Artificial Intelligence Methods for Optimization of the Software Testing Process: with Several Industrial Case Studies

Sahar Tahvili, PhD

AI Product Development Leader, Test Framework & AI

IARIA

ERICSSON

Background

Education:

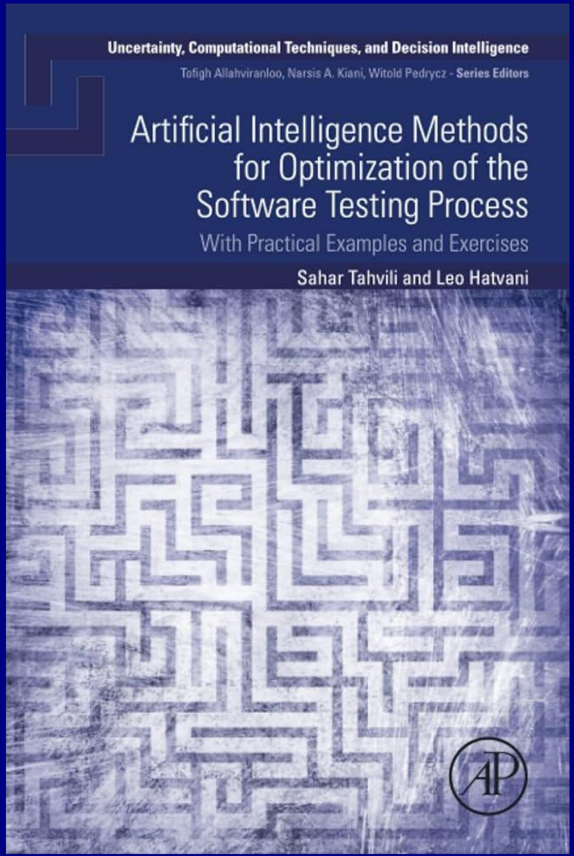
- **Doctor of Philosophy in Computer Science,** thesis entitled " *Multi-Criteria Optimization of System Integration Testing*", Mälardalen University, Sweden, 2018.
- **Bachelor and Master of Philosophy in Applied Mathematics**

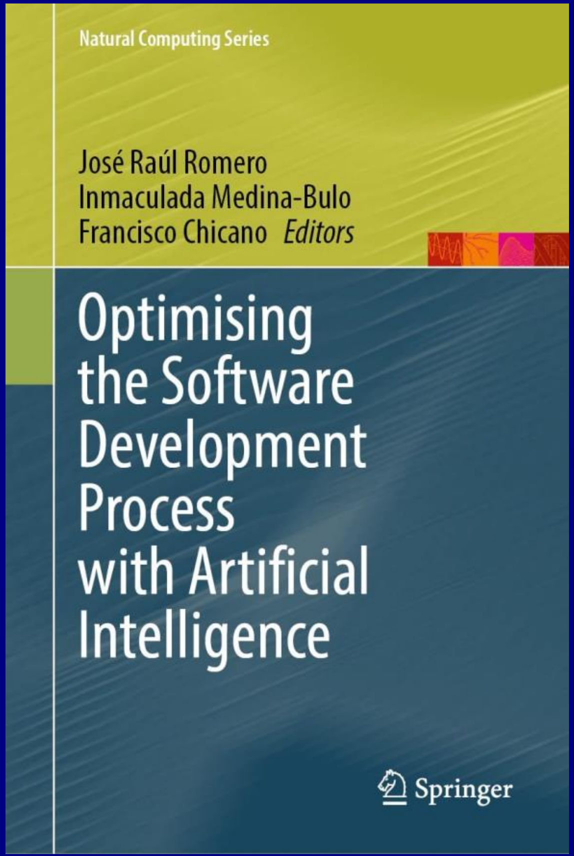
Research Area:

- Software Testing
- Optimization
- Artificial intelligence
- Natural language processing

Publication

- Sahar Tahvili and Leo Hatvani. "Artificial Intelligence Methods for Optimization of the Software Testing Process With Practical Examples and Exercises". Elsevier, July 2022. This book has been published as a peer-reviewed manuscript by Elsevier (Academic Press). The presented results in this book have not been published before. Chapter 5 of this book includes 8 industrial use cases. Furthermore, both the proposal and manuscript underwent multiple revisions before publication.
- Michael Felderer, Eduard Paul Enoiu, and Sahar Tahvili. "Artificial Intelligence Techniques in System Testing". In optimizing the Software Development Process with Artificial Intelligence. Springer, July 2023.





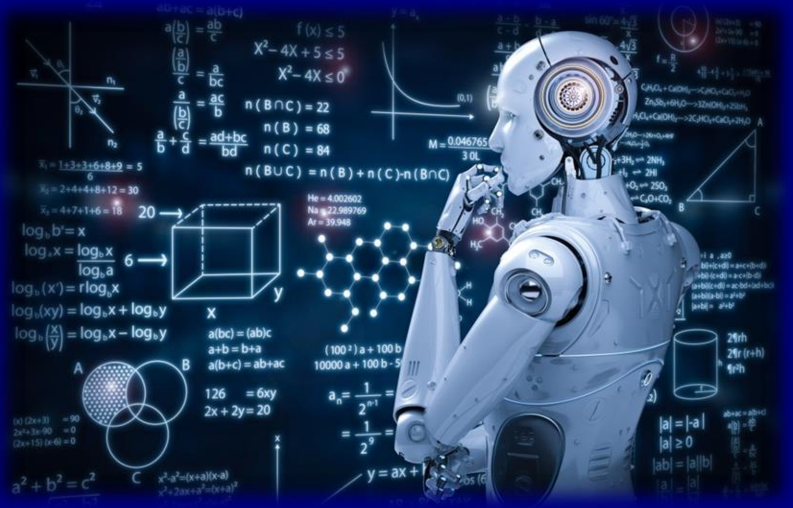

ACM Education & Learning Center @acmeducation · Mar 21
This book presents one of the first empirical studies in the field, contrasting theoretical assumptions on innovations in a real industrial environment with a large set of use cases from developed and developing testing processes at various large industries.
1 83

ACM Education & Learning Center @acmeducation · Mar 21
It also explores specific comparative methodologies, focusing on developed/developing AI-based solutions; serves as a guideline for conducting industrial research in the AI & software testing domain; explains all proposed solutions through real industrial case studies.
85

ACM on X

Challenges

- For testing any software applications (such as Telecom, and safety-critical systems) a set of test cases needs to be generated.
- Test automation requires deep knowledge in the domain (Telecom, safety-critical systems), programming (Java, C#), and software testing.
- A set of test environments (testbed, test channel) are required.
- A mapping between the test scripts and test environments must be done manually and daily, which is not scalable.
- Manual scheduling of the test scripts to the test environments has required a team where they need to master:
  - The capacity, capability, and configuration for each test environment
  - The functional dependencies between test scripts
  - The duration of each test script (based on the lines of the code)

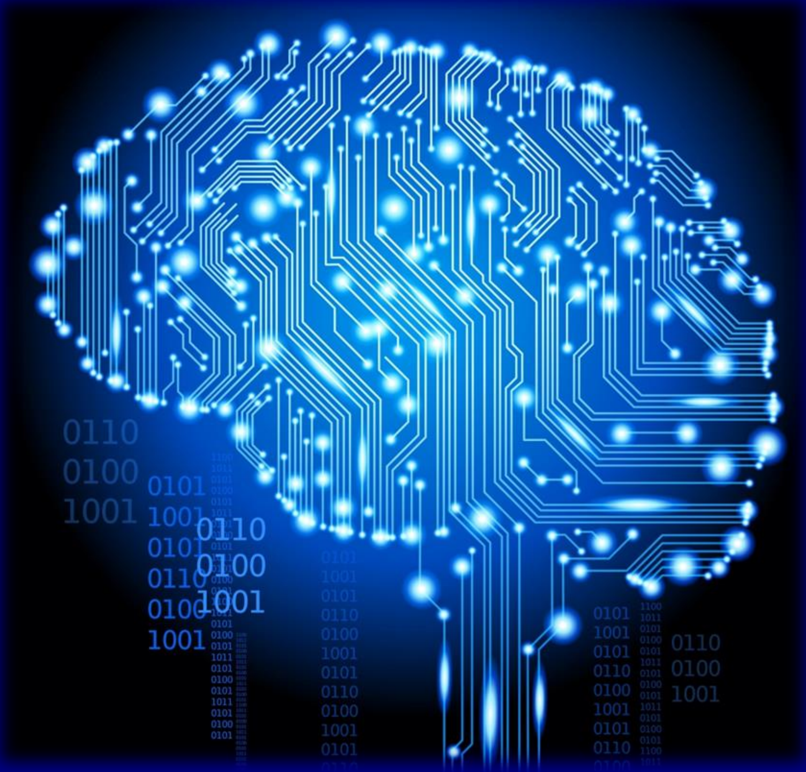


Risk

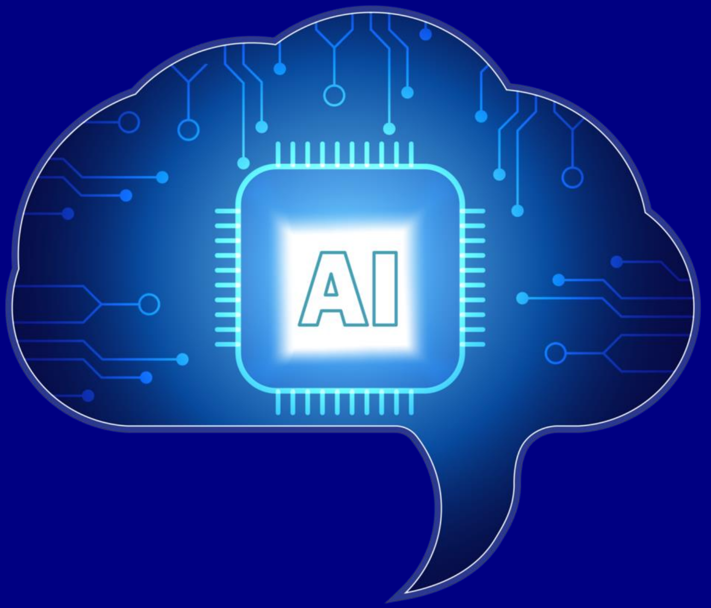
- Time and resource-consuming manual process.
- Requires deep knowledge in the domain, software testing, and test automation.
- Can not handle a large set of features and test environments.
- Paying no attention to the dependencies between test cases can lead to unnecessary failures between test cases by up to 40%.
- Running a test case on the wrong test environment leads directly to test failure.

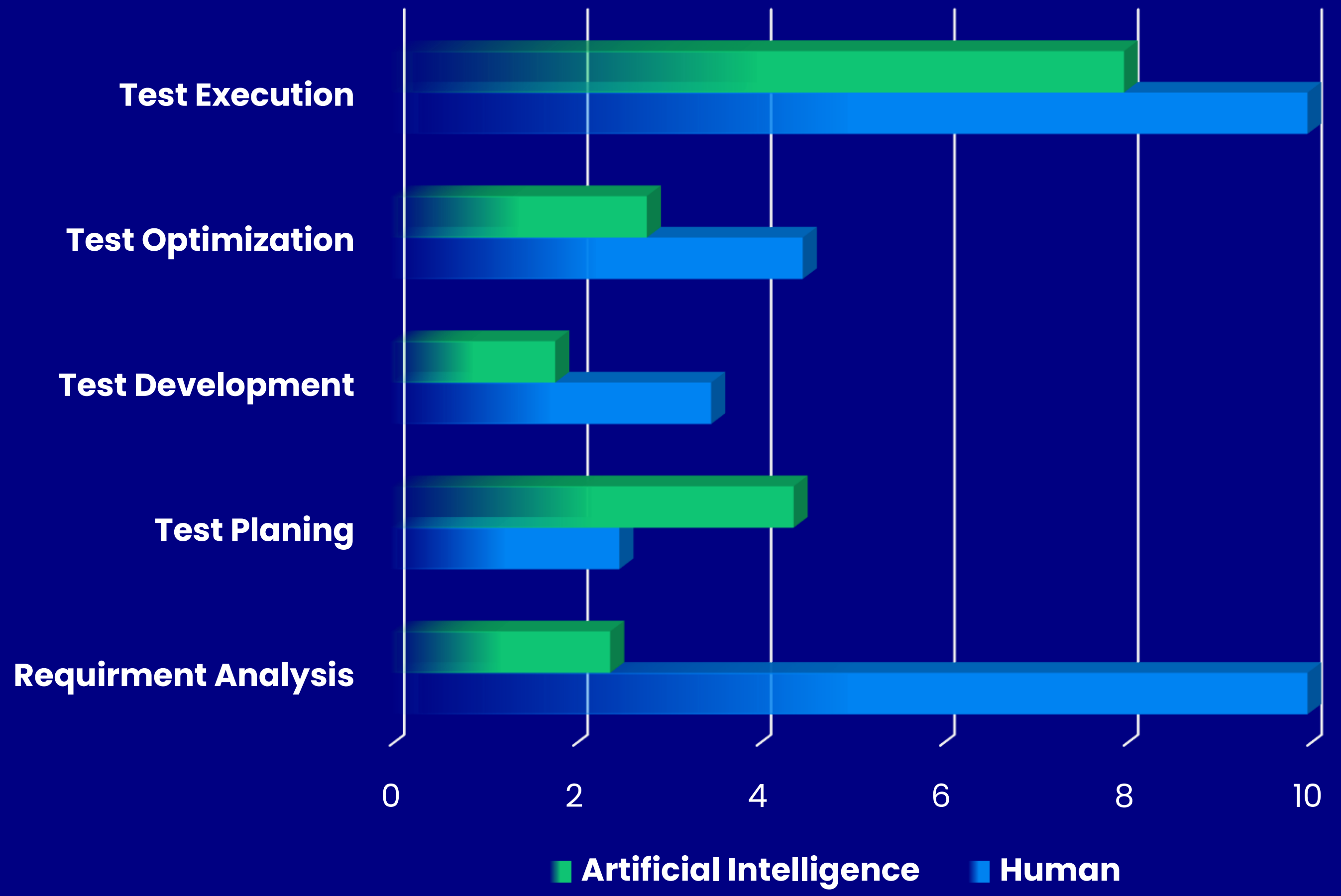
Optimization Problems

- **Test Environment Optimization**
  - Dynamic scheduling and building test environments
- **Test Execution Optimization**
  - Parallel test execution
  - Faster troubleshooting processes
- **AI-based Test Management System**
  - Test automation
  - CI/CD Configuration Selection



A segment of the impact of artificial intelligence in software development during 2020–2021





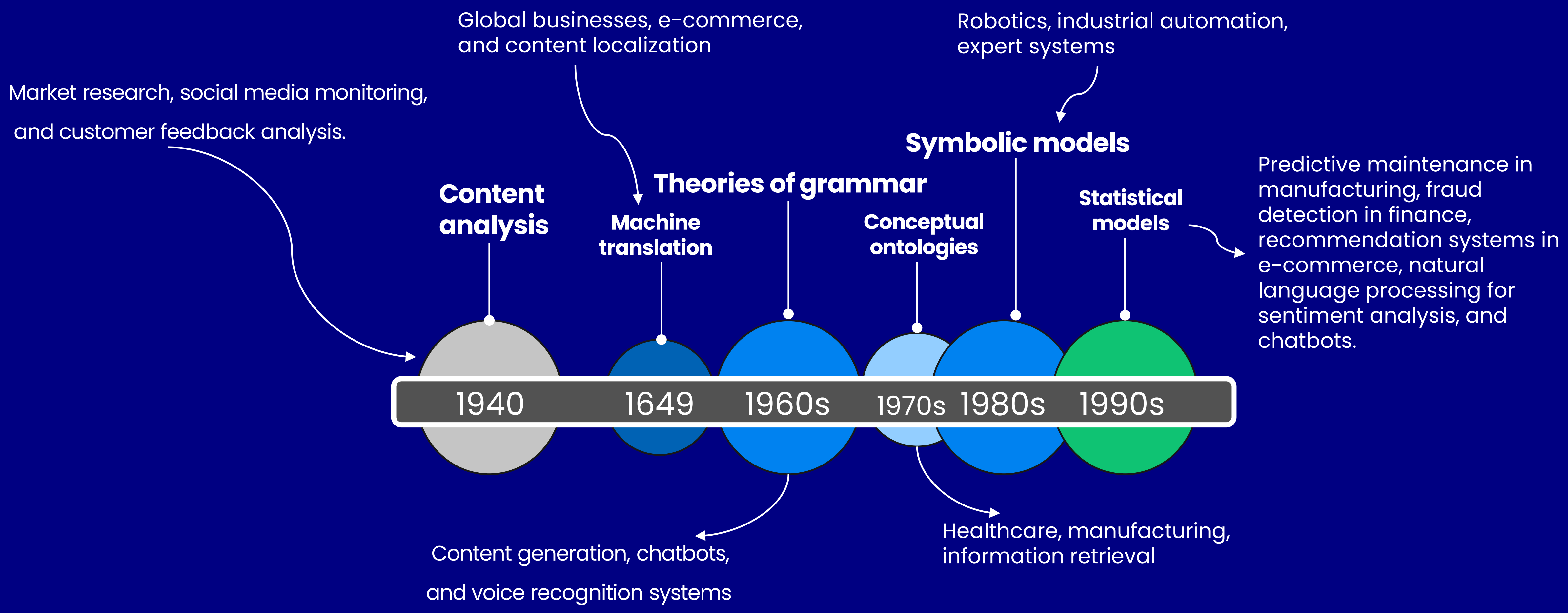
Source: Tahvili. S and Hatvani. L, *Artificial Intelligence Methods for Optimization of the Software Testing Process: With Practical Examples and Exercises, 2022*

AI-aided Content Analysis: Transforming Unstructured Text into Structured Data for Efficient Research

Drivers

Collaborative, automated physical world	Connected intelligent machines	The internet of senses

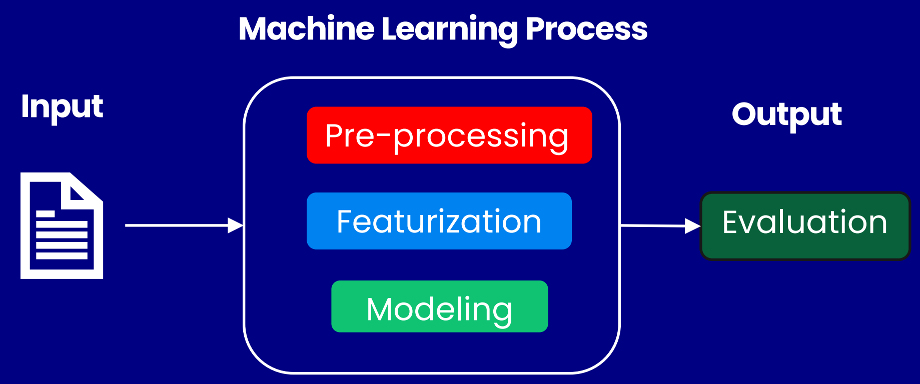
A review of the history of text analytics



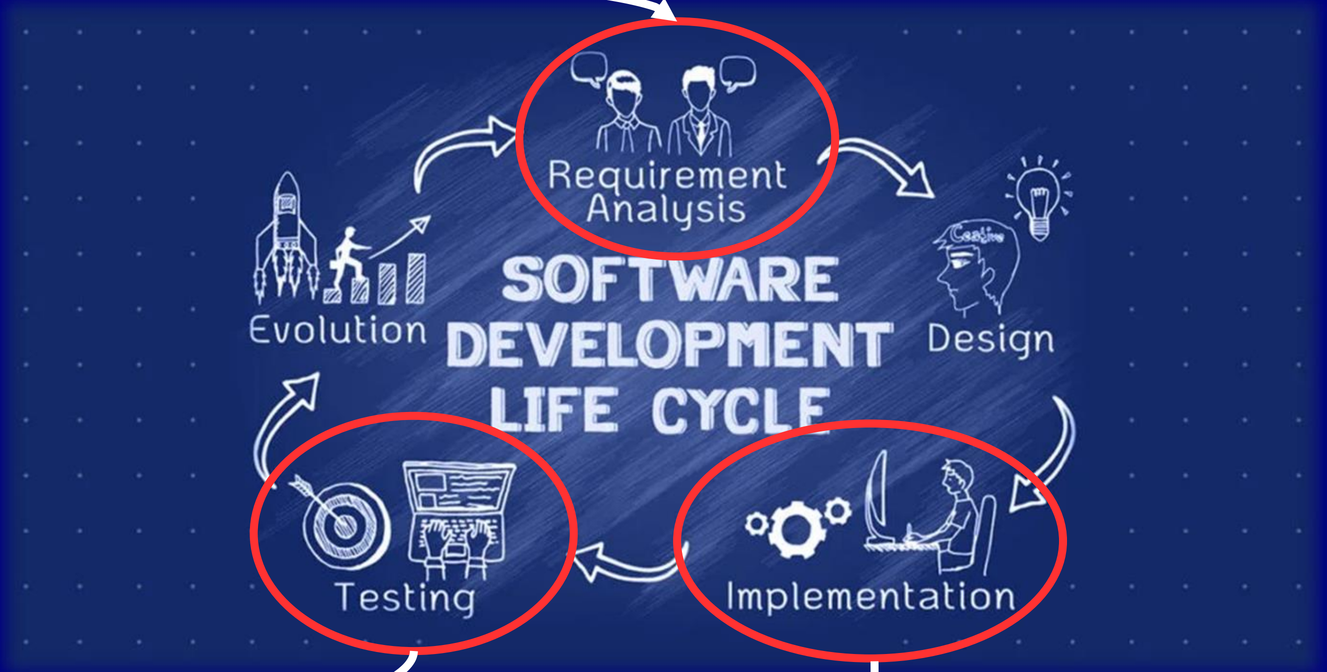
Source: Tahvili. S, Hatvani. L, *Artificial Intelligence Methods for Optimization of the Software Testing Process: With Practical Examples and Exercises*

Machine Learning

- NLP can be used to extract and analyze requirements from natural language documents, user feedback, and emails. It helps in understanding and prioritizing customer needs more efficiently.



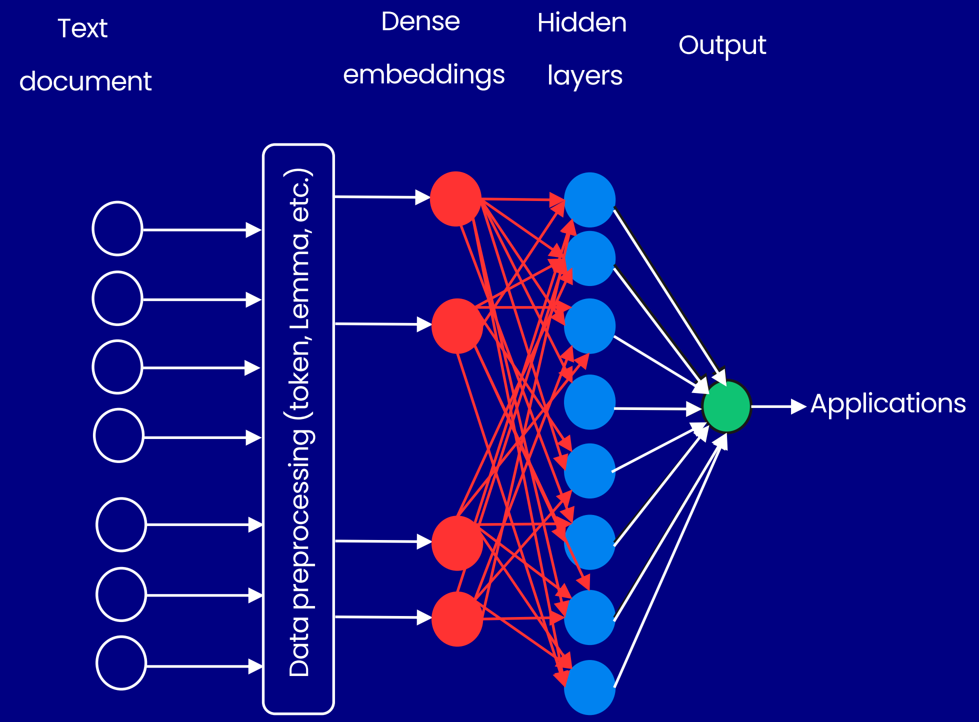
A schematic diagram of a machine learning-based approach for text mining.



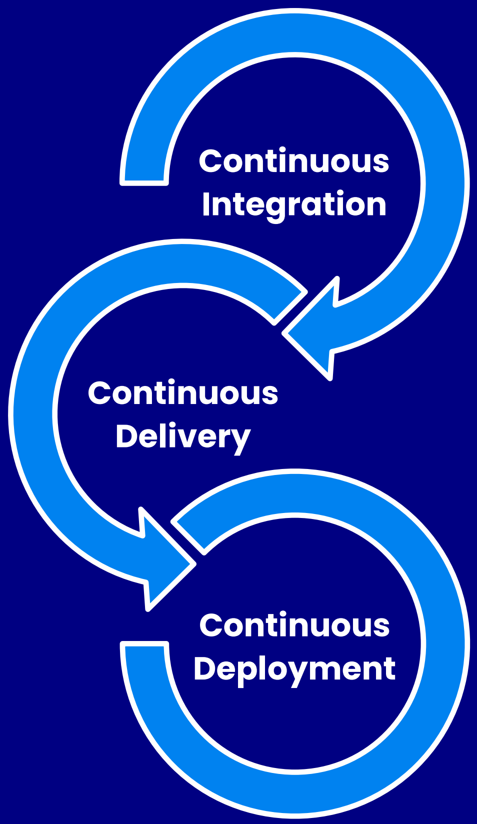
- NLP can assist in test case generation by analyzing natural language requirements and converting them into test scenarios.
- Machine Learning can be applied to automated testing, where it learns from historical testing data to identify patterns and potential issues in the software.

- Machine Learning can be used for code generation, code completion, and code review. It can suggest code snippets and help identify potential coding errors.

Deep Learning



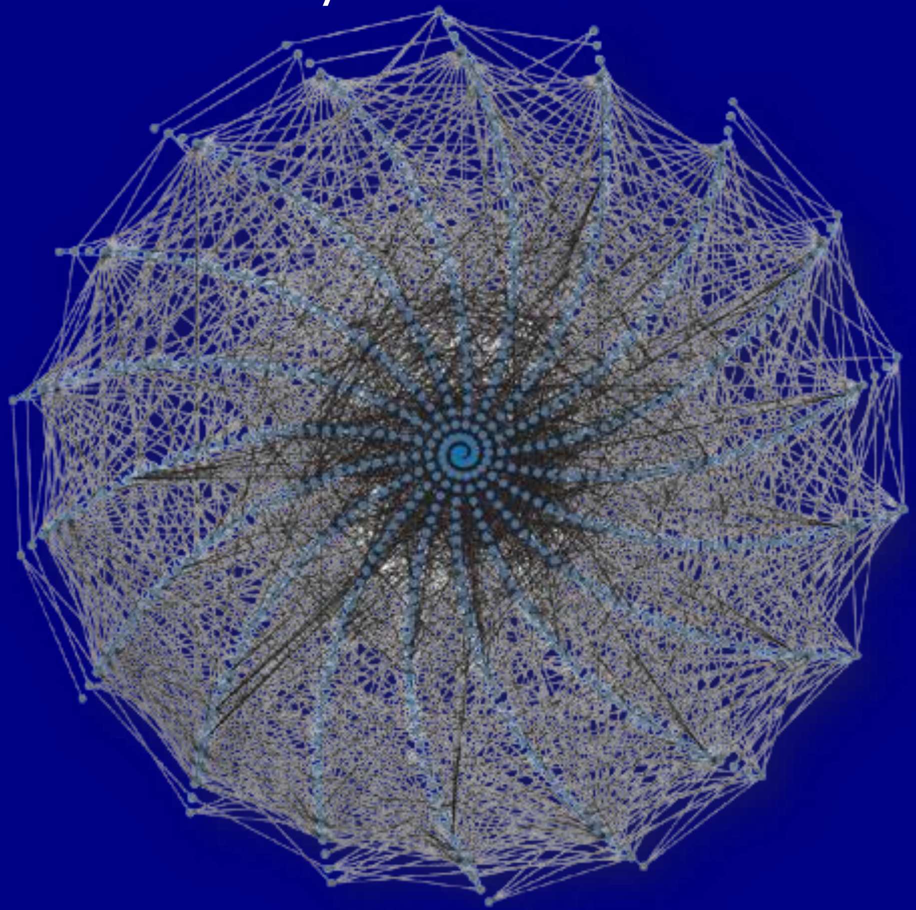
The neural system architecture of word embeddings



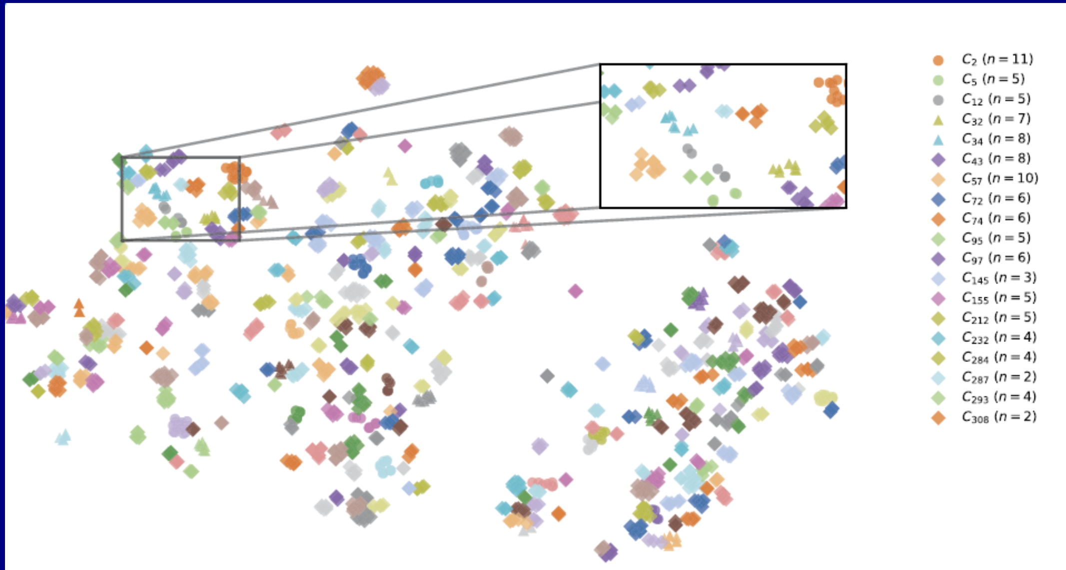
- **Continuous integration and deployment**: Deep learning models can optimize CI/CD pipelines, identifying performance bottlenecks and potential issues in the deployment process.
- **Anomaly detection in server logs**: Deep learning can be used to detect unusual patterns in server logs, helping in the early diagnosis of system problems.

Case Study 1:

- Automated Test Case Generation
- Test Data Generation
- Semantic Analysis

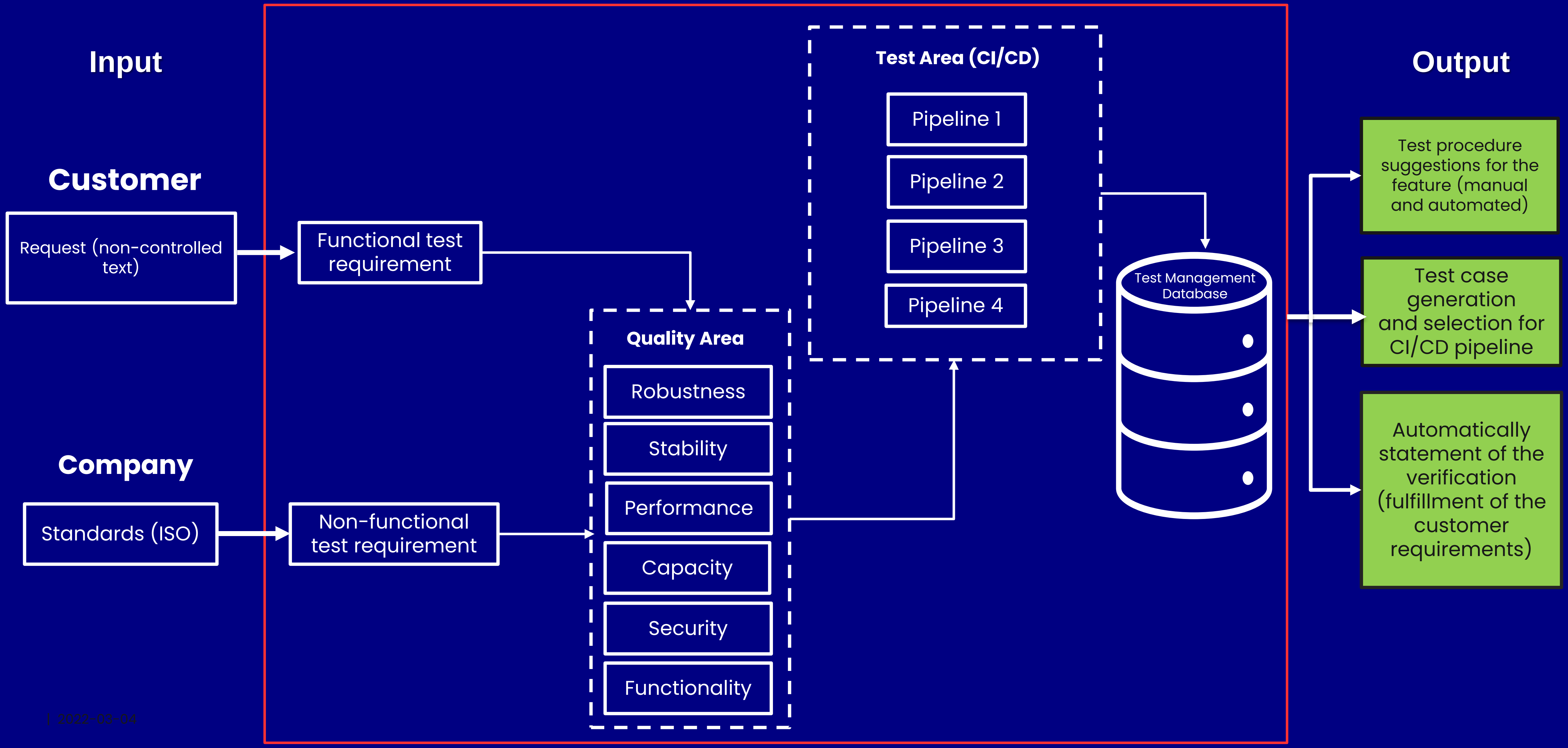


The spiral graph shows the dependencies between the requirements.



The clustered requirements specifications using the SBERT model. Minimizing test execution failure based on the dependencies up to 40%.

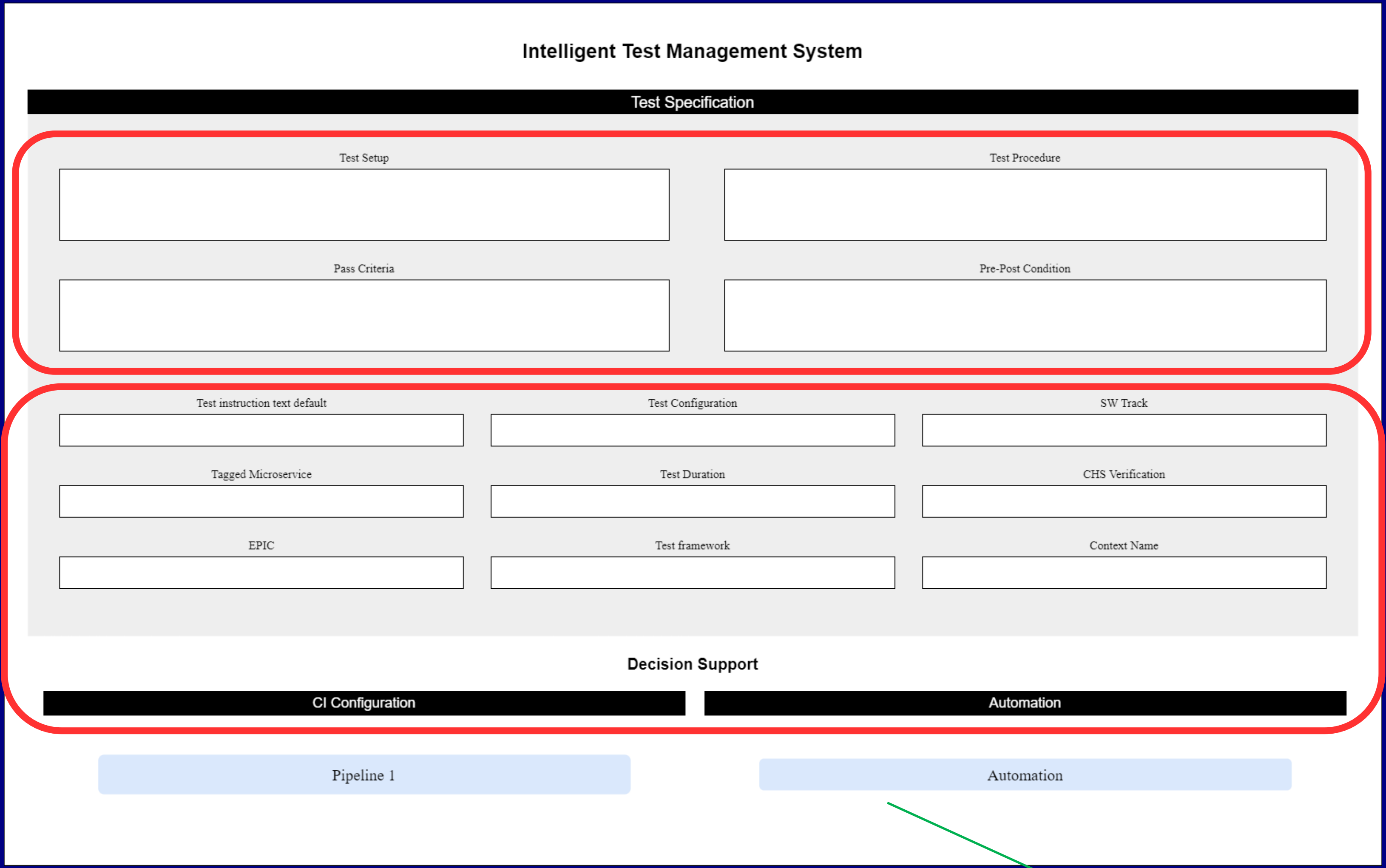
Case Study 2:
Intelligence Test Management System



Intelligent Test Management System

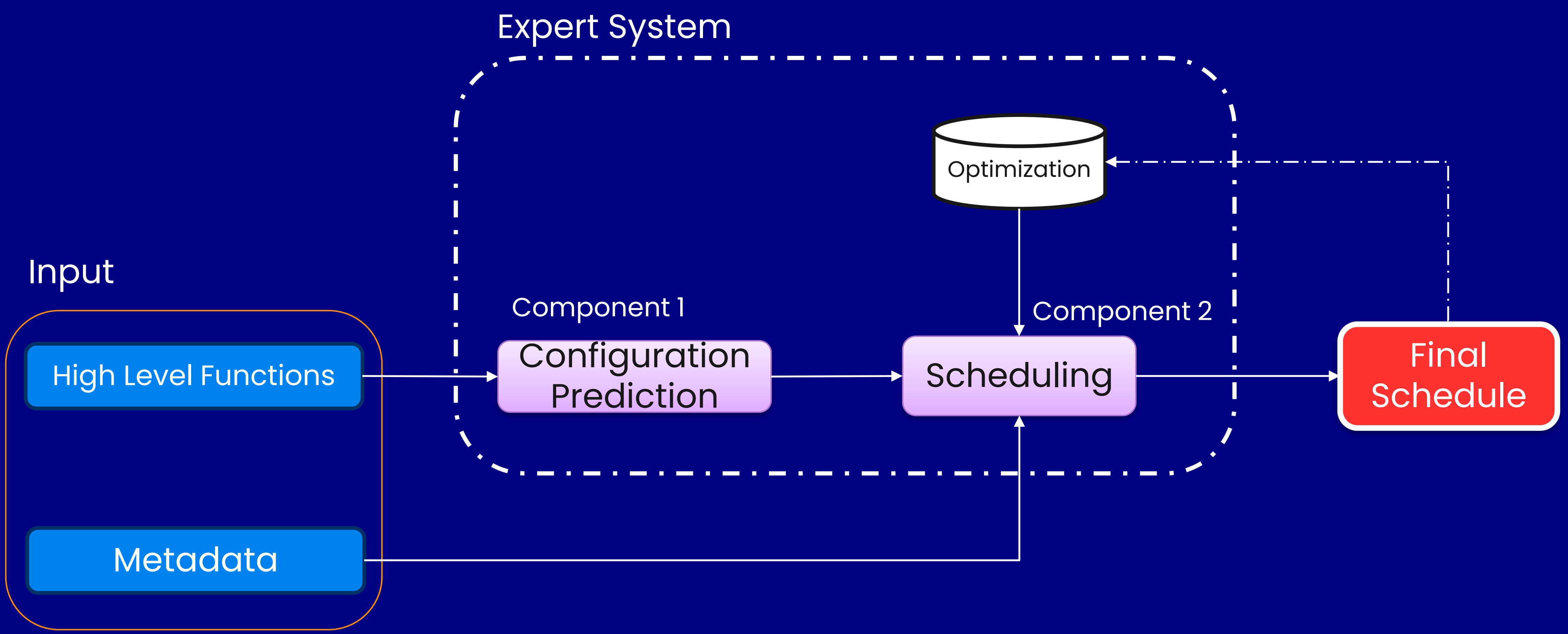
Reads multiple types of inputs (controlled, noncontrolled natural text)

The proposed solution can create a test case when the minimal requirements are specified.

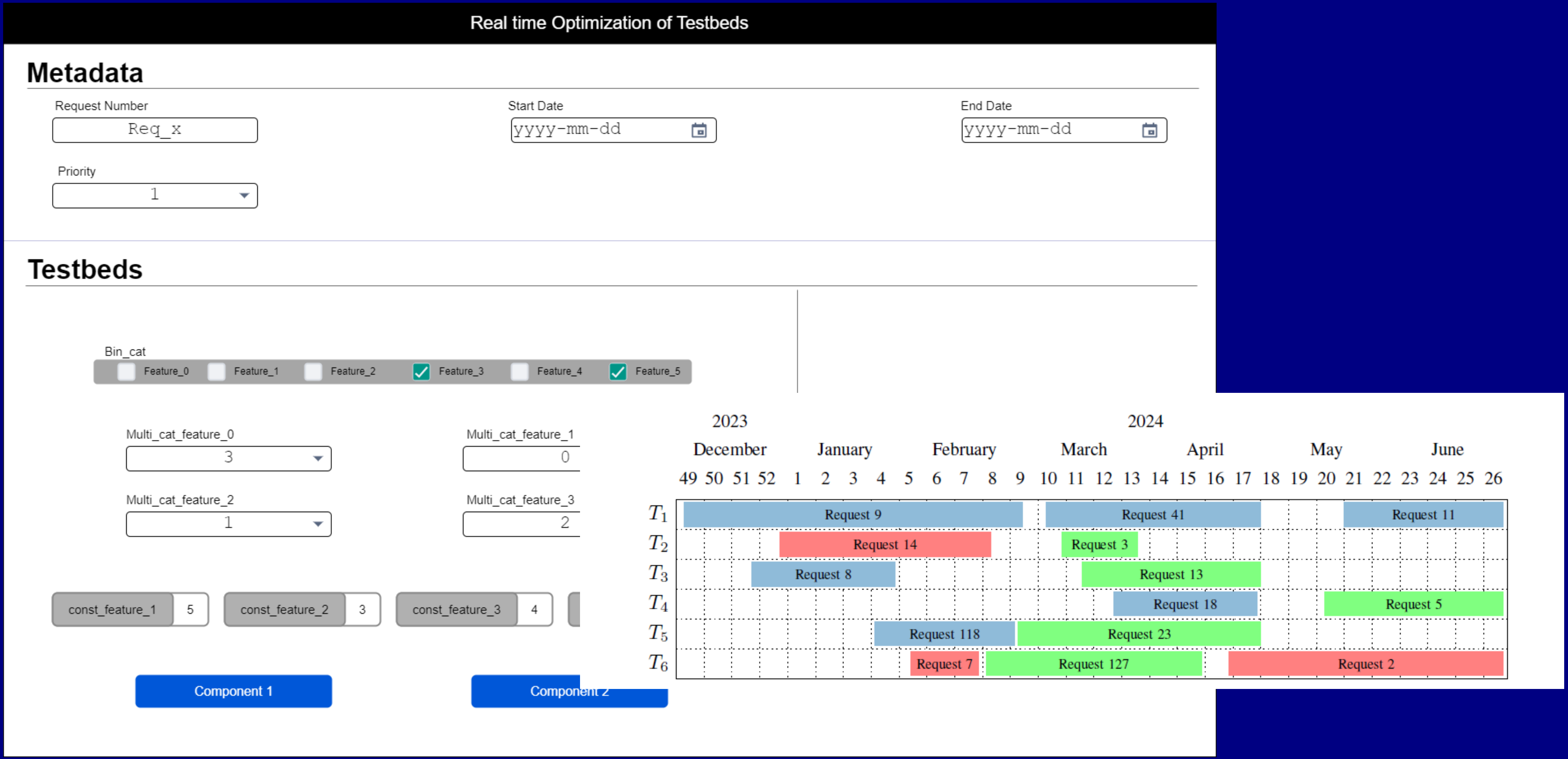


A decision maker to test automation using the **recommended pipeline** and **measuring the complexity of the inserted input by the end-user**

Case Study 3:
Dynamic scheduling and building test environments



Real-time Optimization of Testbeds for Cloudified Radio Access Networks Using Artificial Intelligence



Some of the common artificial intelligence challenges in the industry

Architecture

Architecture

- AI strategies
- Cyber security risk
- Supporting IT systems
- Integration with Cloud

Complexity

Lack of AI explainability

- Lack of skilled workers
- Operational complexity
- High computing power
- Meta learning

Training data

- Data management
- Data governance
- Insufficient data
- Inaccurate data

Data

Ongoing fee

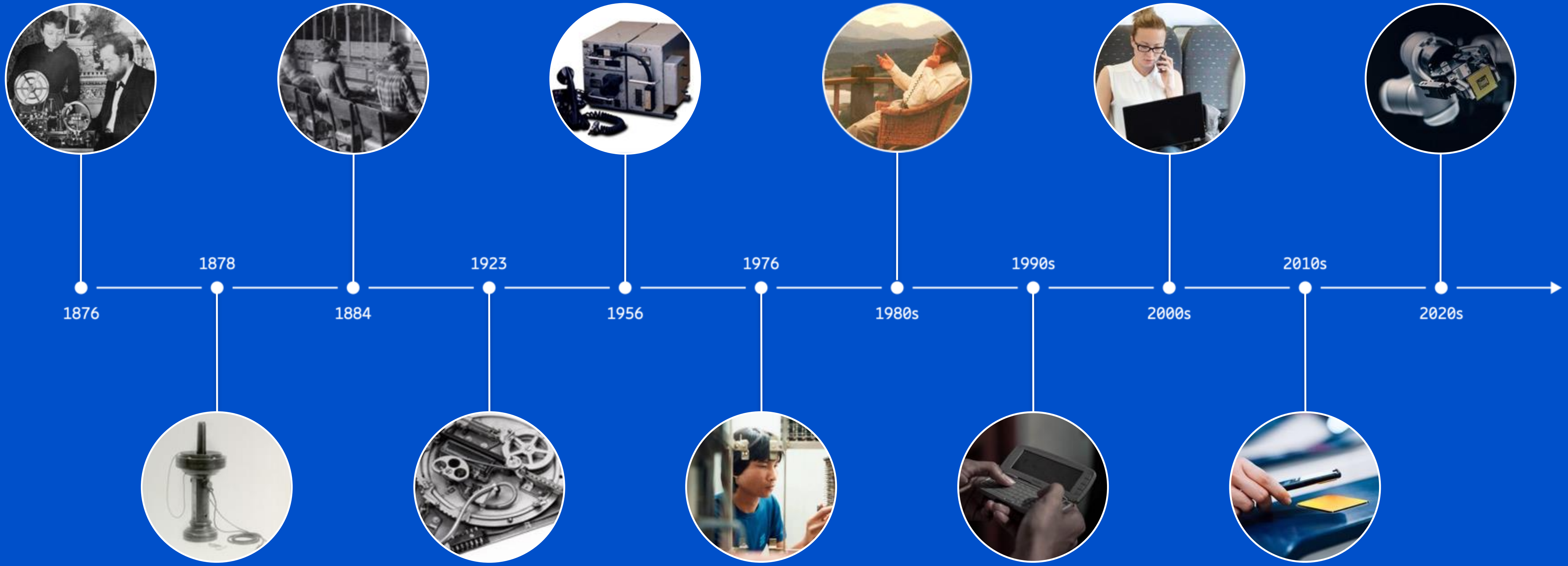
- Implementation cost
- Maintenance cost
- Quantifying ROI

Cost

Source: Tahvili. S and Hatvani. L, *Artificial Intelligence Methods for Optimization of the Software Testing Process: With Practical Examples and Exercises*

Be part of our global team at Ericsson:

*Imagine a better, brighter tomorrow, You can be the one who makes it possible, **145 years** of innovation*



sahar.tahvili@ericsson.com