



## Agility in Testing – QA Leads Perspective to PLM Software Testing

Paresh Kulkarni  
Tata Consultancy Services,  
Nyati Tiara, Pune, Maharashtra, India  
[Paresh.k@tcs.com](mailto:Paresh.k@tcs.com)

### Abstract:

Agility is the buzzword in any competitive industry, with business requirements changing rapidly to keep pace with business needs. Be it hardware or software components, testing of the same is of paramount importance to ensure the components work as desired and support business needs. For the purpose of our discussion, we are considering non-automated testing of software products such as Product Lifecycle Management (PLM), ERP to name a couple. Deployment of PLM enables reduction of lifecycle of product development and life cycle management. These products are deployed on variety of platforms and integrated with other domain specific products e.g. CAD software in automotive industries, embedded software for electronic industries etc. These scenarios engender multiple requirements which are interdependent.

PLM products includes

- CAD software – For design and drafting of the wide industry products,
- Web applications – For access over the web,
- Visualization software – makes easy to send design files in special compressed format to higher management over the web and
- Other third-party soft wares such as application sharing and conferencing soft wares integrated with core PLM software and workflow processes for approvals.

Needless to say, for successful release of such a product within schedule and cost budgets, QA processes have to be agile. Agility here pertains to testing of these software products for new developed applications or for migrated applications which calls for varied requirements to be satisfied like adopting proper test case management processes, document control applications, high end server machines, problem reporting tool and good data security policies.

Our paper would like to highlight some of the best practices deployed in the areas of testcase management and use of technology to help manage successful release and enhancement of PLM products. Some new practices such as Test environment administration, Testcase management using keywords and high level usecase preparation vis-à-vis agility in testing would be discussed.

### Keywords:

We would like to list few keywords in this section.

PLM - Product Lifecycle Management  
Agility  
Visualization  
Testcase Management  
PR tool  
CAD  
SDLC  
QA tools  
Test Environment Administration (TEA)

## 1 Introduction

Product Lifecycle Management (PLM) is an integrated business approach to manage the creation and dissemination of engineering data throughout an enterprise and is process of managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. Leading products in this technology are Teamcenter from Siemens PLM software, ematrix from MatrixOne, Windchill from PTC

Product lifecycle management (PLM) is the title commonly applied to a set of application software that enables the New Product Development (NPD) business process.

Within PLM there are four primary areas;

- Product and Portfolio Management (PPM)
- Product Design (CAD)
- Manufacturing Planning (MPM)
- Product Data Management (PDM)

Once developed and implemented, it will enable the optimization of the product development process resulting in shorter time to market, higher quality and reduced costs for companies. The use of PLM framework to leverage innovation and creativity stages into the development and business stages of new products is proving beneficial to organizations across various domains such as pharmaceutical, aerospace, electronics, retail business and so on.

This international conference will bring together Test engineers, QA managers, and users of PLM. Via such forums and with the involvement of all stakeholders, we hope to shape the future of this new field and advance the science and practice of enterprise systems testing. The program will consist of presentations on all aspects of PLM software testing, including QA managers viewpoint, Data migration testing.

There will be a special emphasis on enhancement in the enterprise software product that has a strong influence on the entire lifecycle. Integration between hitherto different domains, tasks and tools will present new challenges to QA lead for software testing.

## 2 Understanding PLM Application Components

### 2.1 Bill of Materials Management (BOM)

It gives product structure and process information for knowing Bill of Materials (BOM) at any stage of product development relevant to customer needs. Bill of Materials includes different product components supplied by co-developers of product, subassemblies, and miscellaneous parts such as nuts and bolts. With increasingly complex products, shorter lead times, and more advanced and varied product offerings, it is progressively more difficult to maintain the accuracy of your bill of materials (BOM) information. You are challenged to manage engineering change across configurations, synchronize downstream BOMs, and extend BOM changes across variants. This leaves your company struggling to improve your quality and time to market, while reducing development, manufacturing and product lifecycle costs in order to remain competitive.

Bill of Materials							
<input type="button" value="Exit"/> <input type="button" value="Cost"/> <input type="button" value="Print Costing"/>		Finished Product Code: <input type="text" value="Manuf001"/>	Manufactured Item Number 1				
		Suppliers Currency Amount: <input type="text" value="22.00"/>	Quantity Finished Products: <input type="text" value="1"/>				
		Total Stock Materials Cost: <input type="text" value="245.06"/>					
		Costs Added In Production: <input type="text" value="31.04"/>					
		Total Cost Of Finished Product: <input type="text" value="276.10"/>					
Raw Material Code	Raw Material Description	Qty Of Goods Produced	Qty Of Materials	Materials Unit Cost	Materials Unit Foreign Cost	Adde P	
▶ Test001	Test Stock Item 001	1.00	3.00	53.47	4.50		
Test002	New Test Stock Item 002	1.00	2.00	24.83	2.50		
Test003	New Test Stock Item 003	1.00	1.00	35.00	3.50		
*							

Figure 1. Bill of Material

## 2.2 Community Collaboration

Web-based collaboration environment that extends the reach of PLM product lifecycle collaboration capabilities to users throughout the extended enterprise, enhancing team performance to increase product and program productivity. Community web application will integrate different PLM applications via web along with use of single sign on (SSO) to facilitate use of Software as a Service (SAAS).

## 2.3 Process Management

provides engineering teams with the vaulting, global sharing, and workgroup management capabilities they need to capture, manage, and leverage geometry and engineering data created by multiple CAD, CAM and CAE systems.

## 2.4 Product Visualization

Built on top of industry-standard JT (Jupiter text) technology, PLM visualization tools deliver high-end visualization and virtual prototyping capabilities that extended enterprises can use to visually configure and virtually test product innovations comprised of design components created under multiple CAD systems. Example .jt file is as below

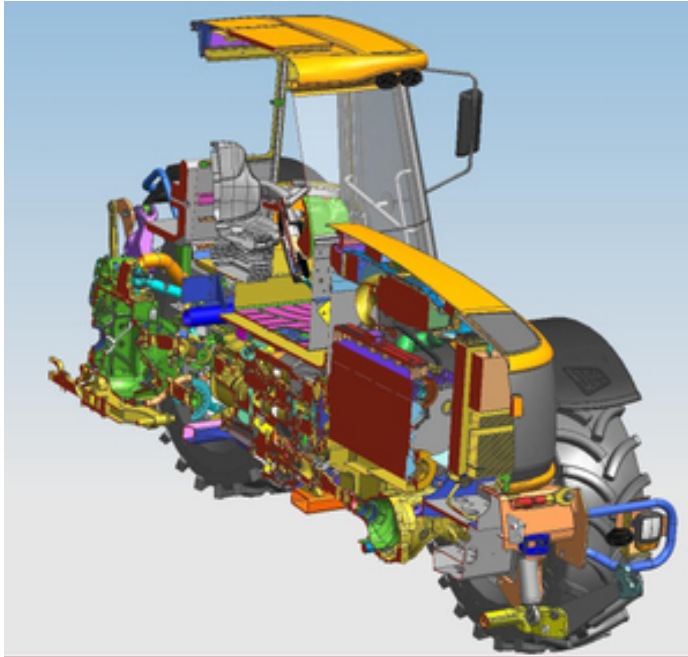


Figure2. Visualization jt image

## 2.5 Program and Project Management

Establishing control over the product lifecycle is critical if manufacturers are to transform their processes of innovation and maximize value. PLM offers a complete infrastructure for strategic planning, program and project management directly linked to detailed execution across the product lifecycle.

## 2.6 Analysis and Reporting

PLM provides easy-to-use tools that enterprises can integrate into their PLM environment to gain visibility into their business processes and managed data by generating HTML and Text based reports

## 2.7 Simulation Data Management

PLM provides a single organized and secure source of simulation data and processes that can be embedded into the product lifecycle to better assess product performance and quality while improving development efficiency.

## 2.8 Vendor Management

PLM offers a proven portfolio of configurable supplier relationship management solutions that unites the product definition knowledge with procurement functions to deliver business value throughout the lifecycle.

## 2.9 Requirements Management

PLM aligns the strategic intent of your products with the correct set of requirements to ensure the successful delivery of products that meet the needs of markets, customers, and regulatory agencies.

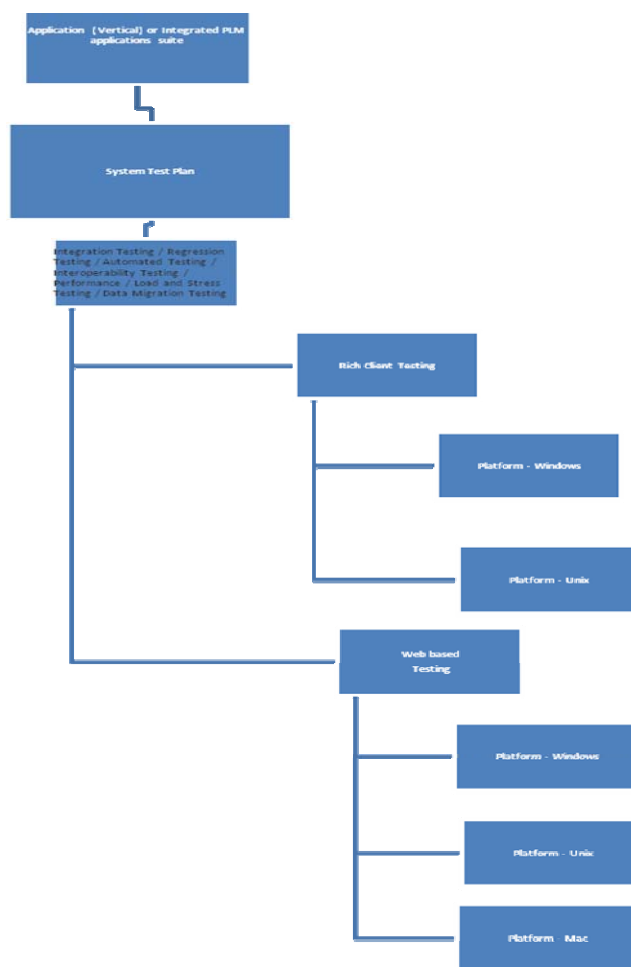
There are more vertical applications under development for different industries such as Aerospace and Defense, Mechatronics, Embedded systems and so on.

### 3. Testing Approach to PLM

PLM as whole is integrated product of different applications and hence software testing is very exhaustive with separate application testing and then integration testing.

Software testing includes defining the Test Strategy, Planning and Management, System Testing, Integration Testing, Regression Testing, Automated Testing, Interoperability Testing, Performance, Load and Stress Testing, and finally, Data Migration Testing.

Generic approach to test PLM applications is shown in the next figure.



After test complete, a final test run for QA complete is done before media verification on CD. Once CD is burnt PLM product is shipped for beta testing.

### 3 Need for Agility in Testing

Agile practices are explained below in comparison with traditional waterfall model of software development.

- **Build frequency:** Gone are the days when most teams could wait until the final stage of development to integrate all of the pieces of an application. In today's world, it is common to develop, unit test, and integrate software in a short time span. I have seen many Waterfall-based projects where builds happen two to three times per day.
- **Iterative development (small releases) and frequent feedback:** In traditional Waterfall-style projects, software is delivered for system testing or acceptance testing after it has been developed. Few large or mission-critical projects can afford to work that way today. Customers often expect to monitor the quality of constructed software through continuous code reviews, encouraging the use of multiple iterations in shorter development cycles. The delivered functionality is prioritized based on the customer's requirements and development feasibility, and the customer provides feedback on each delivered iteration. Feedback has a direct impact on the next iteration and there are fewer (if any) surprises for the customer when the final product is received.
- **Change control:** It has been argued that the basic disadvantage of the Waterfall model is that the requirements phase is closed before development begins. This process allows no room for changes later in the development cycle, even though the need for changes almost inevitably becomes apparent in these later stages. Many Waterfall-based IT shops have adopted a change control process based on how critical the change is, as well as the effort required to implement the change. An effective change control process is defined and documented to ensure that changes are effectively managed and controlled. (A *change*, in this context, can mean a revision to the program scope, deliverables, milestones or levels of services, new clarity configurations, or enhancements that affect the cost, schedule, resources, quality or conformance of the services to the agreed specifications.)

All of these typically agile best practices complement Waterfall-based development and make it more accessible and responsive to the needs of the client, while also breaking down the rigid separation of cycles that characterizes the Waterfall model.

#### 3.1 Agility applied to PLM software testing using web tools

Continuous optimization and enhancement of PLM products driven by increasing demand for PLM technology from different domain industries is reducing cycle time of PLM software release. In this short span, delivering best quality product which satisfies domain specific legal and functional requirements can be achieved only thorough agile testing methodologies. Such resilient testing methodologies include use of various testing and data management tools and practices that are making huge improvements in transforming traditional testing procedure.

Some of these tools and practices sufficient to achieve testing goals are discussed below



### 3.1.1 *Testcase Management Tool*

It is strongly recommended that PLM Company should have its own web tool for managing testcases which should be maintained and enhanced. This testcase management tool is heart of QA activities as it will have following information

1. Testcases for all the products in a hierarchical tree structure in suits for each of the release. These hierarchies can be further divided as  
Product Category > Main Release > Quick Release > Features > Categories (Functional requirements)
2. Ability to apply these testcases on different platforms ( windows, UNIX variants)
3. Ability to update these testcases after execution and link them with Problem reports in case of failure
4. Controlled access to employees for security
5. Selective report generation ability for QA managers
6. Practice area for testcases which can be accessed by developers and new users without logging in
7. Linked review procedure workflow for approval of managers

Test case categories can be further divided into

- **Feature Stability Testing:** A very small suite of testcases which would test basic functionality at the start of test execution after QA handoff. Priority1 PR should be raised if this test case fails.
- **Overall Testing:** A relatively comprehensive test scenarios are considered under this suite with complete coverage of developers and customers expectations as reflected in SRS and FS (Functional specifications)
- **Regression Testing:** This is suite of testcases to know the impact of new development or migration data on existing functionality. Only priority2 and priority3 PRs are expected in case of failures of these test cases

Discussion on Priority for Problem Report is here. There are lots of things about this tool which are beyond the scope of this paper.

### 3.1.2 *Problem Management*

ITIL (Information Technology Infrastructure Library) gives description of number of IT practices which can be tailor made to achieve different goals in software development. Problem management is one of the practices which basically aims at identifying the incidents causing problems can be tailor made to include problem reporting so as to have a common tool ( Problem Reporting Tool as we would call here) which will achieve both the objectives

1. To identify root cause of enlisted problem so as to resolve the error known thereby and
2. To update and report test execution status of testcases for all the stakeholders



PR tool (again a web tool ) should allow users to assign problem as per the functionality to concerned developer and be linked to mail server in order to broadcast the new problem raised or changed status of problem to all the stakeholders. Ability to search for PR and ability to link two or more PRs by making their status as Duplicate PR of master PR is also advanced feature of this tool.

Raised problem will have Priority attribute to decide its severity

**Priority 1:** PR with high severity (show stopper) and must be addressed first. Also reply regarding its status should be given by developer within 24 hours.

**Priority 2:** PR with moderate severity but a must fix problem. It hampers the functionality but is not showstopper

**Priority 3:** PR with low priority which can be fixed later. Problem may not impact expected functionality.

Each problem report will have status to it throughout its existence indicating whether it is resolved or in process or can not resolved like wise depending on the incident in development of software functionality.

Developer may change status of the PR to Fixed, Duplicate, Rejected, System Configuration Error, Can not be fixed, User error to classify it to manage and get the response from concerned QA person who in turn closes the issue by changing the status again to Tested, Accepted and Reopened.

### **3.1.3 SDLC document management**

It is nothing but a PDM system with web version to manage documents related to software development like SRS, FS, SCS (Software combination specification), SDD (Software design document), STP (Software testing plan), TDS (Test design specification) or any other related document with default security features. All the stakeholders will have access to SDLC with inbuilt workflow managing the approval cycle for documents.

Starting off, test lead will access SRS; FS uploaded by developer from SDLC and prepare detailed TDS enlisting test cases in a hierarchy with brief description. After TDS approval from QA managers and stakeholders, Test Lead will create test cases in Test case management tool and may proceed with applying test cases on required platforms.

### **3.1.4 Web mail server**

All of above tools are integrated with mail server for workflow notifications enabling stakeholders to promptly take action. SSO enabled web applications will further reduce the time to navigate across different applications

### 3.1.5 *QA Portal*

Home portal of QA will host information about people, processes such as requests for new login to tools, templates required for preparing documents (SRS,TDS etc), Release information about products, Knowledge bank regarding the product, System Test Plans, Links for launching tools and so on. It is the home portal where QA activities schedules can be accessed.

## 3.2 **Agility applied to PLM software testing using Best Practices**

Use of above tools will contribute towards achieving agility when used along with some of the best practices mentioned as below.

### 3.2.1 *Test Environment Administration (TEA)*

Each QA subgroup has Test Environment Administrator (TEA) who is responsible for establishing environments for testing. TEAs play an important role in downloading and installing the kit from common location and making available different platforms for test execution. QA manager gives the topology required for testing to TEA and raises request in helpdesk for the installation of PLM application.

TEA also has administrative access to all the applications and thereby can control Access Manager Applications. All other QA stakeholders need to raise helpdesk request to make changes at administrative level.

Having separate TEA helps in reducing actual test cycle time by test engineer and allowing test engineer to focus on functional testing. TEAs are meant for QA group's PLM Environment management and are separate from organizations Information System Management (ISM) administrators who are responsible for making hardware available for testing and for software installations on machines.

### 3.2.2 *Use of best hardware available*

Large number of applications including different CAD soft wares demand for high end machines to reflect production scenarios to manage CAD data effectively. In addition to this, PLM software requires to be certified for different windows and UNIX variants and hence large number of High end machines and necessary third party software products are indispensable.

### 3.2.3 *Quick release mechanism to accommodate customer requirements changes*

After the main release to accommodate customer enhancements or changes in requirements quick releases can be scheduled like service packs. This allows for limited scope to test execution with some regression testing completing the QA exercise.

### 3.2.4 *High level testing*

Initial test execution completion is called as Test complete which is then followed by QA complete in order to certify the build for media verification. High level testing is activity carried out before QA complete which includes test execution for high level use cases



designed by experienced QA or development authority which cover all the scenarios on broad level to ensure desired enhancement and objectives of PLM applications are met.

### ***3.2.5 Close interaction between developer and test engineer***

Offline interaction of test engineer with developer is encouraged to certain extent in order to clear doubts regarding desired results or methods. This leads to faster resolution of the problem without redundant efforts.

### ***3.2.6 Test case design standards***

Test case design is also expected to be standardized across organization. Links for set-up instructions files are created for Feature level set-up or category level Set-up or test case level set-up which would show up in test case.

Ability to copy and paste test cases to another test suite in different release could be provided in test case management tool to reduce duplication effort on part of Test Lead.

### ***3.2.7 Maintenance of Test Data***

Common pool of test data such as CAD files, visualization files, huge assembly structures, different types of datasets (Files) should be shared at common location for reproducing the scenarios exactly. This would prove helpful especially during migration testing as data creation may require to be repeated for different cycles.

### ***3.2.8 Knowledge sharing among QA team***

Rotation on different functional areas and knowledge sharing is value adding practice as it expands the scope of Test Engineers towards such an integrated application. Often test scenarios include navigation from one application to another and involve two or more users working at different locations sharing their desktops and hence this practice would avoid possible hiccups.

### 3.3 Role of QA Manager in achieving agility in Testing

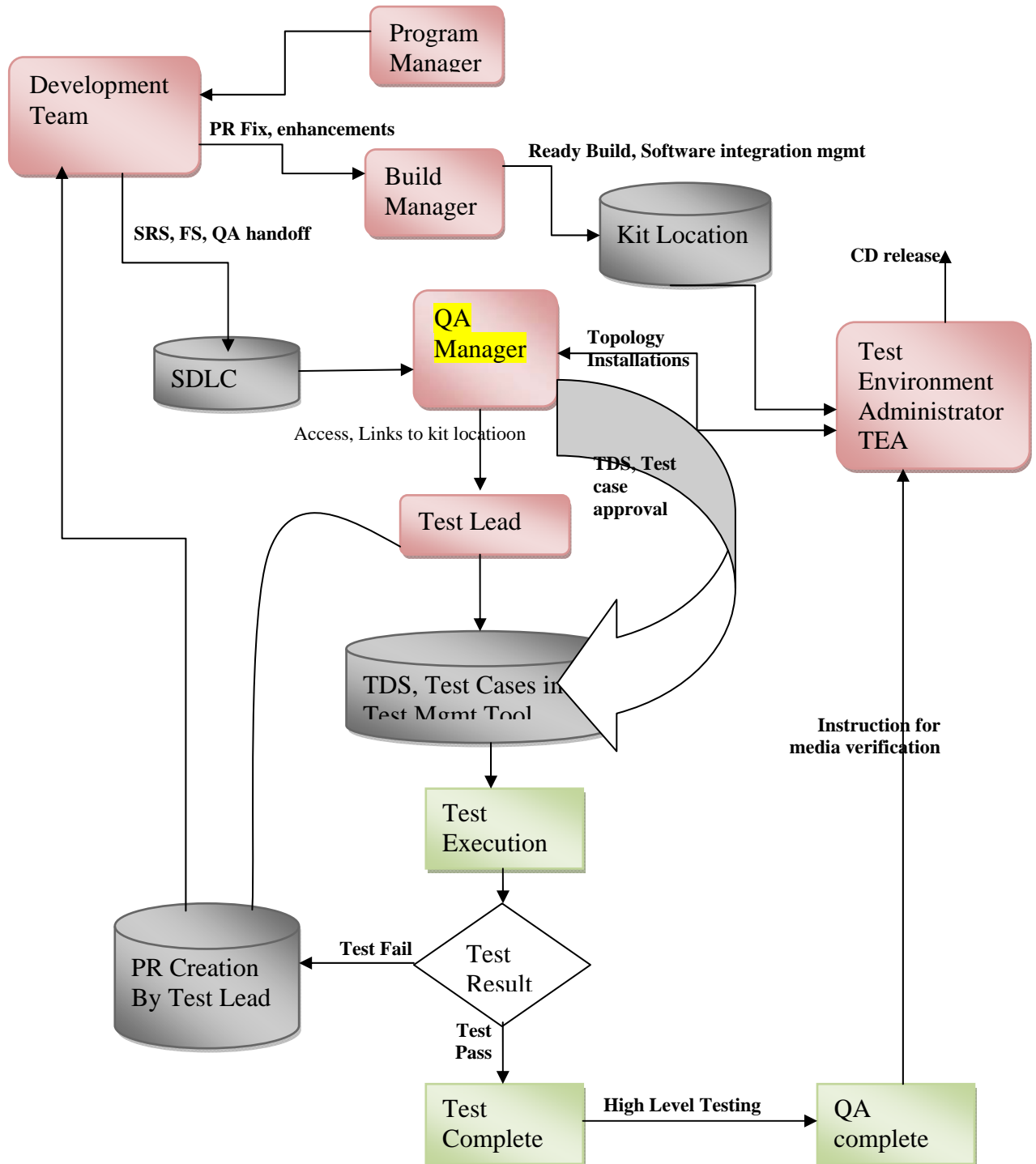


Figure 4. QA activity flow for PLM software release



Active role of QA Manager in decision making and tracking QA activities will bring agility towards shorter product release cycles

QA manager is key driver in planning, motivating, executing and following with development the progress of testing of PLM software.

Activities of QA Manager can be summarized as

1. Providing System test plan
2. Assigning application areas for testing to test leads
3. Review and approval of Test Documents, test cases prepared by test leads
4. Granting selective access to different QA tools to QA team members
5. Raising topology requests and installation requests to TEA
6. Holding weekly status meeting and guiding QA team
7. Raising issues with development team to clarify technical matters
8. Reporting top managers or QA directors about QA status and signing off for Test Complete and QA Complete
9. Updating QA portals with release status

Good communication and agility in executing testing process are virtues of efficient QA manager.

### **3.4 Quality Certifications**

Ever increasing competition in the international market has forced organization to adapt to the recent quality practices to patch with the industry requirements. There are certain established Models and Standards of SEI (Software Engineering Institute) of Carnegie Mellon University. Such as SEI authorized CMM, CMMI, and PCMM.

It's a model of 5 levels of process 'maturity' that determine effectiveness in delivering quality software. Organizations can receive CMMI ratings by undergoing assessments by qualified auditors. In Level 5 – the focus is on continuous process improvement. The impact of new processes and technologies can be predicted and effectively implemented when required. Quality Assurance Institute (QAI), USA was established in 1980 as a professional association formed to represent the software quality assurance industry. Today, Software Certifications, administered by QAI, is global. ISO 9126 defines six high-level quality characteristics that can be used in software evaluation. It includes functionality, reliability, usability, efficiency, maintainability, and portability. It's clear that raising the quality of software without recourse to elaborate techniques or complex methodologies is still a needed and practical solution.



#### 4 References

- [1] Siemens PLM official website - [http://www.plm.automation.siemens.com/en\\_gb/](http://www.plm.automation.siemens.com/en_gb/)
- [2] Image-Figure1.Source  
<http://images.google.co.in/imgres?imgurl=http://www.ibyss.co.za/IAM/ScreenShots/Production/BillOfMaterials.jpg&imgrefurl=http://www.ibyss.co.za/IAM/ScreenShots/Production/BillOfMaterials.htm&h=691&w=916&sz=59&hl=en&start=25&tbid=YtucMknI79VZ9M:&tbnh=111&tbnw=147&prev=/images%3Fq%3DBill%2Bof%2Bmaterial%26start%3D20%26gbv%3D2%26ndsp%3D20%26hl%3Den%26sa%3DN>
- [3] Test2008: 2008: Conference Proceeding Template.  
<http://www.javaworld.com/javaworld/jw-03-2008>
- [4] [www.cpdm.iisc.ernet.in/plm06/](http://www.cpdm.iisc.ernet.in/plm06/) - about PLM

## Biography



**Paresh Kulkarni**

Paresh joined TCS,Pune in August 2004 from VNIT Nagpur. Worked in software testing for Microsoft products like Livemeeting, and test automation. For around 2 years worked for teamcenter QA in Siemens PLM as a contract employee. Kishor Repala - Joined TCS in Jan 2006 and worked in Microsoft ODC as a test engineer. Aditya Tondon- Worked in Capgemini India as a software engineer in testing of OSS-BSS in telecom domain for 2 yrs. Joined TCS in March 2008.