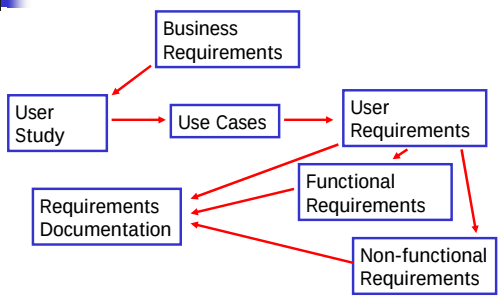
CSE 403
Lecture 14

Safety and Security Requirements

User requirements



Non-functional requirements

- Requirements beyond user interaction with the system
- Kulak and Guiney
  - Availability, cost of ownership, maintainability, data integrity, extensibility, installability, reuse, operability, performance, portability, quality, robustness, scalability

Non-functionality requirements

- Wiegers
  - Performance requirements
  - Safety requirements
  - Security requirements
  - Software quality attributes

Software Safety

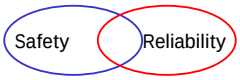
- Safety critical applications
  - Where bugs can kill
- Famous cases
  - Therac-25 radiation therapy machine
  - US Air traffic control which failed in UK
    - Reflected map on Greenwich Median
  - US Aviation software failed in Israel
    - Encountered negative altitudes over Dead Sea

Safety critical systems

- Very high cost of failure
- Software component of a large system
  - e.g. nuclear reactor
- Characteristics of software lead to failures
- Safety requirements
  - Low probability of failure (risk analysis)
  - Understood failure modes

Software Safety

- Safety vs. Reliability



- System hazard analysis
  - High risk tasks
  - Safety critical operator errors
- Design of Human-Machine Interface

Specifying safety requirements

- Component reliability
- Fail to safe state
- Formal guarantees or validation
- Positive measures
- Decouple safety critical components
  - Safety kernel
- Redundancy

UI for Safety

- System failures generally complex with humans involved
- Hard to clarify degree of user error
- Very complicated design space
  - Design for very boring environment
  - Design for crisis operation
- Take into account human cognitive abilities

Security requirements

- Applications are run in a hostile world
- Application compromise vs. system compromise
- Example requirements
  - Only authenticated users can change data
  - Application can change security permissions or execute programs
  - Malicious user cannot crash system with bad data
- Threat analysis

Threat modeling

- The STRIDE Threat Model
  - Spoofing identity
  - Tampering with data
  - Repudiation
    - Allow users to deny having performed actions
  - Information disclosure
  - Denial of service
  - Elevation of privilege

Approaches to security requirements

- Security audit / validation
- Implementation limitations
  - No use of ‘gets’
  - No use of unsafe calls on user input
- Restricted operation modes
- Safe defaults

Security requirements for multiplayer games

- Cheating ruins game play (and consequently market)
- Threats
  - Players introducing counterfeit weapons
  - Sending packet of death across network
  - Using profiling tools to detect areas of activity in dungeons

Threat analysis for Classroom Presenter

Classroom Presenter

Useful references

- Writing Secure Code, Michael Howard and David LeBlanc (2nd Edition)
  - Good book, but strongly oriented towards Windows
- Safeware: System Safety and Computers, Nancy Leveson
  - Defines the field of software safety