Jama software®

EBOOK

Best Practices Guide for Writing Requirements

Learn how to overcome common challenges and leverage state-of-the-art automation using industry-leading guidance.

Table of Contents

Introduction

Development teams face many challenges in today's competitive, fast-paced market. In this eBook, we'll explore how to navigate a path from a high-level market need or problem to a detailed requirement for an actual product, system, or software. We'll cover best practices for writing requirements in depth and demonstrate how to write those requirements in a manner such that all stakeholders have a clear understanding of the needs around the development of any complex system, software, or product. Additionally, we'll demonstrate how to leverage new tools and automation, such as Jama Connect Advisor™, to increase accuracy and speed in the development process.

WHY EFFECTIVE REQUIREMENTS MATTER

Better requirements lead to clearer, more effective communication between stakeholders. This drives the entire organization toward greater transparency, less rework, and, accelerated development... without sacrificing quality. While writing requirements is both an art and a science that will vary by context, there are a few best practices to consider.

In an ideal world, every individual user, business, and functional requirement would exhibit the following qualities: complete, correct, feasible, necessary, prioritized, unambiguous, verifiable, consistent, modifiable, and traceable. We'll dive deeper into these qualities later in this eBook.

1 Common Challenges of Writing Requirements

Common Challenge #1

The designer/customer disconnect

The first common challenge organizations face when writing requirements is that designers aren't exactly sure what they're supposed to deliver and what the actual end user needs. From the perspective of a designer, the detail that a product marketer or a customer can provide is sometimes insufficient, and many questions remain unanswered. The designers either go on to build something that doesn't result in a successful outcome, or they ask a series of questions to which the product marketer doesn't have clear answers.

Common Challenge #2

Too many details, too little innovation

The second common challenge organizations face when writing requirements is the opposite of the first problem. Teams are given very granular details that over constrain them. In this scenario, designers have little freedom to innovate. This often takes the form of specs from a previous product that you're reusing, or a customer RFP, where customer requirements are highly detailed. And while at times this is necessary, it doesn't often lead to innovation. Designers often end up in a pattern of just doing what they're told in this scenario, instead of thinking about how they could creatively solve a problem.

Common Challenge #3

Right requirement, wrong product

The third common challenge organizations face is where the team builds a product, but it just simply doesn't meet the original requirements. The team may have done well in gathering the problem statement and figuring out the constraints, but inevitably the product somehow drifts outside of what they were originally trying to solve. As design challenges come up, trade-offs are made, and the specifications related to the product slowly drift outside of the scope of an "acceptable" product. In this scenario, the team is so focused on building 'something,' they end up not building the right 'something.'

2 Overcoming Challenges in Writing Requirements

Understanding the Problem You're Trying to Solve

Clearly defining the problem is key to aligning all stakeholders. So, creating problem statements is a great first step in understanding the context of who you're trying to help, what problem they have, and how often it occurs.

This statement gives the team context so when they do get into the weeds of detailed requirements gathering, they're able to trace it back to the original problem or the need that you're trying to fulfill.

Legend has it that Henry Ford may have once said, "If I had asked people what they wanted, they would have said faster horses."

This quote aptly demonstrates how stakeholder needs and problem statements are not the same thing as customer requirements or feature requests. A customer feature request in Henry Ford's day may have been a faster horse or a horse that didn't require as much food. And while those are important requests, they usually don't lead to innovation—like the invention of the automobile.

Interestingly, the problem that Henry Ford solved was not inventing the automobile (as it had already been invented), he solved the affordability problem. The problem he identified (and solved) was that the high cost of acquisition prohibited most people from owning one. Henry Ford made automobiles accessible to more people by revolutionizing the way they were built and bringing down the cost. And this was made possible by truly understanding the problem that they were trying to solve before they set out to solve it.

The Five Whys

A great way to distill the problem your trying to solve is by using the "Five Whys" method. The theory here, is that if you ask "Why?" five times, you'll eventually get to the root cause of the problem.

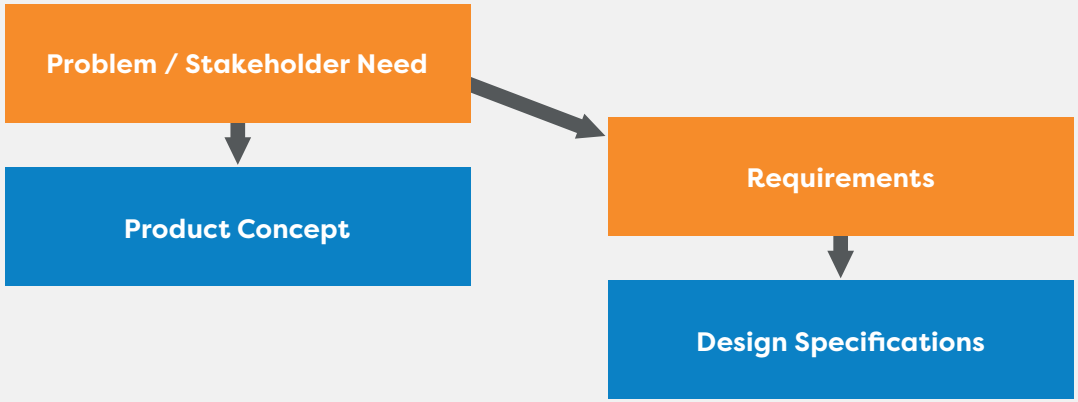
Defining Your Requirements Hierarchy and Taxonomy

Once you've clearly defined the problem, you need to now break it down into more granular needs or requirements. This may take many different forms, like a products requirements document (PRD), a requirements spec, a very long Excel sheet, or a database of requirements. However, remember that using a documents to manage requirements introduces an abundance of risk to your development process, whereas a formal requirements management platform helps maintain upstream and downstream traceability. Clear traceability allows teams to maintain a formal change management process; perform impact analysis; and reveals interdependencies with the process — making it easier to bring in the right decision-makers at the right time.

In order to improve requirements management as a whole, your teams must establish a clear hierarchy and taxonomy. It's important that teams understand the difference between requirements and design as it's often something people get confused about.

Requirements:

A requirement should communicate what the product, system, or software needs to do. The requirements are the tool that we use to identify the right product to build, and ensure we're building it in the right way.



Design:

The design specification, on the other hand, is the response to the requirements and should indicate what the product, system, or software does. Specifications are not useful to identify the right product to build, but they are useful to communicate what the product is and how it works.

This delineation is important because using traditional design specifications as requirements can lead to a lot of problems, like over constrained designers. The goal is to capture the needs of the products, system, or software users and customers without constraining or specifying design.

EXAMPLES:

Poorly worded requirements with design embedded in the statement: *The software algorithm will take two audio signals, each of 100dB dynamic range, and combines them into a single output signal with at least 140dB dynamic range so that we can save power.*

Well written requirement that demonstrates need without constraining design: *The solution shall consume less than 20mA while in operation which is below industry standard of competing devices.*

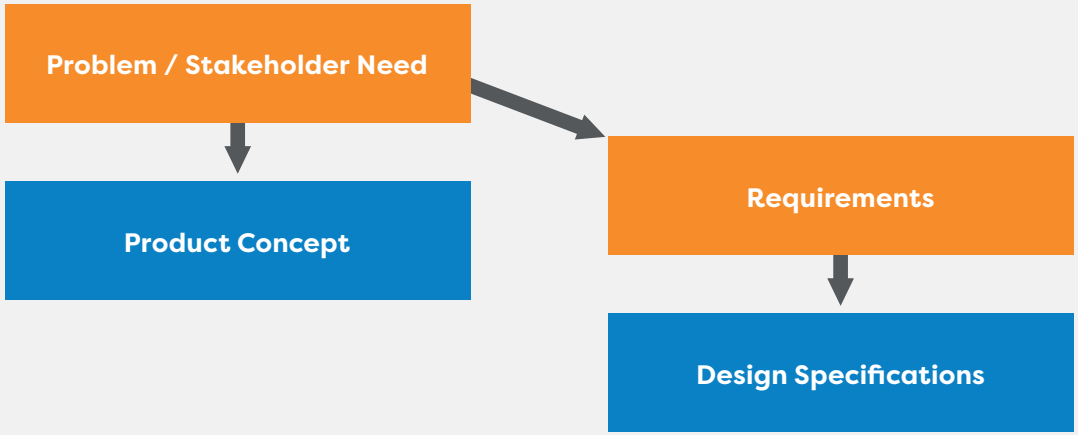
EBOOK

The Jama Software® Guide to Requirements Traceability highlights the importance of tracing requirements without the headaches and risks of a traceability matrix in Excel, but also how to do so in a way that sets your organization up for future success.

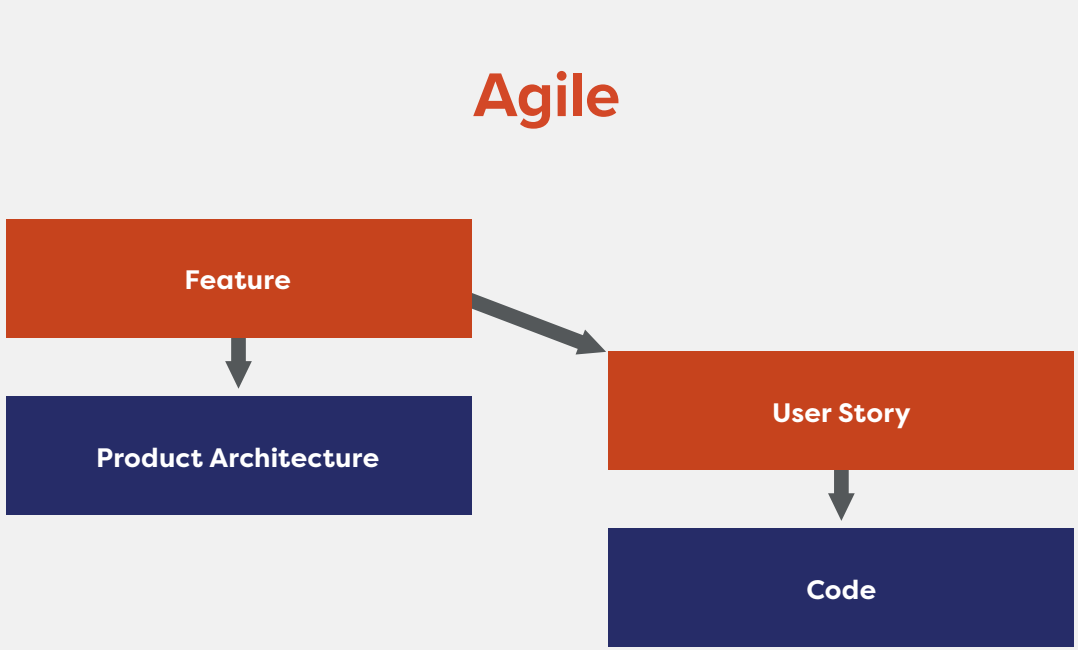
Learn how traceability helps teams:

- Accurately assess the impact of changes
- Track the full history of product development
- Keep everyone in sync
- Consistently improve the quality of the products being built

It's important to come up with a taxonomy of your traceability and your hierarchy. Here are a few examples:

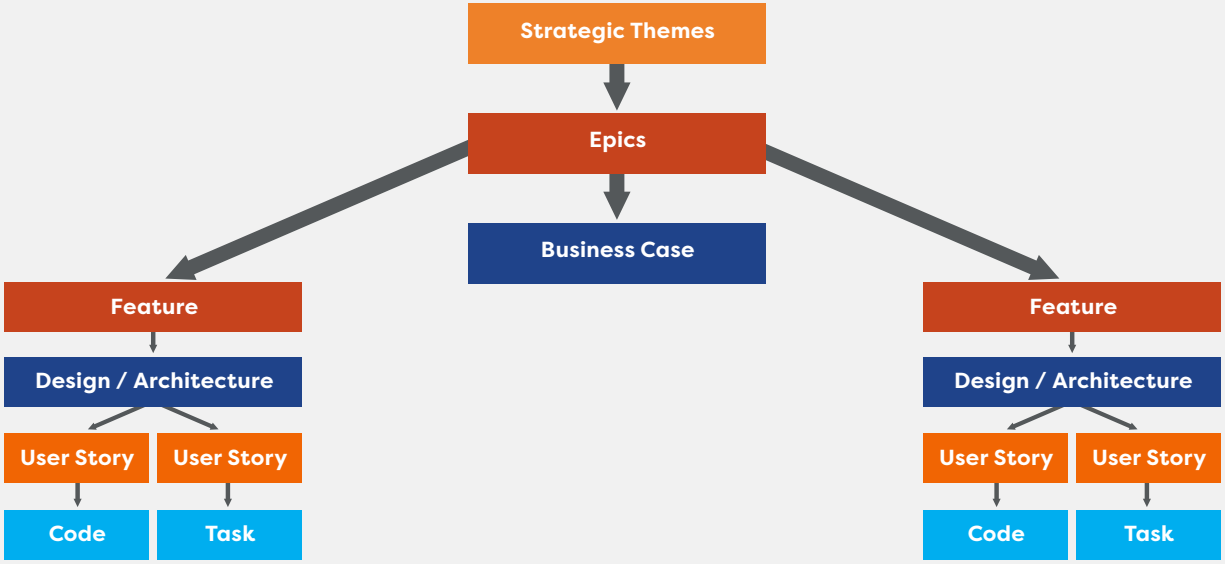


This is a very basic example where you have your problem or your stakeholder need. That decomposes down to requirements, and the blue boxes relate more to the design. Once you've defined the problem or stakeholder need, you might start coming up with a conceptual product or a concept of operations (CONOPS). Whereas from the requirements perspective, you would then be able to start creating actual design specifications from those requirements.



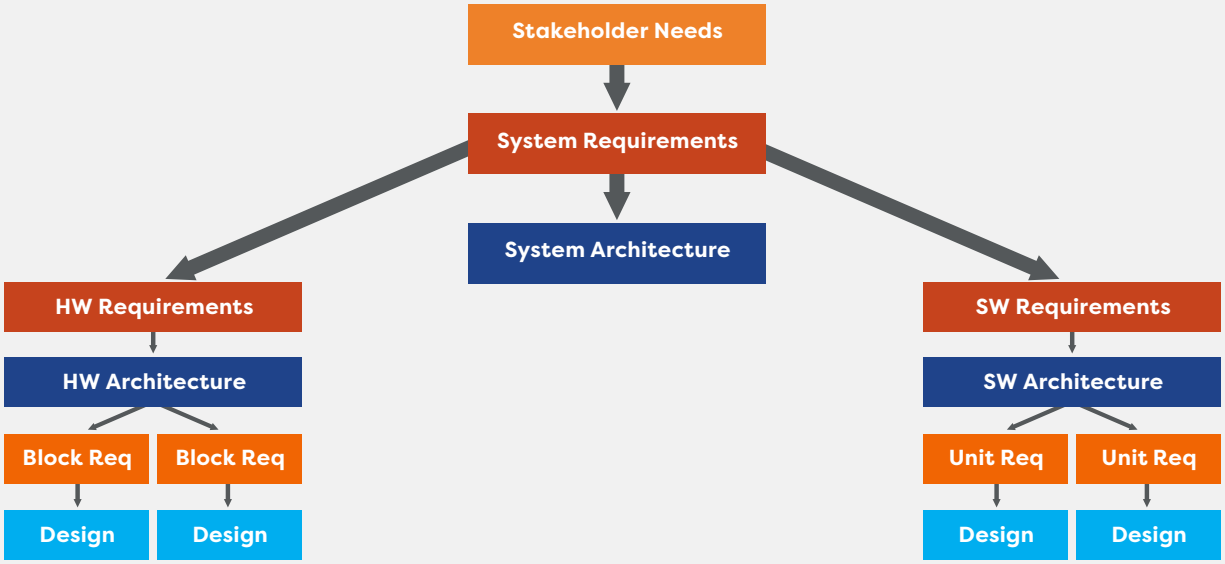
You can also look at this in Agile terms. You might author a list of features or epics that you want to decompose down into user stories. The same principles apply but the semantics may be different. The process of how and when you do this documentation may also differ.

Scaled Agile



Again, if your team is using the Agile methodology instead of defining everything up front, you may experiment and build up your documentation of requirements through several iterations and feedback loops. Often, when making products or systems that are extremely complex, they don't easily fit into a two-level construct. In that case, you would build more layers into the traceability. In this example, we demonstrate strategic themes going down to epics, which decompose into features and user stories. You may notice that there are two branches on either side. That might represent two different modules or subsystems that are living independently but may be interconnected up to that higher-level solution that you're implementing.

Complex Regulated System



We can also look at this with a system engineering perspective for a complex regulated system. Teams who are building life-critical applications or products, for example, are given a very specific method to follow by regulatory bodies. In this case, if teams miss a requirement, or don't properly design and test it, it could lead to the loss of life. Traceability is critical to building safety-critical products like airplanes or cars. You must then determine what taxonomy best fits with the product you're building — keeping in mind that the industry that you're building in is also very important.

“Through Jama Connect, we have been able to fully document our functional and technical requirements for multiple products/projects. For any project in which you need to document detailed requirements, especially software development, Jama would be well suited for use.”

Thai Trinh, Senior Business Analyst
IT & Services Company

Bridge the Communication Gap with Traceability

Irrespective of your industry or your methodology, it's important to clearly define the requirements traceability and differentiate between requirements and design. Because oftentimes, a semblance of a model exists, but people on different teams are using different terms referring to the same thing.

It's critical to be clear on the semantics and have a glossary to define what each of these levels mean. This should be in both your requirements management plan and your requirements management tool. This glossary should define what each of these levels mean and who's responsible for the process and outcomes. When you establish traceability at requirement levels, you're able to decompose the needs into more granular details and demonstrate where you have verification and validation coverage.

Establishing traceability allows you to separate documenting the need from documenting the solution.

Embracing Change

If you're using Agile methodology and iterating often, it's important to be able to change course very quickly and very often. You'll need to be able to see the ripple effects of changing a higher-level epic, and what user stories or tests will be impacted. Traceability makes you quicker and faster down the line when you're ready to develop these products.

With Jama Connect, teams can implement Live Traceability™ which enables any engineer at any time to see the most up-to-date and complete upstream and downstream information for any requirement — no matter the stage of systems development or how many siloed tools and teams it spans. To learn more about the importance of tracing requirements in a way that sets your organization up for success, download this free eBook, "The Jama Software Guide to Requirements Traceability"

3 Improving the Quality of Your Requirements

The “golden rule” of requirements authoring is that you must have clear and effective communication with your stakeholders. If someone new to your company picks up a requirement document, are they going to be able to understand what the requirement is and what the requirement means?

You must also have clear and open communication with your stakeholders. You may have a perfectly written set of requirements, but if you aren’t sharing and collaborating that requirement with your colleagues, you’re going to run into problems. The teams that are going to take your requirements and turn them into a designed solution need to be part of that process. Below we have provided recommendations – not hard rules – for writing requirements. Use these as guidelines but you’ll want to tailor them to your organization’s specific and unique needs - because well written requirements require a mix of both science and art.

EBOOK

Requirements are the foundation of a smooth-running process and are the essential inputs to your mission-critical projects. Effectively managing the flow of changes and refinements early in your lifecycle will significantly reduce both quality issues downstream and the volatility that plagues so many projects.

Download our free eBook Best Practices Guide to Requirements and Requirements Management to learn more about:

- The business value of better requirements
- The four fundamentals of requirements management
- Finding the right level of detail in requirements

Characteristics of Effective Requirements

Necessary

Each requirement should be necessary or required.

- Each requirement should document a capability that the stakeholders really need or one that's required for conformance to an external system requirement or a standard.
- Every requirement should originate from a source that has the authority to specify requirements. Trace each requirement back to specific voice-of-the-customer input, such as a problem statement, stakeholder need, or industry standard.

Unambiguous

Each requirement should be simple, concise, and clearly defined.

- All readers of a requirement statement should arrive at a single, consistent interpretation of it, but natural language is highly prone to ambiguity. Write requirements in simple, concise, straightforward language appropriate to the user domain. Define all specialized terms and those that might confuse readers in a glossary.

EXAMPLE:

Ambiguous: "The car shall be blue"
Unambigious: "The car shall be sky blue (#7BAFD4)"

Feasible

Each requirement must be possible to implement.

- It must be possible to implement each requirement within the known capabilities and limitations of the system and its operating environment.
- To avoid specifying unattainable requirements, have a designer or developer work with marketing or the business analysts throughout the elicitation process.
- The technical resource can provide a reality check on what can and cannot be done technically and what can be done only at excessive cost.
- Incremental development approaches and proof-of-concept prototypes are ways to evaluate requirement feasibility.

Verifiable

Each requirement should be measurable or testable.

- See whether you can devise a few tests or use other verification approaches, such as inspection or demonstration, to determine whether the product properly implements each requirement.
- If a requirement isn't verifiable, determining whether it was correctly implemented becomes a matter of opinion, not objective analysis. Requirements that are incomplete, inconsistent, infeasible, or ambiguous are also unverifiable.
- The key here is that you need to add a qualifying objective, or you need to change this requirement statement to make it more measurable or testable. It's always a good idea to avoid adverbs, also known as any word that ends with "ly." Adverbs typically are not testable.

EXAMPLE:

Incorrect: "The car must run quickly"
Correct: "The car must be able to reach 100 mph in less than 10 seconds"

Correct

Each requirement must accurately describe the functionality to be built

- The reference for correctness is the source of the requirement, such as an actual user or a high-level system requirement. A software requirement that conflicts with its parent system requirement is not correct.
- Only product users can determine the correctness of requirements, which is why users, or their close surrogates should review the requirements.

Reviewing and Discussing Requirements

Reviewing and discussing requirements leads to a shared understanding of what you're setting out to build and why.

Taking time up-front to review needs & requirements:

- Gives you feedback and makes you a better author
- Increases shared understanding amongst team
- Helps define acceptance criteria and ensure testability
- Reduces surprises and missed requirements

4 Templates for Writing Effective Requirements

Requirement Templates

[Trigger]	[Precondition]	Actor	Action	[Object]
When a collision is detected	and the passenger airbag switch is on	the system	SHALL detonate	the passenger airbags

User	Need	Reason / Purpose	Objective
As a driver	I need to know my vehicle speed	so that I can abide by traffic laws	without losing vision of the road

5 Leveraging Automation for Writing Clear, Concise Requirements, Every Time

Leveraging Automation for Writing Clear, Concise Requirements, Every Time

Successful product delivery starts with having the right user needs and requirements. Efficient, precise, and professionally written requirements form the foundation of the product development process so that various teams (design, software, and hardware systems) can all work together with a shared and clear understanding of the project goals.

Jama Connect Advisor™ is a state-of-the-art requirements authoring guide and optimizer powered by natural language processing for engineering that helps a system engineer or a product developer write effective, well-organized requirement specifications based on industry-accepted INCOSE (International Council on Systems Engineering) rules and the EARS (Easy Approach to Requirements Syntax) notation.

With Jama Connect Advisor, teams can:

Leverage natural language processing for efficient assessment of alignment to industry-leading practices for requirements authoring based on INCOSE Rules and EARS Notation: The Easy Approach to Requirements Syntax.

Get advice delivered during the authoring task, and directly within Jama Connect to optimize workflow efficiency. Have a direct integration within Jama Connect cloud as an add-on capability

Why Use Jama Connect Advisor?

- Improve the quality and usability of your requirements
- Save time authoring, reviewing, and updating requirement statements.
- Continuously enhance team requirement authoring skills with regular use.
- Deliver programs and projects on time and on budget with long-term success.

To learn more about Jama Connect Advisor, download our datasheet >>

Conclusion

Writing requirements and requirements management can appear to be a complex topic, but at its core, it's a simple concept. It helps teams answer the question: Does everyone — from business leaders to product managers and project leaders to developers, QA managers and testers — understand what is being built and why?

When everyone is collaborating and has full context and visibility into the discussions, decisions, and changes involved in product development, they maintain high quality and almost always ensure success.

To learn more about how Jama Software can help your team improve the quality of your requirements writing and requirements management, **get in touch with us today.**

If you're already a Jama Software customer and struggling with writing quality requirements, sign up for a **Requirements Quality Assessment** to help understand your Requirements Quality Score, along with with recommended actions for improvement.

"You won't learn how to write good requirements from reading a book... You need practice. Constructive feedback from reviewers can help anyone become a better writer. In fact, it's essential."

Karl Weigers, *"Writing High Quality Requirements"*

ABOUT JAMA SOFTWARE

Jama Software® is focused on maximizing innovation success in multidisciplinary engineering organizations. Numerous firsts for humanity in fields such as fuel cells, electrification, space, software-defined vehicles, surgical robotics, and more all rely on Jama Connect® requirements management software to minimize the risk of defects, rework, cost overruns, and recalls. Using Jama Connect, engineering organizations can now intelligently manage the development process by leveraging Live Traceability™ across best-of-breed tools to measurably improve outcomes. Our rapidly growing customer base spans the automotive, medical device, life sciences, semiconductor, aerospace & defense, industrial manufacturing, consumer electronics, financial services, and insurance industries.

To learn more, visit jamasoftware.com