
Requirements Engineering for Safety-critical Systems

Discusses an approach to
requirements engineering for deriving
safety and dependability requirements

November 2004

©Ian Sommerville 2004

Slide 1

Objective

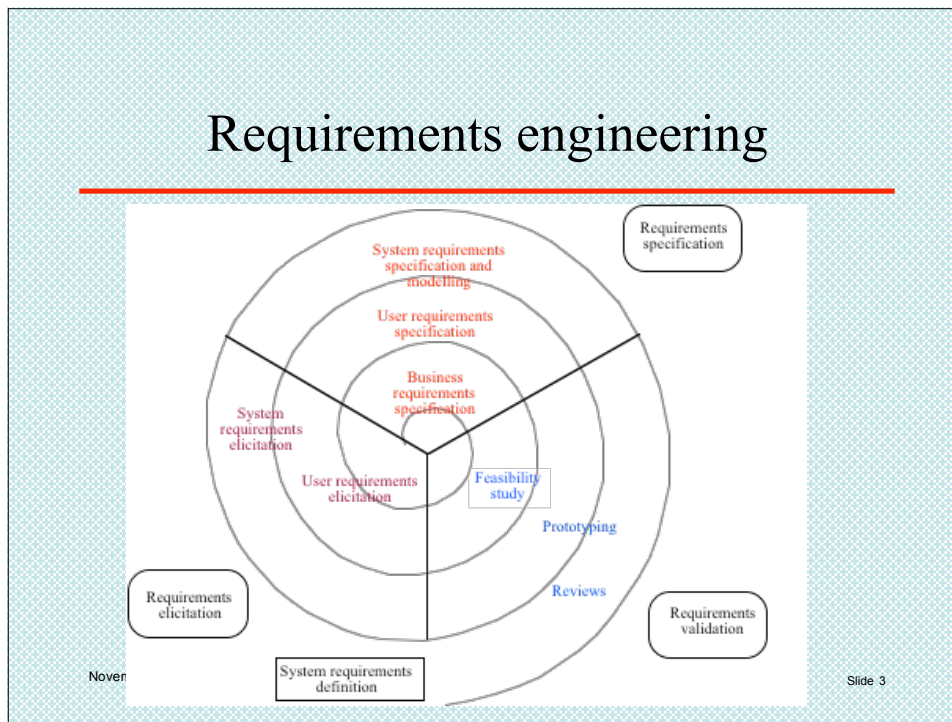
- To introduce the issues involved in safety requirements engineering
 - Using an example, to discuss risk-driven approaches to eliciting safety requirements
 - To discuss how the scope of safety-critical systems is broadening and to introduce an approach to requirements elicitation that integrates safety requirements engineering more closely with other requirements engineering activities

November 2004

©Ian Sommerville 2004

Slide 2

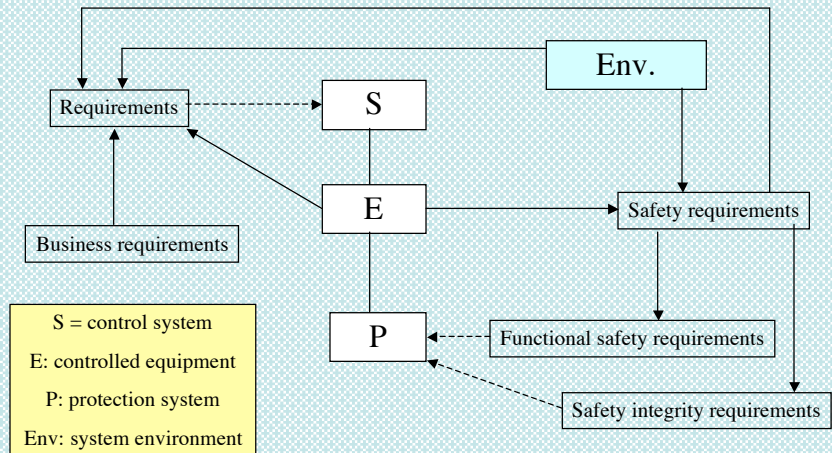
Requirements engineering



Business requirements

- Requirements that define how the business intends to benefit from the system
- Requirements that define how the risks to the business from the system and other sources may be minimised
- Safety requirements should therefore be thought of as **BUSINESS REQUIREMENTS** rather than system requirements

Safety requirements (Protection system)



November 2004

©lan Sommerville 2004

Slide 5

Safety requirements

- Business requirements that identify risks to the system, undesirable conditions to be avoided or fundamental safety features that must be included
 - At the same level as other business requirements for a system so should be expressed in an abstract way
- Specific functional requirements that define how risks are to be reduced and undesirable conditions avoided
 - Equivalent to system requirements - more detailed and specific
- The more detailed safety requirements are generated from the abstract safety requirements

November 2004

©lan Sommerville 2004

Slide 6

Accident trajectories

- Accidents or critical incidents rarely have a single cause. Generally, they arise because several events occur simultaneously
 - Loss of data in a critical system
 - User mistypes command and instructs data to be deleted
 - System does not check or ask for confirmation of destructive action
 - No backup of data available
- An accident trajectory is a sequence of undesirable events that coincide in time. It may be initiated by:
 - Human error
 - Hardware or software failure
 - Inappropriate system behaviour as a consequence of a wrong requirement

November 2004

©Ian Sommerville 2004

Slide 7

Warsaw airbus accident

- In September 1993, an Airbus 320 was landing in bad weather. The braking system failed to deploy and the aircraft overshot the runway. 2 fatalities, many injuries
- Active event:
 - Landing on one wheel that was aquaplaning
- Accident trajectory
 - Failure of control software to recognise that aircraft was on the ground
 - Reverse thrust deployment not enabled
 - Braking system failed to stop aircraft in time
 - **Runway overshoot**

November 2004

©Ian Sommerville 2004

Slide 8

Defensive layers

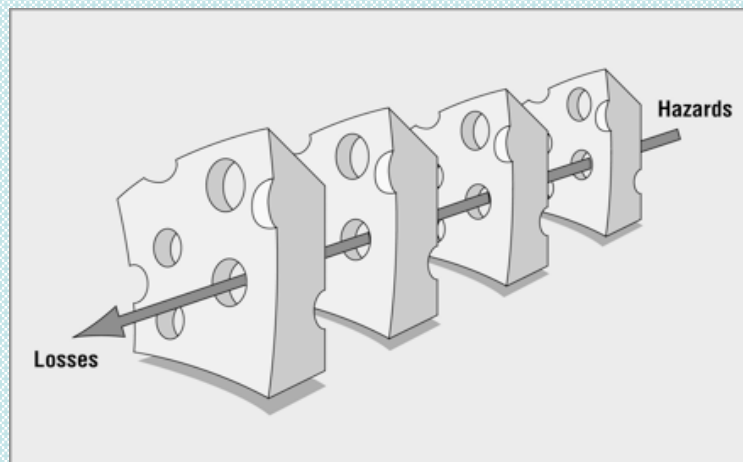
- Defensive layers (protection) in a system are intended to intercept accident trajectories
- Safety-critical systems should have many defensive layers:
 - some are **engineered** - alarms, physical barriers, automatic shutdowns,
 - others rely on **people** - surgeons, anesthetists, pilots, control room operators,
 - and others depend on **procedures** and administrative controls.
- In an ideal world, each defensive layer would be intact so that (assuming correct operation), the accident trajectory would be blocked
- In reality, these layers are more like slices of Swiss cheese, having many holes- although, unlike in the cheese, these holes are continually opening, shutting, and shifting their location.

November 2004

©Ian Sommerville 2004

Slide 9

Reason's Swiss Cheese Model



November 2004

©Ian Sommerville 2004

Slide 10

Active and latent failures

- Active failures
 - Active failures are unsafe conditions that initiate an accident trajectory (human error, hardware/software failure, procedural violations, etc.).
 - Active failures have a direct and often short-lived effect on the integrity of the defenses.
- Latent conditions
 - Fundamental vulnerabilities in one or more layers of the socio-technical system such as system faults, system and process misfit, alarm overload, inadequate maintenance, etc.
 - Latent conditions may lie dormant within the system for many years before they combine with active failures and local triggers to create an accident opportunity.

November 2004

©Ian Sommerville 2004

Slide 11

Incident reduction

- Reduce the number of latent conditions in the different layers of the system (plug the holes)
 - If the number of faults in a software system is reduced, this increases the strength of the defensive layer
 - However, this technical approach ON ITS OWN cannot be completely effective as it is practically impossible to reduce the number of latent conditions in the system to zero
- Increase the number of defensive layers and hence reduce the probability of an accident trajectory occurring
- This is the classic approach to safety engineering - introduce high-quality defences to protect against conditions that compromise system safety

November 2004

©Ian Sommerville 2004

Slide 12

Active failure reduction

- A complementary approach is to tackle active failure reduction ie reduce the likelihood that an active failure will arise to initiate an accident trajectory
- This is ONLY possible when the development of safety requirements is integrated with the development of system requirements
- SAFE requirements are requirements that are intended to reduce active failures in a system

November 2004

©Ian Sommerville 2004

Slide 13

Safe requirements

- Safe requirements
 - Active failure avoidance
 - Active failure detection
 - Active failure recovery
- Safe requirements cannot be divorced from the design of the system
 - The operator shall input a value between 1 and 10 (unsafe)
 - The operator shall input a value between 1 and 10 by choosing the value from a menu whose options are single values from 1 to 10 (safe)

November 2004

©Ian Sommerville 2004

Slide 14

Integrated safety engineering

- The model of safety requirements engineering developed for protection systems (IEC 61508) is not a general model for safety critical software
- It tends to focus on defences in the system (dealing with failures) rather than avoiding the initial conditions that result in a failure
- An integrated approach, based on safety requirements as business requirements, is likely to be more cost-effective

November 2004

©Ian Sommerville 2004

Slide 15

A personal insulin pump

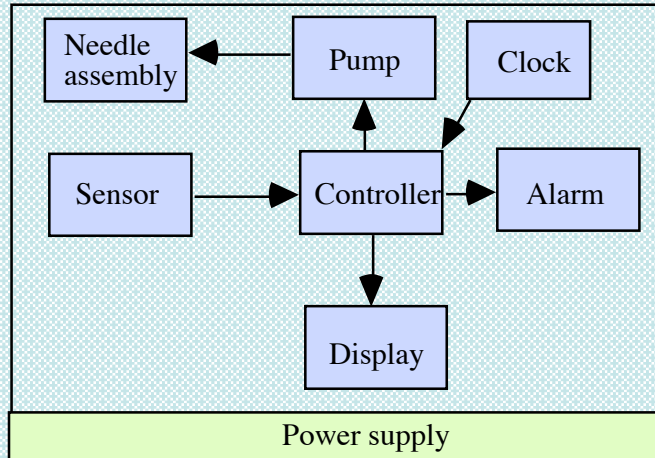
- A personal insulin pump is an external device that mimics the function of the pancreas
- It uses an embedded sensor to measure the blood sugar level at periodic intervals and then injects insulin to maintain the blood sugar at a 'normal' level.
- I will draw on this example at various points in the tutorial to illustrate aspects of safety requirements specification

November 2004

©Ian Sommerville 2004

Slide 16

Insulin pump components



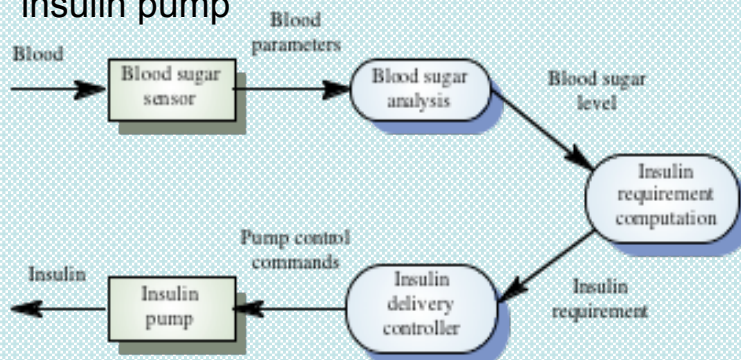
November 2004

©Ian Sommerville 2004

Slide 17

Insulin delivery system

- Data flow model of software-controlled insulin pump



November 2004

©Ian Sommerville 2004

Slide 18

Concept of operation

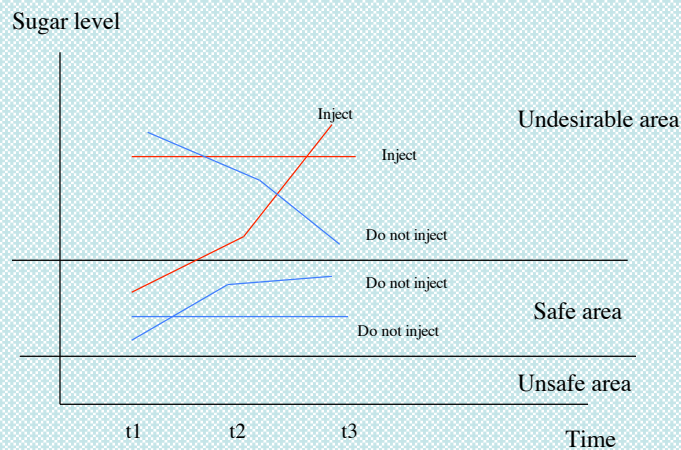
- Using readings from the embedded sensor, the system automatically measures the level of glucose in the sufferer's body
- Consecutive readings are compared and, if they indicate that the level of glucose is rising (see next slide) then insulin is injected to counteract this rise
- The ideal situation is a consistent level of sugar that is within some 'safe' band

November 2004

©Ian Sommerville 2004

Slide 19

Glucose measurements



November 2004

©Ian Sommerville 2004

Slide 20

Scenario 1 - reading in unsafe area

- If the blood sugar reading is below some minimum level this is potentially dangerous
- Insulin is never delivered if a reading is below this level
- An audible alarm is sounded and a warning message displayed

November 2004

©Ian Sommerville 2004

Slide 21

Scenario 2 - reading in safe area

- Do not deliver insulin if the sugar level is stable or falling
- Do not deliver insulin if the sugar level is increasing but the rate of increase is decreasing (i.e. the curve is flattening)
- Deliver insulin if the sugar level is increasing and the rate is stable or increasing
- Amount of insulin delivered depends on the rate of increase of blood sugar

November 2004

©Ian Sommerville 2004

Slide 22

Scenario 3 - reading in undesirable area

- Do not deliver insulin if the sugar level is decreasing and the rate of decrease is stable or increasing
- Deliver a fixed quantity of insulin if the sugar level is stable or if the sugar level is decreasing and the rate of decrease is decreasing (i.e. the curve is flattening in the high zone)
- Deliver insulin if the sugar level is increasing. The amount delivered depends on the rate of increase

November 2004

©Ian Sommerville 2004

Slide 23

Safety requirements

- We shall explore in the next presentation how a risk-based approach may be used to derive safety requirements for the insulin pump control system

November 2004

©Ian Sommerville 2004

Slide 24