Safety Critical Systems

Adam Young

Overview

- What is a Safety Critical System?
- Design of a Safety Critical System.
  - Risk Analysis.
  - Verification.
- Industry Standards.

What is a Safety Critical System

- System considered safety critical if failure causes:
  - Injury or death to a person.
  - Damage to or loss of equipment.
  - Damage to the environment.

Examples


A 591204
PHYSIO-CONTROL
LIFEPAK 9P MONITOR • DEFIBRILLATOR • PACEMAKER
PACER
LEAD SELECT
ON
1
RATE
ECG SIZE
ENERGY SELECT
2
HR ALARM
CHARGE
3
CODE SUMMARY
RECORD
SYNC
STERNUM
APEX

Case Study: Therac 25

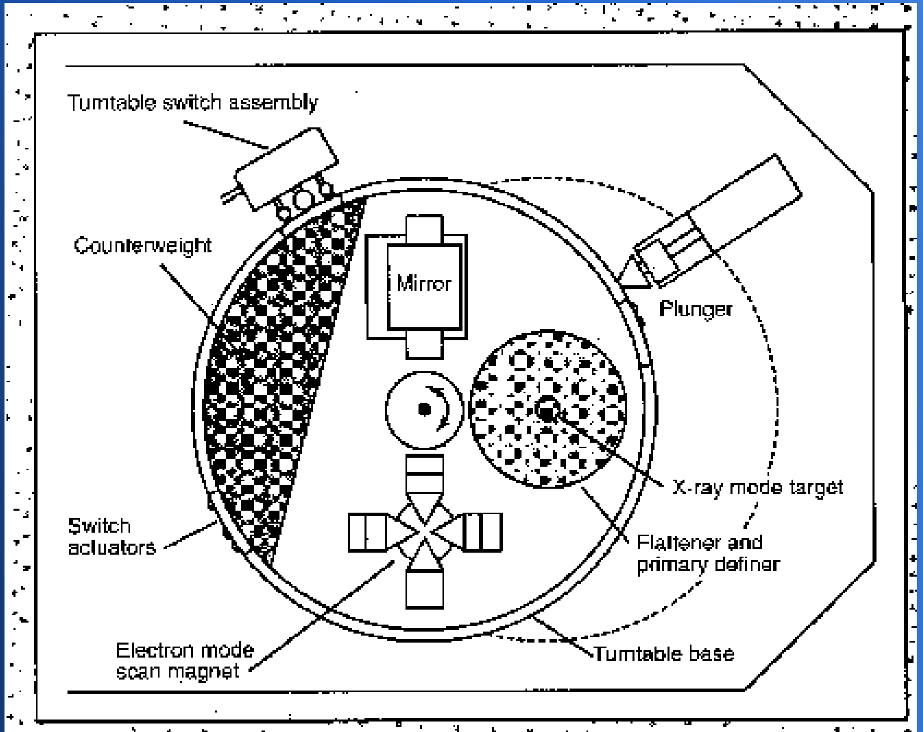
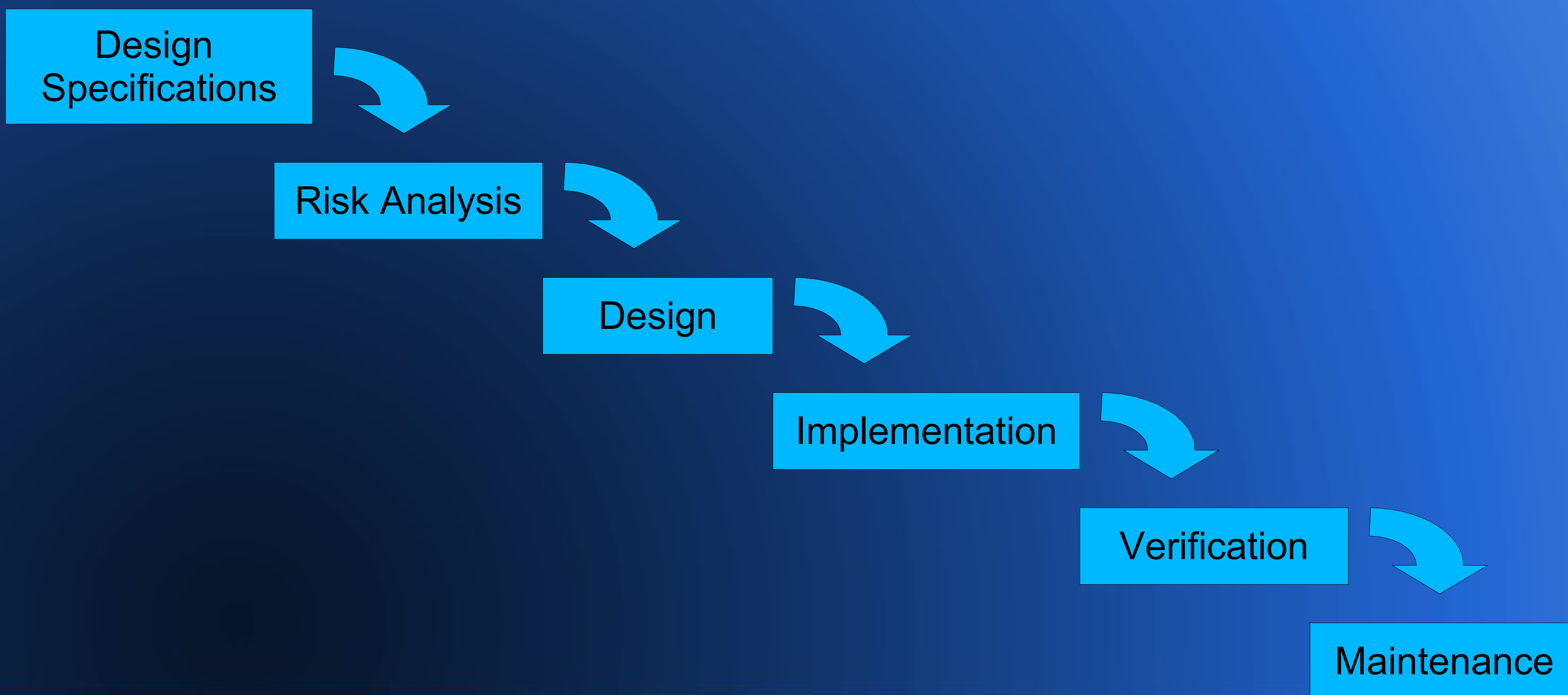


Figure B. Upper turntable assembly.

Design of Safety Critical Systems

- Design process similar to waterfall process.



Design: Requirements Specifications

- Requirements specifications methods are still the same.
- Important for requirements specifications to be:
  - Concentrated on functionality not safety in this stage.
  - Well documented.

Design: Risk Analysis

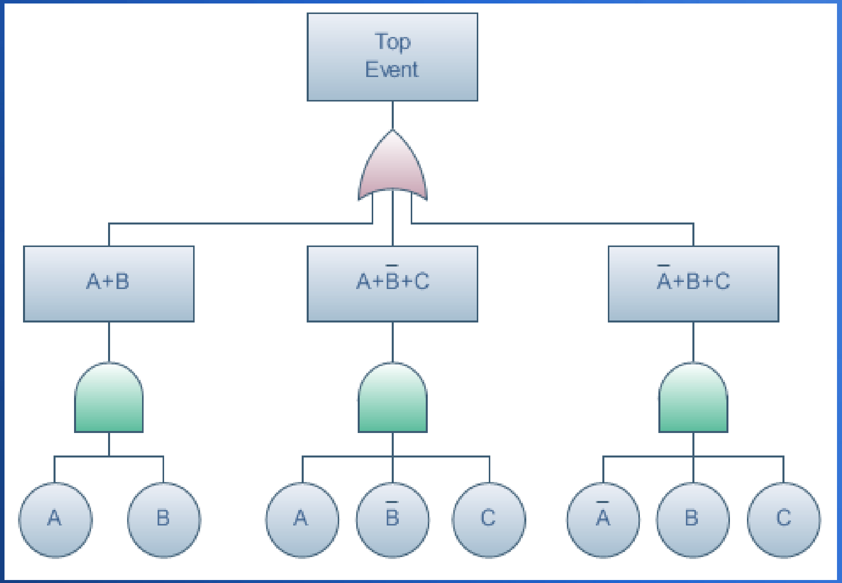
- Risk analysis is one of the most important steps.
- Goal is to find hazards in the system and determine their severity.
- Hazards produce their own set of design specifications that must be followed.

Design: Risk Analysis

- Failure Modes and Effects Analysis.
  - Uses block diagrams of the system.
  - Determines failure mode of each block.
  - Determines effect of each failure.
  - Failures prioritized by their consequences.
  - Provides insight to safety, cost, performance, quality and reliability.
  - Formal templates for the process exist.

Design: Risk Analysis

- Fault Tree Analysis.
  - Each individual failure has a tree model.
  - Each possible cause of failure composes tree.
  - Hard to use for software.



Design: Risk Analysis

- Determining all hazards in a complex system is hard.
- Software brings a higher level of complexity.
- Reliable vs. Trustworthy system.

Design: Risk Analysis

- Reliability Regimes:
  - Fail-Operational.
  - Fail-Safe.
  - Fail-Secure.
  - Fail-Passive.
  - Fail-Tolerant.

Design/Implementation

- Some design done during risk analysis.
- Common proven methods often used:
  - Top down design, bottom up implementation . (v model)
  - Spiral Method.

Implementation: Limitations

- Limitations/guidelines placed on programming practice to reduce risk of fault.
- Examples:
  - Restricted used of recursion.
  - Recommendations for nesting depth.
  - Pointers to functions can not be used.
  - Many more (too many to list).

Verification

- Large scale and time consuming aspect of a safety critical system.
- Often carried out by a dedicated team.
- Quality of verification dependent on quality of documentation (both specs and risks).
- Actual method of verification is project specific.
- Many companies provide dedicated verification software specifically for safety critical systems.

Verification

- Expensive and adds greatly to the cost of a safety critical software system.
- Software usually in the range of 50-70% of the project budget for a safety critical system.
- Fully delivered safety critical system code can cost $90 a line.

Maintenance

- Extent of maintenance needed is determined by reliability regime used.
- Standards provide guidelines

  for maintenance of systems.

Industry Standards

- DO-178 (Aerospace Industry)
- MIRSA C (Automobile Industry)
  - JSF++
  - MISRA C++
- IEEE/EIA 12207

Conclusion

- Design of a safety critical system is a complex procedure.
- Software only adds to complication
- Planning and thorough documentation are key.
- Many standards provide guidelines for projects.

References

- http://en.wikipedia.org/wiki/Safety_engineering
- http://en.wikipedia.org/wiki/Life-critical_system
- http://www.cs.virginia.edu/papers/safecomp.97.pdf
- http://www.adaworld.com/pdfs/criticalsafetyada.pdf
- http://en.wikipedia.org/wiki/Independent_software_verification_and_validation
- http://www.misra-cpp.org/
- www.ldra.com/nologindownload.asp?id=134

Questions?