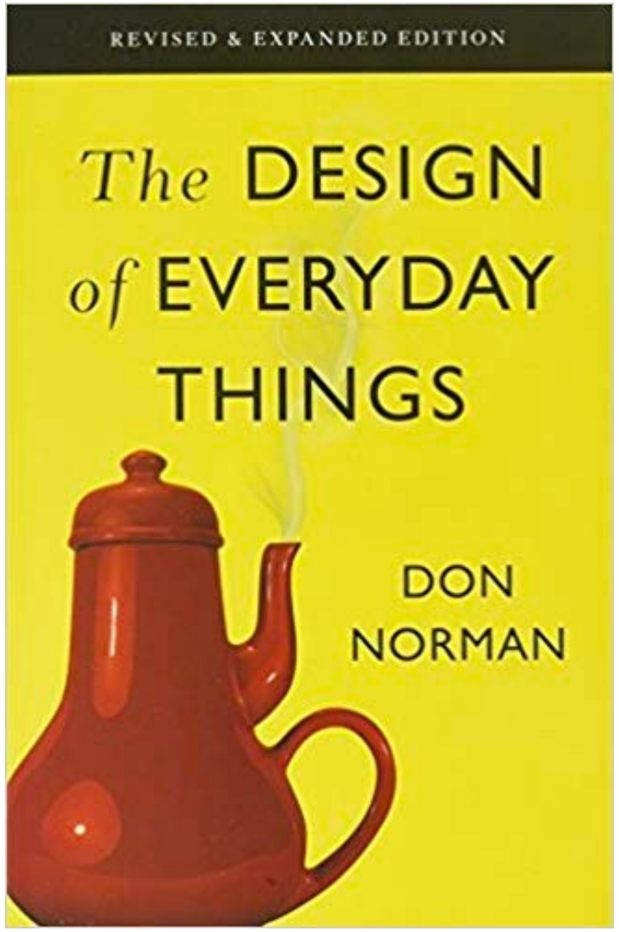
– 3e. Q2. If she performs the action, producing

will Abby see progress toward the subgoal?
- Yes/no/maybe. Why? Consider Abby's *Self-Efficacy* & ...

User Centered Design

User-centered design tries to optimize the product around how **users can, want, or need to use the product**, rather than forcing the users to change their behavior to **accommodate the product.**

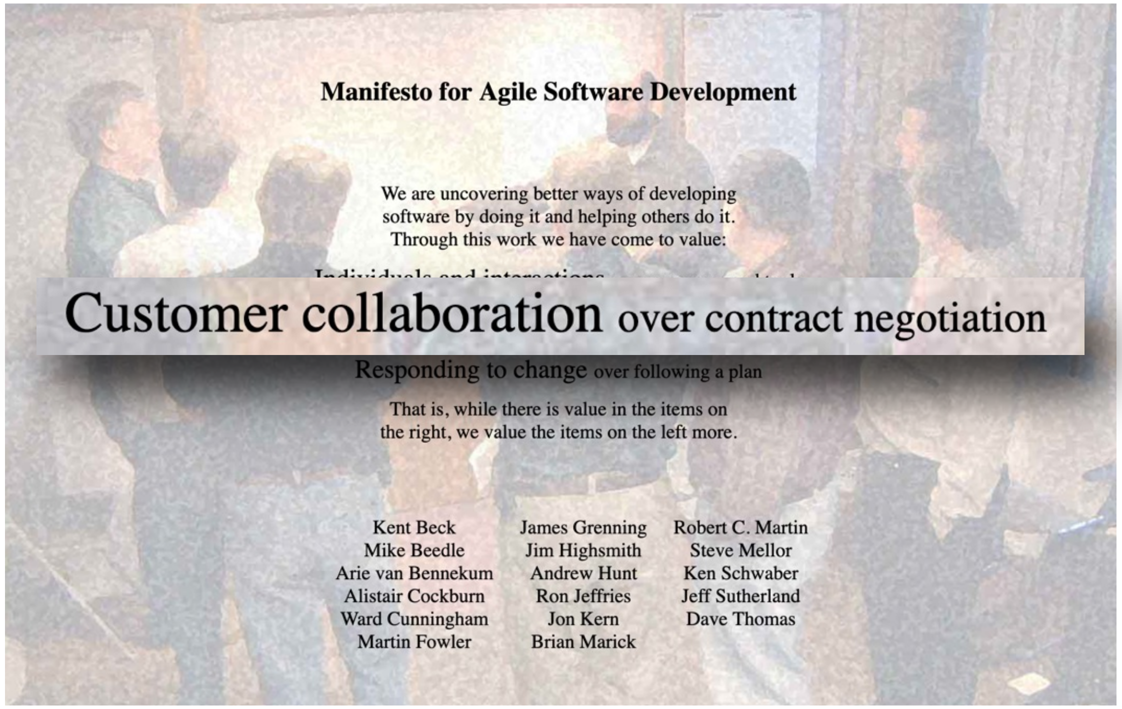
-Wikipedia



Agile

User Centered Design

Agile customer representative


Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

2. Does my software amplify positive or negative behavior for users and society at large?

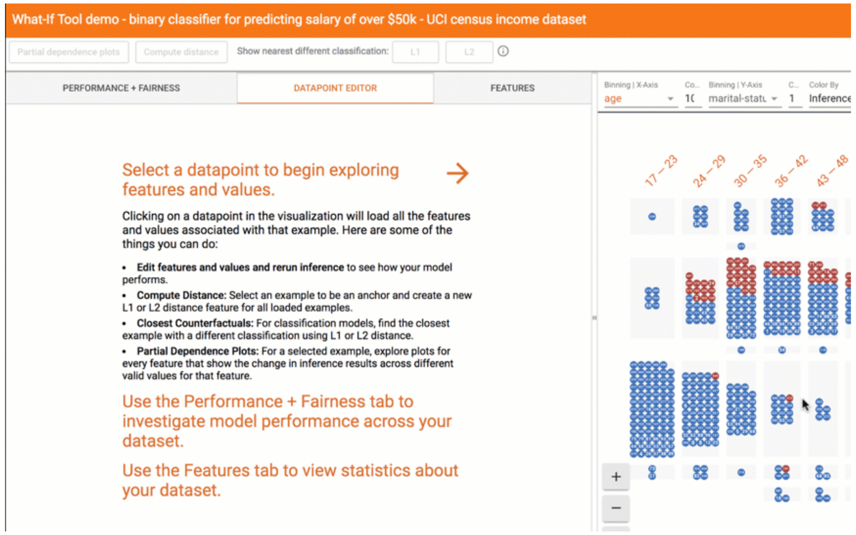
What if…

https://pair-code.github.io/what-if-tool/

What If…
you could inspect a machine learning model, with minimal coding required?

What if…

https://pair-code.github.io/what-if-tool/



Select a datapoint to begin exploring features and values.

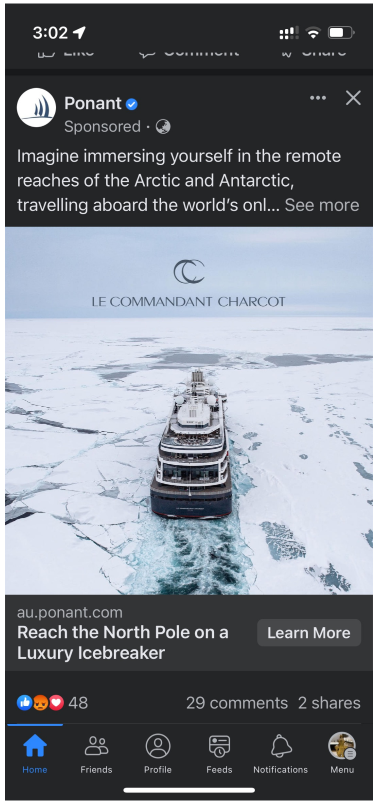
Clicking on a datapoint in the visualization will load all the features and values associated with that example. Here are some of the things you can do:

- **Edit features and values and rerun inference** to see how your model performs.
- **Compute Distance:** Select an example to be an anchor and create a new L1 or L2 distance feature for all loaded examples.
- **Closest Counterfactuals:** For classification models, find the closest example with a different classification using L1 or L2 distance.
- **Partial Dependence Plots:** For a selected example, explore plots for every feature that show the change in inference results across different valid values for that feature.

Use the Performance + Fairness tab to investigate model performance across your dataset.

Use the Features tab to view statistics about your dataset.

Explain "why" to customers



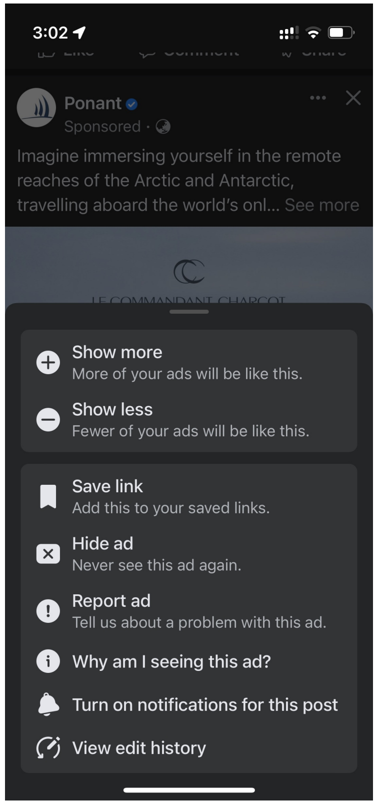
3:02

Ponant
Sponsored ·

Imagine immersing yourself in the remote reaches of the Arctic and Antarctic, travelling aboard the world's onl... See more

au.ponant.com
Reach the North Pole on a Luxury Icebreaker
Learn More

48 29 comments 2 shares



3:02

Show more
More of your ads will be like this.

Show less
Fewer of your ads will be like this.

Save link
Add this to your saved links.

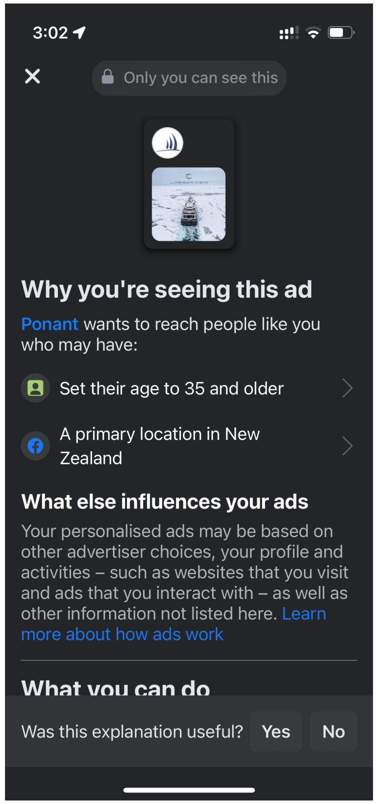
Hide ad
Never see this ad again.

Report ad
Tell us about a problem with this ad.

Why am I seeing this ad?

Turn on notifications for this post

View edit history



3:02

Only you can see this

Why you're seeing this ad

Ponant wants to reach people like you who may have:

Set their age to 35 and older

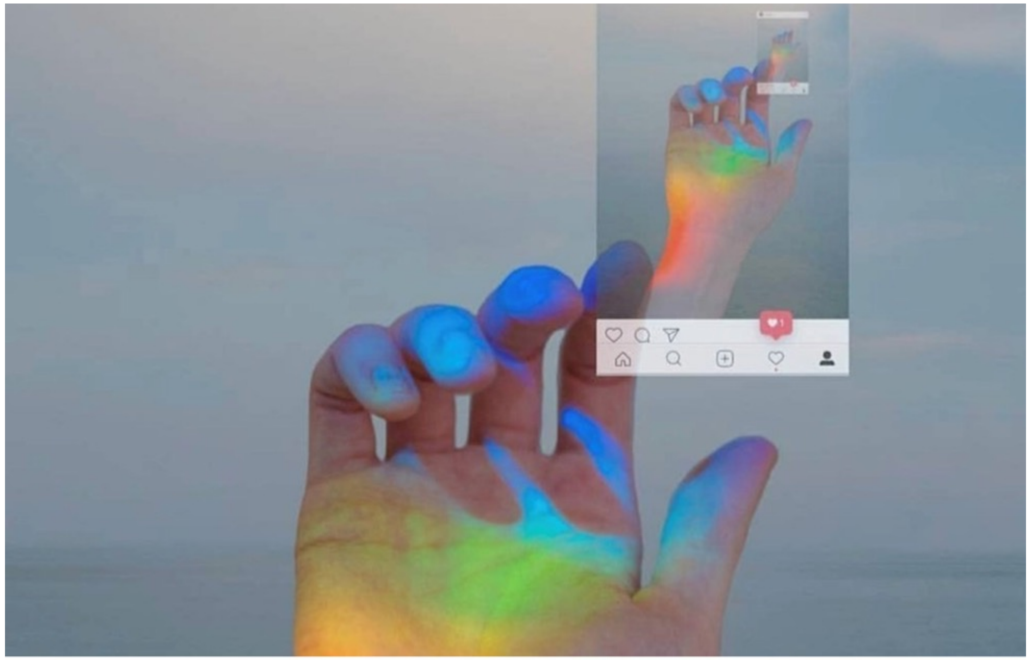
A primary location in New Zealand

What else influences your ads

Your personalised ads may be based on other advertiser choices, your profile and activities – such as websites that you visit and ads that you interact with – as well as other information not listed here. Learn more about how ads work

What you can do

Was this explanation useful? Yes No


@dovneon

What Instagram removing likes may mean for influencers and our self-esteem

SCIENCE & TECH - FEATURE

The decision could have a positive impact on the way people use the platform, but harm those trying to use it professionally

Anil Dash on how to prevent abuse

http://anildash.com/2011/07/20/if_your_websites_full_of_assholes_its_your_fault-2/

You should have real humans dedicated to monitoring and responding to your community.

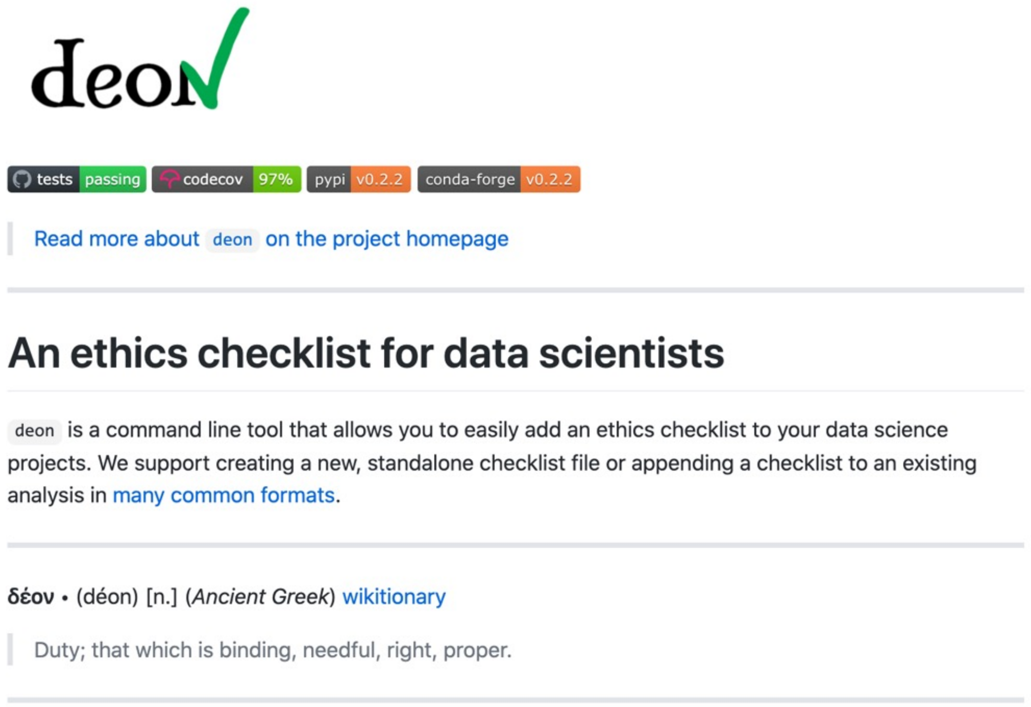
You should have community policies about what is and isn’t acceptable behavior.

Your site should have accountable identities.

You should have the technology to easily identify and stop bad behaviors.

You should make a budget that supports having a good community, or you should find another line of work.

Deon
https://github.com/drivendataorg/deon



deon

tests passing | codecov 97% | pypi v0.2.2 | conda-forge v0.2.2

> Read more about `deon` on the project homepage

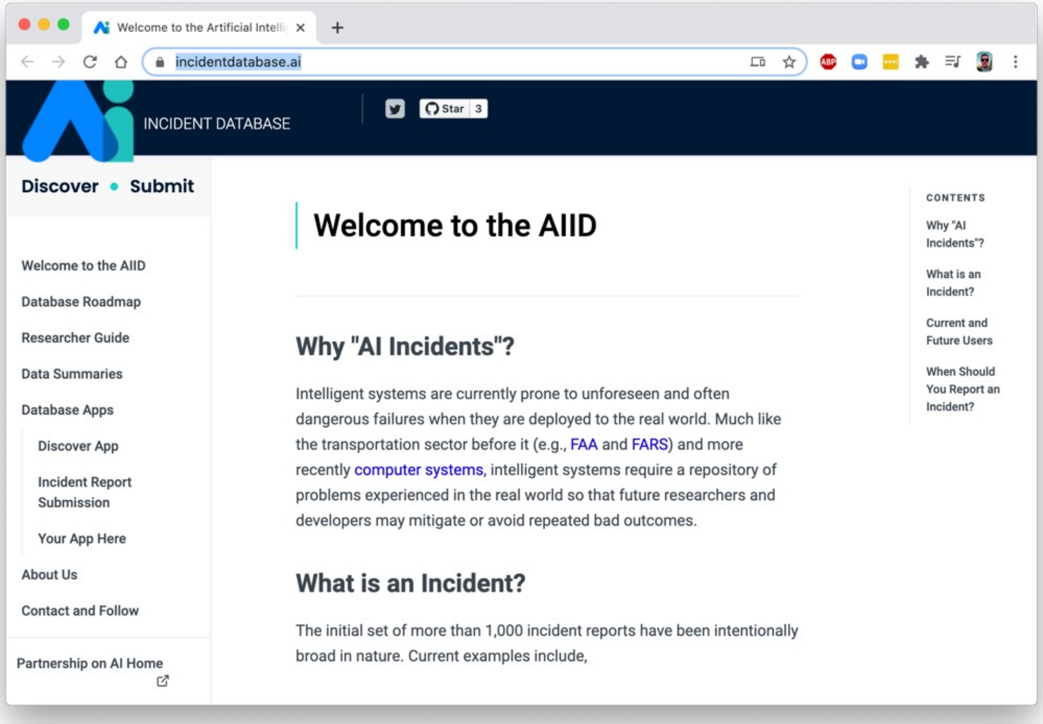
An ethics checklist for data scientists

`deon` is a command line tool that allows you to easily add an ethics checklist to your data science projects. We support creating a new, standalone checklist file or appending a checklist to an existing analysis in many common formats.

δέον • (déon) [n.] (*Ancient Greek*) wikitionary

> Duty; that which is binding, needful, right, proper.

AI Incident Database



Welcome to the AIID

Why "AI Incidents"?

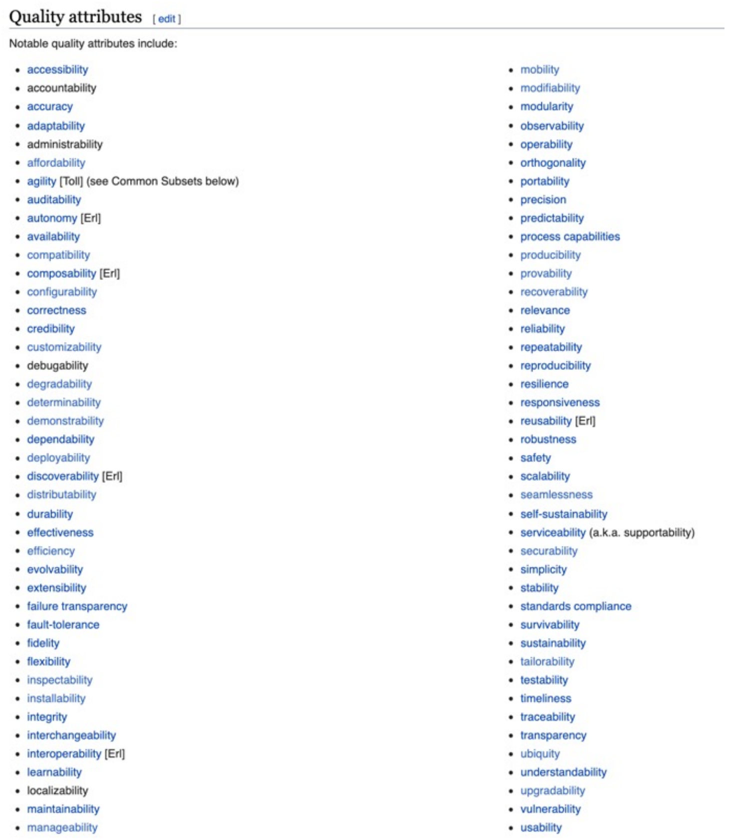
Intelligent systems are currently prone to unforeseen and often dangerous failures when they are deployed to the real world. Much like the transportation sector before it (e.g., FAA and FARS) and more recently computer systems, intelligent systems require a repository of problems experienced in the real world so that future researchers and developers may mitigate or avoid repeated bad outcomes.

What is an Incident?

The initial set of more than 1,000 incident reports have been intentionally broad in nature. Current examples include,

3. Will my software's **quality** impact the **humanity** of others?

Quality has long been considered



Quality attributes [edit]

Notable quality attributes include:

- accessibility
- accountability
- accuracy
- adaptability
- administrability
- affordability
- agility [Toll] (see Common Subsets below)
- auditability
- autonomy [Erl]
- availability
- compatibility
- composability [Erl]
- configurability
- correctness
- credibility
- customizability
- debugability
- degradability
- determinability
- demonstrability
- dependability
- deployability
- discoverability [Erl]
- distributability
- durability
- effectiveness
- efficiency
- evolvability
- extensibility
- failure transparency
- fault-tolerance
- fidelity
- flexibility
- inspectability
- installability
- integrity
- interchangeability
- interoperability [Erl]
- learnability
- localizability
- maintainability
- manageability
- mobility
- modifiability
- modularity
- observability
- operability
- orthogonality
- portability
- precision
- predictability
- process capabilities
- producibility
- provability
- recoverability
- relevance
- reliability
- repeatability
- reproducibility
- resilience
- responsiveness
- reusability [Erl]
- robustness
- safety
- scalability
- seamlessness
- self-sustainability
- serviceability (a.k.a. supportability)
- securability
- simplicity
- stability
- standards compliance
- survivability
- sustainability
- tailorability
- testability
- timeliness
- traceability
- transparency
- ubiquity
- understandability
- upgradability
- vulnerability
- usability

Engineering ethics.

Ethics applies and is formalized in many professional fields: medical, legal, business, and engineering.

The first codes of engineering ethics were formally adopted by American engineering societies in 1912-1914. In 1946 the National Society of Professional Engineers (NSPE) adopted their first formal Canons of Ethics.

https://www.engineersaustralia.org.au/publications/code-ethics

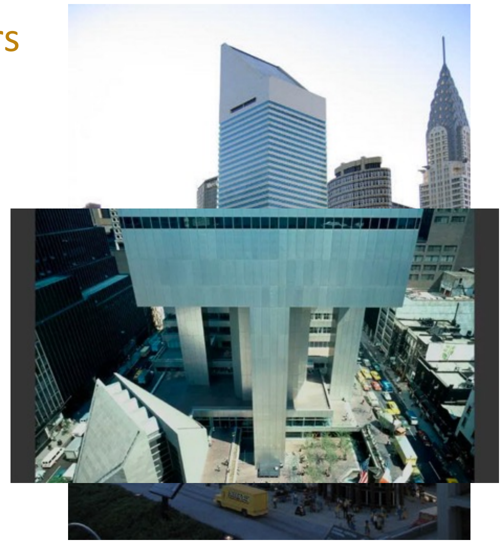
“hold paramount safety, health and welfare of the public”

Citigroup Center, Designed by Structural engineer William LeMessurier

Followed calculations required by building codes

Civil Engineering student Diane Hartley realized there was a problem

Tests showed that winds needed to bring it down would happen every 55 years



Professional Ethics

Professional ethics encompass the personal, and corporate standards of behavior expected by professionals.

First three "professions"

-Divinity

-Law

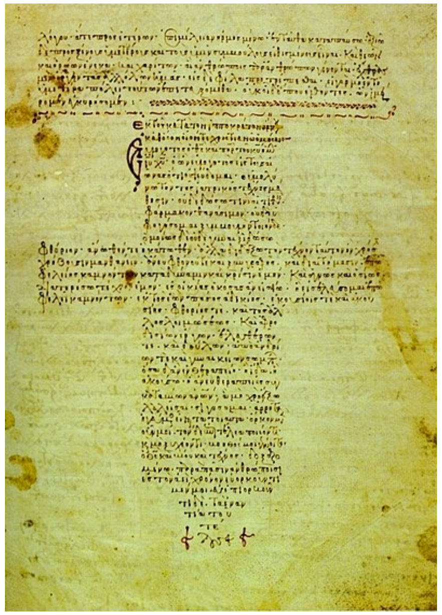
-Medicine

Medicine - Intrinsic

Hippocratic Oath

~450BC

“Do no Harm”



Law -Extrinsic

Bar regulates behavior

Oath to follow rules

Malpractice

Legal Malpractice

Not every mistake is legal malpractice. For malpractice to exist:

Attorney must handle a case inappropriately

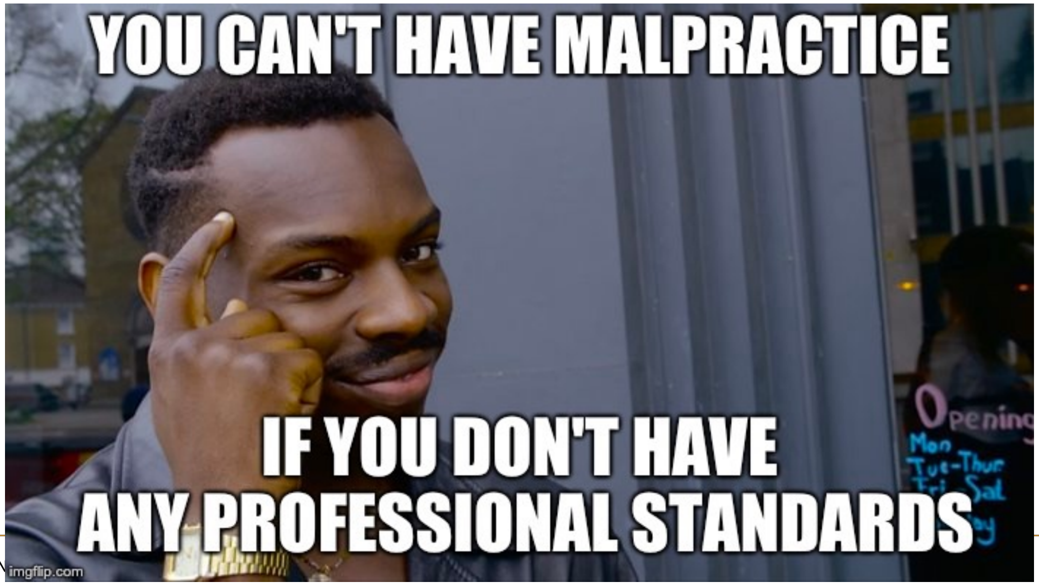
due to negligence or with intent to harm

And cause damages to a client

Malpractice vs. Negligence

Negligence is a failure to exercise the care that a **reasonably prudent person** would exercise in like circumstances.

Malpractice is a type of negligence; it is often called "professional negligence". It occurs when a **licensed professional** (like a doctor, lawyer or accountant) fails to provide services as per the **standards set by the governing body** ("standard of care"), subsequently causing harm to the plaintiff.



Bioengineering Ethics:

- Respect for Autonomy
- Beneficence
- Nonmaleficence
- Justice

Professional Engineers

17-313 Foundations of SOFTWARE ENGINEERING

What {is / could be} the role of **professional engineers** in software?

By ----PCStuff 03:47, 31 July 2006 (UTC), CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=10340855

https://en.wikipedia.org/wiki/Engineer%27s_Ring

Will software quality impact human flourishing?

Most traditional emphasis of “engineering ethics”

What can we learn from other professions?

Should software have ”Professional Engineers”?

How do we define “safety critical systems”?

How much testing is enough? How can we convince others to do that much testing?

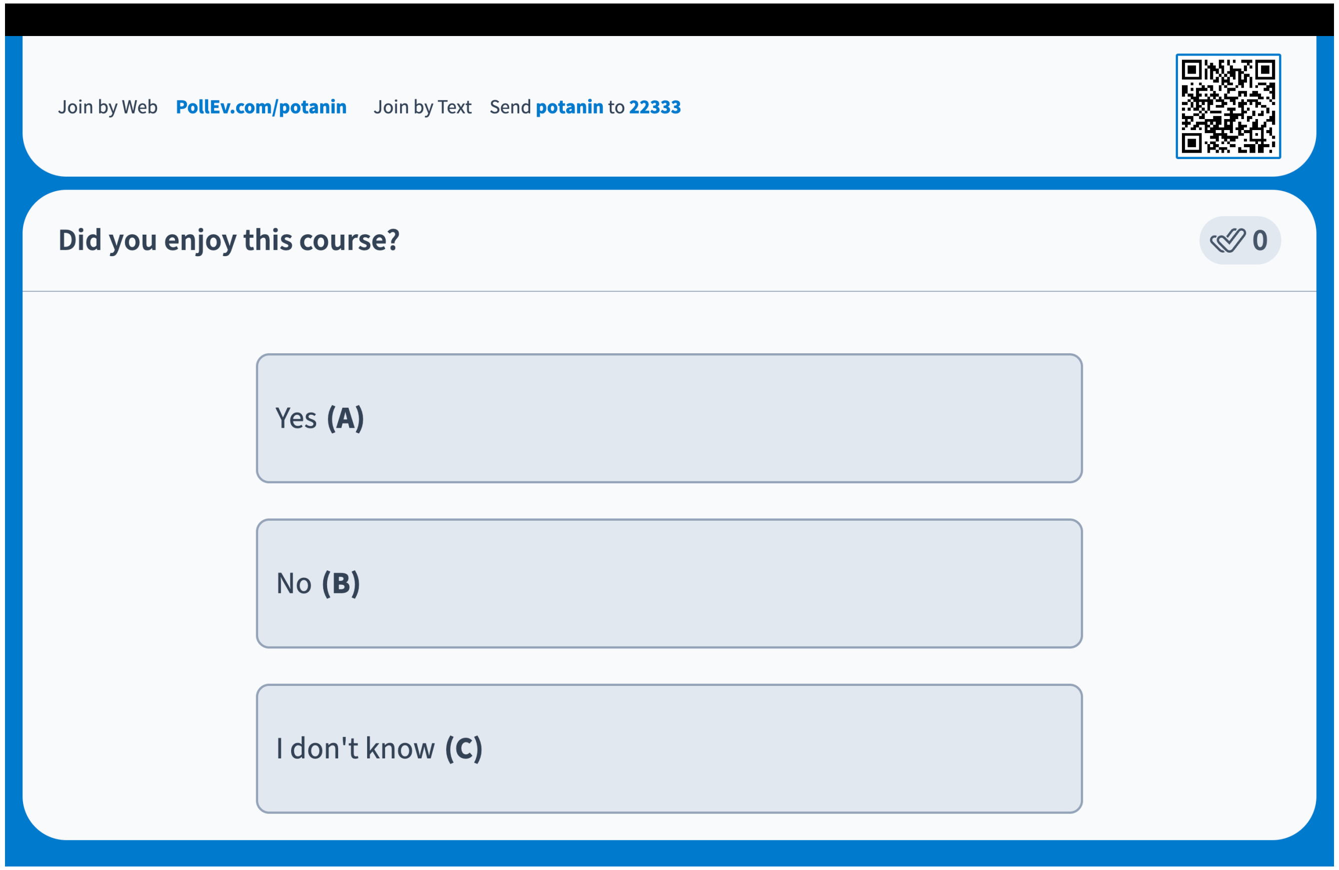
These questions are the **start** of the **conversation**, but as technology evolves, we must be **vigilant** to ensure we are promoting human flourishing

Three questions to promote human flourishing

1. Does my software respect the **humanity** of the **users**?
2. Does my software **amplify positive** behavior, or **negative** behavior for users and society at large?
3. Will my software's **quality** impact the **humanity** of others?

https://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world?language=en

Poll Everywhere Time!



Join by Web **PollEv.com/potanin** Join by Text Send **potanin** to **22333**

Did you enjoy this course?

Yes **(A)**

No **(B)**

I don't know **(C)**