White Paper

PTC®

Improve Quality and Decrease Time to Market with Better Requirements Management

Requirements Engineering: Right Requirements, Right Products

> “Nearly 20% of development cost is due to rework because of ill-defined and poorly managed requirements.”
>
> Software Engineering Institute
> US Department of Labor, Bureau of Labor Statistics

Requirements engineering presents many challenges for engineering and IT organizations today. These challenges stem from a lack of visibility into changing requirements, poor communication of change, and lack of traceability between requirements and other artifacts throughout the software development lifecycle.

Improved requirements management gets better products to market faster while improving the overall quality of your software.

A common source of easily accessible, up-to-date requirements enables members of the global software development project team to work more efficiently. Analyzing the impact of changes to requirements before they are made, and alerting project team members when changes are made, enables change to be managed more effectively. Leveraging the requirements assets and all related artifacts between projects over time minimizes rework and allows project teams and the organization as a whole to streamline processes, increase productivity and dramatically reduce time-to-market.

This paper provides a thorough and detailed examination of a comprehensive requirements engineering solution. The paper begins by summarizing the high-level benefits of an effective requirements engineering approach, the role of requirements engineering within Application Lifecycle Management (ALM) and how PTC Integrity can seamlessly and effectively manage requirements through its single platform, single architecture approach. This paper will also examine how requirements are authored, captured and traced through the downstream lifecycle, how companies can utilize best practices such as parallel development and reuse in relation to requirements, and how configuration management concepts such as versioning and base lining can be leveraged to achieve advanced requirements management capabilities.

The State of Requirements Management Tools

Most requirements management solutions are disconnected from the development process.

When separate tools are used for requirements development and management across systems, software and hardware, data does not flow smoothly between the engineering disciplines or between requirements, specifications, design and test artifacts. Therefore, it is critical for organizations to have a robust requirements engineering solution that is connected to the development system and artifacts across disciplines in order to deliver smarter, more innovative products on time and within budget.

Separate repositories for requirements and development data also limit the ability of analysts to get a view into the progress of their requirements or upper management to get a cohesive picture of activities across the organization.

Although many organizations are interested in requirements engineering and recognize it as a critical capability, the overall market share for requirements management tools is relatively small. This discrepancy can be partly attributed to most available requirements management tools being too restrictive, forcing organizations into development processes that don't fit their unique situation. Existing solutions prescribe traditional waterfall processes or Agile processes, while most development environments consist of modified or hybrid processes or multiple teams using mixed processes. Flexibility to accommodate this situation and evolve as processes improve and adapt is critical in a requirements management solution.

As the pace of global software development accelerates, the need to support parallel development projects and leverage configuration management concepts and practices is moving outward from the development group to the other domains in the application lifecycle. Highly dynamic projects need to leverage project assets and stay connected to the activities of related projects, as changes now ripple across project boundaries. Tools that support traceability, which traditionally is essential from requirement to code, need to extend this support across projects and versions and all the elements that comprise each deliverable and asset — be it source code, test case or requirement — in the system.

PTC Integrity and Requirements Engineering: A Unified Approach

PTC Integrity has customizable traceability keeping all stakeholders informed of changes to requirements throughout the development process, fostering collaboration between engineering disciplines and roles. Reuse and requirements change management are coupled with real-time metrics and full traceability between system requirements and downstream software and hardware requirements, leading to superior product quality, full auditability and more rapid time-to-market.

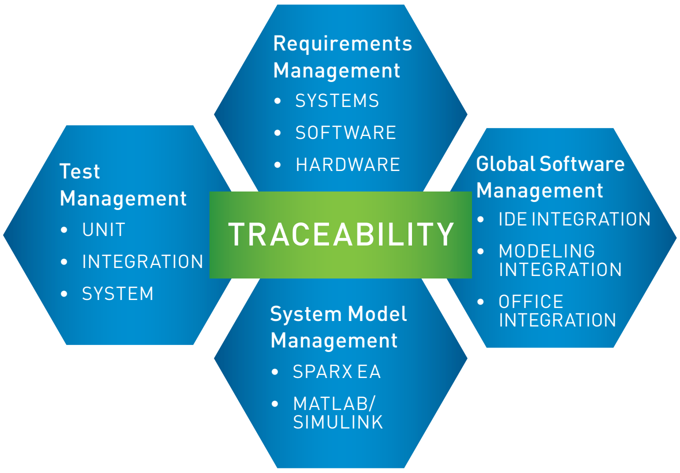
PTC Integrity also enables you to capture, store and manage requirements as a unified part of the development process. PTC Integrity's capabilities for requirements management is built as an extension of PTC Integrity's powerful process management and workflow engine. Analyst, development, quality and build teams collaborate using a single lifecycle management platform for requirements capture, traceability, change management and management of development, testing and deployment tasks.

This integrated approach is not only more cost effective, it also facilitates intra-team and intra-project communication. Analysts can easily determine the impact of a proposed change by reviewing the status of all work in progress to implement the requirement. The development team has easy access to up-to-date requirements and is automatically notified of any changes. Project managers can assess the impact of change, as it occurs, within reused assets. Senior management can get a connected view into all phases of their development projects using powerful querying, charting, reporting and dash-boarding features.

PTC Integrity provides a clear and visible connection between development objects such as source code or documentation and their associated requirements. This traceability satisfies any compliance initiatives and better leverages the assets across the organization.

PTC Integrity includes all the features demanded of a powerful requirements management solution but takes each one to the next level, providing users with the control to manage change effectively through:

- Document view and rich text support – You can prescribe and capture requirements into the system in a familiar document-style interface in addition to the traditional flat list and hierarchical tree views of the data;
- Named relationship fields – You can define how your requirements, test cases and development activities are related;
- Suspect link support and automatic notification – You can establish rules that determine which changes to requirements result in related development activities being automatically flagged for further investigation;



- Traceability and impact analysis through relationships – You can navigate the hierarchy of requirements and their associated development activities for traceability and impact analysis of changes; and
- Historical navigation and reporting – You can see the contents of a single requirement or an entire document at an arbitrary point in its history. You can see how the contents changed between any two points in time and you can start new work, branching the document and its contents, from any baseline or selected point in time.

Offering capabilities far beyond those found in other requirements management tools, PTC Integrity's advanced single ALM architecture capabilities include:

- Requirements change management – Integrated change management capabilities allow you to control requirements churn, stay informed on project scope and more effectively delegate, authorize and assign work across team members;
- Requirements reuse and persistence – You can logically associate groups of requirements and reuse them in a parallel development scenario while maintaining full traceability, history and genealogy;
- Requirements versioning – You can access a data model that enables configuration management of requirements, test cases and other artifacts within the system that not only mirrors how source code assets are managed but also enables full traceability to those assets; and
- Requirements baselines – You can securely identify a document or a set of requirements at any point in history and not only navigate the system based on that identifier but begin new work from those existing baselines.

The remainder of this paper provides a more in-depth review of the core capabilities of PTC's Integrity Requirements Engineering offering.

Authoring and Capturing Requirements

Authoring and capturing requirements are recorded in PTC Integrity using special item types. There are a number of ways that requirements can be captured and viewed within the PTC Integrity system, thereby enabling users to work in an environment and style that is comfortable for them.

Document View

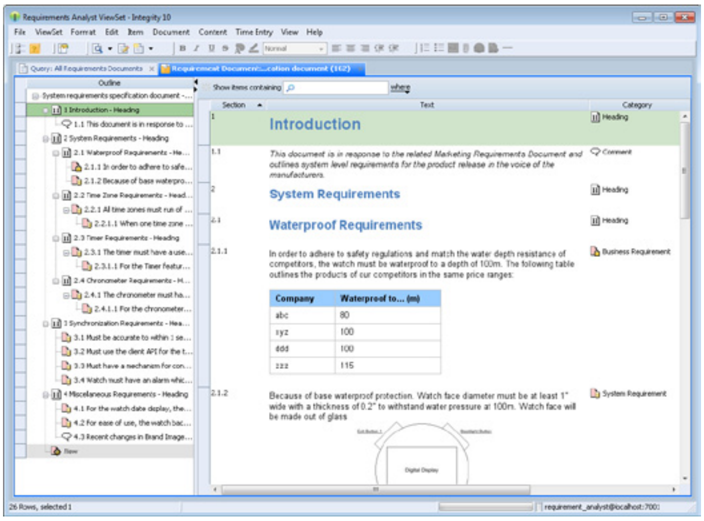
You can enter your requirements information directly in PTC Integrity through its document view. This feature provides analysts with a familiar format for prescribing and authoring requirements. Users can employ rich text mark ups (e.g., bold face, underline, etc.), create tables and embed images and other objects directly into their requirements documents.

List View

For some users, dealing with requirements in a flat list is a more productive way to work. Using the list view is an easy way to spot unwanted duplication in your requirements set or to perform batch operations. For example, you can easily assign work to users, perform quick edits and create traces on multiple items in this view.

Hierarchical Tree View

When structure and relationships are the primary areas of interest rather than the complete content of the item, the hierarchical tree view enables users to view, navigate and create links between requirements and from requirements to any other object in the system. This view is also an effective way to see the rollup of metrics and status at each level in the hierarchy so as to quickly spot and correct items that need attention.


Requirements Analyst ViewSet - Integrity 10
File ViewSet Format Edit Item Document Content Time Entry View Help
Normal
Query: All Requirements Documents
Requirement Document:...cation document (162)
Outline
Show items containing
where
Section
Text
Category
Digital Display

System requirements specification document -...
1 Introduction - Heading
1.1 This document is in response to ...
2 System Requirements - Heading
2.1 Waterproof Requirements - He...
2.1.1 In order to adhere to safe...
2.1.2 Because of base waterpro...
2.2 Time Zone Requirements - Head...
2.2.1 All time zones must run of ...
2.2.1.1 When one time zone ...
2.3 Timer Requirements - Heading
2.3.1 The timer must have a use...
2.3.1.1 For the Timer featur...
2.4 Chronometer Requirements - H...
2.4.1 The chronometer must ha...
2.4.1.1 For the chronometer...
3 Synchronization Requirements - Hea...
3.1 Must be accurate to within 1 se...
3.2 Must use the client API for the t...
3.3 Must have a mechanism for con...
3.4 Watch must have an alarm whic...
4 Miscelaneous Requirements - Heading
4.1 For the watch date display, the...
4.2 For ease of use, the watch bac...
4.3 Recent changes in Brand Image...
New

1 Introduction — Heading

1.1 *This document is in response to the related Marketing Requirements Document and outlines system level requirements for the product release in the voice of the manufacturers.* — Comment

2 System Requirements — Heading

2.1 Waterproof Requirements — Heading

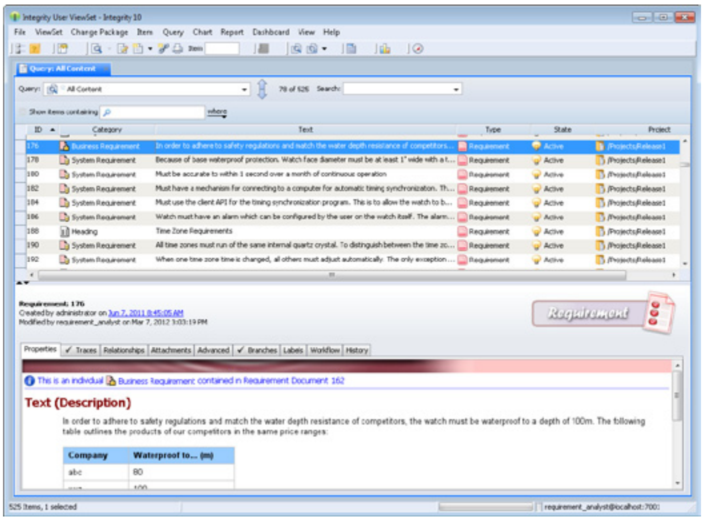
2.1.1 In order to adhere to safety regulations and match the water depth resistance of competitors, the watch must be waterproof to a depth of 100m. The following table outlines the products of our competitors in the same price ranges: — Business Requirement

Company	Waterproof to... (m)
abc	80
xyz	100
ddd	100
zzz	115

2.1.2 Because of base waterproof protection. Watch face diameter must be at least 1" wide with a thickness of 0.2" to withstand water pressure at 100m. Watch face will be made out of glass — System Requirement

26 Rows, selected 1 | requirement_analyst@localhost:700:

Document view supports structure, context-based authoring, rich text and inline images.



ID	Category	Text	Type	State	Project
176	Business Requirement	In order to adhere to safety regulations and match the water depth resistance of competitors...	Requirement	Active	/Projects/Release1
178	System Requirement	Because of base waterproof protection. Watch face diameter must be at least 1" wide with a t...	Requirement	Active	/Projects/Release1
180	System Requirement	Must be accurate to within 1 second over a month of continuous operation	Requirement	Active	/Projects/Release1
182	System Requirement	Must have a mechanism for connecting to a computer for automatic timing synchronizaton. Th...	Requirement	Active	/Projects/Release1
184	System Requirement	Must use the client API for the timing synchronization program. This is to allow the watch to b...	Requirement	Active	/Projects/Release1
186	System Requirement	Watch must have an alarm which can be configured by the user on the watch itself. The alarm...	Requirement	Active	/Projects/Release1
188	Heading	Time Zone Requirements	Requirement	Active	/Projects/Release1
190	System Requirement	All time zones must run of the same internal quartz crystal. To distinguish between the time zo...	Requirement	Active	/Projects/Release1
192	System Requirement	When one time zone time is changed, all others must adjust automatically. The only exception...	Requirement	Active	/Projects/Release1

Requirement: 176
Created by administrator on Jun 7, 2011 8:45:05 AM
Modified by requirement_analyst on Mar 7, 2012 3:03:19 PM

This is an individual Business Requirement contained in Requirement Document 162

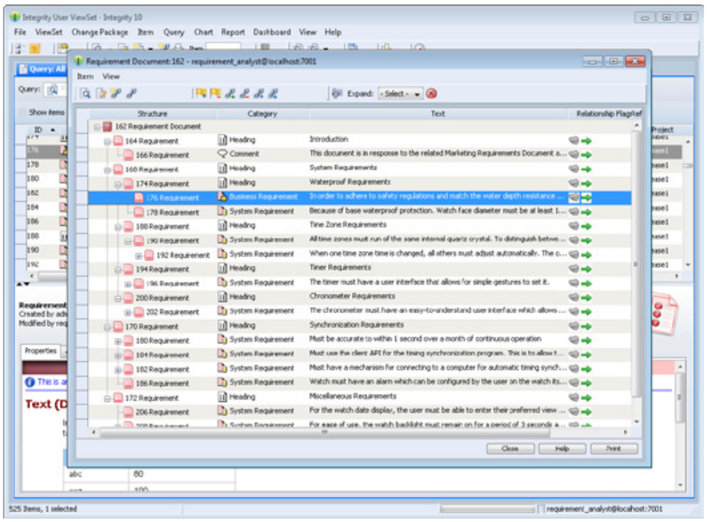
Text (Description)

In order to adhere to safety regulations and match the water depth resistance of competitors, the watch must be waterproof to a depth of 100m. The following table outlines the products of our competitors in the same price ranges:

Company	Waterproof to... (m)
abc	80

525 Items, 1 selected | requirement_analyst@localhost:700:

List view supports filtering, sorting and inline editing.


Integrity User ViewSet - Integrity 10
File ViewSet Change Package Item Query Chart Report Dashboard View Help
Requirement Document 162 - requirement_analyst@localhost:7001
Item View
Expand: - Select -
Close
Help
Print

Structure	Category	Text	Relationship Flag/Ref
162 Requirement Document			
164 Requirement	Heading	Introduction	
166 Requirement	Comment	This document is in response to the related Marketing Requirements Document a...	
160 Requirement	Heading	System Requirements	
174 Requirement	Heading	Waterproof Requirements	
176 Requirement	Business Requirement	In order to adhere to safety regulations and match the water depth resistance ...	
178 Requirement	System Requirement	Because of base waterproof protection. Watch face diameter must be at least 1...	
188 Requirement	Heading	Time Zone Requirements	
190 Requirement	System Requirement	All time zones must run of the same internal quartz crystal. To distinguish betwe...	
192 Requirement	System Requirement	When one time zone time is changed, all others must adjust automatically. The o...	
194 Requirement	Heading	Timer Requirements	
196 Requirement	System Requirement	The timer must have a user interface that allows for simple gestures to set it.	
200 Requirement	Heading	Chronometer Requirements	
202 Requirement	System Requirement	The chronometer must have an easy-to-understand user interface which allows ...	
170 Requirement	Heading	Synchronization Requirements	
180 Requirement	System Requirement	Must be accurate to within 1 second over a month of continuous operation	
184 Requirement	System Requirement	Must use the client API for the timing synchronization program. This is to allow t...	
182 Requirement	System Requirement	Must have a mechanism for connecting to a computer for automatic timing synch...	
186 Requirement	System Requirement	Watch must have an alarm which can be configured by the user on the watch its...	
172 Requirement	Heading	Miscellaneous Requirements	
206 Requirement	System Requirement	For the watch date display, the user must be able to enter their preferred view ...	
[illegible]	System Requirement	For ease of use, the watch backlight must remain on for a period of 3 seconds a...	

525 Items, 1 selected | requirement_analyst@localhost:7001

Relationship view supports hierarchical decomposition.

MS Word, Excel and Project Integrations

For organizations that have used conventional methods such as Microsoft Word or Excel to document requirements, PTC Integrity's integration with these tools allows them to leverage those assets and utilize their familiar authoring tool to build requirements. Completed requirements are imported into the system and PTC Integrity then reproduces the document structure through a linked set of items. Documents can be further exported, modified and re-imported supporting the full round trip with Microsoft Word or users can use the Edit-in-Word functionality to author directly in the Microsoft environment. Using PTC Integrity's reporting engine, requirements documents can be re-created when necessary or, with the Microsoft Excel and Microsoft Project integrations, bidirectional synchronization between PTC Integrity and the external tool can be sustained.

Other Requirements System Import

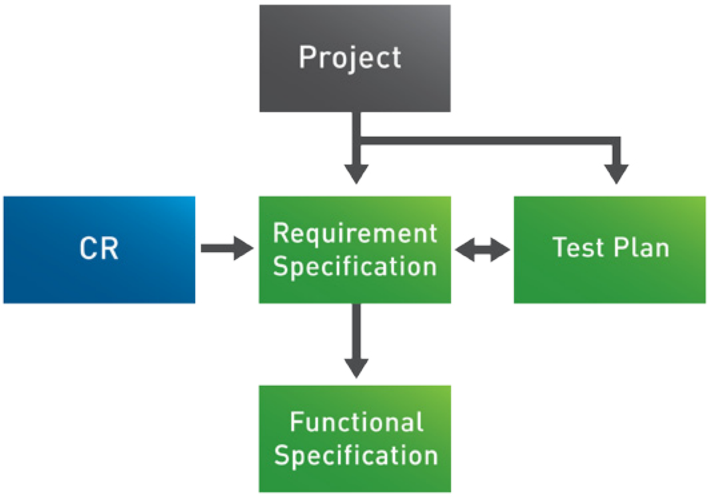
The integration of PTC Integrity with requirements tools such as IBM Rational DOORS or HP Quality Center or HP ALM enables organizations with existing investments in those products to connect front-end requirements directly to related development activities. For organizations with existing requirements management investments, the integration with PTC Integrity provides engineers, developers and QA representatives with a cost-effective way to bridge the requirements and development phases of the lifecycle through an integrated solution. PTC Integrity also fully supports the Requirements Interchange Format (RIF, ReqIF) for the exchange of requirements data with any third party system subscribing to this industry standard. While not as effective as the single-system approach, PTC recognizes that this may be a viable transitional approach in some cases.

Traceability of Requirements through the Downstream Lifecycle

PTC Integrity can seamlessly trace and navigate interactively from projects through various levels of requirements to design features and specifications, assigned tasks, testing and deployment activities, as well as view activity in context with associated source code changes — all within in a single system and user interface.

This navigation of linked items provides traceability critical for compliance, accountability and audits. It enables you to root out individual components that consistently cause problems. Also, you can confirm that all requirements have functional specifications and test cases so that they are developed and tested before the product is released or project milestones are reached. In a complex parallel development environment, each Software Requirements Specification (SRS) document is linked to a test plan as well as a functional or design specification. Individual requirements within the SRS are linked to other requirements in the same document and to individual functional specifications and test cases

During the lifecycle of a development project, changes may occur that have a potential impact on other aspects of the project. Business climates change, priorities shift and environments evolve; these fluctuations have effects throughout the application lifecycle. For example, if a change is made to a requirement, it could affect the functional specifications, tasks and test cases associated with the requirement; coding delays could affect the functional specification the requirement is associated with, which in turn could affect other tasks and tests associated with the specification. Ideally, you need to determine what the overall impact of a change will be before it occurs, but definitely as soon as it is identified.



The PTC Integrity template for global software development inclusive of Requirements Engineering contains a number of item types to capture information across the application lifecycle. A Project item represents a software project and tracks its progress. A Requirement Specification document contains requirements, business, user, functional, non-functional and system needs content. A Functional Specification document tracks the specification and design of features that will implement the requirements. Process items like Tasks, not shown in the illustration below, allow for the assignment of specific development work and track any defects uncovered in testing that need to be fixed. A Test Plan document contains the test cases and links to items, which track testing work and test results that need to be performed to validate a given requirement.

You can customize PTC Integrity's item types by adding fields or field values, adding different types of relationships, or removing those relationships that do not pertain to your business. You can also create your own item types with their own metadata, workflow and behavior to augment the out of-the-box configuration.

Integrating Your Processes

Whether your requirement is a User Story, Functional Requirement, Market Requirement, Industry Constraint or has another name each requirement item type in the system has its own workflow. These separate processes are integrated using rules so that they constrain and/or update each other. By unifying the requirements process with development and test processes, users capture and enforce process controls and interactions throughout the development lifecycle — ensuring a coordinated effort across the development, QA and deployment phases. Integrated workflows connect distinct cultural and organizational silos and prevent them from working in isolation. By unifying and coordinating processes across every stage of the lifecycle, teams are aware of process dependencies.

The following is a sample of simple cross-workflow rules that could be used in PTC Integrity configurations:

- Design work on a functional specification associated with a requirement cannot be initiated until the requirement is approved;
- Once work starts on a functional specification, the associated requirement is automatically moved to the "In Progress" state;
- A functional specification cannot be moved to the "Complete" state unless all related Task items are in the "Complete" state;
- When all related functional specifications move to the "Complete" state, the parent requirement is automatically moved to the "Implemented" state; and/or
- A requirement cannot be moved to the "Satisfied" state unless all related functional specifications are in the "Complete" state.

PTC Integrity's workflow capability goes far beyond a simple level of association. Despite all the similarities between different kinds of objects stored in an ALM repository, there is one area where you may want the objects to have significant differences: the workflow.

Within PTC Integrity, workflow can also be attached to tasks or change requests related to the artifact, rather than being part of the artifact itself. Individual requirements are managed artifacts and hence are treated more like source code files than process items; as such, it makes sense that change requests are used to control their evolution as well. This integrated change management paradigm allows for better control of requirements churn and project scope, and enables more effective delegation, authorization and assignment of work across team members.

This model of association allows PTC Integrity to accommodate multiple assigned users who must coordinate activities around a single workflow. There may be multiple workflows and a change request attached to a particular artifact in the system applies to that artifact and to all artifacts underneath it in the hierarchy.

Analyzing the Impact of Changes to Requirements

To ensure that your final deliverables match your final specifications, a coordinated effort is required across all areas of the development organization, and tight control over change must be achieved. Changes that have the potential to affect other activities must be carefully evaluated and managed — and the impacts must be resolved to maintain the project's integrity after the change.

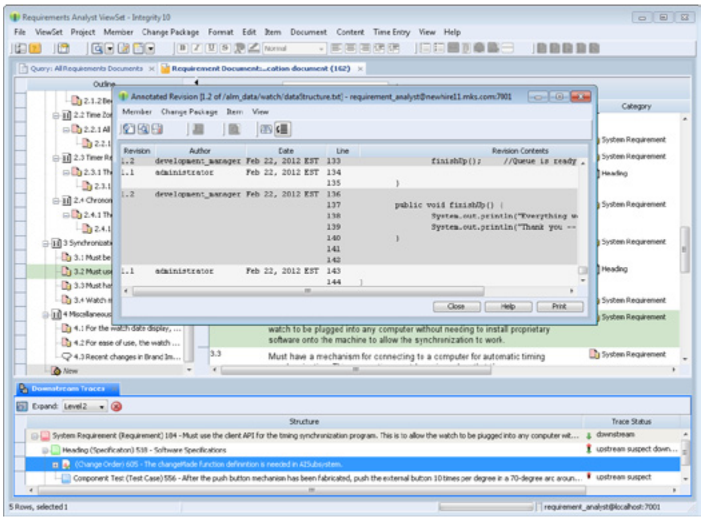
Understanding the impact of change and then effectively managing it, is critical for the successful delivery of software applications. Changes to existing requirements, or the addition of new requirements, can severely impact the total project's delivery schedule. For example, if a change is made to a requirement, it could affect the design specifications, tasks, and test cases associated with the change; coding that is delayed could affect the requirement the code is associated with, which in turn could affect other tasks and tests associated with the feature. Project teams must be able to easily assess the resulting impact of these changes.

Traditional requirements engineering systems rely on a traceability matrix. Maintaining the links and dependencies between the items in the matrix can be a manual, highly time-consuming process; interpreting the matrix in projects of realistic size can be extremely difficult. Since PTC Integrity integrates requirements with all downstream development activities, you can navigate the potential impact of a change across all aspects of the application lifecycle, from requirements to coding and testing. By navigating up or down the relationships hierarchy in the Relationships view, you can evaluate the impact of a potential change before making it. This type of analysis is available to anyone in the organization and the information you see is always current and automatically kept up-to-date.

Each requirement should be traceable to a specific project objective. This traceability ensures that the software product satisfies all strategic goals and that individual requirements do not include inappropriate or extraneous functionality. It is important to know the source of all requirements and functionality so it can be verified as necessary, accurate and complete. Using the Relationships view, you can navigate through the network of related items, tracing each requirement to its originator upstream and to each downstream design specification and feature.

This traceability also enables you to review all aspects of your development process to satisfy an auditor, show compliance with government and safety regulations, or analyze your own processes.

The Relationships view enables you to uncover information such as why a particular piece of code was changed and thereby trace the relationship. For example, you could trace backward from a source code change to its associated requirement; or you could show the code changes made to implement a requirement by tracing forward from the requirement to the code changes. In addition to navigating the Relationships view, you can also print various types of traceability reports.


Requirements Analyst ViewSet - Integrity 10
File ViewSet Project Member Change Package Format Edit Item Document Content Time Entry View Help
Normal
Query: All Requirements Documents
Requirement Document:...cation document (162)
Outline
2.1.2 Be
2.2 Time Zo
2.2.1 All
2.2.1
2.3 Timer R
2.3.1 Th
2.3.1
2.4 Chrono
2.4.1 Th
2.4.1
3 Synchronizat
3.1 Must be
3.2 Must us
3.3 Must ha
3.4 Watch
4 Miscellaneous
4.1 For the watch date display, ...
4.2 For ease of use, the watch ...
4.3 Recent changes in Brand Im...
New
Category
System Requirement
System Requirement
Heading
System Requirement
System Requirement
Heading
System Requirement
System Requirement
System Requirement
Member Change Package Item View
Close
Help
Print
Downstream Traces
Expand: Level2
5 Rows, selected 1
requirement_analyst@localhost:7001

Annotated Revision [1.2 of /alm_data/watch/dataStructure.txt] - requirement_analyst@newhire11.mks.com:7001

```
Revision  Author               Date               Line   Revision Contents
1.2       development_manager  Feb 22, 2012 EST   133              finishUp();     //Queue is ready
1.1       administrator        Feb 22, 2012 EST   134
                                                  135        }
1.2       development_manager  Feb 22, 2012 EST   136
                                                  137        public void finishUp() {
                                                  138                System.out.println("Everything w
                                                  139                System.out.println("Thank you --
                                                  140        }
                                                  141
                                                  142
1.1       administrator        Feb 22, 2012 EST   143
                                                  144    }
```

watch to be plugged into any computer without needing to install proprietary software onto the machine to allow the synchronization to work.

3.3 Must have a mechanism for connecting to a computer for automatic timing

Structure	Trace Status
System Requirement (Requirement) 184 - Must use the client API for the timing synchronization program. This is to allow the watch to be plugged into any computer wit...	downstream
Heading (Specification) 518 - Software Specifications	upstream suspect down...
(Change Order) 605 - The changeMade function definition is needed in AISubsystem.	
Component Test (Test Case) 556 - After the push button mechanism has been fabricated, push the external button 10 times per degree in a 70-degree arc aroun...	upstream suspect

Tracing relationships from requirement to source code.

PTC Integrity supports change management through integrated workflow processes, automatic flagging of suspect relationships and managed suspect relationship resolution. Whenever you revise a Requirements item, all items directly linked are automatically flagged as suspect. When you review your suspect items, if you make changes to key fields, all items that are directly related to the item you have changed are flagged as suspect. At each stage of change, its impact can be analyzed, providing a high level of control.

The automatic flagging of suspect links is controlled through PTC Integrity Requirements triggers and field rules. You can revise the existing rules or add your own. In addition, you can easily configure email notifications so that you are automatically notified whenever any of the items you are responsible for are flagged as suspect.

Requirements Reuse

Requirements reuse provides users with the unique ability to share a requirement across projects without incurring unnecessary duplication of artifacts within a repository. Shared requirements can track with the ongoing change by the author or remain at a static point in time as the needs of the project dictate. Further, change to a shared requirement can be made by anyone and the system handles the branching and evolution of that requirement appropriately.

The concept of reuse is a familiar one within the software development realm, but there are various definitions and use cases that must be taken into consideration when implementing a solution to address requirements reuse. Let's first look at the various parts of a requirement: data, metadata and relationships.

Data

Describes an object, and is relevant to the object itself. An example of data may be a summary or description of a requirement.

Metadata

This is data about the data, which aids in organizing or using the object within a process. Metadata typically describes the current state of the object and has the same scope as the data itself. For instance, metadata may describe the State/Stage within a requirement workflow (i.e., Approved, Rejected, Satisfied, Tested).

Relationships

This characteristic of a requirement allows you to model:

- Structure (i.e., Consists Of, Includes);
- History (i.e., Revision Of, Derived From);
- Conceptual links or traces (i.e., Satisfies); and/or
- References (i.e., Defined By, Decomposes To).

Any given requirement can have information in each of the data, metadata and relationships categories, and when requirements are shared, any or all of the information can also be shared.

By leveraging the powerful data model within PTC Integrity, users may reuse individual requirements or groups of requirements.

Reuse can occur within a number of scenarios leveraging the various parts of a requirement listed above.

Reuse with Change Notification

In this situation, a requirement with all its related information (i.e., data, metadata and relationships) is reused in its entirety. Project state determines the state of the requirements at the time of reuse, and any change to requirements in a reuse scenario causes a "ripple" effect, flagging all artifacts related to those requirements as suspect.

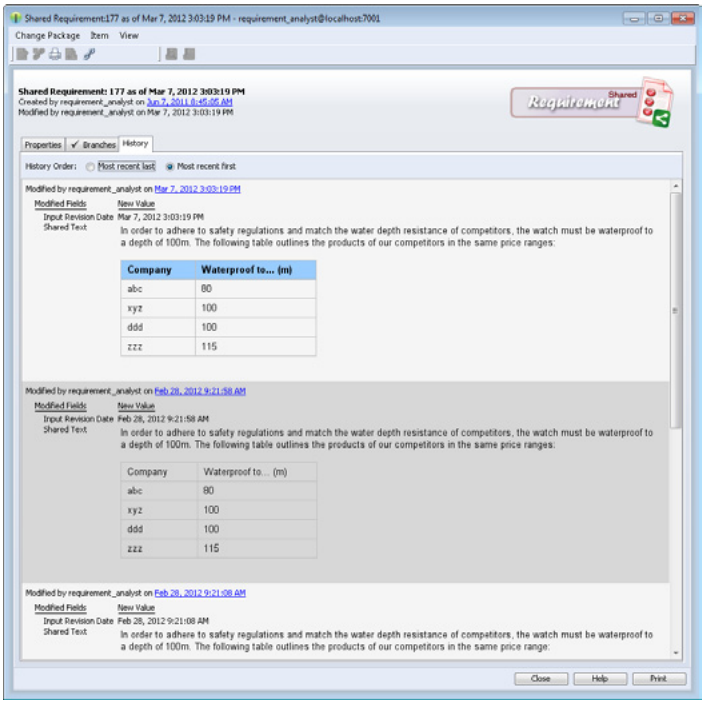
Reuse with Change Control

Reuse with Change Control is similar to Reuse with Change Notification in that data, metadata and relationships are reused in their entirety. The difference is that two projects sharing the same requirement only share it until the point in time when one project needs to change it. When the information changes, a new version is created and only items referencing that new version are declared suspect. All other projects or documents are unaffected.

Reuse with Annotations

In the two previously discussed reuse paradigms, the requirements and related information (i.e., data, metadata and relationships) are reused in their entirety. In Reuse with Annotations, only some of the information belonging to a requirement is identified as a candidate for sharing and reuse. The rest of the information is specific to the project or document. The shared information is held in the repository while the other information belongs to the project or document reference. Each instance of the requirement being reused has its own metadata and relationships. The project or document state is, or can be, independent of the state of the requirements that are contained within it.

New versions of the requirement are automatically created when the shared information in the repository is changed. These changes that trigger revisions can suspect other references, as well as other items in the system, by the ripple effect of that change. For example, changes to requirements may affect test cases or functional specifications downstream.


Shared Requirement:177 as of Mar 7, 2012 3:03:19 PM - requirement_analyst@localhost:7001
Change Package
Item
View
Requirement
Shared
Properties
✓ Branches
History
Close
Help
Print

Shared Requirement: 177 as of Mar 7, 2012 3:03:19 PM
Created by requirement_analyst on Jun 7, 2011 0:45:05 AM
Modified by requirement_analyst on Mar 7, 2012 3:03:19 PM

History Order: ○ Most recent last ● Most recent first

Modified by requirement_analyst on Mar 7, 2012 3:03:19 PM

Modified Fields	New Value
Input Revision Date	Mar 7, 2012 3:03:19 PM
Shared Text	In order to adhere to safety regulations and match the water depth resistance of competitors, the watch must be waterproof to a depth of 100m. The following table outlines the products of our competitors in the same price ranges:

Company	Waterproof to... (m)
abc	80
xyz	100
ddd	100
zzz	115

Modified by requirement_analyst on Feb 28, 2012 9:21:58 AM

Modified Fields	New Value
Input Revision Date	Feb 28, 2012 9:21:58 AM
Shared Text	In order to adhere to safety regulations and match the water depth resistance of competitors, the watch must be waterproof to a depth of 100m. The following table outlines the products of our competitors in the same price ranges:

Company	Waterproof to... (m)
abc	80
xyz	100
ddd	100
zzz	115

Modified by requirement_analyst on Feb 28, 2012 9:21:08 AM

Modified Fields	New Value
Input Revision Date	Feb 28, 2012 9:21:08 AM
Shared Text	In order to adhere to safety regulations and match the water depth resistance of competitors, the watch must be waterproof to a depth of 100m. The following table outlines the products of our competitors in the same price range:

Every change to every object is captured and viewable as part of the audit history.

There are other models of reuse that can be described here, but your business will determine which model is most effective for you. The requirements management tool should, and in the case of PTC Integrity does, allow you to implement the model that is most effective for your business challenges.

The PTC Integrity template for global software development inclusive of Requirements Engineering demonstrates a model with the benefits of all the above scenarios and it can be used out-of-the-box or configured to address the specific needs of the business.

Requirements History, Versions and Baselines

When you implement a complex reuse scenario, or even a system where requirements persist release after release, you must version your requirements much like the development organization versions source code. The term "version" may mean different things to different people, so let's define the term and show how it relates to similar terms like history, baselines and milestones.

Consider a system where requirements are captured within requirements documents but are stored as individual items within the repository.

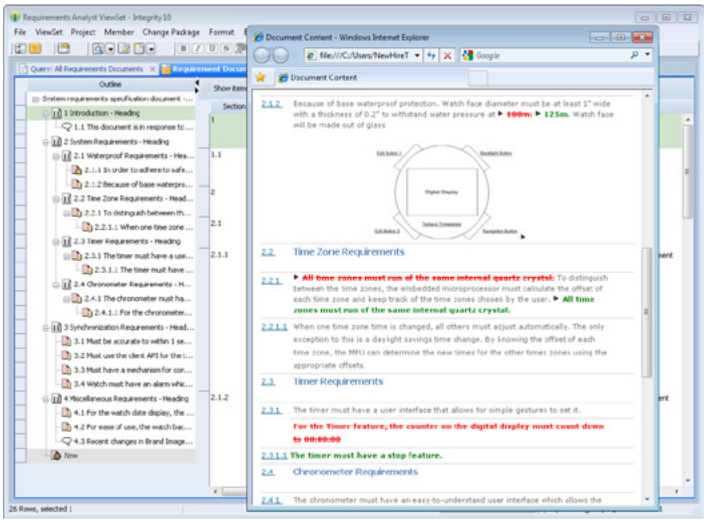
History is the term used to describe the audit trail for an individual item. All changes made to the item — whether that be to data, metadata or relationships, are captured in the item's history. From here you can discover answers to the "Who, When and What" questions with respect to change on that item.

Baseline is a very similar concept to version but has a much different scope. Individual requirements are often organized into groups or sets. In PTC Integrity, these sets are called documents and a baseline is a meaningful point in a document's history.

Some organizations use a slightly different definition for baseline. Rather than being a "snapshot in time" for a given document, a baseline is a goal to work toward. For purposes of this discussion, we will call the goal-oriented baseline a milestone in order to distinguish between the two terms.

Tools on the market today for requirements management often indicate that they allow the versioning of individual requirements. However, many of these tools support versioning by way of cloning or copying the entire requirement, and fewer go so far as to relate the copy to the original. The PTC Integrity solution offers true versioning of the requirements artifacts with support for branches (or the ability to begin new work from a historical version of a requirement or document), as well as the ability to capture the complete genealogy or version history of that artifact.

This ability provides an additional dimension to the traceability question: not only can you trace a requirement through implementation and deployment within the application lifecycle, but you can also trace that individual requirement's evolution over time and across projects, getting a true picture of its use within your organization.



System requirements specification document - ...
1 Introduction - Heading
1.1 This document is in response to ...
2 System Requirements - Heading
2.1 Waterproof Requirements - Hea...
2.1.1 In order to adhere to safe...
2.1.2 Because of base waterpro...
2.2 Time Zone Requirements - Head...
2.2.1 To distinguish between th...
2.2.1.1 When one time zone ...
2.3 Timer Requirements - Heading
2.3.1 The timer must have a use...
2.3.1.1 The timer must have ...
2.4 Chronometer Requirements - H...
2.4.1 The chronometer must ha...
2.4.1.1 For the chronometer...
3 Synchronization Requirements - Head...
3.1 Must be accurate to within 1 se...
3.2 Must use the client API for the t...
3.3 Must have a mechanism for con...
3.4 Watch must have an alarm whic...
4 Miscellaneous Requirements - Heading
4.1 For the watch date display, the ...
4.2 For ease of use, the watch bac...
4.3 Recent changes in Brand Image...
New

Section
1
1.1
2
2.1
2.1.1
2.1.2

2.1.2. Because of base waterproof protection. Watch face diameter must be at least 1" wide with a thickness of 0.2" to withstand water pressure at ▶ **100m:** ▶ **125m.** Watch face will be made out of glass

2.2. Time Zone Requirements

2.2.1. ▶ **All time zones must run of the same internal quartz crystal:** To distinguish between the time zones, the embedded microprocessor must calculate the offset of each time zone and keep track of the time zones chosen by the user. ▶ **All time zones must run of the same internal quartz crystal.**

2.2.1.1 When one time zone time is changed, all others must adjust automatically. The only exception to this is a daylight savings time change. By knowing the offset of each time zone, the MPU can determine the new times for the other times zones using the appropriate offsets.

2.3. Timer Requirements

2.3.1. The timer must have a user interface that allows for simple gestures to set it.

For the Timer feature, the counter on the digital display must count down to 00:00:00

2.3.1.1 **The timer must have a stop feature.**

2.4. Chronometer Requirements

2.4.1. The chronometer must have an easy-to-understand user interface which allows the

Advanced reporting features allow visibility into changes to requirements over specified time periods.

Viewing Historical Items

As mentioned above, PTC Integrity maintains the history for all requirements and represents an audit trail of all changes applied over time. You can use this information to either view and navigate through the item or to produce historical reports that reflect the state of the requirement (or requirements documents) at any point in time – whether denoted by a date, version identifier or baseline.

For example, if you wanted to see the SRS for PTC Integrity 10.0 as of its Project Plan baseline, you could view it in the document view or run a historical report based on that baseline.

Historical reports allow for baselines to be compared. For example, you could use the Project Acceptance and Project Completion milestone dates and run a historical report to see how the content of a development project changed during its lifecycle.

This portion of a comparison report shows all changes to the document including changes, added requirements, and dropped requirements.

This historical report is just one such example, and all defined reports – either by PTC Integrity, administrators or end users – can be run in a historical context as shown.

Navigating the system as of a particular point in time – be it for metrics analysis or reuse – and being able to report on change across historical items is invaluable for controlling churn and scope creep within a project and for leveraging prior work across projects. Further, being able to reuse or branch those historical requirements to begin new work enables you to leverage your requirements assets across multiple projects.

Summary

Requirements management is an integral part of developing successful software products. Effective requirements management ensures that product and business objectives are met by providing approved and up-to-date requirements accessible to all members of the project team. It also ensures that projects are delivered on schedule by providing a means to analyze and communicate changes to requirements.

PTC Integrity is the only ALM platform available today to provide a single unified solution for managing requirements in direct relationship to all downstream development, QA and deployment phases of the lifecycle. Analysts, developers, testers and release managers collaborate via a single platform, unified interface and common process. This facilitates a seamless, real-time flow of requirements data between requirements producers and consumers, and allows traceability through every stage of the development process.

For organizations with requirements engineering needs beyond the capabilities of most requirements management tools, PTC Integrity's advanced architecture and single platform design support sophisticated usage – including requirements versioning, baselining and reuse – that enables you to perform complex reuse scenarios, parallel development and requirements configuration management tasks.

PTC Integrity is the solution for organizations seeking the powerful combination of requirements and process management. Whether you are looking to:

- Lower development costs through reduction in rework
- Increase team collaboration and productivity
- Improve software and product quality
- Achieve downstream traceability
- Increase team collaboration and productivity

PTC has a solution to satisfy your needs.

PTC Integrity Business Unit Locations

North America	1 800 613 7535
United Kingdom	+44 (0) 1252 453 400
Germany	+49 (0) 711 3517 750
Asia Pacific	+65 6830 8338
Japan	+81 3 5422 9503

For more information visit: PTC .com/product/integrity.

J0347–Improve Quality Requirements Management-WP–EN–0912