Semiconductor Manufacturing Equipment Software

Simulator Development Capability

Mindteck

Pushing the limits of possibility

www.mindteck.com

UNITED STATES ▪ UK ▪ BAHRAIN ▪ INDIA ▪ SINGAPORE ▪ MALAYSIA

Projects

Enhancing existing simulation capability to accommodate new devices

Customer Inputs:

- Tool software and Simulator source code and documentation.
- Documentation/ product manuals of new devices to be added in the simulation.

Challenges:

- Reverse engineer existing simulator and tool software source code.
- Identify whether existing interfaces for similar devices can be re-used or new abstract interfaces need to be created.
- Create stubs for the tool & simulator interfaces to work with.
- Deliver the capability in 3 months as customer wanted to switch over to the alternate device vendor. This required all software to be shipped for complete testing on their simulators

Mindteck's Approach:

- Engineer deputed early to customer's onsite location to get an understanding of all the device operations and extracting the tool and simulator interfaces and prepare the requirements document and high level design document for the offshore team.
- Offshore team starts work with the tool interface and prepares stubs to abstract the tool and simulator.
- Delivered 1st incremental release to customer with basic device capabilities 6 weeks from start date.
- Delivered 2nd incremental release to customer with 90% device capabilities 9 weeks from start date.
- Delivered the final release in 11 weeks after integration testing on the customer's simulator.

Benefits:

- Delivered 1 week ahead of delivery schedule.
- Customer had the simulation capability for the new devices in place before shipping the tool with new devices to end customers.

Developing fault injection feature on the customer's existing simulator

Customer Inputs:

- Tool software and Simulator source code and documentation.

Challenges:

- Reverse engineer existing simulator and tool software source code.
- Derive a list of devices which requires fault injection capability.
- Provide an additional framework to the simulator to setup external programmable logic to sequence and trigger the fault injection as customized by the user.
- Deliver the enhancement in 5 months as the customer needs to test next release alpha builds on simulated scenarios of wafer scrap issues recorded earlier at customer fabs to prevent wafer scrap issues in the field.

Mindteck's Approach:

- An engineer is deputed early to customer's onsite location to derive a list of devices requiring fault injection capability and understand the nature and scenarios of fault injection.
- Offshore team works on providing the additional framework for fault simulation.
- Independent fault injection module is delivered as a demo 8 weeks from start date.
- Team work on enhancements suggested in the fault injection capabilities. 2nd interim release of independent fault injection module is delivered 12 weeks from start date.
- Team works on integrating the independent fault injection module with the existing simulator. 1st Integrated release is delivered 16 weeks from the start date.
- Final release containing bug fixes and minor enhancement delivered to customer 21 weeks from the start date.

Benefits:

- Customer had the fault injection capability to reliably test alpha software for specific wafer scrap scenarios prior to testing on tool and releasing software to customers.
- Got repeat order for developing graphic editor for developing simulation code.

Developing a graphic editor to help engineers fine-tune/create new simulation logic with ease on the customer's existing simulators

Customer Inputs:

- Tool software source code, Simulator source code and documentation.

Challenges:

- Deliver the graphic editor capability to customer in 4 months
- Reverse engineer existing simulator and tool software source code.
- Identify devices being simulated
- Model graphic elements for the devices being simulated.
- Develop a graphic editor which would allow users to use the widgets representing simulation devices and set up their simulation logic by graphically interconnecting them and storing the resulting simulation logic in an XML file.
- Modifying the existing simulator framework to read the XML file as per configuration and set up the simulator operation.

Mindteck's Approach:

- An engineer is deputed early to customer's onsite location to derive a list of devices being simulated and work on customer requirements for the graphic editor.
- Offshore team 1 works on with a RAD tool to quickly create a GUI prototype for demo in parallel.
- Prototype is demoed 3 weeks from start date.
- Enhancements/additional requirements are captured and final GUI prototype is demoed by team 1, 5 weeks after the start date.
- Offshore Team 2 works in parallel modifying the existing simulator framework to read a static XML file and set up simulator operation as per the embedded logic. Team 2 delivers first incremental release 6 weeks from the start date.
- Team 2 delivers the 2nd incremental release 9 weeks from the start date.
- Team 1 delivers the first GUI editor release 12 weeks from the start date.
- Offshore teams 1 and 2 deliver the 1st integrated release 16 weeks from the start date.
- Offshore team fixes bugs from previous releases, incorporates minor enhancements and delivers the final integrated release 18 weeks from the start date

Benefits:

- Customer got the flexibility to model simulation logic and create variations based on different tool configurations. This effectively increased the simulation coverage for the tool for different configurations.

Developing simulation capability from ground up for existing tools without simulation framework

Customer Inputs:

- Tool software source code and documentation.

Challenges:

- The customer had recently acquired a small wafer processing equipment manufacturing company – and this brought a couple of new tools to the customer's portfolio. These tools did not have simulation capability and required complete simulation capability in 6 months' time.
- The challenge was to develop a generic simulator from ground up which could be re-used for the new set of tools.

Mindteck's Approach:

- An engineer was deputed to the customer's onsite location to get acquainted with the tool hardware/software and device operation.
- The onsite engineer prepared a list of devices requiring simulation.
- The onsite engineer reverse engineered existing tool software to extract device interfaces and documented the same for the offshore team.
- The offshore team created stubs for tool interfaces and started work on building generic device simulator.
- The offshore team modeled simulator basic building blocks and architected the generic simulator to be scalable using these basic building blocks.
- First incremental release was delivered 8 weeks from start date with basic simulator building blocks and demo simulation logic (which was hardwired).
- The offshore team then created a generic client to interface with the tool software.
- Second incremental release contained integration of generic simulation client with the tool software. This was delivered 13 weeks from start date.
- Thereafter the offshore team added capability to create and interconnect simulation building blocks modeling the devices at runtime from a static XML based configuration file created manually.
- Third incremental release added the capability to create and interconnect simulation building blocks modeling the devices using from the XML based configuration file. This was delivered 17 weeks from the start date.
- Fourth incremental release added simulation support to the remaining devices. This was delivered 24 weeks after the start date.
- Final release with minor enhancement and bug fixes was delivered to the customer 28 weeks from the start date

Benefits:

- Customer was able to deliver the generic simulator for the new product 2 weeks before the commit date.
- Mindteck got a repeat order for building a simulator for another variant of this tool. This was done in half the time and at 40% of the cost since the existing simulation framework was reused to deliver this project.