**Case Study Addendum: Enhancing Moodle Platform Testing for Kodolányi János University**

**Author:** Bayanmunkh Ganbat **Supervisor:** Prof. László Pitlik (critical aspects, quality assurance)  
**Date:** 2025.02.18  
**Email:** [pitlik.laszlo@kodolanyi.hu](mailto:pitlik.laszlo@kodolanyi.hu)

**Initial Documents:**

* [IT-Security Risk Index for Home Workers](https://miau.my-x.hu/miau/319/itsec_index_for_home_workers.docx)🡨Is this documentation more concrete than the initial (benchmark) documentation?
* <https://miau.my-x.hu/miau/320/moodle_neptun_tests/Neptun_testing1.pdf<--thes>e steps are concrete enough, these steps are reproducible
* <https://miau.my-x.hu/miau/320/moodle_neptun_tests/Neptun_testing2.docx<--these> steps are concrete enough, these steps are reproducible
* Case Study Addendum - Neptun\_pl.docx🡨empty bubbles need examples/annexes and never more empty bubbles: <https://miau.my-x.hu/miau/320/moodle_neptun_tests/> (see: the\_single\_position\_for\_empty\_bubbles\_is\_already\_in\_use\_\*.docx
* [Index of /miau/320/moodle\_cubes\_logic](https://miau.my-x.hu/miau/320/moodle_cubes_logic/)
* [Index of /miau/320/moodle\_neptun\_tests](https://miau.my-x.hu/miau/320/moodle_neptun_tests/)|

## **Introduction**

This case study outlines a structured approach to creating a benchmark for deriving IT-security risk index values🡨what is the role of the Moodle System here and now?. It includes three primary layers:

1. Establishing index values through benchmarking.
2. Exploring automation aspects of the benchmarking process.
3. Developing and testing the automated components.

The goal is to refine a conceptual model for IT security risk assessment, ensuring its applicability and automation potential while validating its functionality through rigorous testing.

## **Layer 1: Index-Values (Benchmarking) with Critical Aspects**

### **Concept for IT Security Risk Index Software**

#### **Key Attributes for IT-Security Risk Assessment**

A robust IT-security risk index should incorporate the following attributes:

1. **Device Security** - Availability of antivirus, firewall, and encryption.
2. **Network Security** - Router security, VPN usage, and encryption.
3. **Software Updates** - Frequency of updates and patches.
4. **User Awareness** - Knowledge of security best practices.
5. **Access Control** - Password strength and multi-factor authentication.
6. **Data Backup** - Backup frequency and reliability.
7. **Physical Security** - Secure storage and access control for devices.
8. **Incident Response** - Incident preparedness and response plans.
9. **Remote Access Security** - Security measures for remote connections.
10. **Third-Party Risk** - Security of external applications and services.

#### **Calculation Model**

Each attribute receives a score (0-8), with weights assigned based on importance:

* Example Calculation: Where weights sum to 1.
* **Critical Aspects:**
  + Measurement units for raw scores are undefined.
  + Weight assignments are subjective.
  + Attribute directions (higher vs. lower risk) lack clarity.

### **Recommendations for Improvement**

* Standardize measurement units and define scoring criteria.
* Base weights on empirical data rather than subjective input.
* Clearly define how each attribute influences risk.

## **Layer 2: Automations with Critical Aspects**

### **Automation Strategy**

* Develop an HTML-based interface to collect attribute values.
* Implement JavaScript for automatic risk index calculation.
* Store results dynamically for trend analysis.

### **Code Implementation**

An HTML form enables score selection, and JavaScript calculates the risk index:

#### **Critical Aspects:**

* The form lacks validation for incorrect or missing inputs.
* The JavaScript calculation is hardcoded with fixed weights.
* No data storage or reporting functionality is included.

### **Recommendations for Improvement**

* Implement form validation to ensure valid inputs.
* Enable dynamic weight adjustments based on risk model refinements.
* Store results in a database for tracking security trends.

## **Layer 3: Testing with Critical Aspects**

### **Testing Strategy**

A comprehensive testing framework is essential to validate the automation components:

1. **Form Validation Tests**
   * Ensure dropdowns accept valid values (0-8).
   * Prevent form submission with missing inputs.
2. **Calculation Accuracy Tests**
   * Compare outputs with manual calculations.
   * Test edge cases (all 0s, all 8s).
3. **UI/UX Testing**
   * Verify proper alignment and readability.
   * Ensure risk index displays correctly.
4. **Error Handling Tests**
   * Simulate missing form elements.
   * Check for unexpected input errors.
5. **Cross-Browser Compatibility Tests**
   * Validate performance across Chrome, Firefox, Safari, and Edge.
6. **Performance Testing**
   * Measure response time for index calculation.
7. **Security Tests**
   * Check for vulnerabilities in form inputs.
   * Ensure safe handling of user data.

### **Recommendations for Improvement**

* Integrate unit and system testing using automated test scripts.
* Implement dynamic reporting for test results.
* Address security concerns with input sanitization.

## **Conclusions**

A well-structured final thesis on IT security risk indexing should:

1. Establish a clear, evidence-based benchmark.
2. Ensure automation aspects are logically sound and data-driven.
3. Rigorously test the model to verify accuracy, usability, and security.

By refining these aspects, the benchmark concept can serve as a strong foundation for further research and real-world application in IT security risk management.