Ethics in RE

Evan Citulsky
July 20th 2017
Contents

• Introduction
• A Brief Look at Ethics
• Software Engineering Code of Ethics
• Three Ways RE Can Be Unethical
• RE Ethical Dilemmas
• Value Stories
• Conclusion
Ethics

• Morals
  • Right, wrong, good, bad, evil to an individual

• Ethics
  • A subset of morals in which a society agrees upon
  • Code of conduct in a society
  • Can vary in different parts of the world
    • Eastern and Western Views
Why Ethics

• Software systems
  • Surround us in our daily lives
  • Support human lives
  • Carry out decisions on a persons behalf
  • Provide a blind truth of information (We trust what the computer spits back out to us)
Ethics and Software Systems

• Software Systems CANNOT poses morals
  • This would imply true AI

• It may be possible to program a code of ethics
  • Either built on set of agreed upon societal morals, or the programmer’s morals
    • What morals to use?
  • How to handle ethical dilemmas?
    • More on this coming up
Software Engineering Code of Ethics (SECE)

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

2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest.

3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
Software Engineering Code of Ethics (SECE)

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.
Requirements Engineering

• RE is the process of discovering, defining, documenting, and maintaining requirements for a system

• An important step in software engineering that is sometimes rushed or completely skipped

• Without RE, can a developer guarantee that a system will behave ethically?

• SECE can be applied to the RE process
Three Ways RE Can Be Unethical

• Toxic Requirements
• Improper RE
• Deception
Toxic Requirements

- Software Requirements and the Ethics of Software Engineering (Caper Jones)
- Requirements
  - That can have long term problems
  - That can have hazardous unknown effects
  - That have potential to cause the system to crash
    - May be a “one-in-a-million” scenario
- All requirements should be investigated for these problems
Toxic Requirements Examples

• Famous Y2K bug
  • Original requirement (1960’s) of date format was mm/dd/yy to save on memory
    • Supported only the current century
  • Did not change the requirement when memory became cheap
  • Lead to wide spread panic in the late 90’s, with a rush to repair software
Toxic Requirements Examples

• Los Angeles Air Traffic Control system (2004)
  • Without warning crashed while handling approximately 400 planes
    • Lost main voice communication system (VSCS)
    • Backup system crashed within minutes
  • What went wrong?
    • Timer inside VSCS control system that starts counting down from $2^{32}$ and stops once it hits zero. Enough to last about 50 days
    • FAA procedure to reset system every 30 days
      • Did not happen
    • Was this human error?
Improper RE

• Requirements Engineering may be skipped or rushed process due to:
  • Time
  • Management
  • Laziness

• Consequences of improper RE can lead to devastating situations caused by
  • Not knowing “side effects” of requirements
  • Not verifying requirements
  • Not implementing requirements
  • And the list goes on...
Improper RE Example

• Radiation Treatment
  • Computerized treatment planning system for radiation treatments
    • Responsible for calculating shielding blocks during radiation
    • Limitation of being able to input a maximum of four shielding blocks
  • Engineers discovered that shielding blocks could be inputted in groups with some changes to the software
  • August 2000 - Update applied
  • March 2001 - Error discovered, 28 patients affected
    • New input method results in incorrect radiation dosage calculation, patients experience overexposure
  • August 2003 - 17 patients deceased
Deception

• Software developers can easily choose to be unethical by purposefully deceiving the user or public
  • Malware are common examples of such software
• Stakeholders may set out requirements that are designed to deceive the public
Deception Example

Deception Example

• Volkswagen Turbo Diesel Injection (TDI) 2009-2015 models
  • Requirement for TDI engine to expel no more than government regulated emissions level.
  • TDI engine was NOT designed to satisfy this requirement
    • Improper RE performed?
    • Could not meet requirement?
  • Software designed with the following requirements
    • Detect when an emissions test is being performed
    • Turn ON all emissions control features when a test is detected
    • Turn OFF some emissions control features when not testing to improve drivability, power and “funness”
RE Ethical Dilemmas

• What does a RE analyst do when presented with a requirement that leads to an ethical dilemma??
• Ethical dilemmas can also occur as functional requirements, where the software must make the ethical decision
• Situations
  • When and where to store log files
  • How much protection is enough
  • Choice between two or more human lives
  • Questionably ethical design purpose
RE Ethical Dilemmas Examples

Self Driving Cars

• Consider the following scenario
  • Autonomous Vehicle approaches a group of pedestrians at a crosswalk at a speed in which it cannot stop.
  • The vehicle can either
    1. Crash into the pedestrians, killing them
    2. Crash into a wall, killing the occupants of the vehicle
  • The software system must decide which lives of sacrifice
RE Ethical Dilemmas Examples

Autonomous Killer Robots (AKR)

- The concept of AKRs is to replace humans on the front lines of battle with autonomous machines.
  - Must have the capability to identify *friendly*, *enemy* and *neutral* targets
- At first, the idea of AKRs doesn’t seem that bad...
  - Save “good” human lives by sacrificing machines
  - Save “good” human lives by killing only “evil” humans
RE Ethical Dilemmas Examples

Autonomous Killer Robots (AKR)

• Problems?
  • The system designer defines what constitutes as a *friendly, enemy* or *neutral* target
  • An emotional disconnect between AKR operators and the targets that are destroyed
  • Difficulty maintaining transparent accountability
  • AKRs vs AKRs?
Value Stories

• How can we incorporate Ethics into the RE process?
• Putting human values into RE – Detweiler and Harbers
  • Use existing values elicitation techniques to identify stakeholder values
  • Translate those values into a suitable form to be used for later RE processes
Conclusions & Future Work

• Ethics must be taken into account in all phases of the RE process
  • RE is the best method to discover unethical properties of a system
• Toxic requirements should be avoided or eliminated
• Deceptive requirements should also be avoided or eliminated
• Developing software systems that make ethical decision is a very active field of research
• New processes are being developed to incorporate ethics into RE
Thank You!
References


