



ASSESSMENT OF RISKS INTRODUCED TO SAFETY CRITICAL SOFTWARE BY AGILE PRACTICES - A SOFTWARE ENGINEER'S PERSPECTIVE

Janusz Górski, Katarzyna Łukasiewicz

Department of Software Engineering,

Faculty of Electronics, Telecommunications and Informatics,

Gdańsk University of Technology

10-13 September 2012, KKIO, Kraków, Poland

The problem

- Safety-critical software
 - demand for lowering the development cost and shortening time to market
 - Agile vs. 'heavyweight' practices
 - obligation to assure safety and demonstrate that safety level is acceptable
 - Conformance with standards
 - Examples:



Standards For Validation Of Automated Systems



IEC 62304 (*Medical device software – software lifecycle processes*)





ISO 13485 (*Medical devices – quality management systems*)

- Safety (assurance) cases
 - e.g. FDA recommendation for medical devices
- Experience: agile processes reduce development cost and time to market
- **Question: How to introduce agile practices to safety critical software development while still maintaining safety assurance and safety demonstration at acceptable level?**

Objective

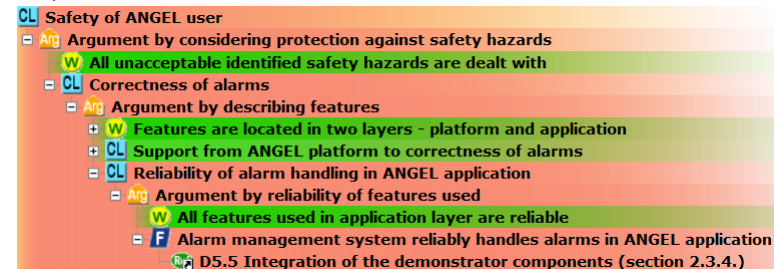
- To develop a method supporting introduction of agile practices to safety-critical projects with the intention to lower the development effort and to shorten the delivery time while maintaining acceptable level of safety assurance in accordance with the norms and the standards applicable for a given product

Approach (1)

- Identification of safety criteria
 - Standards, guidelines, recommendations
- Representing safety criteria as argument patterns for demonstrating conformance
 - TRUST-IT methodology  and NOR-STA services 
- Building a knowledge base of SW development practices
 - Inventory of agile practices (initial focus on SCRUM and XP) and plan-driven practices
 - Impact on effort and time
 - Criteria for safety risk analysis of the practices
 - Hazard identification
 - Hazard analysis
 - Particular interest in hazard scenarios that are ‘anchored’ in software development practices (Fault Trees for representing such scenarios)

Approach (2)

- Proposing a method supporting selection of SW development practices in a way that still maintains the possibility to develop a complete argument demonstrating conformance with the safety criteria
 - ▣ Identification of the necessary evidence to be collected to support the argument
 - ▣ Assessment of the 'strength' of the argument (argument appraisal mechanism based on Dempster-Shaffer theory of evidence)
- Developing a tool support for the method
- Validation of the proposed method
 - ▣ Domain selection - Medical devices
 - ▣ Case studies
 - ▣ Insuline pump, ..
 - ▣ Software developers (agile, plan-driven)
 - ▣ Software development companies
 - ▣ Medical Devices subgroup of EWICS (European Workshop on Industrial Computer Systems)



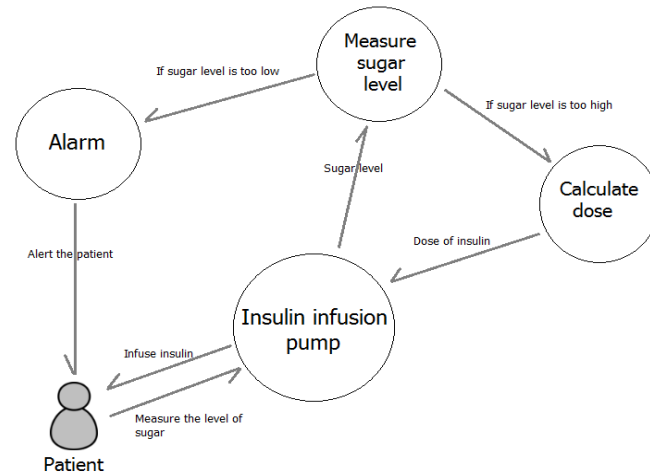
Case study – the focus

- Identification of safety criteria
 - Standards, guidelines, recommendations
- Representing safety criteria in argument patterns for demonstrating conformance
 - TRUST-IT methodology and NOR-STA services
- Developing a knowledge base of SW development practices
 - Inventory of agile practices (initial focus on SCRUM and XP) and plan-driven practices
 - Impact on effort, time
 - Criteria for safety risk analysis of the practices
 - Hazard identification
 - Hazard analysis
 - Particular interest in hazard scenarios that are ‘anchored’ in software development practices (Fault Trees for representing such scenarios)

Case study – the objective and the setting

- Objective
 - To investigate safety risks related to agile practices (Scrum and eXtreme Programming) from the programmers' perspective
- Setting
 - Carried out from March to the end of May 2012 in a group of 31 students (the last year of their master course, specializing in software engineering, 67% already involved in real projects)
 - Work in groups of 2 and 3, forming 12 project groups
 - 6 groups focusing on XP and 6 groups on Scrum
 - Each group was given
 - A specification of an insulin infusion pump
 - A short description of a fictional company called MediSoft which produces software for such pump
 - supporting documents and tools

Case study – the insuline pump



- An insulin pump is a device for patients with diabetes who need to control their blood sugar level by administering insulin. The pump is attached to the patient's body along with a small container filled with insulin. At the correct times, small and precisely calculated amounts of insulin are released from the container into the patient's bloodstream. It can help to keep blood glucose levels steady between meals and during sleep.

Case study – tasks: hazards

- **A. Preparing a list of hazards and hazard scenarios**
 - Identify patient safety hazards following the guidelines adapted for this case
 - Link identified hazards with SW practices by FTA (Fault Tree Analysis) and represent the trees as MS Visio diagrams

Case study – tasks: risk

■ B. Conducting risk assessment

- Having project development phases and tasks (according to the given methodology) represented in the Designsafe tool (*Design Safety Engineerig, inc.*)
 - assess risks and represent the results in the Designsafe tool

Case study – tasks: risk reduction

- **C. Propose a list of additional practices** focusing on reduction of the identified risks
 - A template for the additional practices description

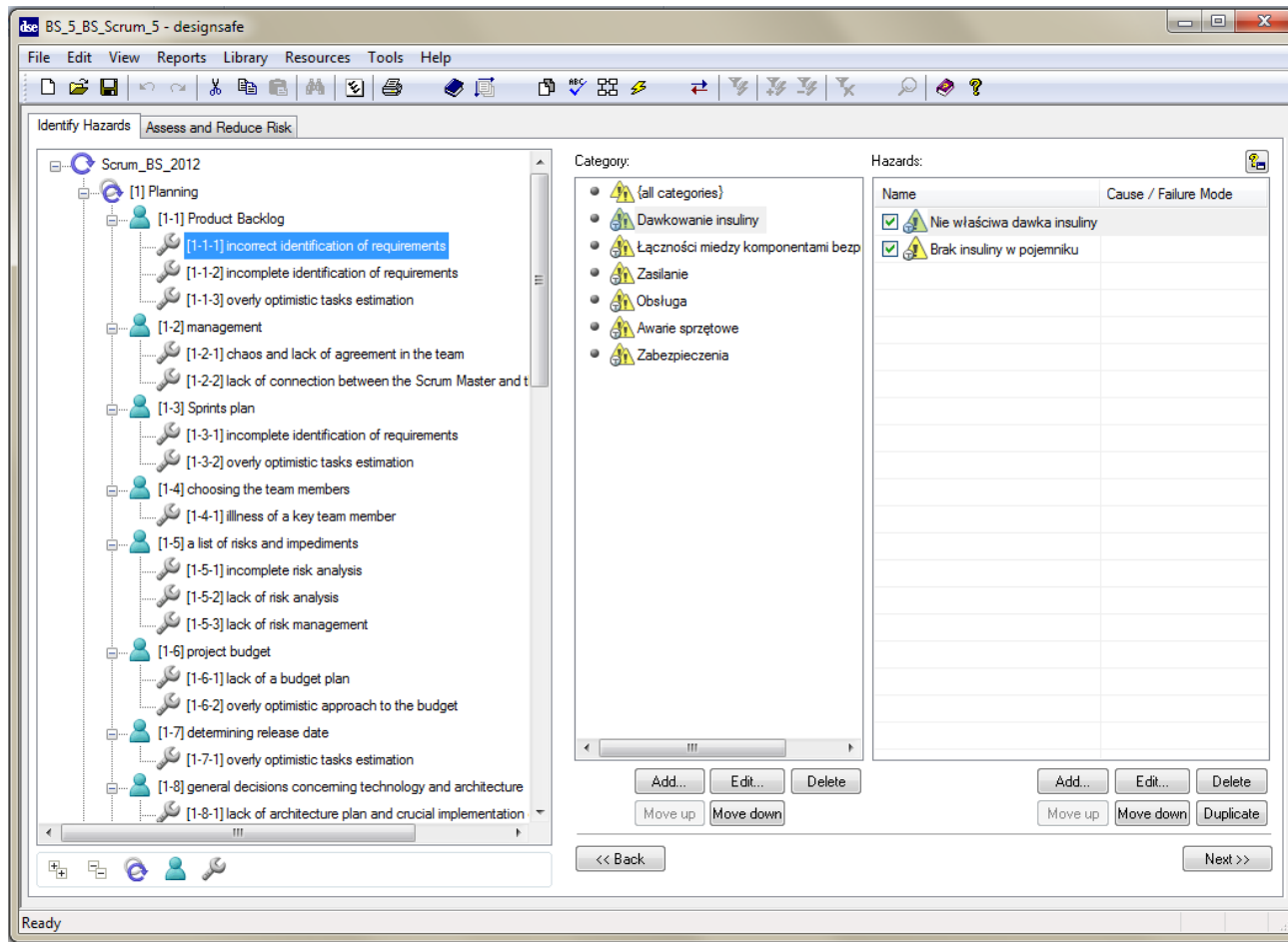
No.	<i>Name of the practice</i>
Description	<i>A description of the proposed practice – what activities it includes, how should they be performed, by whom, at what stages of project.</i>
Related hazards	<i>Which hazards (from your risk analysis) the practice is expected to have influence on.</i>
Expected influence	<i>What is the expected result of implementing the practice, in what way it could reduce the risk, to what extent.</i>
Agility/discipline balance	<i>How the practice will affect the agility of the methodology ie. will it require some alterations in project roles or additional project stages etc.</i>

The case study - results

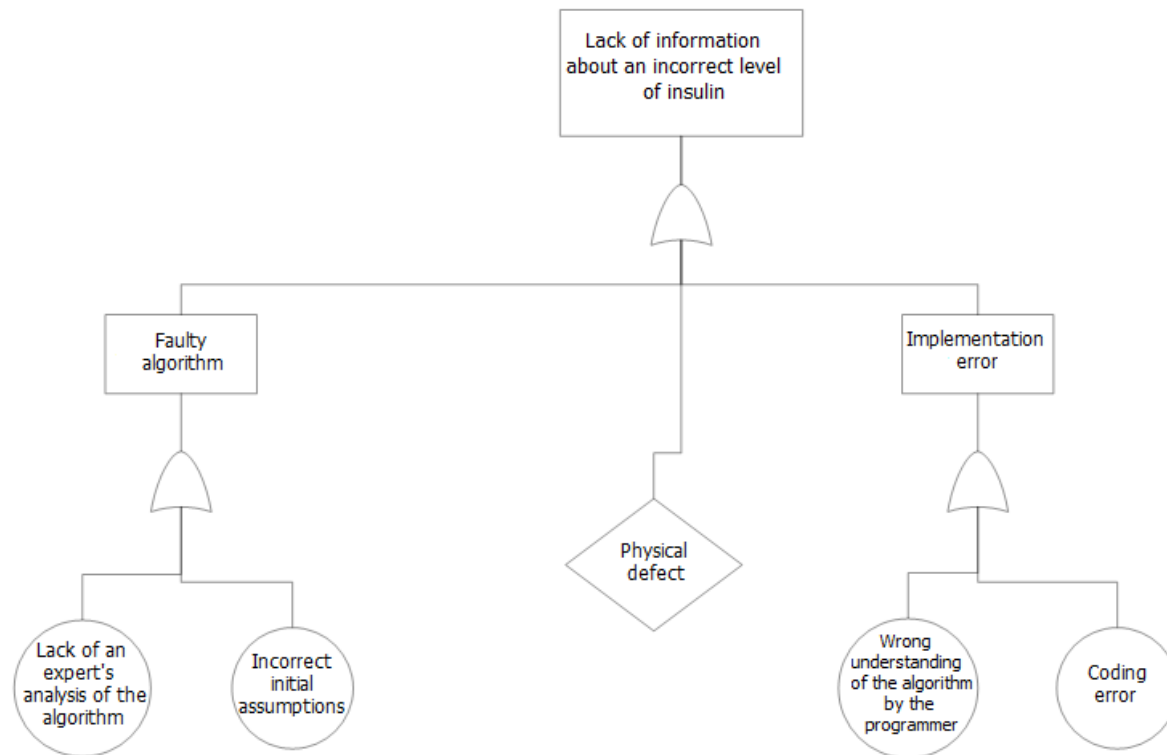
A. List of hazards

- In total, 124 hazards have been distinguished, at different levels of detail.
- There are some hazards that were commonly addressed and that can be divided into ten categories (the most common at the top):
 - Administration of an incorrect dose of insulin
 - User errors (adjusted dose, incorrect configuration, etc.)
 - Error in measuring the level of insulin or sugar
 - Physical / hardware errors
 - Missing or incorrectly administered insulin dose
 - Lack of measurement of insulin or sugar within a prescribed period
 - Errors in alerting system (sugar level, the needle slipped, discharging, etc.)
 - Unauthorized use of the device via radio waves
 - Interruption of system normal activity
 - Incorrect display of data

Example: hazards identification (Scrum)



Example: hazard scenarios (FT)



An example from one of students teams FTA analysis (Kalenik, Kurszewski, Karewska)

The case study - results

B. Risk assessment –for Scrum

- Tasks along with their impediments which were associated with the highest risk:
 - *Product Backlog* - incorrect identification of requirements
 - *Sprints Plan* - incomplete identification of requirements
 - *General decisions concerning technology and architecture* - lack of architecture plan and crucial implementation decisions
 - *General decisions concerning technology and architecture* - incomplete architecture plan and lacking crucial implementation decisions
 - *Providing the requirements (client)* - incorrect identification of requirements
 - *Providing the requirements (client)* - incomplete identification of requirements

Example: risk assessment (Scrum)

Item Id	Machine / Sub-process	User	Task	Hazard Category	Hazard	Cause/Failure Mode	Severity	Probability	Risk Level	Reduce Risk	Severity	
1	1-1-1-1	Planning	Product Backlog	incorrect identification of re	Nieumyślne użycie	Niejasne komunikaty po	Mały ekran urządzenia, nieprzygoto	Minor	Likely	Low	Przygotowanie mechanizmów odpowiedzialnych za prawidłowe wyświetlanie komunikatów, zależne od długości komunikatu.	Minor
2	Planning	Product Backlog	incorrect identification of requirements	Błędy implementacyjne	Wyłączenie pompy	Zbyt późny komunikat (30 min przed wyczerpaniem) o niskim stanie naładowania baterii, forma komunikatu dająca się przeoczyć, brak głośnego komunikatu	Catastrophic	Likely	High	Zwiększenie przedziału czasu ostrzeżenia o niskim stanie naładowania baterii do 12 godzin, wykorzystanie wielu sposobów sygnalizacji (wibracja, dźwięk oraz na ekranie)	Catastrophic	
3	Planning	Product Backlog	incomplete identification of requirements	Hazardy związane z dawkowaniem insuliny	Przerwanie podawania insuliny	Brak odpowiednich zabezpieczeń softwarowych,	Catastrophic	Unlikely	Medium	Zaprojektowanie wykrycia zerwania połączenia, liczenie na bieżąco podawanej dawki,	Catastrophic	
4	Planning	Product Backlog	incomplete identification of requirements	Hazardy związane z dawkowaniem insuliny	Niedostarczenie dawki insuliny	Brak odporności na drgania, niestabilne zachowanie pompy, za słaba bateria aby podać insulinę, brak insuliny w zbiorniku	Catastrophic	Unlikely	Medium	Dokładna analiza techniczna.	Catastrophic	
5	Planning	Product Backlog	incomplete identification of requirements	Nieumyślne użycie	Nieumyślne użycie pilota/pompy	Brak opcji blokady klawiszy	Moderate	Likely	Medium	ustawienie blokady klawiszy	Moderate	
6	Planning	Product Backlog	incomplete identification of requirements	Nieumyślne użycie	„Zepsucie się” insuliny	Za duży zbiornik insuliny, brak informacji o możliwej długości przetrzymywania insuliny w	Catastrophic	Remote	Low	Dodanie mechanizmu zapamiętywania daty ostatniej zmiany fiołki z insuliną,	Catastrophic	

Severity	Probability
<input type="checkbox"/> Catastrophic	<input type="checkbox"/> Very Likely
<input type="checkbox"/> Serious	<input checked="" type="checkbox"/> Likely
<input type="checkbox"/> Moderate	<input type="checkbox"/> Unlikely
<input checked="" type="checkbox"/> Minor	<input type="checkbox"/> Remote

View Risk Scoring System



The case study - results

B. Risk assessment – for XP

- Tasks along with their impediments which were associated with the highest risk:
 - *User Stories* - incomplete identification of requirements
 - *Prototyping* - too general plan for architecture and methods of implementing system
 - *Release scope* : functionalities from previous iteration - large load on errors from the previous iteration
 - *Tests preparation* - incomplete test plan
 - *Unit tests* - low coverage
 - *Acceptance tests* - low coverage

The case study - results

C. Proposed practices

- Most commonly proposed practices:
 - Introducing an expert knowledge into the project
 - Extensive testing (i.e. enhanced acceptance tests, Test Driven Development)
 - Introducing safety standards
 - Improving quality of the methodology practices (i.e. preparing really good User Stories)
 - Keeping high coding standards

The case study - Summary

- General optimism towards agile methodologies and their applicability in safety-critical projects
- According to the participants:
 - Agile methodologies should be regarded as complementary
 - Extensive testing and good identification of requirements are vital
 - A contact with domain experts and potential users is crucial
 - Safety assurance should be incremental – in parallel to iterations in development