# Training Material: Test Effort Estimation and Complexity Analysis in Industrial Control SoftwareLogotyp Mälardalens universitet.

**1. Introduction**

This training material explores the relationship between software complexity and test effort in industrial control systems, focusing on Programmable Logic Controllers (PLCs) programmed using the Function Block Diagram (FBD) language. The analysis is based on **TIQVA[[1]](#footnote-1)**, a specialized tool for measuring software complexity metrics such as Source Lines of Code (SLoC), Cyclomatic Complexity (CC), Halstead Complexity (HC), and Information Flow Complexity (IFC).

**2. Complexity Metrics for FBD Programs**

The complexity metrics employed in this study are tailored to the unique characteristics of the FBD language. These include:

1. **Number of Elements (NoE)**: Measures the size of the program by counting FBD components such as blocks, connections, and variables.
2. **Cyclomatic Complexity (CC)**: Represents the number of independent paths through a program, directly correlating with decision points.
3. **Halstead Complexity (HC)**: Quantifies various dimensions like program length, volume, and effort based on operators and operands.
4. **Information Flow Complexity (IFC)**: Measures coupling and cohesion by analyzing fan-in and fan-out values of software modules.

**3. Test Effort Measurement**

Test effort in this study is evaluated using two proxy metrics:

1. **Number of Test Cases**: Reflects the breadth of coverage achieved by the testing process.
2. **Test Suite Execution Time**: Indicates the time required to execute all test cases for a given program.

These proxies provide insights into testing complexity and its relationship with software characteristics.

**4. Results and Findings**

**4.1 Complexity Measurements**

The TIQVA tool computed complexity metrics across 82 FBD programs. Significant variability was observed, with some programs demonstrating notably higher complexity in terms of NoE, CC, and HC metrics. Such programs often require more extensive testing.

**4.2 Correlation Analysis**

The study found a low to moderate correlation between complexity metrics and test effort. Notably, the NoE metric showed the strongest correlation with test effort, emphasizing the importance of software size as a predictor for testing needs.

**4.3 Prediction Model**

A linear regression model was developed using the complexity metrics to estimate test effort. The model highlighted NoE as the most influential metric, achieving moderate accuracy in predicting the number of test cases and execution time.

**5. Implications**

1. **For Engineers**: TIQVA offers valuable insights into software complexity, aiding resource allocation and planning for testing.
2. **For Researchers**: The study demonstrates the potential for incorporating complexity and test effort metrics into broader software quality models.
3. **For Industry**: Provides evidence of the feasibility of using complexity metrics for test effort estimation in safety-critical environments.

**6. Limitations and Future Work**

While the correlation between complexity and test effort was moderate, other factors such as specification quality and domain expertise likely influence testing needs. Future research should explore additional metrics like entropy and coupling to improve estimation accuracy.

**7. Conclusion**

The study reveals that software complexity, especially size (NoE), has a moderate correlation with test effort in industrial FBD programs. Tools like TIQVA enable engineers to measure complexity and estimate testing needs, supporting informed decision-making in safety-critical domains. Future research should expand the metric set and explore other models to refine these estimations.

1. <https://github.com/amuslija/fbd-complexity-tool> [↑](#footnote-ref-1)