

# Integrating UUT Prognostics into ATML- and SIMICA-Based Test, Diagnostic and Maintenance Processes (Rev G)

## 1671.3

- Describes the UUT components, functions, and failure modes that are covered by diagnostics and **prognostics**
- Describes the characteristics of signals measured or monitored for test, diagnostics, and **prognostics** (IEEE 1641)
- Specifies the location (at the UUT interface) of signals measured or monitored for test, diagnostics, and **prognostics**
- Optionally, contains references to the UUT Description documents for the UUT components

## 1671.1

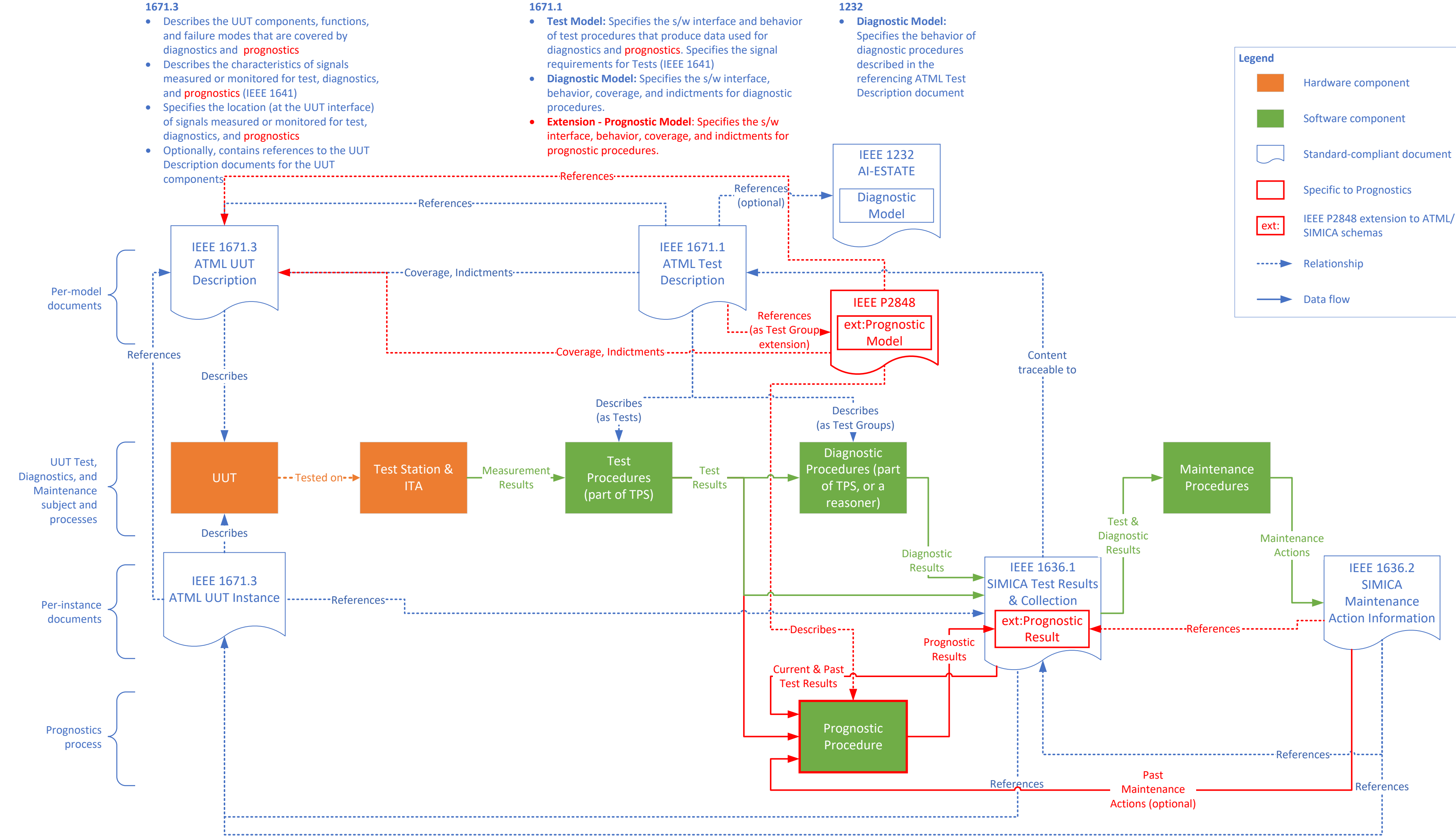
- Test Model:** Specifies the s/w interface and behavior of test procedures that produce data used for diagnostics and **prognostics**. Specifies the signal requirements for Tests (IEEE 1641)
- Diagnostic Model:** Specifies the s/w interface, behavior, coverage, and indictments for diagnostic procedures.
- Extension - Prognostic Model:** Specifies the s/w interface, behavior, coverage, and indictments for prognostic procedures.

## 1232

- Diagnostic Model:** Specifies the behavior of diagnostic procedures described in the referencing ATML Test Description document

### Legend

- Hardware component
- Software component
- Standard-compliant document
- Specific to Prognostics
- ext: IEEE P2848 extension to ATML/SIMICA schemas
- Relationship
- Data flow



## 1671.3

- Identifies a specific UUT instance (ex. by Serial Number)
- Referenced by the SIMICA documents describing the test, diagnosis, **prognosis**, and maintenance history of the UUT instance
- Optionally stores references to SIMICA TestResults documents that record the **estimated RUL** for the UUT instance, for its components, functions, and failure modes
- Optionally, contains references to the UUT Instance documents for the UUT components (ex. WRAs/SRUs) that are installed at a given moment in time

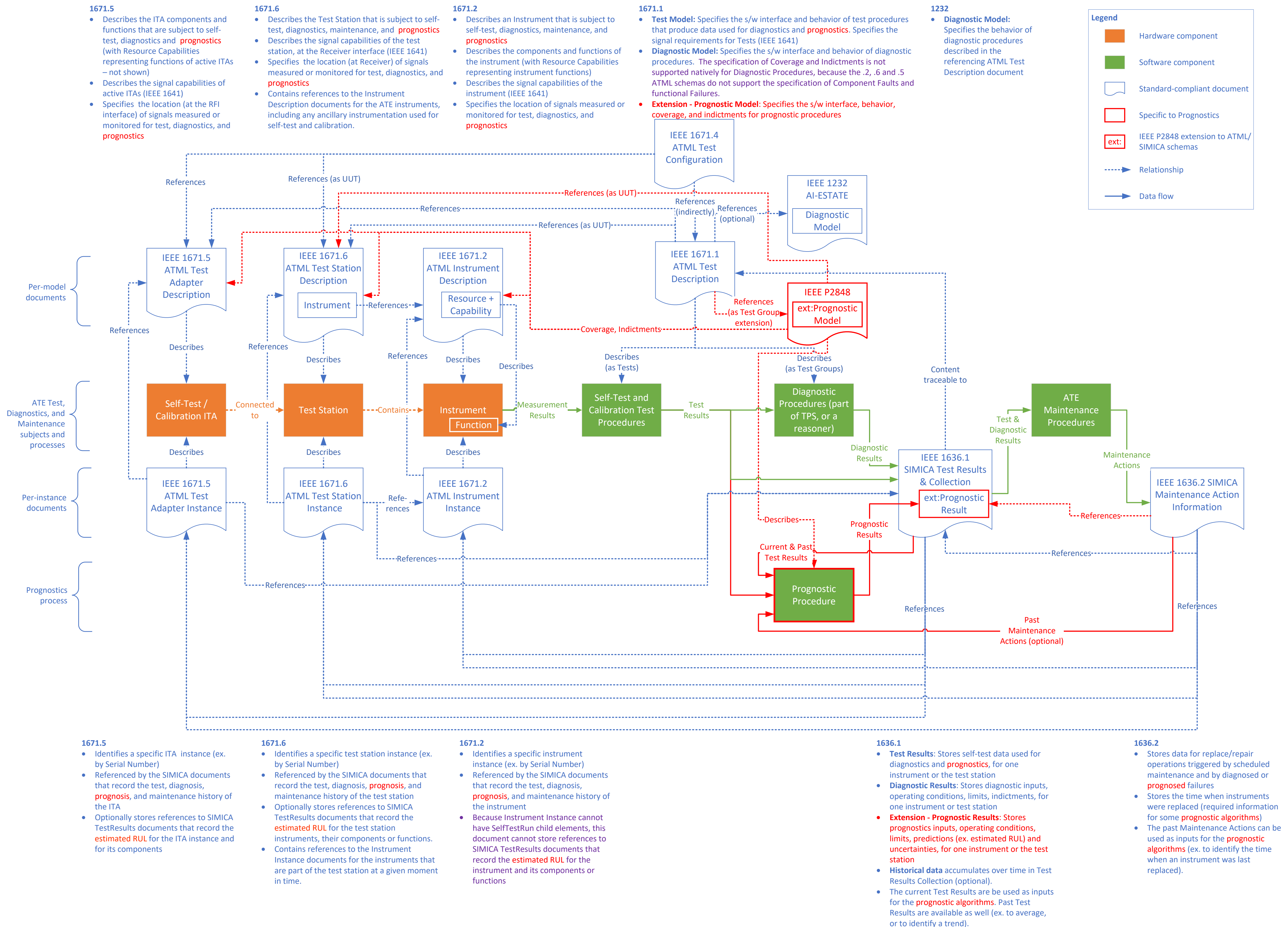
## 1636.1

- Test Results:** Stores UUT test data used for diagnostics and **prognostics**
- Diagnostic Results:** Stores diagnostic inputs, operating conditions, limits, indictments
- Extension - Prognostic Results:** Stores **prognostics inputs, operating conditions, limits, predictions (ex. estimated RUL) and uncertainties**
- Historical data** accumulates over time in Test Results Collection (optional)
- The current Test Results are be used as inputs for the **prognostic algorithms**. Past Test Results are available as well (ex. to average, or to identify a trend).

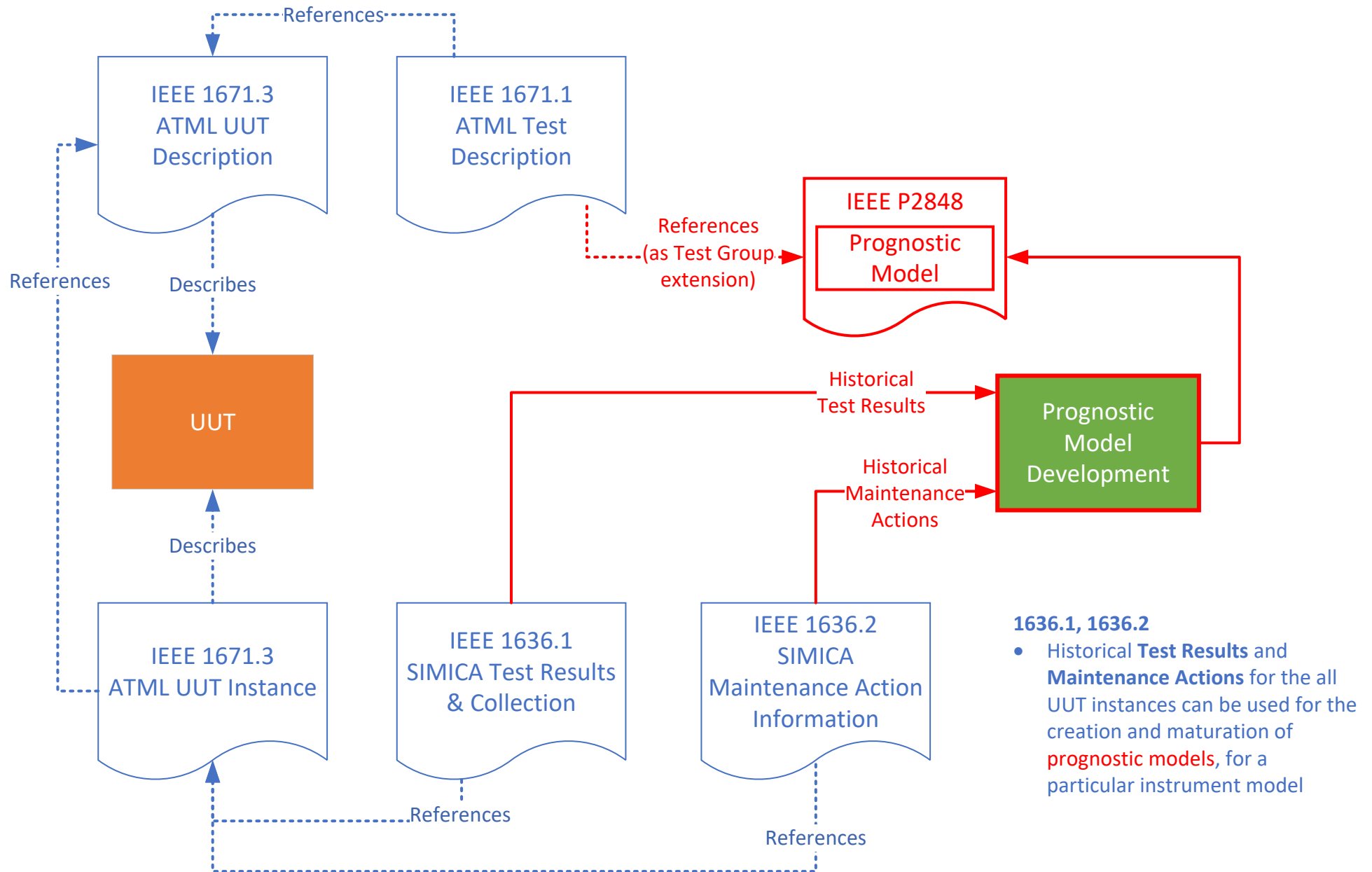
## 1636.2

- Stores data for replace/repair operations triggered by scheduled maintenance and by diagnosed or **prognosed** failures
- Stores time when UUT components were replaced (required information for some **prognostic algorithms**)
- The past Maintenance Actions can be used as inputs for the **prognostic algorithms** (ex. to identify the time when a part was last replaced).

## Integrating ATE Prognostics into ATML- and SIMICA-Based ATS Self-Test, Diagnostic and Maintenance Processes (Rev G)



# Offline Creation & Maturation of the Prognostic Model for UUT (Rev G)



## Revision History

Rev D, 2023-03-08, Ion Neag – First public release

Rev E, 2024-02-24, Ion Neag – updated after review on 2024-02-13; merged into this document the diagram describing the prognosis of the ATE / instruments; changed HardwareItemDescription to TestAdapterDescription; added Test Configuration; separate representations of relationships and data flows; changed labels for better accuracy; revised all text for consistency; removed ATE description data from UUT process; added Coverage and Indictments relationships

Rev F, 2024-04-10, Ion Neag – updated after Eric's review. Changed terminology for the inputs of the Prognostic Procedure, to remove the implication that historical test and maintenance data are used at run time for Prognostic Model creation or maturation. Added new diagram for the offline creation & maturation of the Prognostic Model.

Rev G, 2024-04-11, Ion Neag – separated PrognosticModel into a separate instance documents, conformant to schemas to be defined in the P2848 standard. This is the most likely solution, allowing the model to be updated without revising the Test Description.