

CS 4530

Fundamentals of Software Engineering

Module 18: Engineering Equitable Software

Jonathan Bell, Adeel Bhutta, Mitch Wand
Khoury College of Computer Sciences
With contributions by Saiph Savage

© 2022, released under [CC BY-SA](#)

Learning Objectives for this Lesson

By the end of this lesson, you should be able to...

- List some ways in which software can cause inadvertent harm or amplify inequities, with examples
- Explain some techniques that a software engineer can use to detect and mitigate these harms.

From SE @ Google:

As new as the field of software engineering is, we're newer still at understanding the impact it has on underrepresented people and diverse societies. ... [We must recognize] the increasing imbalance of power between those who make development decisions that impact the world and those who simply must accept and live with those decisions that sometimes disadvantage already marginalized communities globally.

Badly -engineered software can kill people

Therac -25 (1985-1987)

- Bug in software caused 100x greater exposure to radiation than intended
- At least 6 died
- Likely far more suffered: deaths occurred over a period of 2 years!
- Weak accountability in manufacturer's organization



"Therac-25" by Catalina Márquez, Wikimedia commons, CC BY-SA 4.0

Algorithmic sentencing systems can discriminate against Black defendants

Example: the COMPAS Sentencing Tool

	ALL DEFENDANTS	WHITE DEFENDANTS	BLACK DEFENDANTS
Labeled Higher Risk, But Didn't Re-Offend	32.4%	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	37.4%	47.7%	28.0%

Algorithmic bias can discriminate against poorer consumers



THE WALL STREET JOURNAL.



Websites Vary Prices, Deals Based on Users' Information

Getting Different Deals Online
A Journal examination found online retailers adjusted prices by a shopper's location, among other factors

Staples.com
SnapSafe Titan safe
HIGHER PRICE: \$1,199.99
DISCOUNT PRICE: \$1,099.99
DIFFERENCE: 9.1%

Homedepot.com
A 250-foot spool of electrical wiring
Six pricing groups, including:
\$70.80 in Ashtabula, Ohio
\$72.45 in Erie, Pa.
\$77.87 in Monticello, N.Y.

Rosettastone.com
A 20% DISCOUNT
...for buying multiple levels of German lessons, when test-shopping from the U.S. or Canada. But not from the U.K. or Argentina.

Photos: l to r: SnapSafe; Home Depot; Rosetta Stone Source: WSJ testing The Wall Street Journal

SNAPSAFE; HOME DEPOT; ROSETTA STONE

By Jennifer Valentino-DeVries, Jeremy Singer-Vine and Ashkan Soltani

December 24, 2012

<https://www.wsj.com/articles/SB1000142412788732377720457818939181388153>

2017 IEEE European Symposium on Security and Privacy

FairTest: Discovering Unwarranted Associations in Data-Driven Applications*

Florian Tramèr¹, Vaggelis Atlidakis², Roxana Geambasu², Daniel Hsu², Jean-Pierre Hubaux³, Mathias Humbert⁴, Ari Juels⁵, Huang Lin³

¹Stanford, ²Columbia University, ³EPFL, ⁴Saarland University, ⁵Cornell Tech, Jacobs Institute

Abstract—In a world where traditional notions of privacy are increasingly challenged by the myriad companies that collect and analyze our data, it is important that decision-making entities are held accountable for unfair treatments arising from irresponsible data usage. Unfortunately, a lack of appropriate methodologies and tools means that even identifying unfair or discriminatory effects can be a challenge in practice.

We introduce the *unwarranted associations (UA) framework*, a principled methodology for the discovery of unfair, discriminatory, or offensive user treatment in data-driven applications. The UA framework unifies and rationalizes a number of prior attempts at formalizing algorithmic fairness. It uniquely combines multiple investigative primitives and fairness metrics with broad applicability, granular exploration of unfair treatment in user subgroups, and incorporation of natural notions of utility that may account for observed disparities.

We instantiate the UA framework in *FairTest*, the first comprehensive tool that helps developers check data-driven applications for unfair user treatment. It enables scalable and statistically rigorous investigation of associations between application outcomes (such as prices or premiums) and sensitive user attributes (such as race or gender). Furthermore, *FairTest* provides *debugging capabilities* that let programmers rule out potential confounders for observed unfair effects.

We report on use of *FairTest* to investigate and in some cases address disparate impact, offensive labeling, and uneven rates of algorithmic error in four data-driven applications. As examples, our results reveal subtle biases against older populations in the distribution of error in a predictive health application and offensive racial labeling in an image tagger.

1. Introduction

Today's applications collect and mine vast quantities of personal information. Such data can boost applications' utility by personalizing content and recommendations, increase business revenue via targeted product placement, and improve a wide range of socially beneficial services, such as healthcare, disaster response, and crime prevention.

The collection and use of such data raise two important challenges. First, massive data collection is perceived by many as a major threat to traditional notions of individual privacy. Second, the use of personal data for algorithmic

*Work done while the first author was at EPFL.

decision-making can have unintended and harmful consequences, such as unfair or discriminatory treatment of users.

In this paper, we deal with the latter challenge. Despite the personal and societal benefits of today's data-driven world, we argue that companies that collect and use our data have a responsibility to ensure equitable user treatment. Indeed, European and U.S. regulators, as well as various policy and legal scholars, have recently called for increased *algorithmic accountability*, and in particular for decision-making tools to be audited and "tested for fairness" [1], [2].

There have been many recent reports of unfair or discriminatory effects in data-driven applications, mostly qualified as unintended consequences of data heuristics or overlooked bugs. For example, Google's image tagger was found to associate racially offensive labels with images of black people [3]; the developers called the situation a bug and promised to remedy it as soon as possible. In another case [4], *Wall Street Journal* investigators showed that Staples' online pricing algorithm discriminated against lower-income people. They referred to the situation as an "unintended consequence" of Staples's seemingly rational decision to adjust online prices based on user proximity to competitors' stores. This led to higher prices for low-income customers, who generally live farther from these stores.

Staples' intentions aside, it is evidently difficult for programmers to foresee all the subtle implications and risks of data-driven heuristics. Moreover, these risks will only increase as data is passed through increasingly complex machine learning (ML) algorithms whose associations and inferences may be impossible to anticipate.

We argue that such algorithmic biases are new kinds of *bugs*, specific to modern, data-driven applications, that programmers should proactively check for, debug, and fix with the same rigor as they apply to other security and privacy bugs. Such bugs can offend and even harm users, and cause programmers and businesses embarrassment, mistrust, and potentially loss of revenue. They may also be symptoms of a malfunction of a data-driven algorithm, such as a ML algorithm exhibiting poor accuracy for minority groups that are underrepresented in its training set [5].

We refer to such bugs generically as *unwarranted associations*. Understanding and identifying unwarranted associations is an important step towards holding automated decision-making entities *accountable* for unfair practices, thus also providing incentive for the adoption of corrective measures [1], [2], [6], [7].

The Unwarranted Associations Framework. In order to

AI training systems can have serious impacts on climate.

{* AI + ML *}
AI me to the Moon... Carbon footprint for 'training GPT-3' same as driving satellite and back
Get ready for Energy Star stickers on your
Katyanna Quach Wed 4 Nov 2020 // 07:59 UTC

Training OpenAI's giant GPT-3 text-generating model from car to the Moon and back, computer scientists revealed. More specifically, they estimated teaching the neural network Microsoft data center using Nvidia GPUs required energy which using the average carbon intensity of America would have produced 85,000 kg of CO₂ equivalents, the same amount produced by a new car in Europe driving 700,000 km, or 435,000 miles, which is about twice the distance between Earth and the Moon, some 480,000 miles. Phew.

Not to mention bitcoin mining!

Consumption	CO ₂ e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Driving, 1 year	11,023
Avg, 1 year	36,156
Electricity, 1 lifetime	126,000
Model (GPU)	
Training, SRL	39
Experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

"Energy and Policy Considerations for Deep Learning in NLP" Emma Strubell, Ananya Ganesh, Andrew McCallum, in Proceedings of ACL 2019

Poor user interfaces can discriminate against differently -abled people.

Inclusivity and Accessibility: Domino's Pizza LLC v. Robles

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   SHARE



Jul 15 2019	Brief amicus curiae of Washington Legal Foundation filed.
Jul 15 2019	Brief amici curiae of Retail Litigation Center, Inc., et al. filed.
Jul 15 2019	Brief amicus curiae of Cato Institute filed.
Jul 15 2019	Brief amicus curiae of Restaurant Law Center filed.
Jul 15 2019	Brief amici curiae of Chamber of Commerce of the United States of America, et al. filed.

"Domino's Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind" by Brenna Houck, Eater Detroit

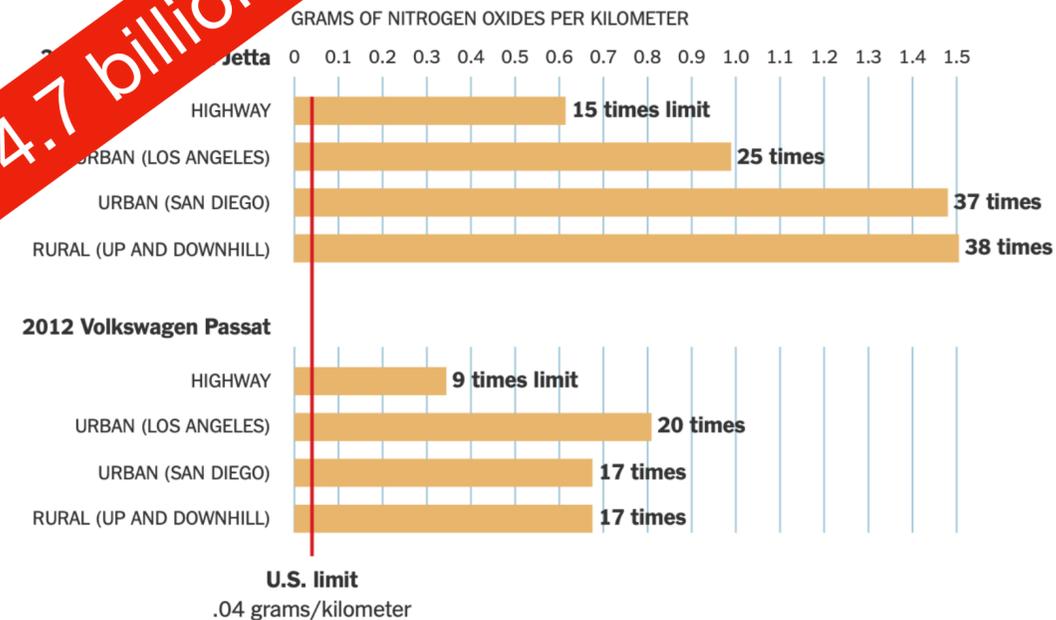
Software Systems can be used to evade regulation.

Example: Volkswagen diesel emissions

The Emissions Tests That Led to the Discovery of VW's Cheating

The on-road testing in May 2014 that led the California Air Resources Board to investigate Volkswagen was conducted by researchers at West Virginia University. They tested emissions from two VW Jetta models equipped with the 2-liter turbocharged 4-cylinder diesel engine. The researchers found that when tested on the road, some cars emitted almost 40 times the regulated levels of nitrogen oxides.

Average emissions of nitrogen oxides in on-road testing



\$14.7 billion settlement

Source: Arvind Thiruvengadam, Center for Alternative Fuels, Engines and Emissions at West Virginia University

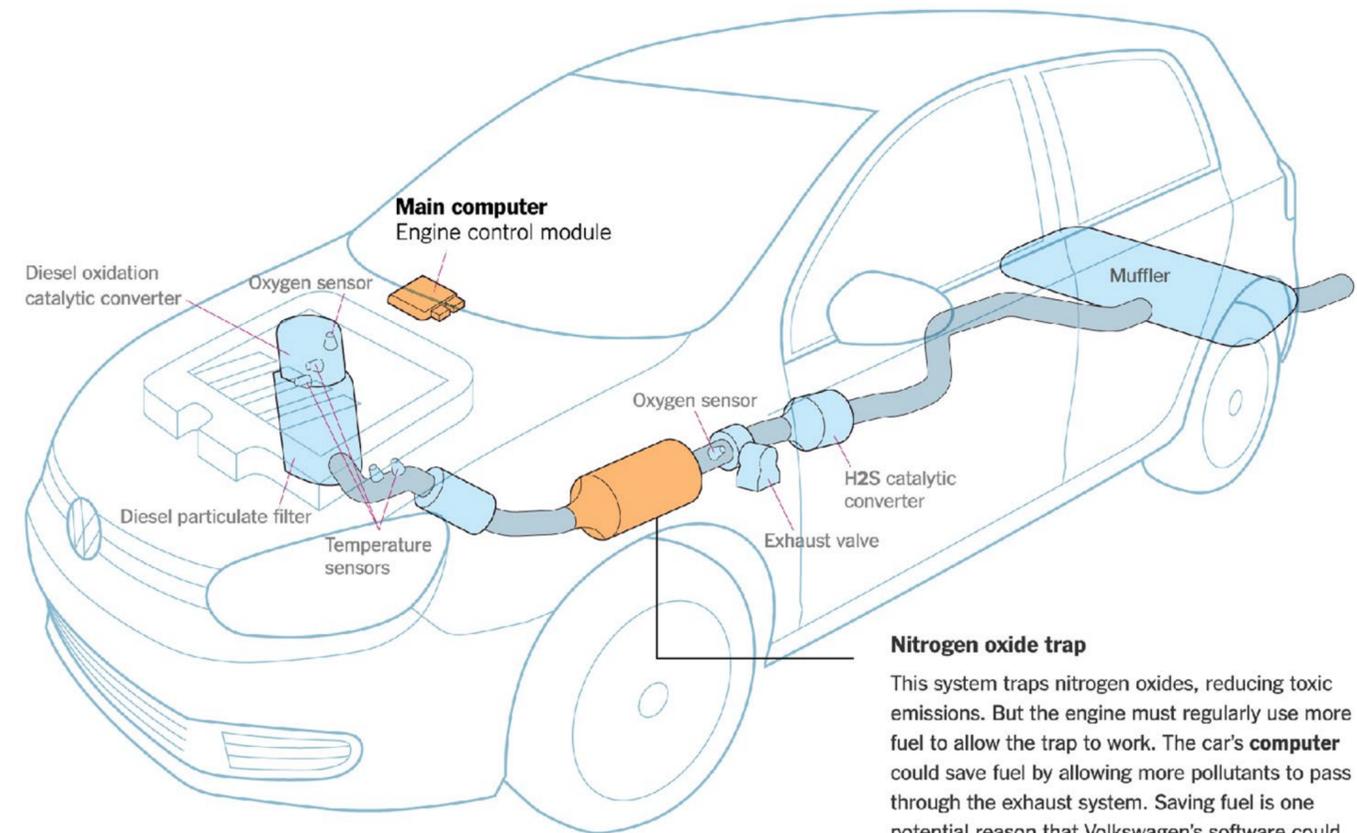


Illustration by Guilbert Gates | Source: Volkswagen, The International Council on Clean Transportation

“How Volkswagen’s ‘Defeat Devices’ Worked” By Guilbert Gates, Jack Ewing, Karl Russell and Derek Watkins

Software can be used in unanticipated ways

- Good twitter:
 - Keeping people informed,
 - Organizing
 - Helping people find communities
 - “Verified identities”
- Bad twitter:
 - Spreading misinformation
 - Bullying
 - Intimidation
 - “Identities for sale, \$7/month”

Inequities can arise in many ways

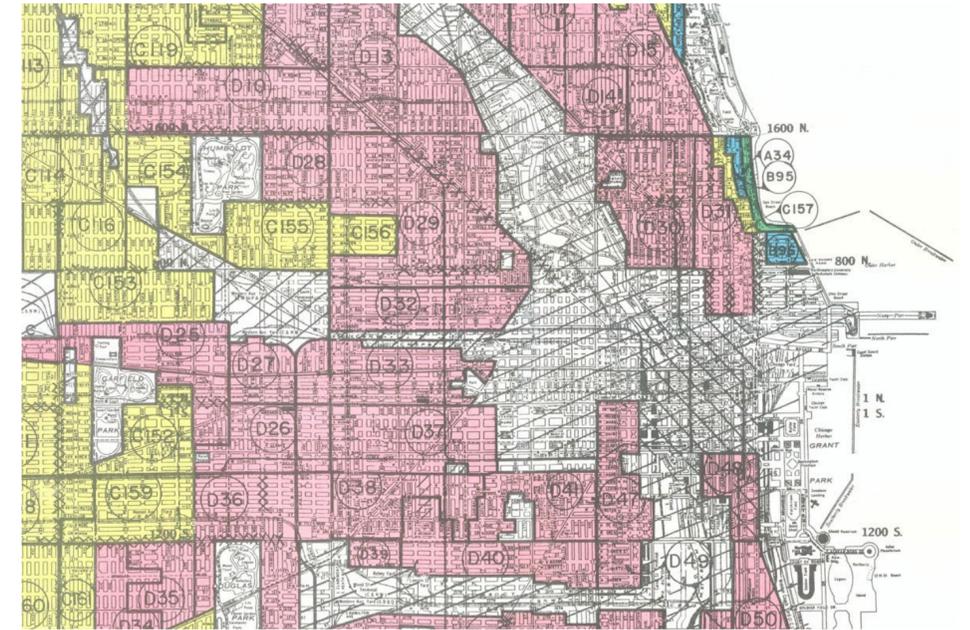
- Intentional (like VW)
- Unintentional bias on the part of the designer (like “sex: male or female”)
- Engineering shortcuts (like Domino’s)
- Unanticipated changes in usage or society (like Twitter)

Engineering Equitable Software Requires Conscious Effort

- Not just “don’t be evil”
- How do we determine what “the right thing” is?
- How do we convince our investors/managers to take this action?

Every design decision advantages some and disadvantages others

- Prioritize short print jobs over long ones?
- Prioritize high-value customers in phone queues?
- Whose interests to prioritize in recommendation systems?
 - The people doing the searching?
 - The people who have materials that they want found?
 - The people running the recommendation system?
- Prioritize people who are “good risks”
 - Red-lining
 - Felt pads



There are often conflicting values that must be reconciled



- What would you generate as an alt-text for this image?
- A Native American man on a horse and a white woman on a horse?
- A man on a horse and a woman on a horse?
- One alternative reinforces racial stereotypes; the other denies the man his identity.

How should we go about analyzing our software for equity?

How can my software make a positive contribution?

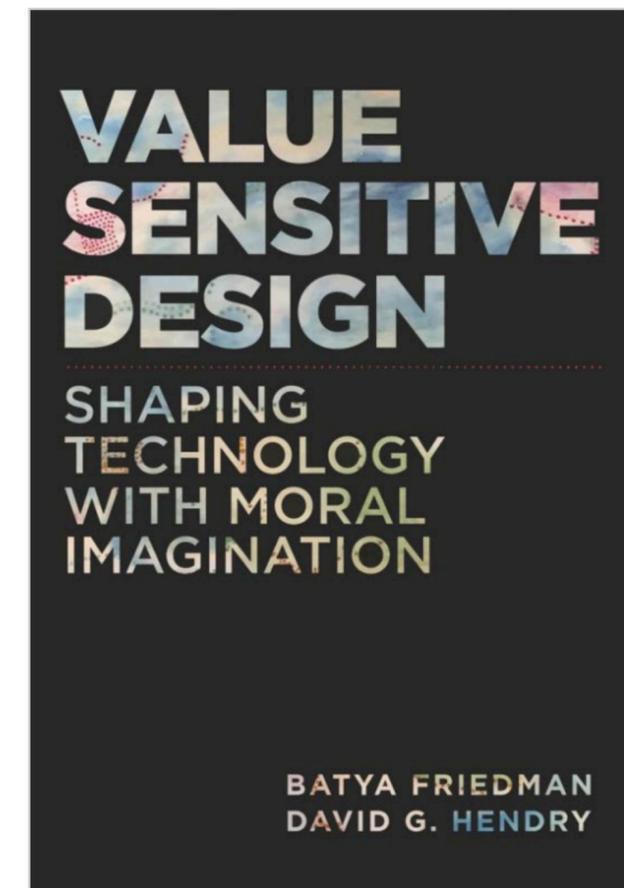
- Can my software make people's jobs easier?
- Can my software make people happier?
- Can my software amplify positive behavior for users and society at large?
- How can my software better achieve these goals?

How can my software cause harm?

- Who will be advantaged by the use of my software, and who be disadvantaged?
- How can my software fail? What are the implications of that failure?
- Who will use my software, and how might different users use it differently?
- How will my software impact those who do not use it directly?

What values might our software promote or diminish?

- Human rights - Inalienable, fundamental rights to which all people are entitled
- Accessibility - Making all people successful users of the technology
- Justice - Procedural justice (process is fair) + distributive justice (outcomes are fair)
- Privacy - An individual's agency in determining what information about them is shared
- Human welfare - Physical, material and psychological well-being



Code of Ethics

ACM's Code of Ethics Software Engineers

1. PUBLIC – Software engineers shall act consistently with the public interest.

2
€
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

1. PUBLIC – Software engineers shall act consistently with the public interest.

4. JUDGMENT – Software engineers shall maintain integrity and independence in their professional judgment.

5. MANAGEMENT – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. PROFESSION – Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7. COLLEAGUES – Software engineers shall be fair to and supportive of their colleagues.

8. SELF – Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

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

TLDR: No

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.

Standards can give more concrete guidance.

- International bodies define standard processes that are designed to protect the public
- By (correctly) following such a standard, you can reduce the chance of harm to users, as well as your ethical (and legal) liability

INTERNATIONAL
STANDARD

IEC
62304

First edition
2006-05

Medical device software –
Software life cycle processes

*This **English-language** version is derived from the original **bilingual** publication by leaving out all French-language pages. Missing page numbers correspond to the French-language pages.*



Reference number
IEC 62304:2006(E)

Standards can give more concrete guidance.

The screenshot shows the FAA website with a green banner at the top containing COVID-19 information. Below the banner is the United States Department of Transportation logo and the FAA logo. The main navigation bar includes links for About, Jobs, News, and a search box. A secondary navigation bar lists various FAA categories: AIRCRAFT, AIR TRAFFIC, AIRPORTS, PILOTS & AIRMEN, DATA & RESEARCH, REGULATIONS, SPACE, and DRONES. The breadcrumb trail reads: FAA Home > Aircraft > Aircraft Certification > Design Approvals. The left sidebar contains a menu for Aircraft Certification, with Design Approvals selected. The main content area features the title "Aircraft Certification Software and Airborne Electronic Hardware" and a detailed paragraph explaining the FAA's role in approving software and hardware for airborne systems. Below this is a "Top Tasks" sidebar with links to various services like "Get Form 337, Major Repair and Alteration" and "Register an aircraft". At the bottom, there is an "Email List Update" section and a "FAA CONTINUED OPERATIONAL SAFETY" banner.

The latest general information on the Coronavirus (COVID-19) is available on [Coronavirus.gov](https://www.cdc.gov/coronavirus). For FAA-specific COVID-19 resources, please visit [faa.gov/coronavirus](https://www.faa.gov/coronavirus).

United States Department of Transportation

Federal Aviation Administration

About Jobs News Search

AIRCRAFT AIR TRAFFIC AIRPORTS PILOTS & AIRMEN DATA & RESEARCH REGULATIONS SPACE DRONES

FAA Home > Aircraft > Aircraft Certification > Design Approvals

Aircraft Certification

- Aircraft Registration
- Airworthiness Certification
- Continued Operational Safety
- Design Approvals**
- Engines and Propellers
- Field Approvals & Supplemental Type Certificates (STCs)
- Rotorcraft
- Small Airplanes
- Supplemental Type Certificates
- International
- Locate an Office
- Production Approvals
- Senior Technical Experts Program (STEP)

Aircraft Safety

General Aviation & Recreational Aircraft

Repair Stations

Aircraft Certification Software and Airborne Electronic Hardware

The Aircraft Certification Service is concerned with the approval of software and airborne electronic hardware for airborne systems (e.g., autopilots, flight controls, engine controls), as well as that used to produce, test, or manufacture equipment to be installed on airborne products. The FAA Aircraft Certification Service develops policy, guidance and training for software and airborne electronic hardware that has an effect on the airborne product (a "product" is an aircraft, an engine, or a propeller).

For a list of people you can contact for additional information regarding Aircraft Certification Software and Airborne Electronic Hardware activities, please visit the [Contacts](#) page.

Email List Update

We are updating the email list used for notification of activities relating to airborne digital systems developed using software and airborne electronic hardware. To be added to the list, send an email to

Top Tasks

- Get Form 337, Major Repair and Alteration
- Register an aircraft
- Look up an N-Number
- Review preliminary accident data
- Find aircraft safety alerts
- Search for SAIBs

FAA CONTINUED OPERATIONAL SAFETY

Continued operational safety ensures the integrity of a product throughout its service life, and includes mandatory requirements for modification, maintenance, inspection and corrective actions.

Standards can give more concrete guidance.

Example: Domino's + ADA

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](#)

A screenshot of a web browser showing the Table of Contents for the WCAG 2.0 Specification. The browser address bar shows "w3.org". The page has a blue sidebar on the left with the text "WC3 Recommendation". The main content area lists the following sections:

- Table of Contents
- Introduction
 - [WCAG 2.0 Layers of Guidance](#)
 - [WCAG 2.0 Supporting Documents](#)
 - [Important Terms in WCAG 2.0](#)
- WCAG 2.0 Guidelines
 - 1 Perceivable
 - [1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.](#)
 - [1.2 Provide alternatives for time-based media.](#)
 - [1.3 Create content that can be presented in different ways \(for example simpler layout\) without losing information or structure.](#)
 - [1.4 Make it easier for users to see and hear content including separating foreground from background.](#)
 - 2 Operable
 - [2.1 Make all functionality available from a keyboard.](#)
 - [2.2 Provide users enough time to read and use content.](#)
 - [2.3 Do not design content in a way that is known to cause seizures.](#)
 - [2.4 Provide ways to help users navigate, find content, and determine where they are.](#)
 - 3 Understandable
 - [3.1 Make text content readable and understandable.](#)
 - [3.2 Make Web pages appear and operate in predictable ways.](#)
 - [3.3 Help users avoid and correct mistakes.](#)
 - 4 Robust
 - [4.1 Maximize compatibility with current and future user agents, including assistive technologies.](#)
- Conformance
 - [Conformance Requirements](#)
 - [Conformance Claims \(Optional\)](#)
 - [Statement of Partial Conformance - Third Party Content](#)
 - [Statement of Partial Conformance - Language](#)
- Appendices
 - Appendix A: [Glossary](#) (Normative)
 - Appendix B: [Acknowledgments](#)
 - Appendix C: [References](#)

What actions can we take?

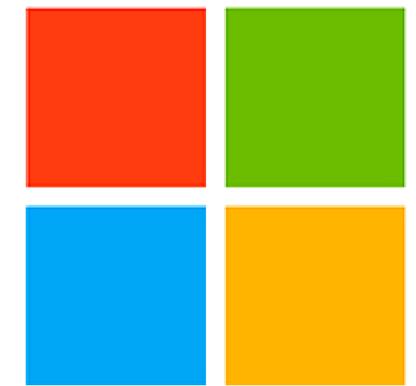
- How can we anticipate the harms our software might do?
- How do we design our software to mitigate unintentional harms?
- How do we work with our teams and employers to minimize the harms that our software might do?
- Everything can and should be iterated on, including the problem itself ... what are we trying to solve?



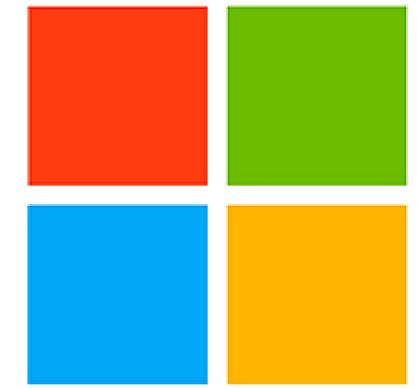
Inclusiveness and Iteration are key

- . For every piece of software you create, you should iterate on it and include a wide range of people to use your software.
- . By including more people you can better detect biases and harm that your software might create on certain populations.

Guidelines from Microsoft on how to create software for people that mitigates harm.



Microsoft



Microsoft

1

INITIALLY

Make clear what
the system can do

Help the users understand what
the AI system is capable of doing.

2

INITIALLY

Make clear how
well the system can
do what it can do.

Help the user understand how
often the AI system may make
mistakes.



INITIALLY

3

DURING INTERACTION

Time services based on context.

Time when to act or interrupt based on the user's current task and environment.

4

DURING INTERACTION

Show contextually relevant information.

Display information relevant to the users' current task and environment.

5

DURING INTERACTION

Match relevant social norms.

Ensure the experience is delivered in a way that users would expect, given their social and cultural context.

6

DURING INTERACTION

Mitigate social biases.

Ensure the AI system's language and behaviors do not reinforce undesirable and unfair stereotypes and biases.



DURING INTERACTION



Microsoft

7

WHEN WRONG

Support efficient invocation.

Make it easy to invoke or request the AI system's services when needed.

8

WHEN WRONG

Support efficient dismissal.

Make it easy to dismiss or ignore undesired system services.

9

WHEN WRONG

Support efficient correction.

Make it easy to edit, refine, or recover when the AI system is wrong.

10

WHEN WRONG

Scope services when in doubt.

Engage in disambiguation or gracefully degrade the AI system's services when uncertain about a user's goals.

11

WHEN WRONG

Make clear why the system did what it did.

Enable the user to access an explanation of why the AI system behaved as it did.

 WHEN WRONG



Microsoft

12
OVER TIME

Remember recent interactions.

Maintain short-term memory and allow the user to make efficient references to that memory.

13
OVER TIME

Learn from user behavior.

Personalize the user's experience by learning from their actions over time.

14
OVER TIME

Update and adapt cautiously.

Limit disruptive changes when updating and adapting the AI system's behaviors.

15
OVER TIME

Encourage granular feedback.

Enable the user to provide feedback indicating their preferences during regular interaction with the AI system.

16
OVER TIME

Convey the consequences of user actions.

Immediately update or convey how user actions will impact future behaviors of the AI system.

17
OVER TIME

Provide global controls.

Allow the user to globally customize what the AI system monitors and how it behaves.

18
OVER TIME

Notify users about changes.

Inform the user when the AI system adds or updates its capabilities.

🕒 OVER TIME



Microsoft

Where does this leave us?

So that we can sleep at night

- Consider the different ways that our software may impact others
- Consider the ways in which our software interacts with the political, social, and economic systems in which we and our users live
- Follow best practices, and actively push to improve them
- Encourage diversity in our development teams
- Engage in honest conversations with our co-workers and supervisors to explore possible ethical issues and their implications.

This lesson was about the harms that software can inflict

You should now be able to...

- Suggest some ways in which software can cause inadvertent harm or amplify inequities, with examples
- Explain why the software engineer has a powerful role to play in avoiding such harms.

Learning Objectives for this Lesson

You should now be able to:

- List some ways in which software can cause inadvertent harm or amplify inequities, with examples
- Explain some techniques that a software engineer can use to detect and mitigate these harms.

Learning Objectives for this Lesson

You should now be able to:

- Explain several of the meanings of “the public interest”.
- List some sources of ethical guidance for a software engineer.
- List several things that a software engineer can do to try to behave in an ethical manner.