Requirements Management
Outline

- Introduction
- Setting the Scene
- Open Question
- Let’s Harmonize
- Requirements Management
  - Management Goals
  - Sub activities of Requirements Management
  - Tasks that requirements managers handle
- Your Reactions
- Requirements Prioritization
  - Decision making in RE
  - The problem of requirements prioritization
    - Different approaches
- Requirements Traceability
ishaya - whoami

- Ishaya Gambo
- Postdoctoral Research Fellow
  - Here in our institute in Tartu
- Senior Lecturer
  - Department of Computer Science & Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

- **Website:** https://www.researchgate.net/profile/Ishaya_Gambo
- **Address:** Room 3009, Narva maantee 18, 51009 Tartu, Estonia
- **Phone/WhatsApp:** +372 545 89 888
“Knowing the stakeholders of a project and **documenting** their aligned requirements at the beginning of the project is **NOT** enough to implement a project successfully”
Setting the Scene

• Software is everywhere
  ✓ Software + hardware …
  ✓ Let’s reflect on our experience as users …
    often it is frustrating …

• Software engineering (SE) is a discipline we expect to change, adapt and accommodate what happens around us in the world

• As software engineers, it becomes necessary to recognize the need for engineering the way we live, feel, think, behave, and not just the software.

• Requirements engineering (RE) establishes the foundation for engineering a successful system/software (Pohl, 2010)
  ✓ it is essential in the production of good quality software systems.

• Stakeholder’s involvement is crucial in RE
  • stakeholders’ aligned requirements need to be documented
  • requirements can change during operation of the system (or software) concerned and should be kept up to date for documentation purposes until the system is decommissioned.
Setting the Scene

• RE in context:
  ✓ Provide methods, techniques, guidelines to:

  ✓ Discover, understand, formulate, analyse and agree on the purpose of a software to be built with emphasis on:

    – *What* problem should be solved
    – *Why* such a problem need to be solved
    – *Who* should be involved in solving that problem
    – *Which* elements are essential to a solution

  ✓ Manage and validate the requirements in the REP
Setting the Scene

Case Study A

CoCoA: Compilation Compiler Advisor

Developed at FBK-IRST (project coordinated by Paolo Avesani 2006), it was used in the Karadar website
Setting the Scene

A quick scenario

• A service provider in the “online music business” would like to promote the access to their online music archive … favour “community activities” …

• … and for this they would like to offer to their customers a web application to access their online music archive …

• Their customer (the users of this web application) should be able to create their own music compilation by accessing the online music archive, … searching for a song by title, genre …

• Users should be able to reuse compilations made by other users
Setting the Scene

The CoCoA case-study

• *What* problem should be solved?

• *Why* such a problem need to be solved?

• *Who* should be involved in solving that problem?

• *Which* elements are essential to a solution
The CoCoA case-study

• **What problem should be solved?**
  - Create on-line music compilation by accessing an online music archive;
    - Feeding the archive with music compilation that can be reused by other users
• **Why such a problem need to be solved?**
  - Service provider goal: building a community of customers
  - Customer goal: enjoy building personalized compilations
• **Who should be involved in solving that problem?**
  - Stakeholders: service provider; customers (end-users); software developers
• **Which elements are essential to a solution**
  - technology to support efficient management of music files
  - technology for building easily understandable user interface
Open Question

• How can we manage the requirements process given the following situation?

  • all requirements were well structured, aligned, and accepted at the start of the project, but changes are introduced by the end of the project or "go live"—and always at the worst times.

  • the requirements also change during operation of the system (or software) concerned and should be kept up to date for documentation purposes until the system is decommissioned.

  • the project becomes too complex:
    • multicultural, multi-stakeholder issues
    • Requirements have to be used by multiple persons
    • Requirements are supposed to be reused.
Let’s Harmonise!

• The more complex the project, the more crucial requirements management (RM)

  • **RM** includes the conscious management of requirements in the classic sense
    • (e.g., by means of assignment of attributes, creation of views, traceability, etc.) as well as the management of changes to requirements.

  • the prior planning and monitoring of the defined RE processes are also part of **RM**, in the sense of:
    • "How do I elicit, document, and review my requirements to be able to continuously report on the status and to react to planned changes?".
• the process of managing existing requirements and requirements-based artifacts.

• this includes documenting, changing, and tracing requirements (Glin, 2014).

• It also includes managing the requirements engineering process, which means planning, controlling, and checking the requirements engineering process.
Requirements Management

• **Justification/Motivation**
  - arise from the necessity to manage requirements and requirements-related activities
    - especially for large systems and software
  - to avoid uncontrolled "fire-fighting" at the RE lifecycle
  - for requirements manager to provide information about the status of the requirements or about the effects of any change requests at any given time.

• **Context**
  - Development project
  - Continued development of an existing system
  - Software-product management or with regard to a continuous, cross-project management of requirements
Observe the system context to detect context changes
Manage the execution of requirements engineering activities
Manage the requirements artefacts
Management goals

• Managing Requirements Artefacts

• Observing system context

• Managing Requirements Engineering Activities
Management goals

• Managing Requirements Artefacts
  – Definition of requirements attribute scheme (assignments of attributes)
  – Requirements traceability
  – Requirements landscape
  – Requirements change management
  – Requirements configuration management
  – Requirements prioritisation
  – Version management
  – Versioning
  – Variant management
  – Reporting

• Observing system context

• Managing Requirements Engineering Activities
Management goals

• Managing Requirements Artefacts

  – Definition of requirements attribute scheme (assignments of attributes): Focuses on:
    – the requirements that have already been accepted, and
    – the source from where the requirements comes from.

  – Requirements traceability: Focuses on:
    – the technical component belonging to a requirement
    – the test cases belonging to a requirement
    – the requirement that is part of the system/product delivered

  – Requirements landscape: Focuses on:
    – the different types of requirements available
    – the levels of detail requirements are documented
Management goals

• Managing Requirements Artefacts…

  – Requirements change management: Focuses on:
    – changes introduced in requirements at the end of the project?
    – changes introduced during operation of the system (or software)

  – Requirements configuration management (Variant management): Focuses on
    – how the two variants of the product differ.

  – Version management: Focuses on:
    – the requirements belonging to a specific software baseline

  – Versioning: Focuses on:
    – the version of the requirement implemented in the system
    – the last person to change the requirement and why did they change it
Management goals

- **Managing Requirements Artefacts…**

  - **Reporting:** Focuses on:
    - proportion of the requirements that has already been implemented and tested
    - the time it takes on the average for a change request to be implemented
    - knowing whether the requirements engineering process has been improved by a specific measure
  
  - **Requirements prioritization:** Focuses on:
    - the requirements that are urgent and important and therefore candidates for the next release
    - the requirement that generates costs that are too high with too few benefits
    - making informed decisions about issues raised during elicitation (tradeoff analysis and negotiation), by:
      - identifying and resolving conflicting concerns
      - assessing and mitigating risks
      - evaluating the alternative options emerging from the previous processes, and selecting the best options
Management goals

- **Managing Requirements Artefacts**
  - Definition of requirements attribute scheme (assignments of attributes)
  - Requirements traceability
  - Requirements landscape
  - Requirements change management
  - Requirements configuration management
  - Requirements prioritisation
  - Version management
  - Versioning
  - Variant management
  - Reporting

- **Observing system context**

- **Managing Requirements Engineering Activities**
Management goals

• Managing Requirements Artefacts
  – Definition of requirements attribute scheme (assignments of attributes)
  – Requirements traceability
  – Requirements landscape
  – Requirements change management
  – Requirements configuration management
  – Requirements prioritisation
  – Version management
  – Versioning
  – Variant management
  – Reporting

• Observing system context
  – A new technology or new competing product emerges
  – A law or standard changes
  – Evolution of stakeholder goals
  – Involvement of additional stakeholders
  – Changes of an organisational policy
  – Changes in the way that external actors (stakeholders or systems) use the system

• Managing Requirements Engineering Activities
Management goals

• Managing Requirements Artefacts
  – Definition of requirements attribute scheme (assignments of attributes)
  – Requirements traceability
  – Requirements landscape
  – Requirements change management
  – Requirements configuration management
  – Requirements prioritisation
  – Version management
  – Versioning
  – Variant management
  – Reporting

• Observing system context
  – A new technology or new competing product emerges
  – A law or standard changes
  – Evolution of stakeholder goals
  – Involvement of additional stakeholders
  – Changes of an organisational policy
  – Changes in the way that external actors (stakeholders or systems) use the system

• Managing Requirements Engineering Activities
Management goals

• Managing Requirements Artefacts
  – Definition of requirements attribute scheme (assignments of attributes)
  – Requirements traceability
  – Requirements landscape
  – Requirements change management
  – Requirements configuration management
  – Requirements prioritisation
  – Version management
  – Versioning
  – Variant management
  – Reporting

• Observing system context
  – A new technology or new competing product emerges
  – A law or standard changes
  – Evolution of stakeholder goals
  – Involvement of additional stakeholders
  – Changes of an organisational policy
  – Changes in the way that external actors (stakeholders or systems) use the system

• Managing Requirements Engineering Activities
  – Phase-oriented approach
    • Same sequence of activities is applied to all requirements artefacts
  – Situative approach
    • Activities to be executed next based on assessment of current status of the existing requirements artefacts
Management goals

• **Managing Requirements Artefacts**
  – Definition of requirements attribute scheme
    *(assignments of attributes)*
  – Requirements traceability
  – Requirements landscape
  – Requirements change management
  – Requirements configuration management
  – Requirements prioritisation
  – Version management
  – Versioning
  – Variant management
  – Reporting

• **Observing system context**
  – A new technology or new completing product emerges
  – A law or standard changes
  – Evolution of stakeholder goals
  – Involvement of additional stakeholders
  – Changes of an organisational policy
  – Changes in the way that external actors (stakeholders or systems) use the system

• **Managing Requirements Engineering Activities**
  – **Phase-oriented approach**
    • Same sequence of activities is applied to all requirements artefacts
  – **Situative approach**
    • Activities to be executed next based on assessment of current status of the existing requirements artefacts
RM Goal in Clear Term

• The goal of RM is to **manage requirements** and **other artifacts related to requirements**
  • (e.g., interview logs and the customer requirements specification) in such a way that the requirements can be systematically scanned, grouped, evaluated, changed, and tracked with reasonable effort.

• RM attempts to meet the needs of many different stakeholders simultaneously.
  • These needs are essentially dependent on the specific project context.
  • They differ, for example, in projects for customer-specific software or product development compared to internal projects performed by the IT department.
Sub activities of RM

1. Observation of the system context to reveal context changes

2. Management and execution of the requirements engineering activities (i.e., requirements management as process management)

3. Management of the requirements and related artifacts during the development process

Tasks that requirements managers handle

- **plan** and **define** the following at the beginning of the RE process:
  
  - The requirement types to be considered, the format in which they must be presented, and the level of detail to which they must be specified.
  
  - The questions to be answer on the basis of his requirements and the views that are necessary for the different stakeholders.
  
  - The criteria to be used to evaluate the requirements to support prioritization.
  
  - Version control for requirements and requirements documents.
  
  - How and when changes should be handled.
Tasks that requirements managers handle…

- **plan** and **define** the following at the beginning of the RE process

  - The requirements and other development artifacts between which traceability must be achieved

  - Decide on how to document requirement variants within the requirements specification

  - The requirement status reports needed, the information they must contain, and the sources (for example, attribute documentation) that can be used to determine this information

  - What the exact RE process (or sequence of activities) for the project should look like, and how the process can be monitored and potentially improved
Your Reaction! – 5 - 7mins


b. RM is worthwhile (valuable) not just for larger projects but also for small projects. What do you think?
Wrap-up

- **Our perspective:**
  - RE = essential in the production of good quality software systems.
  - Stakeholder’s involvement = crucial in RE
  - Documented requirements can change
    - at the end of the project
    - during operation of the system (or software)
  - Requirements changes should be kept up to date for documentation purposes until the system is decommissioned.
  - RM is a complex task: every stakeholder should be able to access up-to-date information at all times and should also be informed about changes that affect them
  - RM simplifies RE: The overarching goal is to manage requirements and other artifacts related to requirements
Wrap-up in picture
Questions?
Outline

- Introduction
- Setting the Scene
- Open Question
- Let’s Harmonize
- Requirements Management
- Management Goals
- Sub activities of Requirements Management
- Tasks that requirements managers handle
- Your Reactions

- Requirements Prioritization
  - Decision making in RE
  - The problem of requirements prioritization
  - Different approaches

- Requirements Traceability
Requirements Prioritization

- Requirements prioritization (RP) is an established activity in RE (Leshob et al., 2019) and an essential subject in determining how requirements are chosen for implementation (Rojas and Macías, 2019), thereby facilitating good decision-making.

- The goal of prioritization is to rank requirements in their order of implementation, thereby providing economic value to customers (Leshob et al., 2019).

- There is more about RP in the Internet...
  - Check now and give your feedback
A data-driven life cycle
The SUPERSEDE approach
The Decision Process

Distributed decision making teams

Decision making process orchestration

- Decide the set of Requirements
- Negotiate on conflicts
- Decide Priority of requirements

User feedback

Context monitoring

Prioritised Requirements and Next Release
Next Release Problem

“Release planning addresses decisions related to prioritizing, selecting and assigning requirements to create a sequence of consecutive product releases that satisfies important technical, resource, budget, and risk constraints”
Problem: [Prioritization of Requirements] To find the best ordering of requirements in each successive release to ensure quality & value of the delivered system, trade-off constraints & end-user satisfaction.
Release Planning

- A good release plan should
  - provide maximum business value by offering the best possible blend of features in the right sequence of releases
  - satisfy the most important stakeholders involved
  - be feasible with available resources
  - reflect existing dependencies between features

- The art of release planning approach relies on human intuition, communication, and capabilities to negotiate between conflicting objectives and constraints

- The science of release planning approach formalizes the problem and applies computational algorithms to generate best solutions

Günther Ruhe and Moshood Omolade Saliu - The Art and Science of Software Release Planning 2009
Problem and Solution Space
Several dimensions

- Multi criteria vs Mono criteria
- Human vs Automated
- Nature of the problem
- Decision maker involved
- Interaction with the Decision maker
- Specific technique
- Machine Learning vs Search-based vs Constraint based
- Pairwise based ranking vs Explicit ordering
Mono / Multi-criteria

**Mono-criteria**
- Consider only 1 aspect
  - For example:
    - maximize the value of the requirements implemented in a software

**Multi-criteria**
- Considering $n$ aspects (e.g., cost and value)?
  - For example:
    - maximize the value of the requirements implemented in a Software and in parallel minimise the cost to implement them

So the same requirements can be seen from different perspectives
Approaches to the Problem

**Machine Learning techniques**
- Here the solution is built incrementally on the bases of the current knowledge and the new knowledge entering in the System

**Search-Based techniques**
- Here the solution is found generating several possible solutions and evaluating it against a fitness function that allow to evaluate “how good” is the found solution

**SAT-based methods**
- This find the solution in a constraints based fashion
Fully automated or Human based?

**Completely automated**
- Given the set of objects to order the algorithm rank them on the bases of their properties

**Human assisted**
- Human is asked to give (contextual) assessment of the objects to rank (considering their properties)
Classification: State-of-the-Art approaches

- **Decision maker Knowledge** refers to the awareness of the requirements attributes and the overall system functionalities to be developed.
Some characteristics

- Pairwise Comparison based approaches
- Non-Pairwise comparison based approaches
- Combining Techniques
- Domain Knowledge based approaches
The domain: software requirements

<table>
<thead>
<tr>
<th>Requirement B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
</tr>
<tr>
<td><strong>Descr.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk</td>
<td>high</td>
</tr>
<tr>
<td>Value</td>
<td>medium</td>
</tr>
</tbody>
</table>

Characteristics of the requirement:
- Cost of implementation
- Risk
- Value for stakeholders
Basics of Prioritization

Need to select what to implement

- Customers (usually) ask for way too much
- Balance time-to-market with amount of functionality
- Decide which features go into the next release

For each requirement/feature, ask:

- How important is this to the customer?
- How much will it cost to implement?
- How risky will it be to attempt to build it?

Perform Triage:

- Some requirements *must* be included
- Some requirements should definitely be excluded
- That leaves a pool of “nice-to-haves”, which we must select from.
Let’s look at some RP Approaches
A Cost-Value Approach

- **Calculate return on investment**
  - Assess each requirement’s importance to the project as a whole
  - Assess the relative cost of each requirement
  - Compute the cost-value trade-off:

![Cost-Value Trade-Off Diagram](image)
A Cost-Value Approach

- **Calculate return on investment**
  - Assess each requirement’s importance to the project as a whole
  - Assess the relative cost of each requirement
  - Compute the cost-value trade-off:

<table>
<thead>
<tr>
<th>Cost (percent)</th>
<th>Value (percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low priority</td>
<td>5</td>
</tr>
<tr>
<td>Medium priority</td>
<td>10</td>
</tr>
<tr>
<td>High priority</td>
<td>15</td>
</tr>
</tbody>
</table>

- **Two approaches:**
  - **Absolute scale (e.g. dollar values)**
    - Requires much domain experience
  - **Relative values (e.g. less/more; a little, somewhat, very)**
    - Much easier to elicit
    - Prioritization becomes a sorting problem
Some complications

- **Hard to quantify differences**
  - easier to say “x is more important than y”...
  - ...than to estimate by how much.

- **Not all requirements comparable**
  - E.g. different level of abstraction
  - E.g. core functionality vs. customer enhancements

- **Requirements may not be independent**
  - No point selecting between X and Y if they are mutually dependent

- **Stakeholders may not be consistent**
  - E.g. If X > Y, and Y > Z, then presumably X > Z?

- **Stakeholders might not agree**
  - Different cost/value assessments for different types of stakeholder
Hierarchical Prioritisation

- Group Requirements into a hierarchy
  - e.g. A goal tree

- Only make comparisons between branches of a single node:

![Diagram showing a hierarchy of requirements and comparisons](attachment:diagram.png)
Analytic Hierarchy Process (AHP)

Requirement B is 3 times more important than A

Ranking computed via eigenvectors
Perini et al. (2007)
Analytic Hierarchy Process (AHP)

Source: Adapted from Karlsson & Ryan 1997

Create n x n matrix (for n requirements)
- For element (x,y) in the matrix enter:
  - 1 - if x and y are of equal value
  - 3 - if x is slightly more preferred than y
  - 5 - if x is strongly more preferred than y
  - 7 - if x is very strongly more preferred than y
  - 9 - if x is extremely more preferred than y
  - (use the intermediate values, 2,4,6,8 if compromise needed)
- ...and for (y,x) enter the reciprocal.

Estimate the eigenvalues:
- E.g. “averaging over normalized columns”
  - Calculate the sum of each column
  - Divide each element in the matrix by the sum of it’s column
  - Calculate the sum of each row
  - Divide each row sum by the number of rows

This gives a value for each requirement:
- ...giving the estimated percentage of total value of the project
AHP example - estimating costs

<table>
<thead>
<tr>
<th></th>
<th>Req1</th>
<th>Req2</th>
<th>Req3</th>
<th>Req4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req1</td>
<td>1</td>
<td>1/3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Req2</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Req3</td>
<td>1/2</td>
<td>1/5</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>Req4</td>
<td>1/4</td>
<td>1/3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Normalise columns

Sum the rows

<table>
<thead>
<tr>
<th></th>
<th>Req1</th>
<th>Req2</th>
<th>Req3</th>
<th>Req4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req1</td>
<td>0.21</td>
<td>0.18</td>
<td>0.18</td>
<td>0.48</td>
</tr>
<tr>
<td>Req2</td>
<td>0.63</td>
<td>0.54</td>
<td>0.45</td>
<td>0.36</td>
</tr>
<tr>
<td>Req3</td>
<td>0.11</td>
<td>0.11</td>
<td>0.09</td>
<td>0.04</td>
</tr>
<tr>
<td>Req4</td>
<td>0.05</td>
<td>0.18</td>
<td>0.27</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Result

<table>
<thead>
<tr>
<th></th>
<th>sum</th>
<th>sum/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req1</td>
<td>1.05</td>
<td>0.26</td>
</tr>
<tr>
<td>Req2</td>
<td>1.98</td>
<td>0.50</td>
</tr>
<tr>
<td>Req3</td>
<td>0.34</td>
<td>0.09</td>
</tr>
<tr>
<td>Req4</td>
<td>0.62</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Req1 - 26% of the cost
Req2 - 50% of the cost
Req3 - 9% of the cost
Req4 - 16% of the cost
Repeat AHP process twice:

- Once to estimate relative value
- Once to estimate relative cost

Plot ROI graph
Other selection criteria

- Above average value
- Below average cost
- Above average in both cost and value
- Above average cost
- Below average value

Risk Exposure:
- High
- Low

Relative Probability

Relative Loss
Machine Learning Approach

- Abstract view of the Process

Objectives:
- Minimize the number of pairwise comparisons
- Approx accuracy

Requirements

Pairwise comparisons from users
Machine Learning Approach

- The Ranking Process

![Diagram showing the ranking process]

Legend:
- Data
- Procedures
- User Activities
Search-Based approach

- Based on *Interactive* Genetic Algorithm (IGA)
  - aims at minimizing the disagreement between a total order of prioritized requirements and the various constraints that are either encoded with the requirements or that are expressed iteratively by the user during the prioritization process

considering user knowledge, considering the possibility of arbitrary constraints and assuring robustness to errors
A genetic algorithm

- A population of candidate solutions, called individuals, to an optimization problem is evolved toward better solutions

- Each candidate solution has a set of properties which can be mutated and altered (via mutation, recombination, …)

- The evolution usually:
  - Starts from a population of randomly generated individuals
  - Iteratively is evolved along generations
  - In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved
The Input

- **Set of Requirements**

- **Domain Knowledge**: available as requirements documentation (e.g., cost of the implementation, value for the stakeholders, dependencies between requirements) that can be converted into total or partial rankings of the requirements

- **Evaluation from users** in terms of orderings between pairs of requirements
SAT-Based Approach

- Based on Satisfiability Modulo Theory and on interactive pairwise user feedback
  - aims at minimizing the **disagreement** between
    - a total order of prioritized requirements
    - and the constraints that are either encoded with the requirements or that are expressed iteratively by the user during the prioritization process
Security Risk Management in Airline Turnaround Sector

- **Check-in passenger information**
  - Risk1: Blacklisted passenger presents fake document, gets checked-in because personnel could be bribed
  - Risk2: Attacker uses phishing email to extract passenger booking number and uses it to check-in to the flight

- **Luggage information**
  - Risk3: The personnel records values lower than actual weight of luggage and ground operations uses the information in the loading of the aircraft
  - Risk4: The personnel accepts luggage and adds contraband items to a passenger’s luggage

- **Fuel slip**
  - Risk5: A malicious insider with access to the computer that stores the fuel slip performs changes to the data contained in the fuel slip
  - Risk6: The attacker intercepts the fuel slip, changes the data contained and sends it to the supplier

- **Cargo assignment**
  - Risk7: A malicious insider with access rights performs changes to the cargo assignment document before it is sent to a service provider
  - Risk8: An attacker hacks the airline mailing list, receives the cargo assignment, changes the data contained and sends the cargo assignment to a service provider

[Matulevičius et al., 2016]
FDSE 2016
Security Risk Management in Airline Turnaround Sector

- **Risk1**: Blacklisted passenger presents fake document, gets checked-in because personnel could be bribed.

- **Risk2**: Attacker uses phishing email to extract passenger booking number and uses it to check-in to the flight.

- **Risk3**: The personnel records values lower than actual weight of luggage and ground operations uses the information in the loading of the aircraft.

- **Risk4**: The personnel accepts luggage and adds contraband items to a passenger’s luggage.

- **Risk5**: A malicious insider with access to the computer that stores the fuel slip performs changes to the data contained.

- **Risk6**: The attacker intercepts the fuel slip, changes the data contained and sends it to the supplier.

- **Risk7**: A malicious insider with access rights performs changes to the cargo assignment document before it is sent to a service provider.

- **Risk8**: An attacker hacks the airline mailing list, receives the cargo assignment, changes the data contained and sends the cargo assignment to a service provider.

[Matulevičius et al., 2016]

FDSE 2016
Security Risk Management in Airline Turnaround Sector

- **Check-in passenger information**
  - **Risk1**: Blacklisted passenger presents fake document, gets checked-in because personnel could be bribed
  - **Risk2**: Attacker uses phishing email to extract passenger booking number and uses it to check-in to the flight

- **Luggage information**
  - **Risk3**: Personnel records values lower than actual weight of luggage and ground operations uses the information in the loading of the aircraft
  - **Risk4**: Personnel accepts luggage and adds contraband items to a passenger’s luggage

- **Fuel slip**
  - **Risk5**: A malicious insider with access to the computer that stores the fuel slip performs changes to the data contained in the fuel slip
  - **Risk6**: The attacker intercepts the fuel slip, changes the data contained and sends it to the supplier
  - **Risk7**: A malicious insider with access rights performs changes to the cargo assignment document before it is sent to a service provider
  - **Risk8**: An attacker hacks the airline mailing list, receives the cargo assignment, changes the data contained and sends the cargo assignment to a service provider

[Matulevičius et al., 2016]
FDSE 2016
Think, reflect and say something!

Why is RP needed?
Exercise
AIS – Airline Information System

- Invoicing
  Rinv01. AIS should create an invoice for a booking.
  Rinv02. AIS should send a created invoice to the customer
  Rinv03. Customer should pay the invoice using AIS
  Rinv04. AIS should close a paid invoice

- Boarding
  Rboa1. AIS should create a boarding pass
  Rboa2. AIS should register a boarded customer

- Booking
  Rboo1. AIS should allow the customer to book an available flight
  Rboo2. Customer should query the airline company for available flights
  Rboo3. Customer should choose the number of seats on the available flights
  Rboo4. AIS should book an airline flight based on the booking info
Exercise
Requirements Prioritisation

- **Perform requirements prioritization**
  - Use the AHP method
  - Prioritize requirements according to
    - **Value**
    - **Cost**
  - Show prioritization results in a plot
Exercise 2

1. Read the article “Requirements Management in Automotive: an Empirical Study on Process Improvement Areas” by Falcini and Lami (2020) and respond to the following: (Paper uploaded on wiki-site)
   i. What are the lessons to learn on RM that can be beneficial to an organization and/or software development project?
   ii. Identify that gaps in the approach the authors used and suggest a possible solution or strategy to fill such gap(s).

2. Decide on any software development project (a case study) of your choice and attempt the following
   i. Describe the initial scenario (textually or with pictures) of the case study
   ii. What is the application domain of your choice project?
   iii. Who are the stakeholders?
   iv. Which are their goals?
   v. What is the main Business requirements of the application domain?
   vi. Express (write out the user requirements), constraints and quality requirements
   vii. Use any of the requirements prioritization approach suitable to you
References


• Martin Glinz: A Glossary of Requirements Engineering Terminology. Version 1.6 May 2014

