Software Disclaimers

Jarek Zylinski

Scenarios

- Friendly surgeon
- Aircraft made by …

Microsoft

Outline

- Scenarios
- What is a disclaimer
- Reasons we get bad software
- What can be done to discourage the making of bad software
- What is needed to make the software better
- Conclusion

What is a disclaimer?

- Definition
  - A repudiation or denial of responsibility or connection
  - Law: A renunciation of one's right or claim.

What is a disclaimer?

(An example of software disclaimer (legal))

IN NO EVENT, UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING, SHALL WSDOT, OR ANY PERSON BE LIABLE FOR ANY LOSS, EXPENSE OR DAMAGE, OF ANY TYPE OR NATURE ARISING OUT OF THE USE OF, OR INABILITY TO USE THIS SOFTWARE OR PROGRAM, INCLUDING, BUT NOT LIMITED TO, CLAIMS, SUITS OR CAUSES OF ACTION INVOLVING ALLEGED INFRINGEMENT OF COPYRIGHTS, PATENTS, TRADEMARKS, TRADE SECRETS, OR UNFAIR COMPETITION.

What is a disclaimer?

(An example of ‘straight forward’ software disclaimer)

Cosmotronic Software Unlimited Inc. does not warrant the functions contained in the program will meet your requirements or that the operation of the program will be uninterrupted or error-free.

However, Cosmotronic Software Unlimited Inc. warrants the diskette(s) on which the program is furnished to be of black color and square shape under normal use for a period of ninety (90) days from the date of purchase.

Note: In no event will Cosmotronic Software Unlimited Inc. or its distributors and their dealers be liable to you for any damages, including any lost profit, lost savings, lost patience or other incidental or consequential damage.

What is a disclaimer?

(Another example of ‘straight forward’ software disclaimer)

We don't claim Interactive EasyFlow is good for anything - if you think it is, great, but it's up to you to decide.

If Interactive EasyFlow doesn't work: tough. If you lose a million because Interactive EasyFlow messes up, it's you that's out of the million, not us.

If you don't like this disclaimer: tough. We reserve the right to do the absolute minimum provided by law, up to and including nothing.

This is basically the same disclaimer that comes with all software packages, but ours is in plain English and theirs is in legalese.

We didn't really want to include a disclaimer at all, but our lawyers insisted. We tried to ignore them, but they threatened us with the shark attack at which point we relented.

Reasons we get bad software

- companies not concerned with quality
  - profit driven
  - negligence
    - software released with known defect
      - Windows 3.1 shipped out with 5000 known bugs
      - Macintosh System 7.0 went out with thousands of bugs
      - percentage of bugs found prior to release → 95 (common)

Reasons we get bad software

- bad management (common myths)
  - ✗ “A general statement of objectives is sufficient to begin writing programs – we can fill in the details later.”
  - ✓ Poor up-front definitions is a major cause of failed software efforts.
  - ✗ “If we get behind schedule, we can add more programmers and catch up.”
  - ✓ Brooks Law: Adding more people to a late project makes it later.
  - ✗ “Once software is ‘working’ maintenance is minimal and can be handled on catch-as-you-can basis”
  - ✓ Budgeted maintenance costs → 10-15%

    Actual maintenance costs → 55-70%

Reasons we get bad software

- Lack of professionals
  - SE historically neglected by Professional Engineers societies
  - SE developing outside engineering community

What can be done to discourage the making of bad software

- lawsuits
  - high costs
  - not enough expert lawyers
  - many programmers/”software engineers” not held accountable because of the lack of professional standards
  - low probability of winning cases when companies hide behind carefully formulated software disclaimers and warranties
    - Family Drug Store vs. Gulf States Computer
- law
  - Article 2B of UCC (Uniform Commercial Code)
- software certification institutions

What is needed to make the software better

- more professionals
  - licensing

Profession	Licensing Required?	Duties Include/May Include
Barbers/ Hairdressers	Generally required	Cutting and styling of hair Applying of dyes to hair Use of proper hygiene
Manicurists	Generally required	Cutting and trimming of nails Grooming of nails Sanitization of equipment

What is needed to make the software better

Profession	Licensing Required?	Duties Include/May Include
Civil Engineers	Mandatory	Design and construction of bridges, roadways, industrial buildings and complexes, military complexes and transit systems, dams, etc..
Escort service employees	Mandatory where lawful (e.g. Nevada)	“Personal services” Regular medical testing “Safe” practices

What is needed to make the software better

Profession	**Licensing Required?**	**Duties Include/May Include**
Software Engineers	No requirement	Design and construction of: -medical diagnostic equipment -medical dosing systems -air traffic control systems -strategic nuclear weapon systems -automotive computer control systems -banking and financial systems
Programmers		Implementation of all items listed above.

What is needed to make the software better

- get more professionalism out of professionals
  - ethics
    - accept individual responsibility (Therac 25)
    - solve the real problem (Family Drug Store vs. Gulf States Computer)
    - be honest about capability (London Ambulance System)

Conclusion

taught by professionals

+

right

curriculum

professional integrity

+

competence

+

responsibility

Education + Ethics =Soft. Eng. =Less Disclaimers

QUESTIONS?

References

[1] The American Heritage® Dictionary of the English Language, Fourth Edition Copyright © 2000 by Houghton Mifflin Company.

[2] Bridge Software, <http://www.wsdot.wa.gov/eesc/bridge/software/index.cfm?fuseaction=disclaimer>

[3] Sam Nitzberg, Ethics in Military and Civilian Software Development, <http://www.iamsam.com/papers/afcea_1999/afcea_ethics_submission.html>

[4] Gotterbarn, Donald, “Ethics Articles”, <http://www-cs.etsu-tn.edu/gotterbarn/articles.htm>

[5] Kaner, Cem, Bad Software. <http://www.badsoftware.com/articles.htm>

[6] Parnas, David, Professional Responsibilities of Software Engineers, 1994

[7] Software Engineering Myths, <http://www.comp.nus.edu.sg/~cs4211/notes/L.1.ppt>

[8] Forester, Tom, and Perry Morrison. Computer Ethics: Cautionary Tales and Ethical Dilemmas in Computing. 1990.

[9] Voas, Jeffrey. Software Certification Laboratories: To Be Or Not To Be Liable?. <http://www.stsc.hill.af.mil/crosstalk/1998/04/certification.pdf>