

## White Paper



### Providing an end-to-end quality assurance process for products

---

Piyush Jain, Roopa Devraj

In today's competitive times cost alone is not going to be a factor behind the success or failure of a product. It is going to be Product Quality. A high quality product will ensure long term relationship between the supplier and the customer ensuing business benefits for both the parties.

Today, management of product quality has become an integral part of product development lifecycle. Every product organization is laying stress on building an effective process that is well supported by systems to enforce stringent quality management of their products.

This paper will explain one such system that provides an integrated and comprehensive approach to product quality management using a workflow based model. The system has been an initiative of the Product Lifecycle Management Center of Excellence group at Infosys, Inc.

## What do Current QA Systems Lack?

Given the bearing quality has on product's success, every product company has invested much resources and money in setting up QA processes and systems that would be sufficient for their needs. However, most companies do not have an integrated product quality assurance system that ensures end-to-end QA process for their products. Most of them have disparate systems to take care of different aspects of QA process such as test management, lab management, release management and customer information management. This disparity leads to numerous issues such as:

- Coordination effort required to ensure smooth flow of information between systems
- Much time spent maintaining the sanity and integrity of data in each system
- Overhead on the maintenance of multiple systems
- Duplication of effort in logging information into different systems
- Missing critical information, which can have a detrimental effect on end-customer deployment.

All of the above can render even the best of processes and systems ineffective in assuring overall quality for the product. Product Quality Management System (PQMS) overcomes the disparity by integrating the key ingredients of the QA processes into one system.

## Introduction

PQMS is an integrated system that supports all the facets of QA process. It is designed on a workflow-based model and can be easily customized to suit the QA processes of the organization.

## Workflow Engine

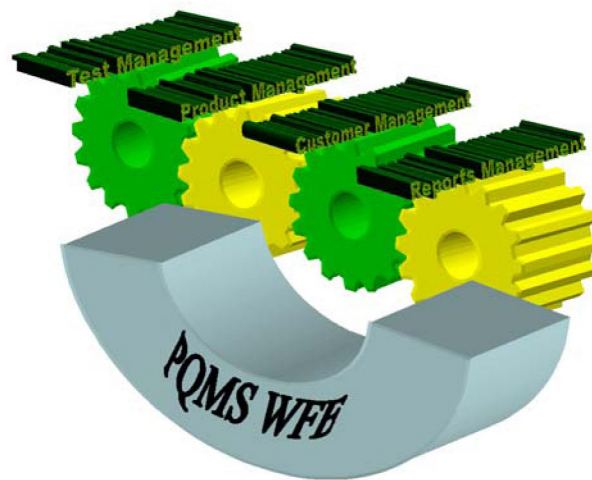


Figure 1 – The PQMS Workflow Engine Consists of Four Components

PQMS has been designed with a versatile workflow engine that provides multi-dimensional support to the entire product quality assurance stage of the PDLC as exhibited in Figure 1 above.

The workflow engine binds the four essential components of the system involving test management, product management, customer management and reports management. The workflow can be customized to suit the QA process needs and provides automated support for end-to-end tracking of the QA process.

## High Level Architecture

PQMS supports four main sub-systems namely [Test Management \(TM\)](#), [Product Management \(PM\)](#), [Customer Management \(CM\)](#) and [Reports Management \(RM\)](#). These sub-systems combined provide an end-to-end quality assurance process coverage for the product.

The flexible architecture of the system allows integration with other sub-systems like automation and defect tracking. It also can be enhanced to provide interfaces for integration with other 3rd party tools and products.

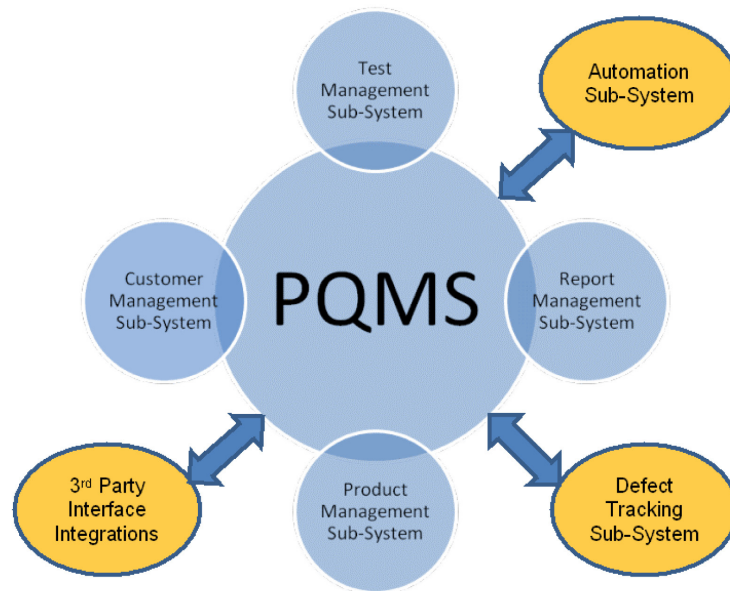


Figure 2 - PQMS High Level Architecture

The following subsections briefly explain the four main sub-systems of PQMS:

### 1. Test Management Sub-System (TM)

This is the core module of PQMS. It has been designed to provide a complete test management support. Its architecture has been defined to support four essential functional activities as depicted in Figure 3 below:

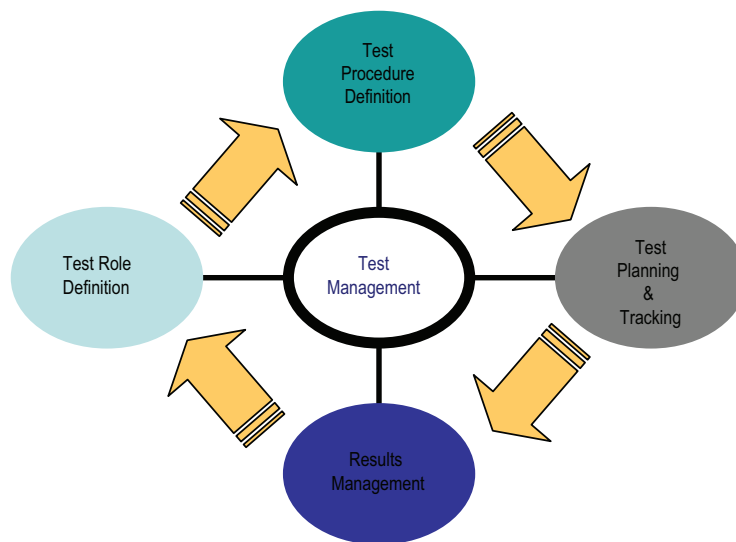


Figure 3 - Test Management sub-System Model

#### Test Role Definitions

This module allows the definition of roles and responsibilities mapping the QA organization structure. The roles can be defined depending upon their intended interface to the system and are classified into two types:

- Active Roles

Roles typifying **Test Manager**, **Test Leads** & **Test Engineers** who are active users of the system for all QA activities are termed as Active Roles.

- Passive Roles

Roles like [Product Manager](#), [Software Development Manager](#), etc. are intended for passive usage of the system in terms of data views and reports and so these are termed as passive roles.

Once the roles and their corresponding responsibilities (access rights) are defined, they are linked through the workflow engine so as to ensure the end-to-end workflow of tasks amongst the roles.

## Test Procedure Definitions

This module allows the creation and maintenance of test procedures which can be defined to contain, but not limited to, the following details:

- Information on feature functionality to be tested
- Information on platform/product to be tested
- Information on customer specific requirement, if any
- Overall setup details with diagram and key notations
- General setup configuration
- List of test cases for the said feature/functionality
  - Test cases contain the following:
    - Purpose of the test case
    - Specific setup settings for the test case
    - Specific configuration settings for the test case
    - Test procedural steps
    - Expected results
    - Actual results
    - Specific information, if any

Test cases are collated in the form of test procedures documents and are tagged with test procedure IDs. These test procedure IDs provide the linkages between test cases procedure headers stored in the test management information base and the actual test procedure documents that are checked into a version controlled system.

The module also provides linkages to the test management module for planning and scheduling of test cases under a test plan for a particular software release.

## Test Planning & Tracking

This module provides the support for:

- Test Planning
- Effort Estimation
- Test Scheduling
- Tracking and Monitoring

This module forms the core of the Test Management process, which facilitates the testing of the release pertaining to a set of customer(s)/product platform(s). The release plan can be defined to cater to different types of testing ranging from regression, sanity, stability, etc.

The module enables the tracking of the variance between the actual efforts versus the estimated efforts for a given release.

The module follows a workflow-based approach to test planning and execution, which can be customized both for the roles/responsibilities and the quality compliance procedures defined for the engineering department.

A high-level test process workflow is depicted in Figure 4 below:

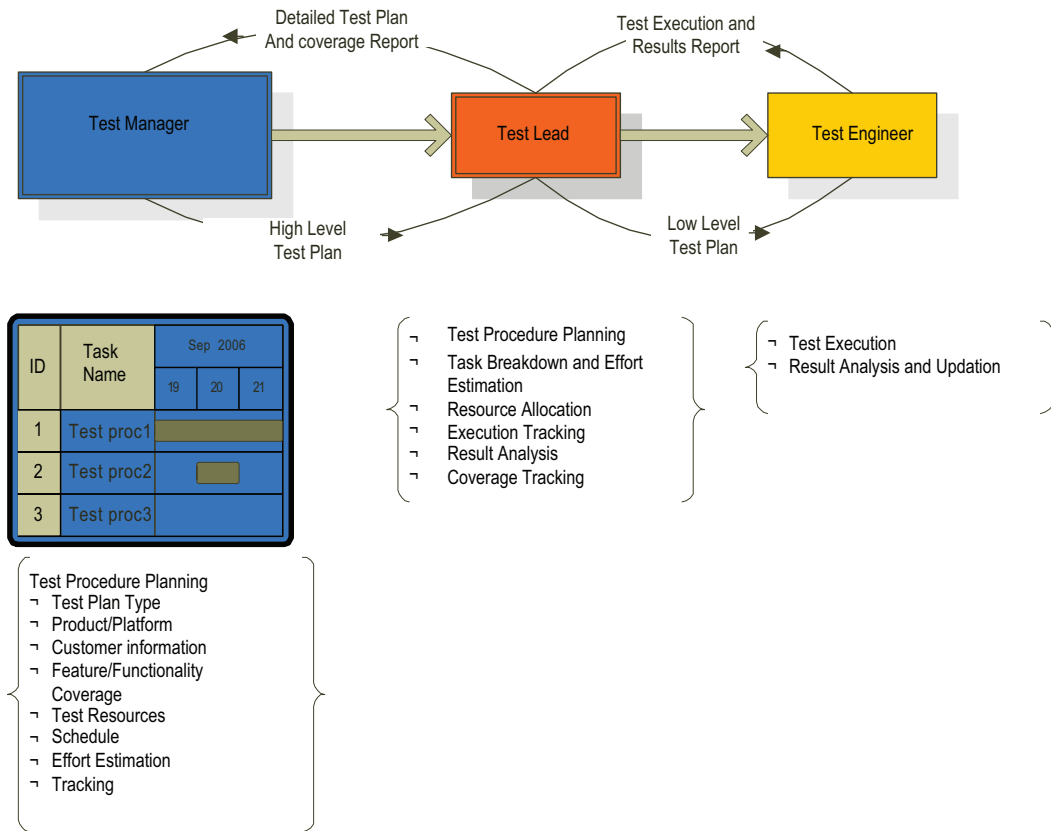


Figure 4 - Test Process Workflow

## Results Management

This module captures the results of the tests executed on a release for a particular product/platform. The module helps in assessing the coverage and effectiveness of testing through the categorization of the results in different ways as shown below.

- Total test cases executed
- Total Pass/Failed/Blocked test cases
- % coverage overall
- % coverage product/platform/module wise breakup
- % failure for the release
- Total issues/bugs reported with severity wise breakup
- Issues/bugs product/platform/module wise breakup
- Repeat issue/bugs list

## 2. Product Management Sub-System (PM)

This sub-system is designed to provide a comprehensive product inventory and configuration management support. One of the essential requirements of any product quality management process is an effective and efficient management of the lab inventory, network models and pertinent device configuration management. This sub-system has been designed to meet this requirement and its key functionalities are listed below.

### Inventory Management

This module provides the following features:

- Maintains comprehensive inventory of all lab equipment
- Maintains inventory to the detail of slot card/ports level
- Reservation of devices usage on setups. Reservation is provided till the card/port level
- Reservation support for 3<sup>rd</sup> party test tools connected to the setups

Future enhancements planned are to have an auto-discovery mechanism built, which can enable the system to automatically detect and gather all relevant information of a network element as soon as it is connected to the network and interfaced with the system.

### Configuration Management

Standard configuration for each device in the lab is stored and maintained through this module. Having standard configurations available on hand saves much time in pre-configuring the devices for well-defined test runs. This also is an essential requirement for the automation framework to enable automated run of tests on the test setup.

### Configuration Analyzer

Coupled with the configuration management, the sub-system also supports a configuration analyzer tool. This versatile tool provides the following features:

- Configuration Parser and Error Detector
  - Parses the configuration provided by the customer and provides details such as:
    - Hardware chassis
    - Card type
    - Port type
    - Number of calls
    - Type of calls
    - Quality of Service, QoS information
  - Detects errors and sanitizes the configurations against standard configurations before they are loaded on the respective devices
- Standard configuration templates
  - Provides bare minimum and default configuration settings.

### Automation Framework Integration

The PQMS system has provisions to integrate the product management and test management sub-systems with an automation framework sub-system. The automation framework could be indigenously developed or be provided by a 3<sup>rd</sup> party.

The integrated system ensures end-to-end automated testing support. Once the test cases are planned and scheduled in the test management sub-system, they invoke the automated sub-system APIs to interface with product management sub-system to reserve and configure the required lab setups. The automated sub-system can then initiate automated test runs and collate the required results.

The architecture of an integrated sub-systems is shown in Figure 5 below:

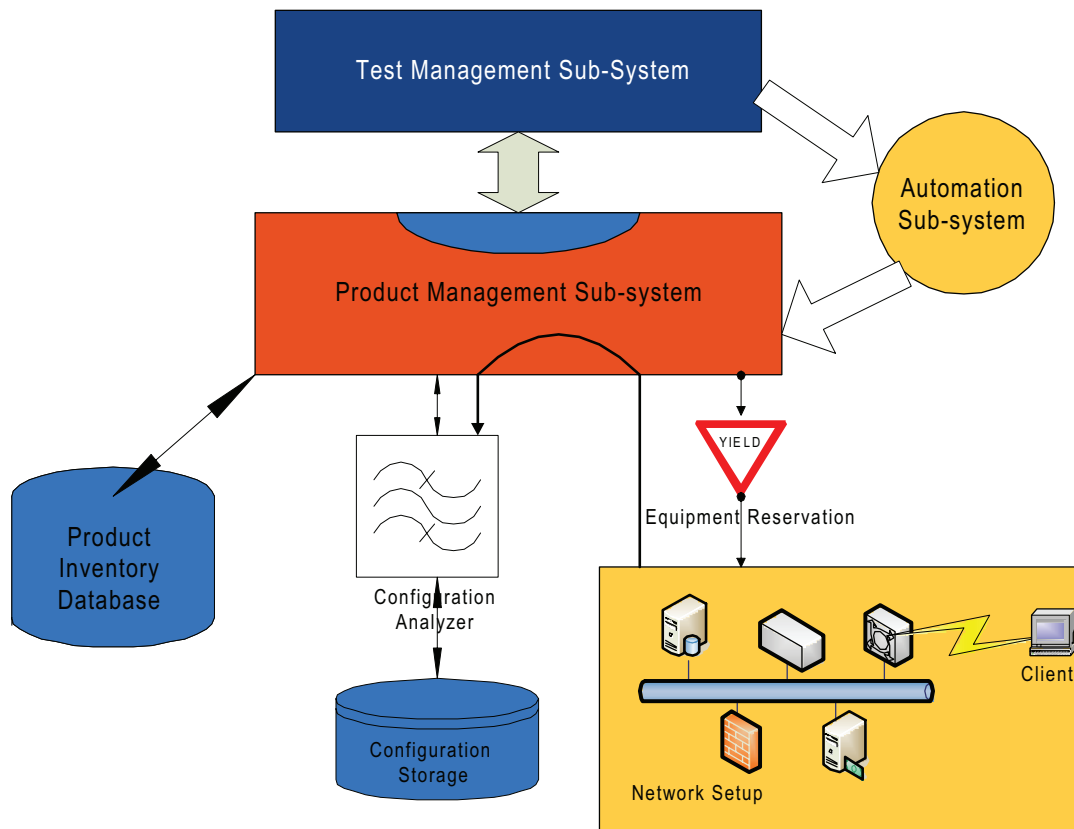


Figure 5 - PM sub-System Architecture

### 3. Customer Management Sub-System (CM)

This sub-system has been specifically designed to map testing requirements to customer defined features. It provides automatic planning of testing procedures for all customer-related features and their deployment scenarios on a release that is targeted for that particular customer.

The customer specific information is derived from various sources such as:

- Features requirements specified by customers
- Information gathered from trials/demos
- Trouble Reports (Field issues) filed by customers

Test procedures are then defined or enhanced to encompass this information.

This sub-system also stores information on the causal analysis of customer-filed problem reports. Causal analysis helps bring out the deficiencies in test coverage that lead to problems being detected in the field. Test scenarios identified during the causal analysis are then added or modified as necessary. This enables a more effective and robust test coverage for future product release.

A high level architecture for the sub-system is shown in Figure 6 below:

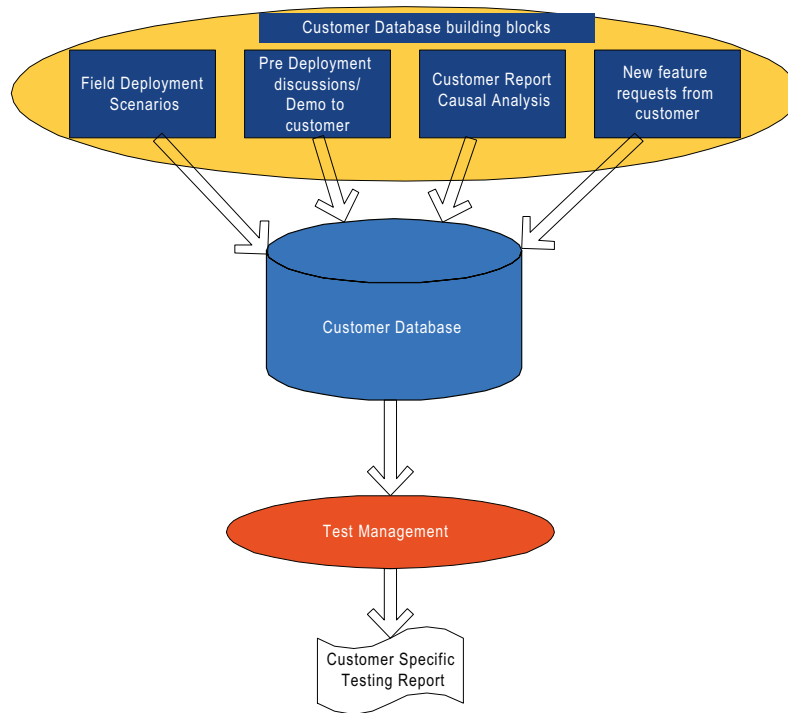


Figure 6 - CM sub-System Architecture

#### 4. Reports Management Sub-System (RM)

Reporting is one the most critical aspect of the system as it provides the interface to various stakeholders on the quality of the product. Different groups perceive different information from the testing process for decision making. At present, the system provides a 3-tier reporting model catering to the three levels of test roles that are involved in the usage of the system. A pictorial representation of the model is shown in Figure 7 below:

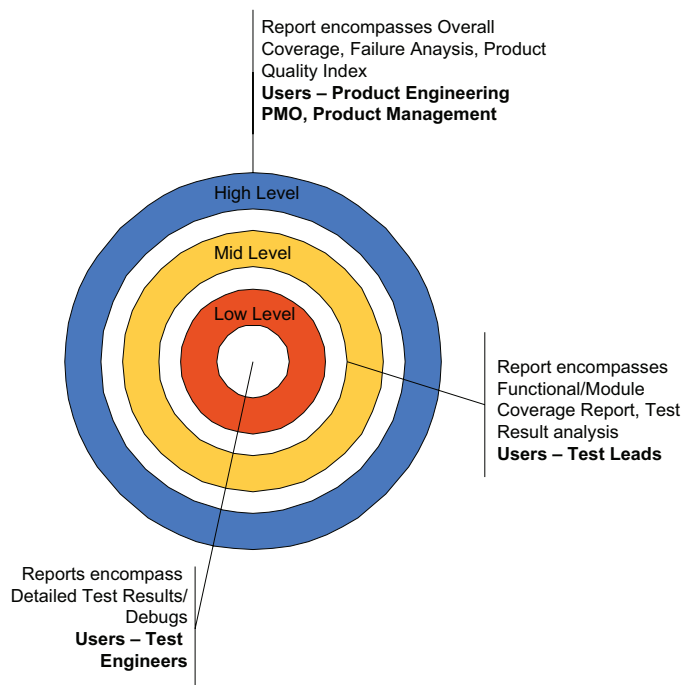


Figure 7 - Three Tier Reporting Model

## Detailed Architecture

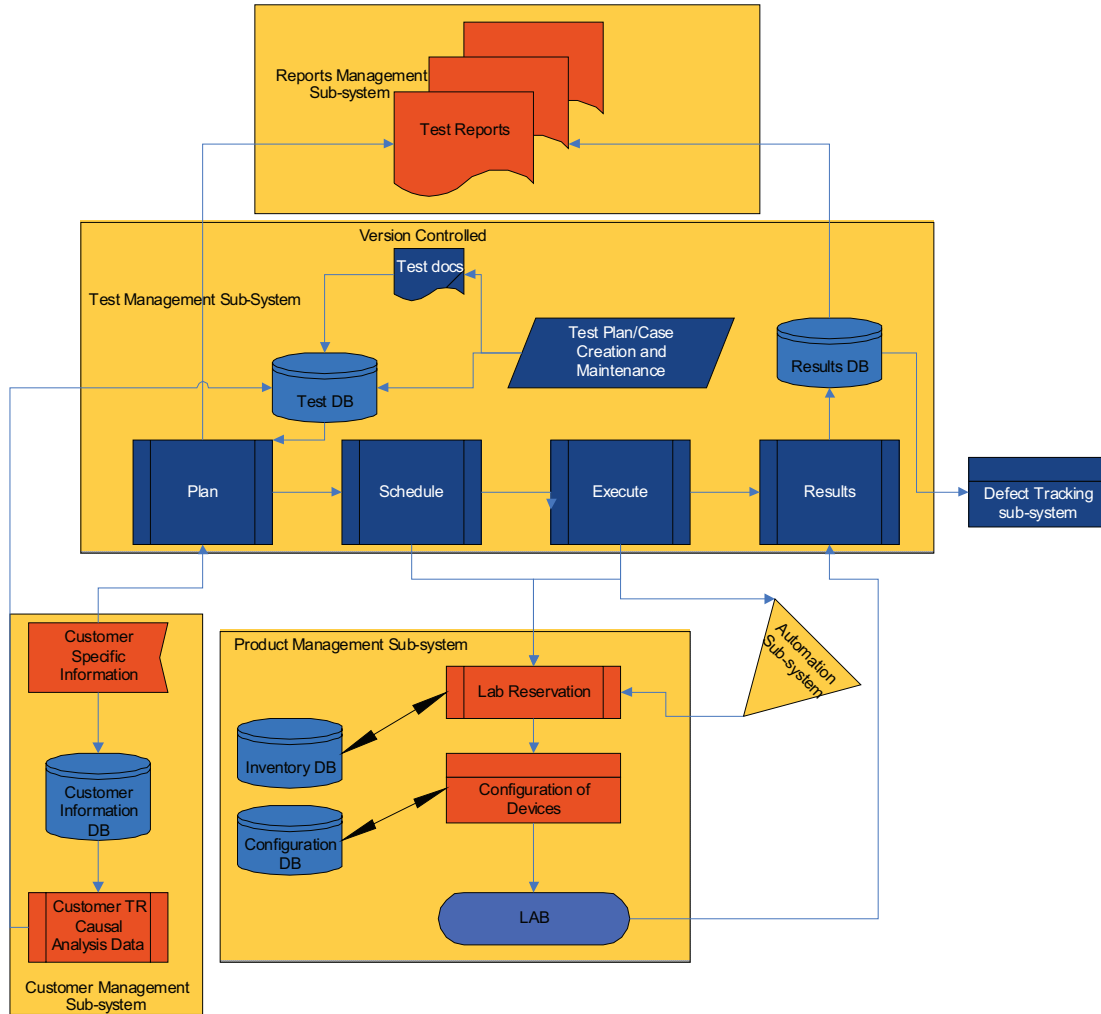
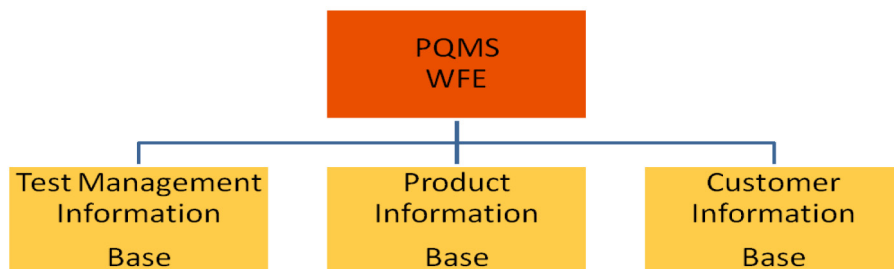


Figure 8 - Detailed PQMS Architecture

## PQMS Information Base

The heart of the system is its information bases, which in conjunction with the workflow engine provide versatility, flexibility and scalability. The system is powered by three primary information bases as depicted below.



These information bases consist of the following:

- Test Management Information Base
  - Test Role Definitions
  - Test Plans and Schedules
  - Test Procedure Definitions
  - Test Results
- Product Information Base
  - Product Inventory Definitions
  - Lab Model Definitions
  - Configuration Details
- Customer Management
  - Customer Definitions
  - Customer-Platform-Test Case Relationships
  - Causal Analysis of Customer Reports

### Anticipated Future Enhancements

- Interface to integrate automation framework
- Interfaces to integrate with defect tracking systems like Bugzilla, etc.
- Interfaces to integrate with 3<sup>rd</sup> party test management systems such as Mercury Test Director, Compuware, etc.

### Summary

PQMS provides improved product quality by tightly integrating into a unified environment, the four key components of the Product Management Cycle that include:

- Quality Assurance
- Release Management
- Customer Quality Management
- Resource Management

PQMS is developed on the concept of integrating processes seamlessly into the engineering workflow through a robust workflow engine that can be easily configured to incorporate a company's process. PQMS achieves process efficiency and better predictability through continuous closed loop corrective actions and can be a powerful tool to improve and manage a product's quality.

---

For more information, contact [PE@infosys.com](mailto:PE@infosys.com)

---



For more information, contact [askus@infosys.com](mailto:askus@infosys.com)

#### About Infosys

Many of the world's most successful organizations rely on Infosys to deliver measurable business value. Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 30 countries build tomorrow's enterprise.

For more information about Infosys (NASDAQ:INFY), visit [www.infosys.com](http://www.infosys.com).