Software Development and Risk Management in the Safety Critical Medical Device Domain

Christin Lindholm

Department of Computer Science
LTH, Lund University
Background

• Software in medical devices
  – Growing economic sector
  – Increasing functionality
  – Regulated by laws and standards
  – Changed law in Sweden 2010
    • Standalone software – medical device
  – Changed standard 2012
    • IEC 60601-1 – medical electrical equipment

“Be validated according to state of the art in the domain”

“Focus on concept of risk”
Software – Medical devices

– Software for medical devices: software that can manage, control or regulate.

– Process software: used for the manufacture, inspection or testing of medical products.

– Medical software: software that provides medical functions independent of the hardware system
Risk management

Risk management
- Procedure for resolving risk
- Closely connected to safety critical domains
- Regulated by law

Process:

1. Risk identification
2. Risk analysis
3. Risk planning
4. Risk monitoring
Ongoing case study

Input to new **software** risk management process development of a medical device
Patient monitor system

1. Pressure sensor – patient skull

2. Patient monitor – to sensor

3. Bedside monitor (new device)
   – PC – Windows
   – Palcom application – Java
   – GUI
Participants

- Intended users (physicians, nurses)
- The development organisation (medical device expert, risk analysis supervisor)
- The researcher (process experts, technical experts)

Step 1
The patient's blood pressure rises above the alarm limit.

Step 2
The alarm is activated after x minutes.

Step 3
The user pauses the alarm and initiates treatment.
Predefined 4 graded scales – Swedish national board of health and welfare

Probability, P
Severity, S
Risk value = P x S

Detectability:
if the fault (hazard) always could be detected before a severe situation occurred

sometimes

never
Used process

• Emphasis on users
  – User dominance
    • Severity assessment
    • Detectability assessment

• Scenario composition impact on outcome
  – Catch unrealistic behaviour

• Detectability
  – Not estimated
    • Imprecise scale
    • Concept not well understood

• System boundary
  – Too narrow
    • Catastrophic consequences -> non existing
Suggestions for improved process

• Users and developers work separately

• Time slot – opinion from all participants

• Scenarios in context
  – Presumptions
    • Working situation

• Detectability
  – Remove detectability from process
  – New scale
    • Direct observable
    • Indirect observable
    • Unobservable
Design improved process

Interesting tracks

- Checklist
- Usability tests – risk management
  Part of the process??