



Test Framework for Mirka Ltd

Faster implementation and simpler support of test rigs

Continuous improvement through standardization and reuse

Increased automation and reduced cost of ownership

Test Framework for Mirka Ltd

Mirka Ltd is a world leader in surface finishing technology and offers a broad range of groundbreaking sanding solutions for the surface finishing and precision industry. The Mirka motto, 'Dedicated to the Finish', states the company's high standards for all to see.

The growing Mirka Power Tools department develops and manufactures advanced sanding and polishing machines, and continually focuses on innovative product and process development. Remaining in the technical forefront, and at the same time maintaining the company's uncompromising quality requires an equally high paced development of test capabilities and coverage.



Developing test rigs from scratch for each new product, and manually controlling them would make the cost of test increase at the same pace as the increase in functionality. Such a proliferation of test rigs would also become hard to maintain and create strong dependencies on specific individuals.

In discussion with Novator Solutions, implementation of a standardized test framework would address all these challenges. Basing all of Mirka's test rigs on a common flexible architecture makes it possible to create a single learning path for new employees to acquire the knowledge needed to maintain the rigs, as well as create new ones. It would also allow for faster development of test rigs, continuous improvement of the architecture, and an ever-increasing automation grade.

Project Overview

Training

The project started with a request for LabVIEW training. An engineer from Novator Solutions held a custom course for the test and verification team at Mirka. During the training, the challenges and costs associated with thoroughly testing a rapidly evolving product portfolio – and the concept of test frameworks as a way to meet those challenges – was discussed



Workshop

A subsequent workshop was scheduled to discuss the general benefits the approach would give Mirka, and identify their specific requirements on such a framework. General requirements are modularity, flexibility and architected according to best practices, as well as specific instrumentation, sequencing and reporting needs were listed.



Implementation

The implementation of the framework and the first rig based on it was done by Novator Solutions engineers because of our experience in developing strong architectures. Mirka simultaneously trained personnel that could act as framework architects and test developers respectively.



Challenge

- Test capability can become bottleneck in development
- Test stations can also have bugs in them
- New product features forces development of new test rigs
- Cost of test increase linearly with test coverage
- Support of old rigs is difficult and expensive

Solution

- Flexible architecture possible to apply on different products for fast rig development
- Modular architecture allows reuse and thus continuous improvements to each module
- The flexibility and modularity allows expansion and repurposing of existing rigs
- Automation capabilities built in from the start, and continuously refined, to allow higher test coverage without increasing cost.
- Common architecture makes it possible to teach new employees how to support all rigs at once

Characteristics

A framework can be defined as a **reusable** codebase of **high quality** that can be **scaled up or down without limiting** its usability, and **seamlessly extended** with new functionality.

Requirements

During the course of the workshop, the following high-level requirements for the test framework were identified:

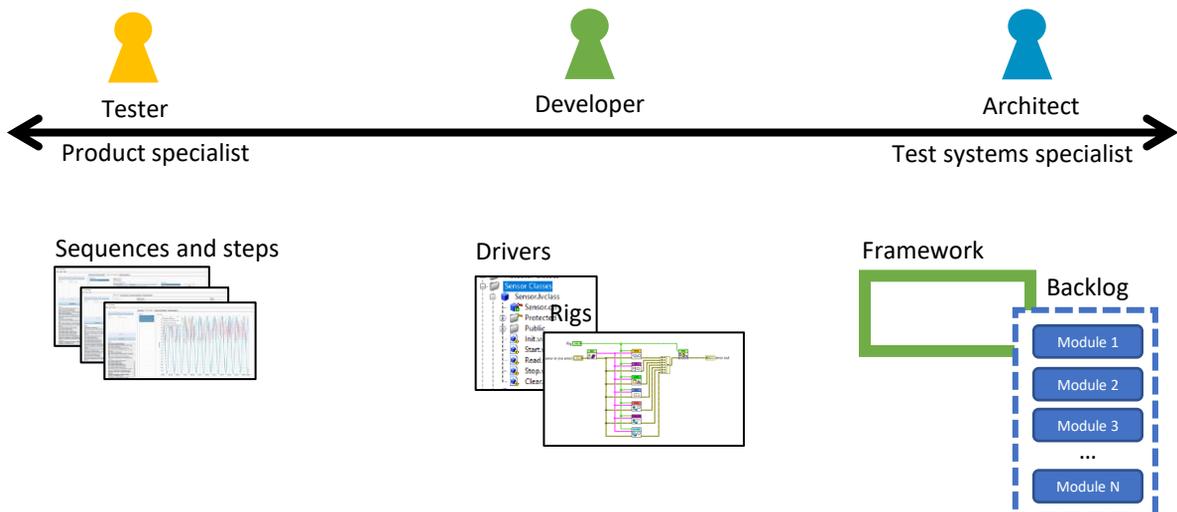
- It should be **modular** – so that individual components can be reused in different configurations
- It should be **flexible** – so that optimal hardware can be selected and test sequences can be created or adapted without changes to the overall architecture
- It should produce **user friendly** rigs, including user interfaces, data storage and stability – so that rigs created become helpful tools to product specialists that can use them effortlessly, and if needed request new features from a test system architect.
- It should be **well architected** according to best practices – so that new architects can be trained according to standardized learning paths, and that maintenance and development of the framework does not become dependent on specific individuals.

This singled out LabVIEW OOP as the ideal software tool because of the inherent object-oriented modularity, broad hardware integration, UI and data storage possibility, and existence of efficient classroom and online training.

Examples of components of a test framework are

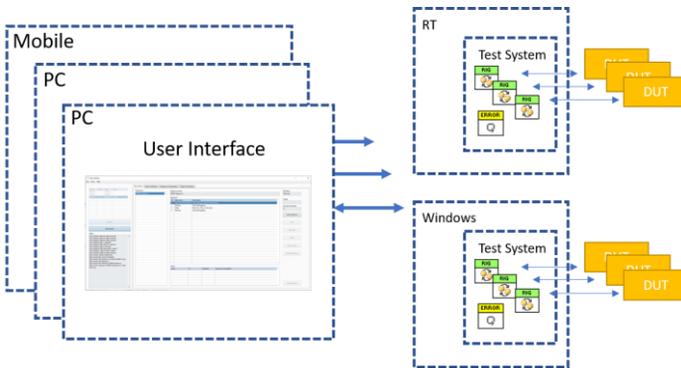
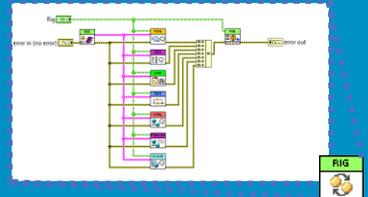
- Architectures
- Documentation
 - Example code
 - Training material
 - Development guidelines
- Communication layer
 - Inter-process communication
 - Network communication
 - Internet communication
 - Database communication
- Well packaged software, such as libraries or object-oriented classes, to solve recurring tasks
 - Hardware drivers
 - Abstraction layers (HAL, MAL)
 - Report generation
 - Test sequencing
 - Monitoring
 - Data logging
 - Storage of metadata
 - Database integration
 - User interface

It was decided that the framework should be treated as a product, not a tool. This means it will have a responsible owner, and a living backlog and bug reporting. This also makes it possible for Mirka to assign and train people for different roles in the test value chain.



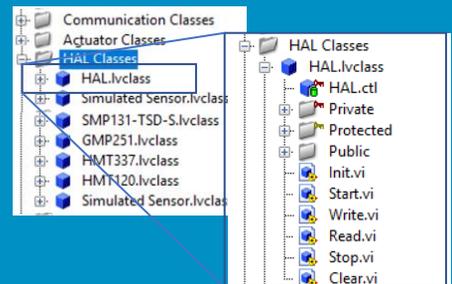
Features

A test rig consists of a LabVIEW OOP class inheriting from an overarching rig class. This rig class launches and manages communication interfaces, hardware layer, I/O-channels, logging, centralized error handling and other functionality needed.



I/O and sequencing are separated in a client-server model. This allows access to the measurements from different devices, also parallel devices. There is also the choice to deploy to either Windows or real-time targets. It also enables usage of either a LabVIEW-based sequence engine and data visualization user interface for development tests, or TestStand for enterprise integration for production tests.

Hardware is integrated through a hardware abstraction layer. Modular NI hardware is the de facto standard for general measurements. This allows for cost efficient way of avoiding downtime as spares can be kept that can be used on multiple systems. Specialized instrumentation can be integrated seamlessly when needed. Sensors built into the Mirka electric sanders are also integrated through the HAL and natively used as instrumentation in the framework.



Other features that new rigs automatically benefit from through reuse and simplified setup are

- Centralized error handling API to allow and enforce consistent error handling.
- User login for increased traceability.
- “Report abstraction layer” for quick customization of reports to fit different products, but also the possibility to easily reconfigure existing rigs if the reporting template or file format are changed.

The process for creating new rigs is

- 1) Reusing existing hardware drivers, or implementing new ones by inheriting from the HAL-class.
- 2) Reusing or developing new test steps, including control, measurements and calculations, and adding the steps to new or existing test sequences.
- 3) Configuring what data should be recorded and to where (report, files on server, database), no programming needed as long as report format has not changed.

In that way, Mirka leverages all work that has been on previous rigs today, and can continue to leverage today's work in the future.