**Requirement Validation and Requirement Management**

**Requirements Validation**

**Requirements validation** is the process of checking that requirements defined for development, define the system that the customer really wants. To check issues related to requirements, we perform requirements validation. We usually use requirements validation to check error at the initial phase of development as the error may increase excessive rework when detected later in the development process.

In the requirements validation process, we perform a different type of test to check the requirements mentioned in the [Software Requirements Specification (SRS)](https://www.geeksforgeeks.org/software-engineering-quality-characteristics-of-a-good-srs/), these checks include:

* Completeness checks
* Consistency checks
* Validity checks
* Realism checks
* Ambiguity checks
* Verifiability

The output of requirements validation is the list of problems and agreed on actions of detected problems. The lists of problems indicate the problem detected during the process of requirement validation. The list of agreed action states the corrective action that should be taken to fix the detected problem.

There are several techniques which are used either individually or in conjunction with other techniques to check to check entire or part of the system:

1. **Test case generation:**  
   Requirement mentioned in SRS document should be testable, the conducted tests reveal the error present in the requirement. It is generally believed that if the test is difficult or impossible to design then, this usually means that requirement will be difficult to implement and it should be reconsidered.
2. **Prototyping:**  
   In this validation techniques the prototype of the system is presented before the end-user or customer, they experiment with the presented model and check if it meets their need. This type of model is generally used to collect feedback about the requirement of the user.
3. **Requirements Reviews:**  
   In this approach, the SRS is carefully reviewed by a group of people including people from both the contractor organisations and the client side, the reviewer systematically analyses the document to check error and ambiguity.
4. **Automated Consistency Analysis:**  
   This approach is used for automatic detection of an error, such as non-determinism, missing cases, a type error, and circular definitions, in requirements specifications.

First, the requirement is structured in formal notation then CASE tool is used to check in-consistency of the system, The report of all inconsistencies is identified and corrective actions are taken.

1. **Walk-through:**  
   A walkthrough does not have a formally defined procedure and does not require a differentiated role assignment.
   * Checking early whether the idea is feasible or not.
   * Obtaining the opinions and suggestion of other people.
   * Checking the approval of others and reaching an agreement.

**Requirements Validation**

It’s a process of ensuring the specified requirements meet the customer's needs. It’s concerned with finding problems with the requirements.

These problems can lead to extensive rework costs when they are discovered in the later stages, or after the system is in service.

The cost of fixing a requirements problem by making a system change is usually much greater than repairing design or code errors. Because a change to the requirements usually means the design and implementation must also be changed and re-tested.

During the requirements validation process, different types of checks should be carried out on the requirements. These checks include:

1. **Validity***checks*: The functions proposed by stakeholders should be aligned with what the system needs to perform. You may find later that there are additional or different functions are required instead.
2. **Consistency***checks*: Requirements in the document shouldn’t conflict or different descriptions of the same function
3. **Completeness***checks*: The document should include all the requirements and constraints.
4. **Realism***checks*: Ensure the requirements can actually be implemented using the knowledge of existing technology, the budget, schedule, etc.
5. **Verifiability**: Requirements should be written so that they can be tested. This means you should be able to write a set of tests that demonstrate that the system meets the specified requirements.

There are some techniques you can use to validate the requirements, and you may use one or more of them together, depending on your needs.

**Requirements Reviews**

A team of system customers; those who interact with the customer to gather requirements, and system developers start reading the requirements in the document and investigate in great detail to check for errors, inconsistency, conflicts, and any ambiguity.

Then they may negotiate with the customer on how to solve the problems and errors found.

**Prototyping**

We’ve discussed [prototyping](https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc#189e) as one of the (non-standalone) software process methodologies which is used as part of a full methodology, and we’ve also mentioned it can be used in requirements engineering.

In this approach to validation, an executable model of the system is demonstrated to the customer and end-users to validate and ensure if it meets their needs.

Prototyping is usually used when the requirements aren’t clear. So, we make a quick design of the system to validate the requirements. If it fails, we then refine it, and check again, until it meets the customer's needs.

This definitely will decrease the cost as a result of having clear, understandable, consistent requirements.

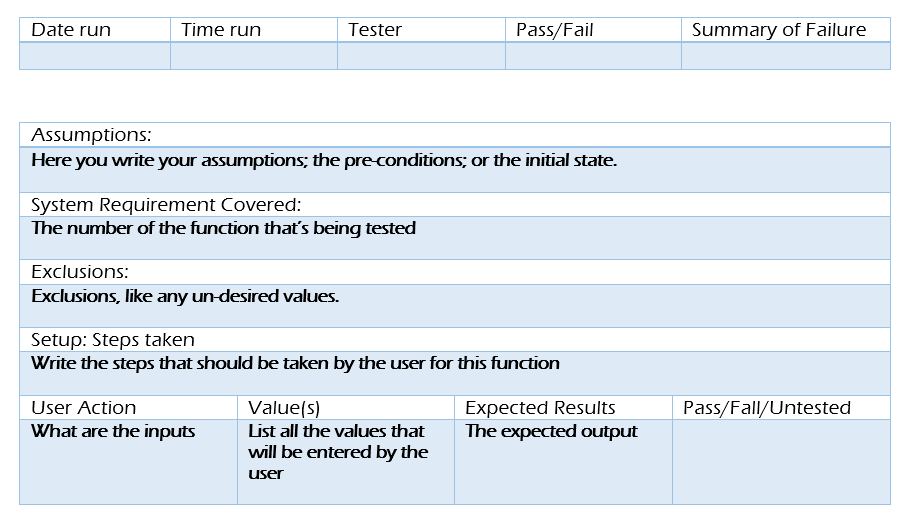
**Test-case Generation**

As we’ve just mentioned, the requirements need to be testable. If the tests for the requirements are added as part of the validation process, this often reveals requirements problems.

If the test is difficult or impossible to design, this usually means that the requirements will be difficult to implement and should be reconsidered.

*The term “tests” here doesn’t mean to write and run some code for every function. It means to write a textual description of the “inputs”, “expected value”, and “steps taken” to perform each function.*

Here’s a template for a test case.



## Requirement Management

## How does requirements management work?

The purpose of requirements management is to ensure product development goals are successfully met. It is a set of techniques for documenting, analyzing, prioritizing, and agreeing on requirements so that engineering teams always have current and approved requirements. Requirements management provides a way to avoid errors by keeping track of changes in requirements and fostering communication with stakeholders from the start of a project throughout the engineering lifecycle.

What is requirement management?

Whether you’re just learning the basics of requirements management, looking to improve your current requirements management process, or interested in benchmarking your process against industry leaders, you’re in the right place.

This article covers what requirements management is, why managing requirements matters, what the requirements management process entails, and how to manage requirements when creating complex, highly regulated products.

### What is a need?

A need is an agreed-to expectation for a product or system to perform some function or possess some quality within specified constraints with acceptable risk. Needs communicate what the stakeholders need and expect from a product or system in order for a given problem or opportunity to be addressed.

### What is a requirement?

A requirement is the result of a formal transformation of one or more needs or parent requirements into an agreed-to obligation for a product or system to perform some function or possess some quality within specified constraints with acceptable risk.

There are different types of needs and requirements that range from a business focus to a user focus to a technical focus.

Business needs and requirements, sometimes referred to as stakeholder needs and requirements, are those derived from the business processes or elicited from the stakeholders including customers, users, and other stakeholders involved in the project. The stakeholder needs represent what the stakeholders need the product to do to address the problem or opportunity the product is to address; stakeholder requirements are stakeholder-defined product requirements that communicate what the stakeholders require of the product to meet their needs. Stakeholder needs are expressed in natural language without the use of “shall”, while stakeholder requirements are communicated with “shall” to make sure they are treated as binding requirements the product will be verified to meet.

Given there are multiple stakeholders, there will be multiple sets of stakeholder needs and requirements. It is up to the project team to elicit these needs and requirements, and resolve conflicts, inconsistencies, and other issues. The results will be an integrated set of needs from which the product requirements will be transformed. The resulting product requirements represents what the product must do in order for the needs to be met. Product requirements are sometimes referred to as system requirements, software requirements, or technical requirements.

### What is requirements management?

Requirements management is the process of gathering, analyzing, verifying, and validating the needs and requirements for the given product or system being developed. Successful requirements management ensures that completed deliverables meet the expectations of the stakeholders. Requirements can be managed using documents, however, complex systems or products in highly regulated industries mitigate risk by using trusted requirements management tools.

### Why is requirements management important?

Requirements management is important because it empowers everyone to clearly understand stakeholder expectations and confidently deliver a product that has been verified to meet the requirements and validated to meet the needs.

Requirements management is a sophisticated process that includes many moving parts and diverse groups of people. Typically, the product management department, specifically the product manager, is responsible for the requirements management process. They work with the stakeholders, including business teams, customers, users, developers, testers, regulators, and quality assurance.

Additionally, a product may have only 100 requirements, or it may have several thousand. This number will depend on the product’s complexity and level of regulation. With all these elements at play, success hinges on the ability to get everyone on the same page, working toward the same goal.

Therefore, the business value of managing requirements is huge, as successful requirements management is key to project success.

Benefits of requirements management include:

* Enhance understanding of stakeholder needs, requirements, and expectations and the problem or opportunity the product means to address
* Gain clarity on scope, budget, and schedule
* Minimize costly, time-consuming rework
* Increase product quality
* Mitigate risk
* Improve likelihood of delivering the right product, within budget and schedule with the required quality.

The importance of requirements management is intensified, however, when building complex or highly regulated products. This is because more time and budget are invested in development. The cost of getting it wrong — be it money, time, or reputation — is too great to risk. Hence, developers in regulated industries, or those who develop products with a lengthy list of needs and requirements, tend to rely on requirements management tools.

#### **Requirements management vs. project management**

While it may seem that requirements management and project management are synonymous, there is a difference. Simply, project management is getting the product built within budget and schedule with the available resources. Requirements management is making sure the product is the right product and that it is built right.

The goal of the product development process is to create a successful product that meets stakeholder, customer, and market needs. The requirements management piece of product development includes managing the needs and requirements so that the product meets stakeholder expectations. Therefore, needs and requirements along with budget and schedule set the scope for the project.

However, the domain of project management encompasses tasks like providing budget, personnel, and resources needed to develop the product.

### Stages of the requirements management process

So, [how do you manage requirements](https://www.jamasoftware.com/li-campaign-asset-rm-best-practices-ebook-thank-you)?

The most successful teams work from a defined requirements management process. Defining the requirements management process is important because requirements change throughout a project. When this happens, that change needs to pass through the same, repeatable process.

There are four main stages of the requirements management process — planning, development, system verification, and system validation — each with important considerations to the overall project. Change management, while not a stage itself, affects nearly every phase of the requirements management process.

#### **Planning Stage**

The product development methodology—waterfall, agile, or Scrum—helps decide how requirements move through the process. Waterfall models are typically linear with development moving from one process area to another with a completed product delivered with the required features and functionality. Agile development, including Scrum, is iterative in nature. In requirements management, agile teams, and those using the Scrum approach, may be working on different sets of requirements concurrently, and delivering a product in increments, each increment adding value with additional features or functionality.

No matter which methodology is employed, the Requirements Management Plan (RMP) is the documented process the team uses throughout product development. It contains information like stakeholder roles and responsibilities, which needs and requirement artifacts will be defined, how traceability will be competed and managed, how needs and [requirements baseline](https://www.jamasoftware.com/blog/defining-requirement-baseline/) will be handled, how interactions (interfaces) with external systems and users will be managed, how changes will be managed, how the product will be verified to meet the requirements, and how the product will be validated to meet the needs.

A successful Requirements Management Plan is visible to and has sign off from stakeholders, as it sets the course and expectations for all stakeholders throughout the product development journey.

#### **Needs and Requirements artifacts**

Part of the RMP is defining the needs and requirements artifacts that will be created during the requirements management process.

Needs and requirements artifacts include the data and information concerning the needs and requirements and related information. Examples include diagrams, and models, integrated set of needs, set of product requirements, use cases, design documents, testing plans and procedures. Requirements artifacts are used throughout the product development lifecycle to:

* Describe the product being built
* Identify the actions needed to develop the product
* Capture the actions performed during development
* Define the testing needed for system verification and system validation
* Assist with stakeholder review, communications, and engagement

While some organizations will communicate this information in a document form (e.g.,  a [Software Requirements Specification (SRS)](https://www.jamasoftware.com/blog/how-to-write-system-requirement-specifications-srs-documents/)), the increasing trend is to manage the needs and requirements within a requirements management software application.

The reason organizations are moving towards a data-centric practice of product development is that is hard, for any document-based approach to be flexible and scalable enough for complex agile projects. This is especially true for highly regulated industries that must prove compliance. Due to numerous factors—lack of consistent updating, human error, incomplete data, version control, the need to establish and maintain traceability, etc., documents simply cannot be relied upon determine whether a need or requirement is fulfilled.

Agile product teams working on complex products in highly regulated industries will have much more success using a [requirements management software](https://www.jamasoftware.com/solutions/requirements-management/) application to streamline the requirements analysis phase of requirements definition and management.

[Modern requirements definition and management solutions, like [Jama Connect®](https://www.jamasoftware.com/platform/jama-connect/), define, establish traceability for, manage, and validate complex product requirements automatically, which simplifies not only requirements analysis, but the overall requirements definition and management process as well.]

It is imperative that needs and requirements and their artifacts are linked to each other and to additional related artifacts. This can be difficult to achieve using documents, as the manual nature lends itself to myriad errors. This is especially true in the case of complex or highly-regulated products where [traceability is a prerequisite for proving compliance](https://www.jamasoftware.com/blog/building-an-audit-trail-through-live-traceability/).

Needs and Requirements attributes

To keep track of the requirements in a given project, each requirement should have a specific list of attributes. Requirements attributes are used to ensure that requirements are organized and uniquely identified across all requirements artifacts.  The attributes also aid in managing the sets of needs and requirements, enabling reporting and dashboards to be defined to provide accurate and timely status of the project.

It is a best practice that the following requirement attributes are included for all needs and requirements:

* Unique Identifier
* Rationale
* Owner
* Type
* Definition status
* Priority
* Criticality
* Compliance
* Version number
* Change history
* Trace data
* Verification status (for requirements)
* Validation status (for needs)

#### **Needs and Requirements Baseline**

A needs and [requirements baseline](https://www.jamasoftware.com/blog/defining-requirement-baseline/) is a point-in-time look at a committed set of agreed-upon, reviewed, and approved needs and requirements – or planned functionality and features – to be included in the product. The purpose is to provide information to stakeholders so they can make informed decisions on and possibly modify the planned functionality and features using change requests.

The RMP defines a baseline strategy including timing and frequency of creation, needs and requirements prioritization (deciding which requirements should be included), publishing, change management, requirements verification, and requirements validation.  In this context needs and requirements verification address the quality of the needs and requirements statements as well as the existence and correctness of the traceability. Needs validation is confirmation with the stakeholders that the integrated set of needs clearly communicates the intent of the stakeholder in terms of what the need the product to do.  Requirements validation is confirmation with the stakeholders that individual requirements and the entire set of requirements clearly communicate the intent of the needs they were transformed from.  The quality of the requirements is dependent upon the quality of the needs from which they were transformed.

Establishing a requirements baseline is important because it implies that:

* Scope has been defined and agreed to
* Formal change control begins
* Staffing levels and budgets are set
* Schedule commitments are made

Typically, baselines are stored in [software requirements specification (SRS) documents](https://www.jamasoftware.com/blog/characteristics-of-excellent-requirements/). However, a complex system might need a variety of software, hardware, and interface requirement specifications to encapsulate the baseline’s components. This can be cumbersome to maintain during initial development and downright impossible during change management.

Alternatively, [working with baselines in a requirements management solution](https://www.jamasoftware.com/blog/baselines-in-jama-connect-to-facilitate-compliance-and-reuse/) allows baselines to be defined as a subset of the requirements already stored in the database. This streamlines the process – from prioritization through stakeholder sign-off.

### Requirements development stage

The development stage is conducted using the organization’s requirements analysis process.

The needs and requirements development stage includes eliciting needs and requirements from the identified stakeholders, properly defining and refining them, and analyzing them to ensure clarity and resolve issues and conflicts. Without successful needs and requirements development, there would very likely be gaps between what stakeholders were expecting and what is ultimately delivered, which could ultimately result in disaster.

#### **Eliciting needs and requirements**

Needs and requirements elicitation – also called [needs and requirements gathering](https://www.jamasoftware.com/blog/what-is-requirements-gathering) – is the act of working with users, customers, and internal business stakeholders to identify stakeholder needs and requirements and get an understanding of the requirements for a product or system.

#### **Needs and requirements definition**

Needs and requirements definition is the time when the gathered needs and requirements are re-written in a clear and traceable way that enables effective communication throughout the product development lifecycle.

There are many [do’s and don’ts for writing needs and requirements](https://www.jamasoftware.com/blog/8-dos-and-donts-for-writing-requirements/), but the following are basic [characteristics of quality needs and requirements](https://resources.jamasoftware.com/requirements-management/best-practices-guide-for-writing-requirements):

* Necessary
* Unambiguous
* Feasible
* Verifiable
* Correct
* Traceable
* Appropriate

Traceability of needs and requirements is very important. At the end of the day, [traceability](https://www.jamasoftware.com/blog/what-is-requirements-traceability-and-why-does-it-matter-for-product-teams/) is the only way to know if a need or requirement has been fulfilled by the design and the built product. Bi-directional traceability—the ability to perform both forward and backward traceability, usually made possible via requirements management tools like Jama Connect—allows teams to understand why a specific need or requirement exists and [easily analyze the impact of change](https://www.jamasoftware.com/blog/change-impact-analysis-2/)s.

Furthermore, products in [regulated industries must demonstrate traceability to prove compliance](https://www.jamasoftware.com/blog/highly-regulated-industries-need-modern-requirements-management/) with standards and regulations. Therefore, when writing requirements, it’s paramount that each requirement maps to all corresponding artifacts.

Many teams use a requirements traceability matrix (RTM) to track requirements and manage the scope of a requirements change. The RTM is static and maintained manually. The problem is that change is ubiquitous in the product development process. When change happens, teams must manually search the RTM document for all related upstream and downstream needs and requirements, dependent requirements, and verification and validation tests that may be affected by the change.

Scouring through an excel spreadsheet for each change may not be that daunting if there are only one hundred or so needs and requirements. However, if the product has hundreds or thousands of needs and requirements—think complex, regulated products—managing the scope of change becomes a cumbersome and time-draining exercise fraught with risk.

Requirements management tools are designed to streamline the process—even for highly complex, regulated products. Jama Connect, specifically, uses the [benefits of living requirements](https://www.jamasoftware.com/blog/requirements-management-living-not-static/) to:

* Easily determine the impact of change
* Automate compliance
* Enable end-to-end process improvement
* Increase productivity
* Reduce product risk
* Increase quality

#### **Requirements analysis**

Sometimes, there is a gap between what stakeholders say they want and what they actually want. The purpose of requirements analysis is to ensure all business, software, and product requirements accurately represent stakeholder needs and requirements. The goal of requirements analysis is to get a clear understanding of stakeholder needs so the deliverable matches stakeholder expectations.

#### **System Verification Stage**

System verification means determining if the finished product meets the baselined product requirements. This is different than system validation (discussed below), which evaluates whether the product satisfies the stakeholder needs. Both are important, but system verification always comes first.

Planning for this stage starts when the product requirements are being defined. Planning includes determining which system verification events are needed to confirm that product requirements are fulfilled.

To ensure successful system verification, the following attributes should be defined for each product requirement before the requirements are baseline.

* Success criteria (what must be proven to show the requirement was met)
* Method (test, demonstration, inspection, or analysis)
* Strategy (approach to be used, operating environment, test environment, system configuration, etc.)

Defining these attributes sets the expectations for the [test and quality assurance](https://www.jamasoftware.com/solutions/test-management/) work that needs to be done, and reduces the risk of rework down the line.

#### **System Validation Stage**

System validation means determining whether the product satisfies its established stakeholder needs. Successful system validation is what leads to acceptance of the product for its intended use by its intended users in the actual operational environment. For highly regulated products, approval for use is dependent upon successful system validation.  Final product acceptance by a customer also depends on successful system validation.

Planning for system validation starts when the integrated set of needs are being defined.  Planning includes, determining which system validation events are needed to confirm that needs are fulfilled. One approach is to define a set of attributes that address system validation for each need.

To ensure successful system validation, the following attributes should be defined for each need before the needs are baselined.

* Success criteria (what must be proven to show the need was met)
* Method (test, demonstration, inspection, or analysis)
* Strategy (approach to be used, operating environment, test environment, system configuration, etc.)

Defining these attributes sets the expectations for the [test and quality assurance](https://www.jamasoftware.com/solutions/test-management/) work that needs to be done, and reduces the risk of rework down the line.

### Challenges with requirements management

[Requirements management has its share of challenges.](https://www.jamasoftware.com/blog/10-product-development-challenges-engineers-face/) One such challenge is change management. Because change is so prevalent in the requirements management process, teams need to address how they will manage change at the beginning of the project.

When building products that have thousands of requirements and countless changes, teams can spend hours circulating, editing, and tracking changes in an attempt to maintain traceability and simply keep development on track.

This is especially challenging when maintaining the needs and requirements in document form. Version control issues are one challenge that results from change. Versioning problems can arise on the artifact itself. For example, someone might be giving feedback on SRS version one but there is already an SRS version two with different, additional edits. What’s more, version control can be even more granular within the document as well. For example, requirement one might be on version three, which is linked to test B on version two.

Add to that consolidating feedback from multiple stakeholders via email or meetings, analyzing the impact of change across various versions of requirements artifacts, and communicating the correct changes and statuses to the right people. It’s a nightmare just thinking about how to keep it all straight.

The following are the [top five challenges of requirements management](https://www.jamasoftware.com/blog/conquering-the-5-biggest-challenges-of-requirements-management-part-i/):

* Last-minute feedback
* Decision rehashing
* Change tax
* Attention deficit
* Mismatched expectations

It’s easy to recognize that problems are compounded when managing requirements using documents - or legacy systems based on documents – instead of a purpose-built requirements management tool. Most of these challenges, and more, can be overcome by switching from a document-centric to a data-centric approach to requirements management to one that is data-centric.

### How to successfully manage requirements in complex products and highly regulated industries

The challenges above are real and mastering them can have significant impact on development timelines and budget. Developers in regulated industries - like [automotive](https://www.jamasoftware.com/solutions/automotive), [aerospace](https://www.jamasoftware.com/solutions/airborne-systems/), [medical device](https://www.jamasoftware.com/solutions/medical-device/), [government](https://www.jamasoftware.com/solutions/government), and [industrial manufacturing](https://www.jamasoftware.com/solutions/industrial-manufacturing/)- are further challenged by the need to prove compliance with regulations and standards.

Standards and regulations can be sources of hundreds of requirements with which a product must comply. Often, not all of the requirements in a regulation or standard apply to your specific product. Standards and regulations contain requirements not only for products, but also requirements concerning system verification and system validation (acceptance, certification, and qualification), and requirements. On the organization developing the product, as well. In addition, requirements within standards and regulations are often poorly written and ambiguous.

When identifying drivers and constraints at the beginning of the project, the project team needs to identify all applicable standards and regulations which requirements within those standards and regulations actually apply to their specific product. Then, they must write well-formed product requirements to address the intent of those regulatory requirements, such that the product will result compliant with the applicable standards and regulations.

Regulations are not requirements. Requirements must be written to adequately satisfy the regulatory standards. A keen understanding of applicable standards and regulations is paramount in writing requirements that will lead to a product’s compliance. Once written, a regulatory requirement should be attributed as such in all requirements artifacts. This can be done be assigning a “compliance” attribute. This provides all team members visibility throughout development, which assists with decision making and change analysis.

In addition to expert knowledge of the standards and regulations, the following are needed to maintain product compliance when building a product:

* Collaboration between teams
* Single source of truth for defining, verifying, and validating compliance requirements
* Standard frameworks aligned to standards and regulations
* Traceability across all development activities and resulting artifacts.
* Simplified audit preparation and data exports

Reporting is key when faced with a compliance audit. Teams must know what data is required for the audit and how to easily access it when needed. Often, audit trails are an afterthought in which teams are scrambling to pull together data from several sources. This is a dangerous practice, though, because it can put delivery timelines and launch deadlines at risk. It can even put the launch at risk of failing altogether. Teams that must demonstrate compliance, must eliminate this risk with a detailed, digital audit trail that can be easily exported whenever needed. Establishing and managing traceability is critical to maintaining the audit trail.

[Many leaders in regulated industries rely on requirements management tools to reduce the risk of failure to comply with standards and regulations during the product development process. Jama Connect, for example, can track standards and regulations and compliance when building products. These industry leaders leverage Jama Connect to keep them at the forefront of innovation:

* [Arteris IP manages complexities of autonomous vehicle development](https://www.jamasoftware.com/blog/how-arteris-ip-leverages-jama-connect-for-autonomous-vehicle-development/)
* [Spaceflight brings the high costs of doing business in space down to Earth](https://resources.jamasoftware.com/aerospace-and-defense-customer-stories/spaceflight-customer-story)
* [Infineon transitions from a document-centric to data-centric development flow](https://resources.jamasoftware.com/customer-story/jama-software-infineon-customer-story)
* [3Shape is changing the shape of dentistry](https://resources.jamasoftware.com/customer-story/how-3shape-is-changing-dentistry-with-jama-connect)

### Benefits of requirements management tools

The advantages of RM tools are many. A modern solution can eliminate the risks and inefficiencies of documents and legacy systems. A valuable RM system, like Jama Connect, can bridge engineering silos throughout the development process, including test and risk activities. An effective requirements management tool helps improve the product development process by:

* + Ensuring quality and compliance
  + Managing risk
  + Increasing efficiency and optimizing processes
  + Making it easy to understand and respond to change
  + Improving traceability
  + Streamlining and accelerating reviews
  + Enabling real-time collaboration and iteration
  + Saving time
  + Improving quality

There are many requirements management tools available, but only a few that can help reap all the benefits listed above. When thinking about the most important features in a requirements management system, first think about what’s being built. The industry and complexity level will help inform the features that will be best for your team. This [guide to buying a requirements management tool](https://resources.jamasoftware.com/requirements-management/selecting-the-right-requirements-management-tool) may also help.

From our experience, and [what we hear from customers](https://www.jamasoftware.com/company/press/customers-rank-jama-software-the-leader-in-requirements-management-software-on-the-g2-grid-reports-for-summer-2021/), the most important features of a requirements management system are:

* + [Live bidirectional traceability](https://www.jamasoftware.com/platform/jama-connect/features/#tab-id-1)
  + [Real-time collaboration communication](https://www.jamasoftware.com/platform/jama-connect/features/#tab-id-2)
  + [Efficient and scalable review process](https://www.jamasoftware.com/platform/jama-connect/features/#tab-id-3)
  + [Simplified test and quality assurance management](https://www.jamasoftware.com/platform/jama-connect/features/#tab-id-4)
  + [Always-on risk analysis](https://www.jamasoftware.com/platform/jama-connect/features/#tab-id-5)
  + [Reusable requirements and baseline catalogues](https://www.jamasoftware.com/platform/jama-connect/features/#tab-id-6)
  + Standardized validation and functional safety kits
  + Comprehensive visibility and compliance reporting
  + Fast implementation
  + Flexible configuration
  + Easy-to-use interface/administration
  + [Automated integrations](https://www.jamasoftware.com/platform/jama-connect/integrations/)
  + Quality assessment of requirements and traceability