

Week: COMP 2120 / COMP 6120

12 of 12

OPEN SOURCE

A/Prof Alex Potanin

Dr Fabian Muehlboeck



Today

- Open Source
- Licenses
- Dependency Management
- Ethics





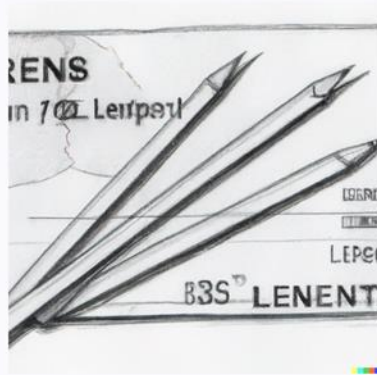
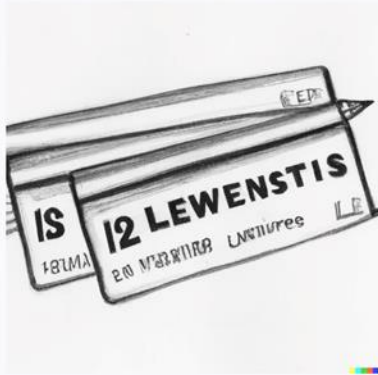
Edit the detailed description

Surprise me

Upload →

pencil drawing of licenses

Generate



Copyright, Licensing, and Open Source

Poll Everywhere Time!

EULAs and you

0

I know what a EULA is and I read all of them

I know what a EULA is and I read some of them

I know what a EULA is and I read one once

I know what a EULA is, but I have never read one

I don't know what a EULA is



[PollEv.com/fabianm](https://pollev.com/fabianm)

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app



Disclaimer

IANAL (I am not a lawyer)

None of the following is meant to be legal advice – you need to consult an actual lawyer for that

You may also be interested in taking courses like LAWS8422

Copyright slides based on:

https://www.ag.gov.au/sites/default/files/2022-07/short_guide_to_copyright.pdf

© Commonwealth of Australia 2016



Copyright

- Certain rights to intangible kinds of property, in particular “original works”, e.g. software
- NOT for ideas – that’s the realm of patents, and their use for software is contentious
- Big differences between jurisdictions, but common themes and harmonisation through treaties (in particular, the Berne Convention and the TRIPS Agreement)



Copyright

- No need to register, no need to explicitly mark
- Why write e.g. “Copyright © 2024 Fabian Muehlboeck”?
 - To give others an idea of whether copyright still applies (has it expired?) and who to contact for a potential license
- Lasts for a long time: minimum for literary works (includes software) in Australia is 70 years after death of author
- Who the creator/author is depends on the kind of work and the circumstances of its creation
 - For software, it is generally whoever wrote the code, except if someone commissioned it, or it was made for an employer, and some other exceptions



Copyright

Economic Rights

- Reproduce (i.e. print/copy)
- Publish/perform/adapt/...
- Grant licenses for above

Can be bought and sold, can be assigned automatically through contracts (e.g. employment)

Moral Rights

- Attribution
- No false Authorship
- Integrity of Authorship

Always stay with creator – not transferable



Copyright - Exceptions

- “Fair Dealing”
 - Research or study
 - Criticism or review
 - Reporting of news
 - Giving of professional advice (lawyer/attorney)
 - Parody and satire
- Additional Limited Exceptions
 - Certain kinds of home copying
 - Archival copies for library services
 - For education
 - To assist access by people with disabilities

Additional
Conditions Apply!



Public Domain

Works in the “public domain” are free to use/reproduce/modify/... for everyone.

Copyright holders can release works into the public domain, or the work enters the public domain when copyright expires



Licensing

To use a kind of work that falls under copyright (i.e. software, videos, images, sound recordings, ...), at least one of the following needs to be true:

- You are the copyright holder
- The work is in the public domain
- You have a license to use it, and the license applies to the use case

It's good to have evidence!



Licensing

As a copyright holder, you can grant licenses with more or less arbitrary limits and conditions (subject to contract law)

You may have seen this in EULAs, LICENSE files on GitHub/GitLab, in code, etc...

Note: license agreements often contain unenforcible clauses (e.g. because of consumer protection laws).

Using standard licenses makes it more likely to achieve standard desirable outcomes.



The MIT License

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This is a “permissive” license.

Other common such licenses: BSD license, Apache license

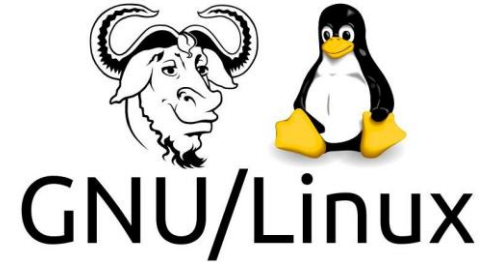


Copyleft



Forces licensees to make derivative work available under the same conditions. “reciprocal”/”viral”

Main example: GPL (GNU Public License), used in many parts of GNU/Linux



If you use code licensed to you under the GPL, you will likely have to also release the result under the GPL (or one of a small number of compatible licenses)



Photo: [Anders Brenna](#), [Teknisk Beta](#), [CC-BY](#)



Free Software



- “Free as in free speech.”
- 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.



Photo: [Anders Brenna](#), [Teknisk Beta](#), [CC-BY](#)



Open Source Software

- More business-driven initiative

“Open source enables a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is higher quality, better reliability, greater flexibility, lower cost, and an end to predatory vendor lock-in.”
[<https://opensource.org/about>]



A screenshot of the Open Source Initiative website showing various sponsor categories. The categories include: Alfred P. Sloan Foundation, Anchor (Google Open Source and Cisco), Premier (Bloomberg Engineering, Capital One, GitHub, Intel, Meta, Red Hat), and Community (CrossCompute, Driven, Loadview, etc.).

A screenshot of the Open Source Initiative website showing various sponsor categories. The categories include: Maintainer (FerretDB, Microsoft, OpenLogic), Supporter (Salesforce, SAS, Tidelift, Blindside, SENTRY, slim.ai, sysdig), Partner (Angular, CodeSee, etc.), and Community (CrossCompute, Driven, Loadview, etc.).





The Open Source Definition

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 Open Source Definition



1.

.

2.

.

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.

.

5.

.

6.

.

7.

.

8.

.

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.

.





The Open Source Definition

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.

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.



FOSS/FLOSS

Free(/Libre) Open-Source Software

Umbrella term that covers both the FSF and OSI definitions

FSF and OSI definitions largely similar,
and for most projects, both apply



Open-Source Development – Why?

- To improve software quality by getting volunteers to contribute to the project
- To satisfy the requirements of licenses for components you want to use
- To satisfy the requirements of your customers (e.g. governments)
- To build an ecosystem around other products
- To foster good will towards your project/organization
- Because you think open/free software is a good idea



Open-Source Development – How?

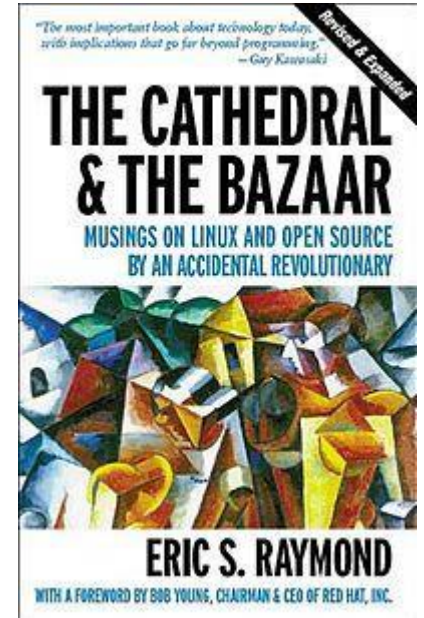
No prescriptions, but two major models:

1) The Cathedral:

Core team of developers works on product out of public view, publishes source code along with releases

2) The Bazaar:

Development process generally open, e.g. on GitHub – anyone can contribute, though maintainers may check and reject patches



Tables have turned



-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 6900 BASIC, and are writing 8900 API, 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



Open-Source License Overview

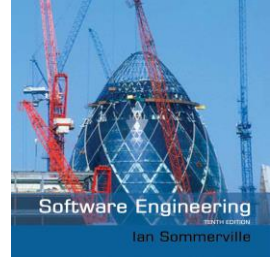
Permissive	Weak Copyleft	Strong Copyleft	Network Protections
MIT, BSD, Apache, ...	LGPL	GPL	AGPL
Use it almost however you want	You need to provide access to the source code of the LGPL-licensed code you are using, and if you modify it, you need to share the result under the LGPL, too	You need to provide source code to the whole product that includes GPL-licensed code, and redistribute it under the GPL (or compatible)	You need to provide source code for a whole software system (including server code) that includes AGPL-licensed code, and redistribute it under the AGPL

Companies often avoid these!

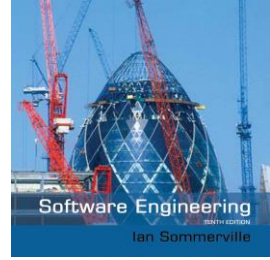


Open source business

- “Free as in free speech, not as in free beer”
vs. reality (know anyone who paid for WinRAR?)
- OSS business model typically not reliant on selling a software product but on selling support for that product.
- Companies often 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.



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.



Why would projects choose one license over another?

[Home](#)

Licenses

Open source licenses grant permission for anybody to use, modify, and share licensed software for any purpose, subject to conditions preserving the provenance and openness of the software. The following licenses are sorted by the number of conditions, from most (GNU AGPLv3) to none (Unlicense). Notice that the popular licenses featured on the [home page](#) (GNU GPLv3 and MIT) fall within this spectrum.

If you're looking for a reference table of every license on choosealicense.com, see the [appendix](#).

GNU AGPLv3

Permissions of this strongest copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. When a modified version is used to provide a service over a network, the complete source code of the modified version must be made available.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Patent use ● Private use 	<ul style="list-style-type: none"> ● Disclose source ● License and copyright notice ● Network use is distribution ● Same license ● State changes 	<ul style="list-style-type: none"> ● Liability ● Warranty

[View full GNU Affero General Public License v3.0 >](#)

GNU GPLv3

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Patent use ● Private use 	<ul style="list-style-type: none"> ● Disclose source ● License and copyright notice ● Same license ● State changes 	<ul style="list-style-type: none"> ● Liability ● Warranty

[View full GNU General Public License v3.0 >](#)

GNU LGPLv3

Permissions of this copyleft license are conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work through interfaces provided by the licensed work may be distributed under different terms and without source code for the

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Patent use ● Private use 	<ul style="list-style-type: none"> ● Disclose source ● License and copyright notice ● Same license (library) ● State changes 	<ul style="list-style-type: none"> ● Liability ● Warranty

[View full GNU Lesser General Public License v3.0 >](#)

Mozilla Public License 2.0

Permissions of this weak copyleft license are conditioned on making available source code of licensed files and modifications of those files under the same license (or in certain cases, one of the GNU licenses). Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Patent use ● Private use 	<ul style="list-style-type: none"> ● Disclose source ● License and copyright notice ● Same license (file) 	<ul style="list-style-type: none"> ● Liability ● Trademark use ● Warranty

[View full Mozilla Public License 2.0 >](#)

Apache License 2.0

A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Patent use ● Private use 	<ul style="list-style-type: none"> ● License and copyright notice ● State changes 	<ul style="list-style-type: none"> ● Liability ● Trademark use ● Warranty

[View full Apache License 2.0 >](#)

MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Private use 	<ul style="list-style-type: none"> ● License and copyright notice 	<ul style="list-style-type: none"> ● Liability ● Warranty

[View full MIT License >](#)

Boost Software License 1.0

A simple permissive license only requiring preservation of copyright and license notices for source (and not binary) distribution. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Conditions	Limitations
<ul style="list-style-type: none"> ● Commercial use ● Distribution ● Modification ● Private use 	<ul style="list-style-type: none"> ● License and copyright notice for source 	<ul style="list-style-type: none"> ● Liability ● Warranty

[View full Boost Software License 1.0 >](#)

The Unlicense

Permissions	Conditions	Limitations

[From <https://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

Poll Everywhere Time!

Join by Web PollEv.com/potanim Join by Text Send [potanim](https://poll-ev.com/potanim) 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

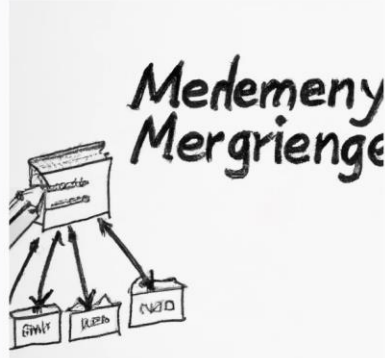
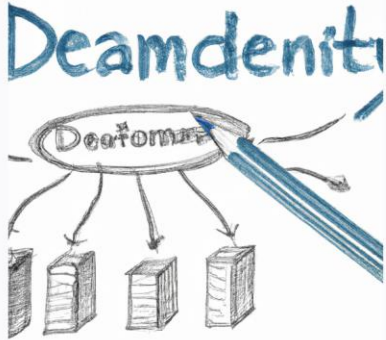
Surprise me

Upload



pencil drawing of dependency management

Generate



Dependency Management

Left-pad (March 22, 2016)



OBSSESSIONS 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 silkenrance opened this issue on Mar 22, 2016 · 193 comments



silkenrance 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/.npm/versions/node/v4.2.2/bin/node" "/home/travis/.npm/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

1.3.0

License

WTFPL

Unpacked Size

9.75 kB

Total Files

10

Issues

3

Pull Requests

7

Last publish

4 years ago



Left-pad (Source Code)



17 lines (11 sloc) | 222 Bytes

```
1  module.exports = leftpad;
2
3  function leftpad (str, len, ch) {
4    str = String(str);
5
6    var i = -1;
7
8    if (!ch && ch !== 0) ch = ' ';
9
10   len = len - str.length;
11
12   while (++i < len) {
13     str = ch + str;
14   }
15
16   return str;
17 }
```



See also: isArray

isArray

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

build passing | downloads 227M/month



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
```

5 lines (4 sloc) | 133 Bytes

```
1 var toString = {}.toString;  
2  
3 module.exports = Array.isArray || function (arr) {  
4   return toString.call(arr) === '[object Array]';  
5 };
```

```
> npm i isArray
```

Repository

github.com/juliangruber/isarray

Homepage

github.com/juliangruber/isarray

Weekly Downloads

50,913,317

Version

2.0.5

License

MIT

Unpacked Size

3.43 kB

Total Files

4

Issues

4

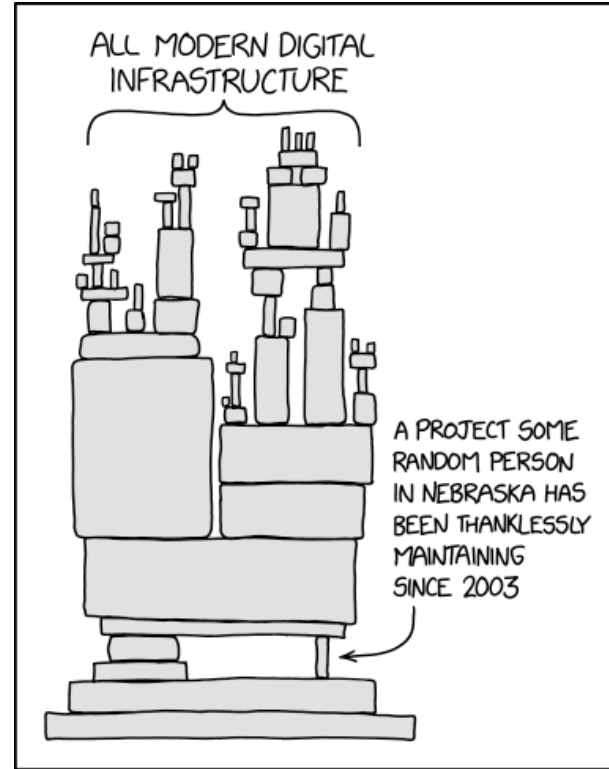
Pull Requests

3



Dependency Management

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



Comic by Randall Munroe
<https://xkcd.com/2347/>



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?
 - Do you have an appropriate license for external dependencies?

Dependencies: Example



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

fast, scalable, distributed revision control system

Other Packages Related to git


45 lines (45 sloc)

```
1 Pillow==8.3
2 PyPDF2==1.2
3 PyYAML==5.4
4 Whoosh==2.7
5 bleach==4.0
6 celery==5.1
7 django-acti
8 django-cele
9 django-colo
10 django-cors
11 django-form
12 django-math
13 django-mode
14 django-mptt
15 django-pure
16 django-qsst
17 django-solo
18 django-stroi
19 django-wldg
20 djangoestf
21 djangoestf
22 drf-yasg==1
23 extract-msg
24 flanker==0.
25 flex==6.14.
26 furl==2.1.2
27 fusepy==3.0
28 gevent==21.
29 graphviz==0
30 gunicorn==2
31 jsonschema=
32 mock==4.0.3
33 node-semver==0.8.0
34 packaging==21.0
```

• 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 from Debian):

- Gerrit Page
- Jonathan Nieder
- Anders Kaseorg

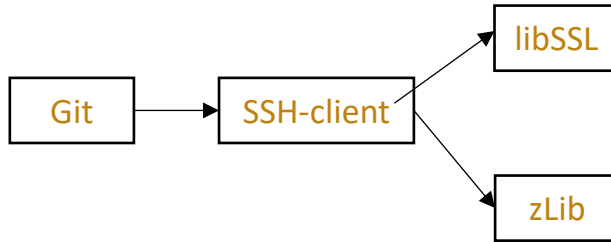
It should generally not be necessary for users to contact the original maintainer.

External Resources:



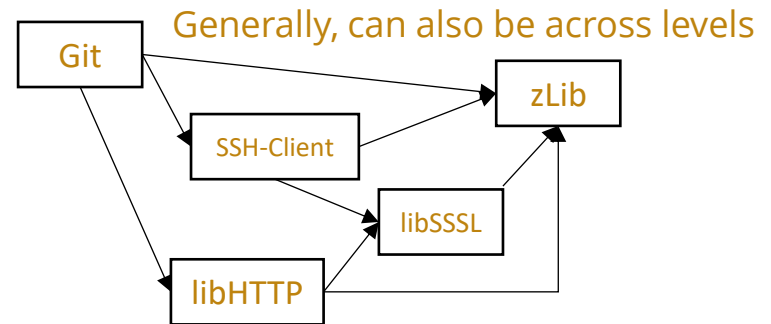
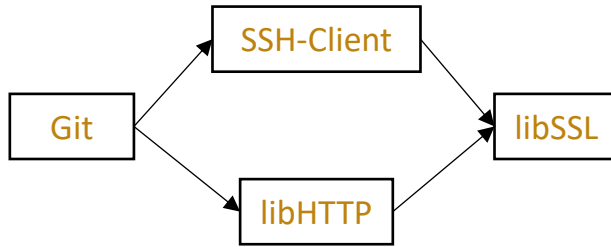
Transitive Dependencies

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



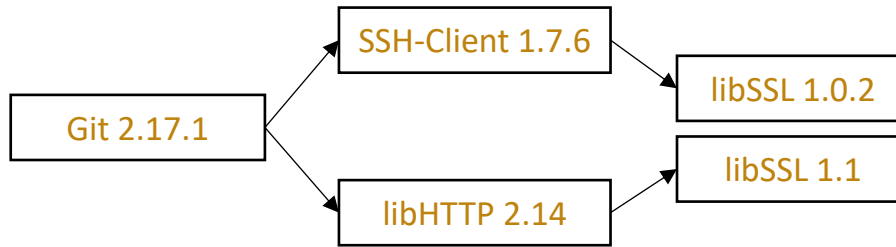
Diamond Dependencies

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



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 and security (!)

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/>



[2.0.0](#) [2.0.0-rc.2](#) [2.0.0-rc.1](#) [1.0.0](#) [1.0.0-beta](#)

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

 Build	Build #5: Manually run by rohanpadhye	9 days ago 16m 43s	...
 README: Add build badge	Build #4: Commit f656b2a pushed by rohanpadhye	2 months ago 18m 12s	...


CMU-313 / Mayan-EDMS Public template

forked from [mayan-edms/Mayan-EDMS](#)

<> Code Issues 70 Pull requests 70 Discussions Actions Projects Security Ins

✓ **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.

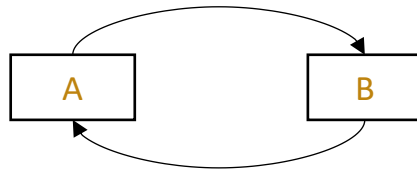
```

mayan/apps/rest_api/dependencies.py
@@ -59,6 +59,9 @@
59 59     PythonDependency(
60 60         module=__name__, name='flex', version_string='==6.14.1'
61 61     )
62 + PythonDependency(
63 +     module=__name__, name='jsonschema', version_string='==3.2.0'
64 + )
62 65     PythonDependency(
63 66         module=__name__, name='swagger-spec-validator', version_string='==2.5.0'
64 67     )

```

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



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 Security

SECURITY

AI hallucinates software packages and devs download them – even if potentially poisoned with malware

84 

Simply look out for libraries imagined by ML and make them real, with actual malicious code. No wait, don't do that

 [Thomas Claburn](#)

Thu 28 Mar 2024 // 07:01 UTC



IN-DEPTH Several big businesses have published source code that incorporates a software package previously hallucinated by generative AI.

Not only that but someone, having spotted this reoccurring hallucination, had turned that made-up dependency into a real one, which was subsequently downloaded and installed thousands of times by developers as a result of the AI's bad advice, we've learned. If the package was laced with actual malware, rather than being a benign test, the results could have been disastrous.

https://www.theregister.com/2024/03/28/ai_bots_hallucinate_software_packages/



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”
 - NEW: Generative AI hallucination squatting

Poll Everywhere Time!

Join by Web PollEv.com/potantin Join by Text Send **potantin** 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

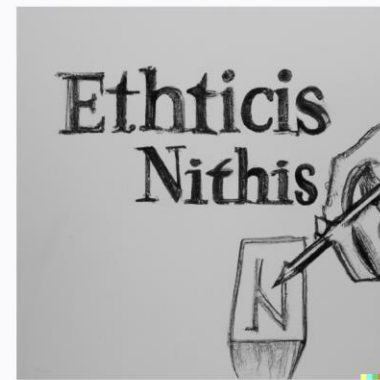
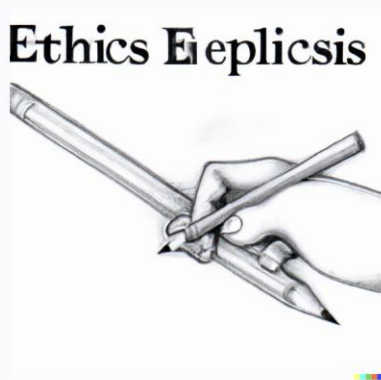
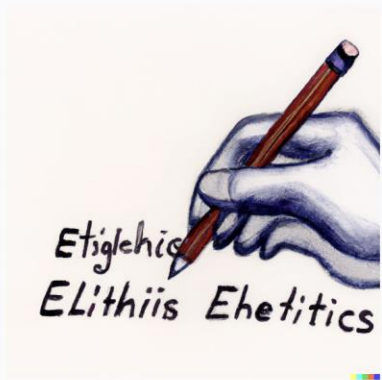
Surprise me

Upload



pencil drawing of ethics

Generate



Ethics



Just because you can,
does not mean you should

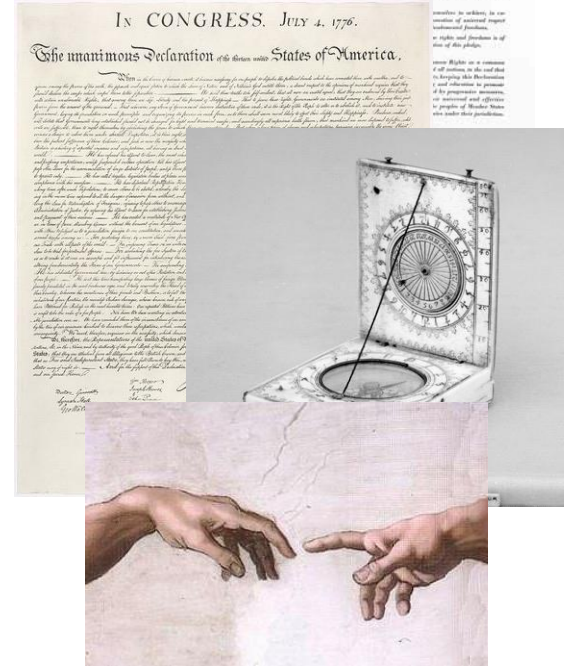


What is Human Flourishing?

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



How does Software Engineering affect Human Flourishing?



Therac-25

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

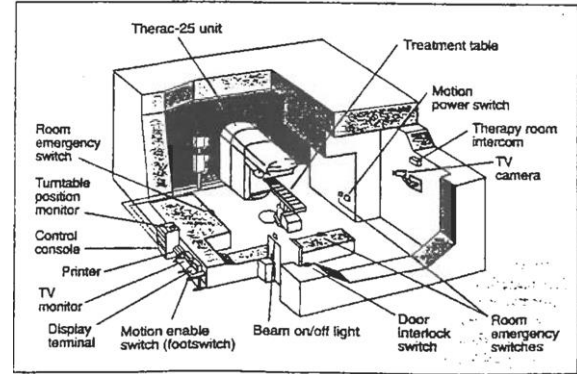
Traced to:

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

GANTRY ROTATION (DEG) 0.000000 0.000000 VERIFIED
COLLIMATOR ROTATION (DEG) 89.200000 89.200000 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: 2012-04-14 SYSTEM: BEAM READY OP.MODE: TREAT AUTO
TIME: 11:49:18 TREAT: TREAT PAUSE 0-DAY 179777
DB ID: 037-14740 REASON: OPERATOR COMMAND: █
    
```

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

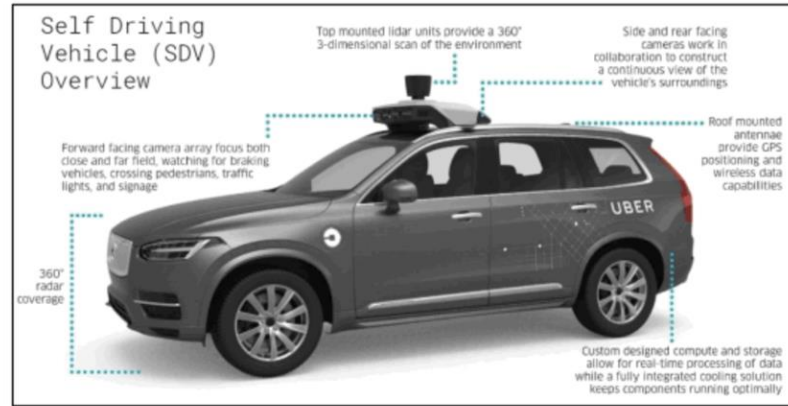
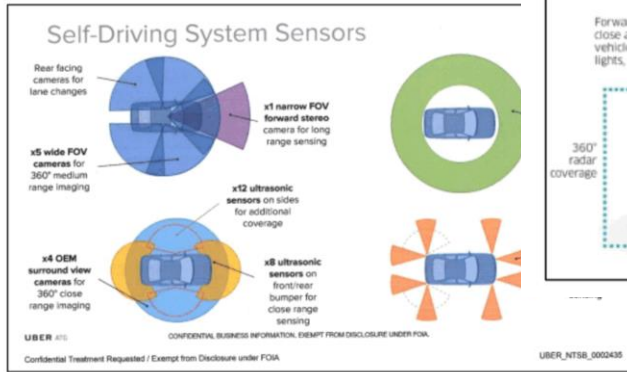
The system had several serious software flaws, the NTSB said.



Steve Dent, @stevetdent
 11.06.19 in [Transportation](#)

25
 Comments

1131
 Shares





Login

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



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

Search

- Facebook privacy
- Transportation
- Enterprise
- Def Con 2019

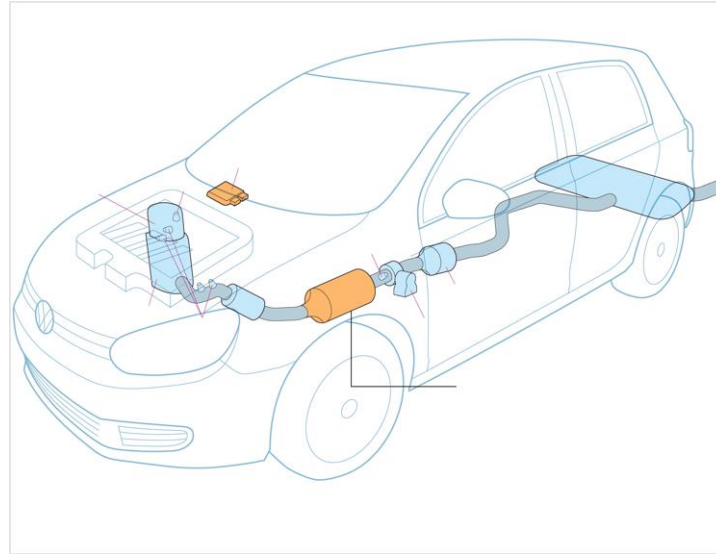
Ingrid Lunden @ingridlunden / 9:51 am EST • January 12, 2019

Comment



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>

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

80

'People like surprises,' executive tells UK Parliament

By [Ana Diaz](#) | [@AnaLikesPikachu](#) | Jun 21, 2019, 9:10am EDT

[f](#) [t](#) [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

[f](#) [t](#) [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

[f](#) [t](#) [s](#) SHARE



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





How Uber Uses Psychological Tricks to Push Its Drivers' Buttons

The company has undertaken an extraordinary experiment in behavioral science to subtly entice an independent work force to maximize its growth.

By NOAM SCHEIBER and graphics by JON HUANG | APRIL 2, 2017

<https://www.nytimes.com/interactive/2017/04/02/technology/uber-drivers-psychological-tricks.html>



Cryptocurrencies: Last Week Tonight with John Oliver (HBO)

<https://youtu.be/g6iDZspbRMg?t=896> (Part 1, 2018)

https://www.youtube.com/watch?v=o7zazuy_Ufl (Part 2, 2023)



How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did

Kashmir Hill Former Staff

Welcome to The Not-So Private Parts where technology & privacy collide



Feb 16, 2012, 11:02am EST

“If we send someone a catalog and say, ‘Congratulations on your first child!’ and they’ve never told us they’re pregnant, that’s going to make some people uncomfortable,” Pole told me. “We are very conservative about compliance with all privacy laws. **But even if you’re following the law, you can do things where people get queasy.**”

“Then we started mixing in all these ads for things we knew pregnant women would never buy, so the baby ads looked random. We’d put an ad for a lawn mower next to diapers. We’d put a coupon for wineglasses next to infant clothes. That way, it looked like all the products were chosen by chance.

“And we found out that as long as a pregnant woman thinks she hasn’t been spied on, she’ll use the coupons. She just assumes that everyone else on her block got the same mailer for diapers and cribs. As long as we don’t spook her, it works.”

via [How Companies Learn Your Secrets - NYTimes.com](#).

<https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/>



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

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 (Po)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 (Po)ICE" Arcieri @bascule

Trying a h... it...

Which will Barack Ob... ithm

TheArtGun COMMS OPEN @TheArtGun
Replying to @bascule
What if we adjust the contrast



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

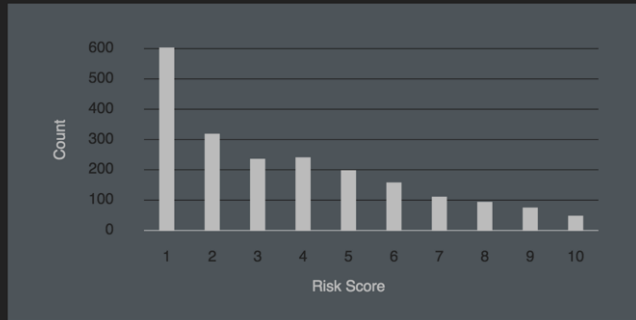
35 Retweets 5 Quote Tweets 102 Likes

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

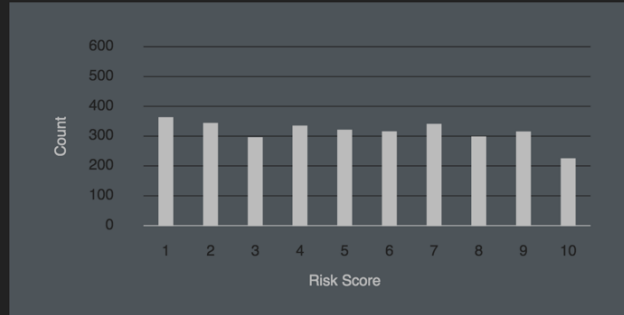
64.7K Retweets 16.3K Quote Tweets 198.6K Likes



White Defendants' Risk Scores



Black Defendants' 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.)

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





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





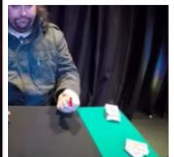

The New York Times | SUBSCRIBE NOW | LOG IN

Make a contribution | Subscribe | FINANCE NAV | US edition

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

We Need To Work Harder To Make Software Engineering More Ethical

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



...it ethics

...T READ

...to fool AI with magic

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



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.



Association for
Computing Machinery



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 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

How can we design products that are more sophisticated about human nature?

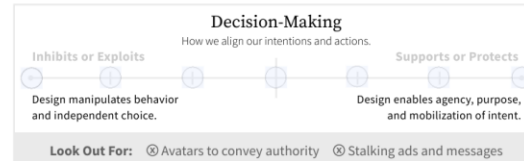
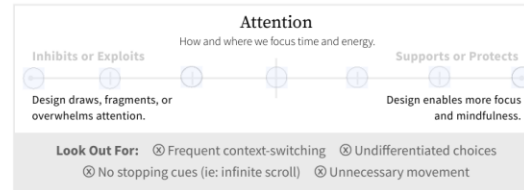
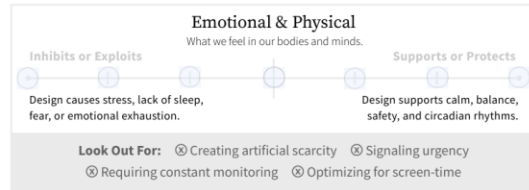
1 Identify a product or feature to design:

What is it?

What value should it bring to the user?

What is your measure of success?

2 For each vulnerability below, plot where your product currently lies on each spectrum of humane design:



https://s3.amazonaws.com/com.appolearning.files/production/uploads/uploaded_file/0fd922be-3bbc-46ea-b1a2-fdeb8eb4e3d5/Humane_Tech_Design_Guide.pdf

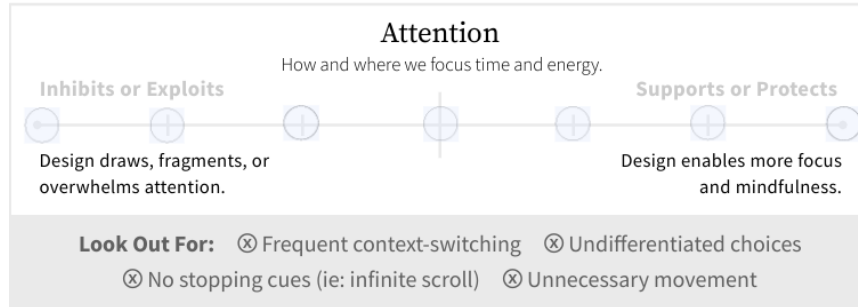


Humane Design Guide

<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 & Physical, Attention, Sense-Making, Decision-Making, Social Reasoning, and Group Dynamics



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



Abby Jones¹



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.

¹Abby 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>

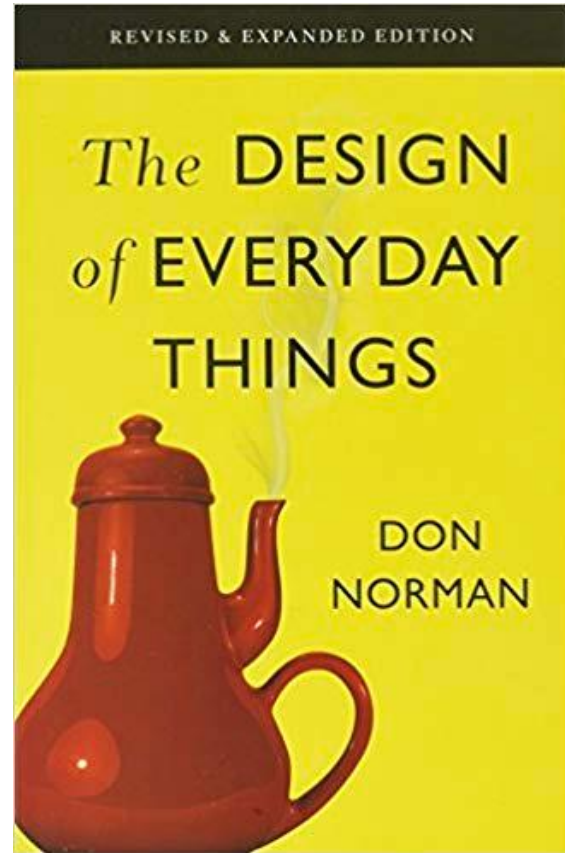
<ul style="list-style-type: none">1. Pick a persona. eg: Abby 2. Pick a use case/scenario in your tool, eg:<ul style="list-style-type: none">– in Book Store Navigator app...– “Find science fiction books”	<ul style="list-style-type: none">3a-b. Pick a Subgoal for that scenario. eg:  <p>Subgoal #1: “See bookstore map”.</p> <p>Q: Will Abby have formed this sub-goal...?</p> <ul style="list-style-type: none">• Yes/no/maybe. Why? Consider Abby’s Motivations...  
<ul style="list-style-type: none">3c-d. Pick an Action for that subgoal.  <p>Action #1: “Tap ‘Browse Off’”:</p> <ul style="list-style-type: none">– Q1. Will Abby know what to do?<ul style="list-style-type: none">• Yes/no/maybe. Why? Consider Abby’s, ... Tinkering  	<ul style="list-style-type: none">– 3e. Q2. If she performs the action, producing   

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

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

A background image showing a group of people in a meeting, with a whiteboard in the background. The image is slightly faded and serves as a backdrop for the text.

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:

Individuals and interactions over processes and tools
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/>



What-If Tool demo - binary classifier for predicting salary of over \$50k - UCI census income dataset

Partial dependence plots Compute distance Show nearest different classification: L1 L2

PERFORMANCE + FAIRNESS **DATAPOINT EDITOR** FEATURES

Binning 1 X Axis: age Co... Binning 1 Y Axis: marital-stat... C... Color By: Inference

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:

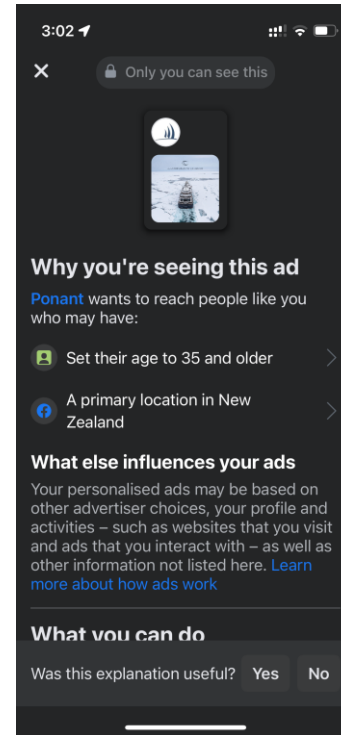
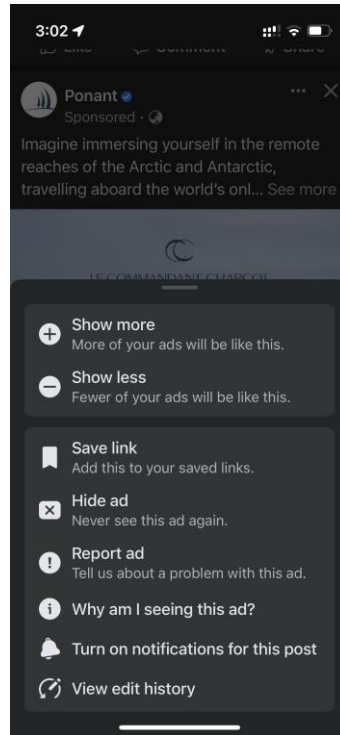
- Edit features and values and rerun inference to see how your model performs.
- Compute Distance: Select an example to be an anchor and create a new L1 or L2 distance feature for all loaded examples.
- Closest Counterfactuals: For classification models, find the closest example with a different classification 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.

The interface shows a grid of data points where each point is a small cluster of blue and red dots. A mouse cursor is hovering over one of the clusters. Above the grid, there are labels for age bins: 17-23, 24-29, 30-35, 36-42, and 43-49. To the right, there are controls for binning and color by.

Explain “why” to customers





@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://aniildash.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>



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*) [wiktionary](#)

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



AI Incident Database



A screenshot of a web browser showing the AI Incident Database website. The browser's address bar contains 'incidentdatabase.ai'. The website has a dark blue header with the logo and 'INCIDENT DATABASE' text. Below the header, there are two tabs: 'Discover' (selected) and 'Submit'. The main content area features a 'Welcome to the AIID' section with a sub-section 'Why "AI Incidents"?' and a paragraph of text. A 'What is an Incident?' section is also visible. On the right side, there is a 'CONTENTS' sidebar with a list of links. The left sidebar contains a navigation menu with items like 'Welcome to the AIID', 'Database Roadmap', 'Researcher Guide', 'Data Summaries', 'Database Apps', 'Discover App', 'Incident Report Submission', 'Your App Here', 'About Us', 'Contact and Follow', and 'Partnership on AI Home'.



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 [Eri]
- availability
- compatibility
- composability [Eri]
- configurability
- correctness
- credibility
- customizability
- debugability
- degradability
- determinability
- demonstrability
- dependability
- deployability
- discoverability [Eri]
- distributability
- durability
- effectiveness
- efficiency
- evolvability
- extensibility
- failure transparency
- fault-tolerance
- fidelity
- flexibility
- inspectability
- installability
- integrity
- interchangeability
- interoperability [Eri]
- 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 [Eri]
- 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>



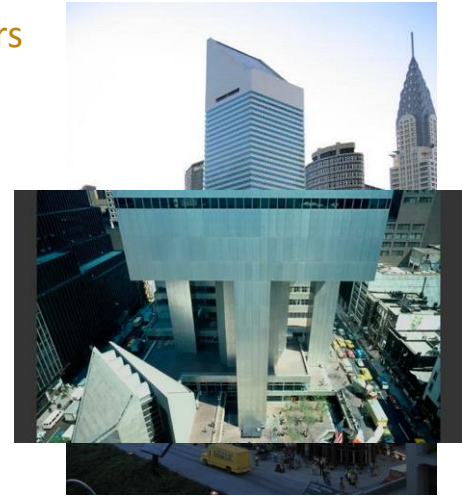
“hold paramount safety, health and welfare of the public”

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

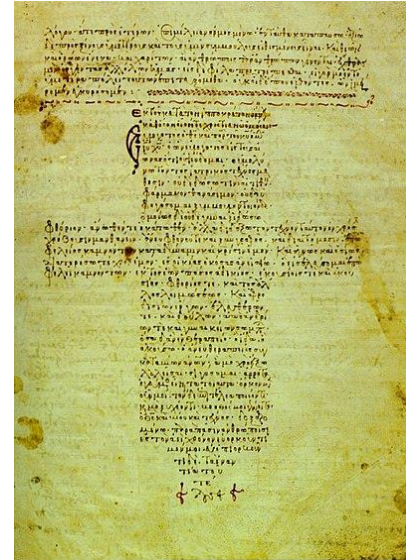


Medicine - Intrinsic

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

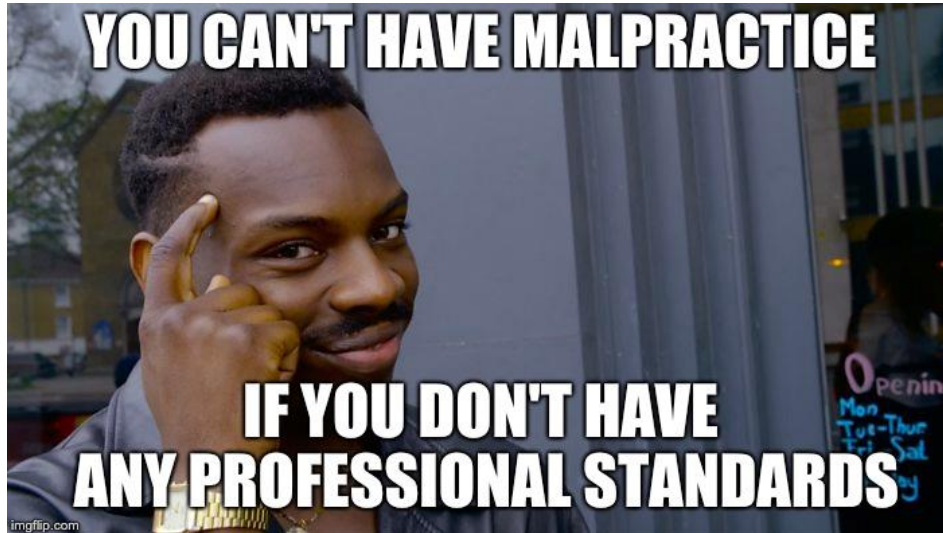


Malpractice vs. Negligence



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

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



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

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



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/potanim Join by Text Send [potanim](https://poll-ev.com/potanim) to 22333



Did you enjoy this course? 0

Yes (A)

No (B)

I don't know (C)