Discuss your views on the role of the Computer Science professional organizations, such as the ACM and IEEE, in self-regulating the software industry?

Would there be advantages and disadvantages to instituting professional standards, including possible licensing, for the computing industry, and if so, by whom?

Advantages:

-guarantee of ability to employers

-possible higher wages to certified

-

Disadvantages:

-discrimination against those without, regardless of ability

-“However, algorithms are not just recommending, they are also being used to make big decisions about people’s lives, such as who gets loans, whose résumés are reviewed by humans for possible employment…”(transparancy)

-denial of jobs to capable workers

-

Construction worker certs? Nope.

Will still not prevent accidents.

Would such standards limit liability and/or improve the societal perception of software developers as more than just hackers or IT?

It might improve societal perception

-“hey software developers are an official thing now”

Limit liability? No. If accident happens ppl are still liable, just accidents should happen less.

How would these standards apply to the individuals, teams, and companies that produce the software and what consequences would there be for failure to follow them codified either in a Code of Ethics?

Individuals

-

Teams

-

Companies

-

What role do you feel the government should play in protecting the interests of both consumers and corporations through legislation given the complex interactions of software compared to other types of products?

Govt can only (and should) make companies liable for anything they release, and the company deals with punishing the employees responsible

Does the degree of accountability depend on the type of software, and who, if anyone, should be held liable for software failures?

No, degree of impact does.

Company bears responsibility for incident and may punish employee

-“Accountability rejects the common deflection of blame to an automated system by ensuring those who deploy an algorithm cannot eschew responsibility for its actions.”(transparancy)

Would releasing code for external review and/or open source mitigate liability for bugs, security vulnerabilities, etc.?

Getting peer reviewed would prevent some accidents, may split liability

-can distribute blame

-breaks trade secrets, potential security leak

dtdtudtfu

A major movement in the software community promotes the idea that software should be open source, i.e. “freely” available. One common license is the MIT version:

[https://mit-license.org](https://mit-license.org/)

Discuss the ramifications to software creation, use, and distribution if all software was governed by the MIT license.

// Creation would be stifled as most developers would become unmotivated to produce from the lack of a guaranteed return on their investments

But…

// Use - highly customized to user’s preferences, can be amended/upgraded by those that know what they’re doing

However…

// Distribution – becomes confusing from all the customized versions

* Easy to “contaminate” with malware and pretend it’s not there
* “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.”(MIT)

Present the advantages and disadvantages if such a policy would be legislated as the intellectual property statute for the entire software industry both from the standpoint of economic feasibility and societal benefit.

Econom Advan:

-users get software cheaper/free, which would garner more users

-publicity/popularity

Econom Disadv:

-basically no profit.

-even if can sell, can modify and resell. Flood of similar products competing, no winners

-thus, how pay employees?

Soc benif advan:

-Promotes users to customize and/or upgrade the software they acquire.

Soc benif disadv:

-Extreme hit to initial developers’ motivation

-“no” return on investment, makes developers unmotivated to produce since there’s no profit in it

-would be much more cost effective to just upgrade someone else’s work to suit the needs of your usage.

*Should* software developers and/or companies have a right to current intellectual property protections, e.g. copyrights and patents?

Should have the option, would keep ownership and profits, may hurt in publicity and/or copies sold

How do these protections extend to content on the web, e.g. social media sites, both for the creator of the content but also for those that may be impacted by the content?

*Should* an alternative IP category be created for software and/or digital media, and what protections would these IP category grant/deny to the owner, who would grant the protections, and who would enforce them?

Should different aspects of software, e.g. the conceptual framework (design and/or API interface), source code (implementation), object code (executables), graphical interface layout (visuals), etc. be protected by different categories or even at all?

How could this new category avoid the complicated cross licensing problems that plague the current patent system?