

# Connect the Dots: 5 Tips on Traceability to Skillfully Control Change and Improve Quality.

## Overview

### Traceability Helps You Stay Connected, Manage Change & Improve Quality. Yes!

What is traceability? It sounds complicated. Why do teams do it? It sounds like a lot of work. Is it really worth the effort? Good questions. The short answer: Yes. And, here's why it's so important: Change happens.

If managed poorly, change will wreak havoc on even the most talented and experienced development teams. If managed skillfully, using tools like traceability, teams are better equipped to assess the impact of changes, track the full history, keep everyone in sync and (deep breath) consistently improve the quality of the products being built – every project, every release. Sign me up.

In some industries, traceability isn't an option, it's mandated. We'd recommend that regardless of the industry you're in or the process you use – whether you're building components for commercial airplanes using waterfall or designing the next killer iPhone app using an agile method – traceability is a best practice that will benefit your team greatly.

In this paper, our goals are to demystify traceability and its related concepts, and provide five practical tips to help you take control and keep everyone in sync.

**Did You Know? “Companies with mature requirements management and traceability processes achieve 75% higher success rates.” – IAG, Business Analysis Benchmark, 2009**

### The Five Tips for Mastering Traceability:

1. Create relationships to connect everyone and everything together with Trace Relationships
2. Ensure you have proper coverage using a Traceability Matrix
3. Assess the impact of a change before it occurs with Impact Analysis
4. Document changes for complete visibility and a detailed audit trail with Version History
5. Keep communication flowing and the team in sync with smarter, real-time Email Notifications

At Jama, we believe concepts are nice, but actions are what really matter. We hope this paper will aid you in taking action to improve your requirements management process using traceability. **Read on...**

# Challenges

## Swimming Upstream or Downstream without a Trace is Risky Business.

**Do these scenarios sound familiar?** You just got a great piece of feedback from your best customer mid-project, and a high-level business requirement needs to change. How will this change impact the functional requirements your developers are working on right now? How will it impact scope for the upcoming release? Your QA team just found a deal breaker of a bug in your most popular new feature and you're two weeks away from launch. Do you ship with the known bug or delay the launch? Who is working on that feature? Who else needs to be notified and weigh in on the decision? What else does it affect? These scenarios occur daily for development teams. So, how do you deal with them? One of the tools in your arsenal is traceability.

**Is traceability worth it?** One common challenge that teams face in implementing traceability is the incremental time and costs involved. There's no question that in order to do traceability well, there is a time investment that's needed upfront to set-up the trace relationships and configure coverage reports. However, the incremental costs incurred with using traceability are small compared to the time and money you will save further along in the development process due to the benefits that traceability provide.

# Benefits

## Your Ticket to Greater Project Success – On Time, On Budget and within Scope.

For most organizations, the benefits outweigh the time required to set-up traceability by at least 2x. With a consistent process, structured templates and the help of a modern requirements management tool, much of the process can be automated and streamlined. Even if you opt to manage it manually, traceability offers several valuable benefits to your organization:

- **Minimize Risk:** Assess risks and the overall impact of a change before it's made
- **Control Scope Changes:** Effectively manage change throughout the process and avoid scope creep
- **Improve Quality:** Ensure quality standards are met or exceeded to achieve industry compliance
- **Reduce Development Costs:** Avoid gold plating and costly engineering rework
- **Increase Productivity:** Keep the team in sync and reduce administrative overhead
- **Complete Test Coverage:** Ensure all requirements are properly tested before a release
- **Greater Visibility:** Traceability offers visibility into the process for the entire team and stakeholders
- **Accelerate Innovation:** Cut product planning and developments cycles in half

---

*“Companies that desire to be Best-in-Class should consider better requirements definition early on. This includes all aspects of requirements management such as better process, tools, workflows, ownership, traceability and change management...” – Maryane Chapman, Director of Integrated Systems Engineering, Pitney Bowes, Aberdeen “System Engineering” Benchmark Study, 2009*

---



# Terminology

## Let's Demystify Traceability and Related Concepts.

Before we dive into the five tips, let's take moment to define a few terms to make sure we understand the lingo.

Terminology	Definition
<a href="#">Traceability</a>	Traceability is a sub-discipline of requirements management. Traceability documents the life of a requirement, tracks every change and links its relationships with other items within a project.
<b>Trace Relationship</b>	A link between items within the scope of a project, used to help assess impact on other items when a change occurs.
<b>Upstream</b>	Upstream relationships, a.k.a. backward traceability, look at the links between detailed functional requirements back up to the original customer need and high-level requirements captured. It's used to ensure that the evolving product remains on track in regards to the goals of the product and what customers need. Helps to avoid scope creep and gold plating.
<b>Downstream</b>	Downstream relationships, a.k.a. forward traceability, look at the links between a high-level requirement and the functional requirements, test cases, tasks, defects and other items that support it. It's used to ensure that you're building the right product.
<a href="#">Traceability Matrix</a>	A traceability matrix is created by associating requirements with the work products that satisfy them. Often it's used to track tests associated with the requirements on which they are based and the product tested to meet the requirement.
<a href="#">Impact Analysis</a>	Using impact analysis, the traceability links between requirements, specifications, design elements, and tests are captured, and these relationships can be analyzed to determine the scope of an initiating change.
<b>Version History</b>	Used for change control, a detailed history of each version of a requirement, and other items, is documented and stored in a unified system of record, enabling complete audit trails used over the lifecycle of the requirement. Required for industry compliance in specific industries such as aerospace and medical devices.
<b>Suspect Links</b>	Suspect links help to manage the impact of requirement changes. A trace relationship (or link) becomes suspect after a requirement in the relationship changes. A suspect links report is often used along with Impact Analysis for assessing impact before making a change.
<a href="#">CMMI</a>	Created by the Software Engineering Institute, CMMI models provide guidance for developing or improving processes that meet the business goals of an organization. As it relates to traceability and requirements management maturity, see Levels 2-3.

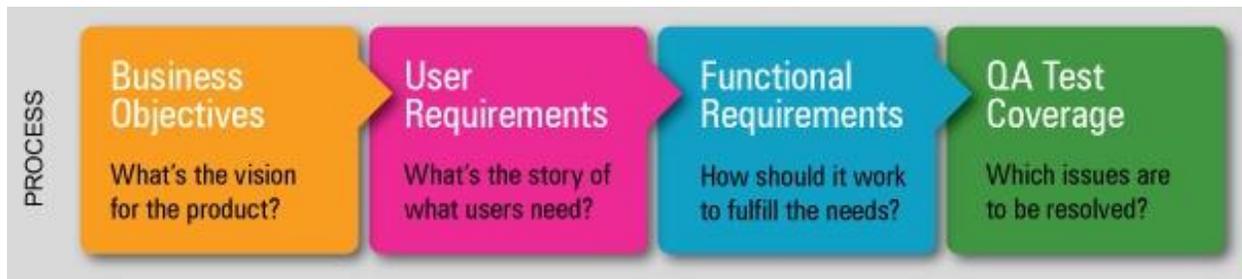
**Note:** For additional background on traceability and a few of the related concepts, click on the term in the above table with blue hyperlinks and it will take you to the related *Wikipedia* page to learn more.



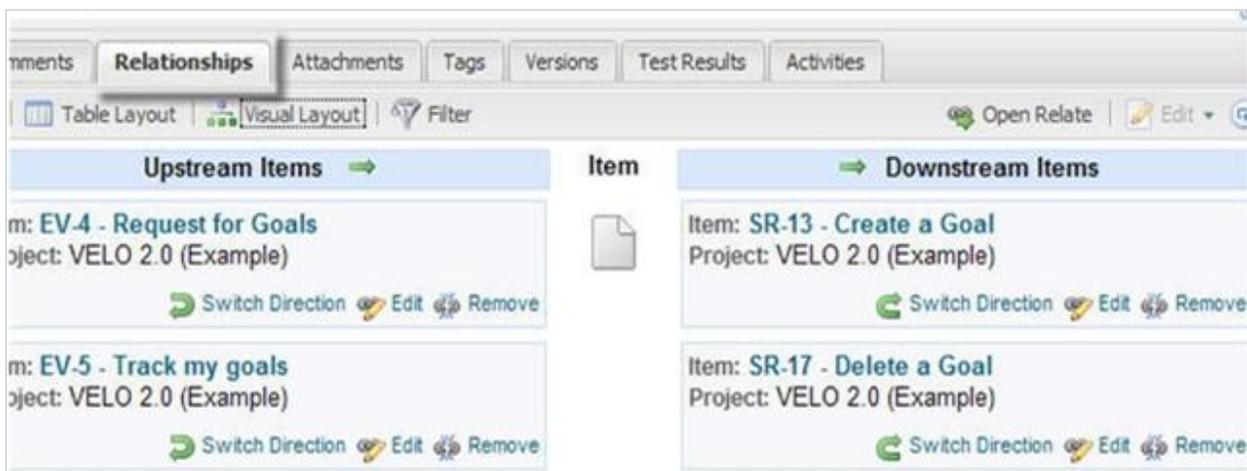
## Tip #1

### It's like the Six Degrees of Separation from Your Business Objectives.

Look at that, we managed to slip in a [Kevin Bacon reference](#). It isn't just because we're fans of the movie *Quicksilver*, it actually has relevance here. As in many aspects of life, your product development success is highly dependent on relationships. All of the details such as user requirements, functional requirements, test cases and other items that define the scope of what you're building are related in some fashion, either directly or indirectly. Here's an example of a common process flow.



Using trace relationships you can connect everything together to map out the interdependencies between the different items. These relationships are the foundation for doing traceability effectively. As an example, here's a screenshot of a Visual Traceability Layout showing both upstream and downstream items related to this requirement (item).



In addition, trace relationships are as much about connecting together the people involved as it is about connecting together all the items. Each requirement in the system has customers, stakeholders and members of your team associated with it. There are analysts who own defining it. There are developers building it. There are QA engineers testing it. And, there are stakeholders and customers who care about its status.

When one item changes it has a ripple effect on other related items and the people associated with the items. Keeping track of this ripple effect is crucial to the success of your projects. It's one of the primary reasons organizations do traceability.

## Tip #2

### Mr. Anderson – Welcome to the Traceability Matrix.

Wow, and now a reference to *The Matrix* movie, we're on fire. In all seriousness, a Traceability Matrix isn't science fiction. It's very real, and can be a valuable report for helping you ensure complete test coverage. For a manual example of a Traceability Matrix, you can build one in Excel such as this one courtesy of Joyce Ludwig.

ID	User Requirements	Forward Traceability
U2	Users shall process retirement claims.	S10, S11, S12
U3	Users shall process survivor claims.	S13

ID	System Requirements	Backward Traceability
S10	The system shall accept requirement data.	U2
S11	The system shall calculate the amount of retirement.	U2
S12	The system shall calculate point-to-point travel time.	U3
S13	The system shall calculate the amount of survivor annuity.	U3

This simple insurance claims system example shows both forward and backward tracing between user and system requirements. User requirement identifiers begin with "U" and system requirements with "S." Tracing S12 to its source shows this requirement is problematic, and should be rewritten to support the processing of survivor claims or the traceability link corrected.

Here is an example of an automated traceability matrix generated from Contour. In this example, the matrix is reporting on the relationships between Features and Test Cases. This is useful to identify gaps in test coverage, which is a popular use of a traceability matrix to ensure each feature is properly tested before its release.

Trace Matrix									
Left: Features					Top: Test Cases				
Matrix Table	2.5.1 Upgrade Patch	Add/Edit Item	BIRT Reports	Configure Sets	DB Test: SQL Server, Oracle	Filter Items	Internationalization	Item Activity	LDAP Configuration
A pick-list can be defined as a multi-select component									↩
Ability to add an attribute to relations	↕	↕							
Access Drafts from My Profile				↕					

## Tip #3

### Impact Analysis – Your Virtual Crystal Ball into the Future.

What if you could anticipate the impact a change would have on your project and the entire team before it occurred? Will this change request send the development team over the edge or do they have bandwidth to squeeze it into the next release? These insights are possible and the tool to get them doesn't require any magical pixie dust, just Impact Analysis. Impact analysis relies on the trace relationships you've set-up, and it reports on the complete picture of all the items that are affected – both directly and indirectly.

Here's an example of an automated impact analysis report for a high-level business requirement. If it were to change, four directly related system requirements are affected and one indirect use case is also impacted.

The screenshot shows the 'Impact Analysis' tool interface. On the left, the 'Target Set' includes System Requirements, Business Requirements, Test Cases, Change Requests, Use Cases, Defects, Issues, and Evidence. On the right, the 'Source Item(s)' table shows a Business Requirement for 'Ability to set goals'. Below that, the 'Direct Relationship' table lists four System Requirements affected by the change. Finally, the '2 Degrees of Separation' table shows one Use Case affected.

Project	ID	Name	Set	Ass
VELO 2.0 (Example)	BR-3	Ability to set goals	Business Requirement	

Project	ID	Name	Set	Ass
VELO 2.0 (Example)	SR-13	Create a Goal	System Requirement	Bar
VELO 2.0 (Example)	SR-17	Delete a Goal	System Requirement	Unas
VELO 2.0 (Example)	SR-15	Share a Goal	System Requirement	Unas
VELO 2.0 (Example)	SR-14	Edit a Goal	System Requirement	Unas

Project	ID	Name	Set	Ass
VELO 2.0 (Example)	UC-5	Create a Goal	Use Case	Cont Adm

Wow, now if only we could apply impact analysis to other aspects of our lives, like the decision to have that second helping of pumpkin pie on Thanksgiving. What's the impact on "Project Fat Tire"? Oops, we better expand the scope on that one and push out the release date until New Year's.

*"Any single change can ripple through many different documents. Having a way through Contour to manage the "ripple effect" is incredibly valuable."* – Mace Volzing, Software Development Manager, Intrapace, Medical Devices



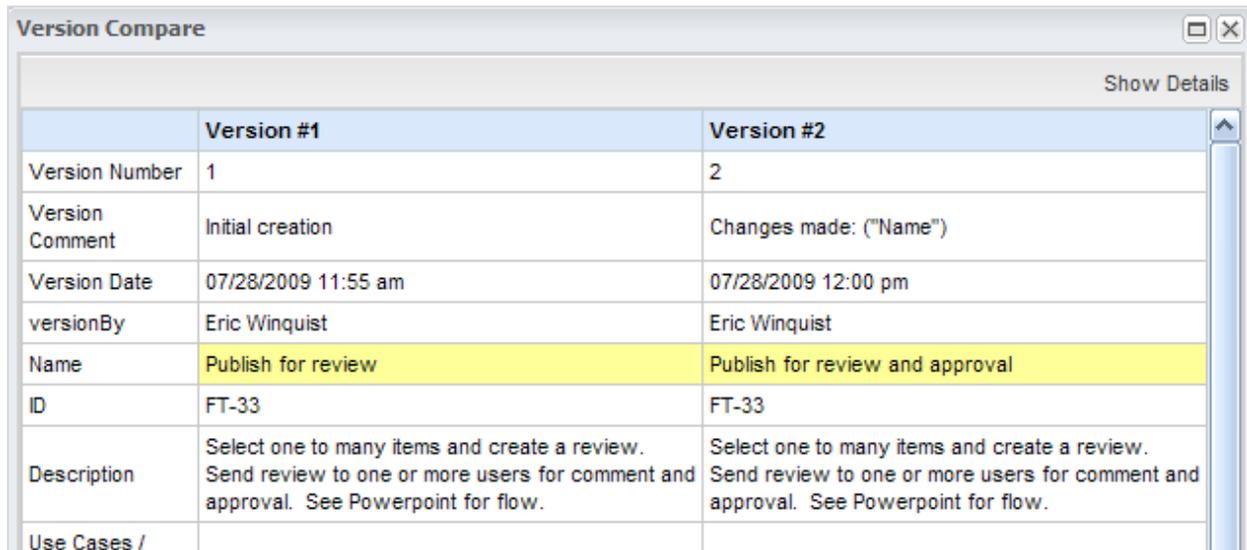
## Tip #4

### Version History – Your Complete Audit Trail in Rich, Glorious Detail. Whoohoo!

If you prefer things at a high-level and don't like to dive into the details, look away. This tip isn't for you. This tip on Version History is for those among us that like to roll-up the sleeves and get deep into the glorious details of every change. It's also been humorously referred to as the "CYA tip", for coverage of a different kind.

Personal motives aside, capturing a complete and detailed record of all changes is a critical element for reaching higher levels of requirements maturity within your process, such as CMMI. It's also helps companies meet industry compliance standards in specific fields such as aerospace and medical devices. One of the benefits of doing traceability is having a comprehensive audit trail of changes, so you can analyze who, what, when and why a change occurred. At the same time, you can easily roll-back to an earlier version if needed because it's all stored in the unified system of record.

Here's an example of seeing a side-by-side comparison of two versions, using an automated process within Contour. For efficiency gains, the specific field that changed is highlighted in yellow, so you don't have to spend time hunting around the full requirements specification document to pinpoint and understand precisely what changed. Viva la details!



	Version #1	Version #2
Version Number	1	2
Version Comment	Initial creation	Changes made: ("Name")
Version Date	07/28/2009 11:55 am	07/28/2009 12:00 pm
versionBy	Eric Winquist	Eric Winquist
Name	Publish for review	Publish for review and approval
ID	FT-33	FT-33
Description	Select one to many items and create a review. Send review to one or more users for comment and approval. See Powerpoint for flow.	Select one to many items and create a review. Send review to one or more users for comment and approval. See Powerpoint for flow.
Use Cases /		

As with the other aspects of traceability, you can track version history manually through static documents using versioning. It's just more cumbersome and time consuming to manage complex projects that way.

*"We use Contour to provide a clear workflow of our requirements, understand impacts when requirements change and ensure proper test coverage." – Christopher Moustier, QA Manager, Wyplay, Home Entertainment Systems*



## Tip #5

### Avoid Noise. Communicate Changes Quickly, Intelligently and to Those Who Care.

How often have you been involved in a project where “change notice paralysis” sets in after about 3-4 weeks of inbox overload? Usually it occurs when the entire team is on a project-wide distribution list and the project manager is on the hook to send out an email with the complete 200-page Software Requirements Specification document attached for every little change that occurs. Right intention, wrong solution. What happens next? People either waste time hunting around in the requirements spec trying to determine if the latest change is relevant to them – which is costly. Or, they tune out the email barrage as noise and become vulnerable to missing a change that is important to what they’re working on – which is even more costly.

**There are smarter ways to keep everyone on the same page.** You need to ensure that everyone that’s impacted by a change is in the loop. At the same time, you don’t want to flood the entire organization with irrelevant emails. What do you do?

In this example using Contour, when a change occurs, you can instantly send a direct link to the specific requirement that changed with version notes to just the relevant groups or individual users that are affected by it. The notification step is then part of the overall change management workflow. Stay in the loop. Avoid noise.

**Edit Item**

### Commit and Notify Users

**Version Notes**

Made edit per customer call to increase parameters on ability to track results against goals set.

**Choose Recipients**

**Groups** **Users**

Click below to add

- Project Team
- Development
- Project Sponsors
- Test Team
- Analysts
- Design Team

**Selected Groups and Users**

Click below to remove

- Contour Administrator

*“Contour is flexible, easy to use and gives us **complete traceability** for all of our requirements.” – Erik Johansson, Software Engineer, W.M. Keck Observatory*



## Let's Automate!

### Accelerate Development by 50% and Improve Quality 2x by Automating Traceability.

You can manage traceability manually using Microsoft Word or Excel documents. That's a real option. For small teams and simple projects, that's probably all you need. We've provided links to a few free templates courtesy of industry experts that you can use to manage traceability manually:

- [Traceability Matrix Template](#) (.xls), courtesy of CDC.gov, requirements management templates
- [Impact Analysis Checklist](#) (.doc), courtesy of Karl Wieggers, *Process Impact*

**What's the right time to automate?** The challenge with a manual solution is it can be extremely time consuming and cumbersome if your projects have any level of complexity – meaning you have many requirements, frequent scope changes or if members of your team are remote working from different locations. In these scenarios, automation can provide a huge boost to productivity, saving you time and money in the long run. Automation also minimizes the risks of human error, which is always possible despite the best intentions and most skillful staff.

**What's the ROI?** The return on investment is different for every company, but through our experience, we've seen as high as a 42:1 cost-to-benefit ratio for a global entertainment company. For most organizations, as a conservative benchmark, you can expect to speed development cycles by at least 50% and improve quality by 2x or more within the first 6 months using a requirements management solution that automates traceability.

If you're interested in automating your process, we recommend you evaluate a few different requirements management tools to find the right one for you. As one option, you can explore the traceability features of Contour. It's designed specifically as a collaborative, Web-based platform to help teams solve the traditional requirements management challenges and automate traceability, so you can reap the rewards:

- Save time and money
- Accelerate development cycles
- Improve quality and compliance

---

## Take Action!

Take the next step toward mastering traceability – watch the 5 minute how-to video.

**The Moment of Truth:** Concepts are nice, but actions rule. Take action today to become a Traceability Master.



[Click to watch the video and access other resources to help you implement traceability.](#)



## About the Authors



### **John Simpson, Director of Customer Outreach & Marketing**

John represents the voice of the customer in Jama's product strategy and communications. He has over 14 years experience working at software and Web technology companies including Microsoft, WebTrends, Omniture and ZAAZ. He has contributed to several books, whitepapers and presentations on marketing and technology.



### **Eric Winqvist, CEO and Co-founder**

Eric founded Jama with the vision of providing customers a more collaborative way to develop new products and eliminate the common frustrations with traditional approaches to requirements management. Eric is an accomplished entrepreneur, business analyst and project manager with over 15 years experience working with a wide range of organizations. Previous to Jama, Eric founded Redside Solutions, a software development consulting firm.

## Let's Build Great Products.

Our team at Jama is working to provide solutions that make it easier to implement a social product development platform. We're collaborating with companies across industries, from agile startups to some of the world's largest, most innovative organizations such as Intel, Merck, Emerson, Amgen, Wells Fargo, Bio-Rad, SMART Technologies and others to design new ways to smash information silos, speed innovation and build great products. Contour, Jama's [Web-based requirements management software](#), is now trusted by thousands of users worldwide managing billions in R&D projects. Join us.

Contact us if you'd like to learn more. Visit [www.jamasoftware.com](http://www.jamasoftware.com) or follow us on [Twitter](#). We love to hear your thoughts and share best practices.

---

*"Prior to Contour, requirements were captured in a document format and lacked **traceability**, search/analysis capabilities and the ability to collaborate across complex project teams." – Ryan Palmer, User Centered Design Group, Intel*



# Roll the Credits

Footnotes on statistics and articles we referenced during our research. Enjoy.

1. Business Analysis Benchmark Study, IAG, 2009  
<http://www.iag.biz/resources/library/business-analysis-benchmark.html>
2. "Systems Engineering – Top Four Design Tips to Increase Profit Margins for Mechatronics and Smart Products", Michelle Boucher, Aberdeen Group, November 2009  
<http://aberdeen.com/summary/report/benchmark/6044-RA-system-engineering-mechatronics.asp>
3. Booz Allen Hamilton: Over \$1 trillion is spent annually on innovation (R&D), "Global Innovation 1000 Report", October 2009 [http://www.booz.com/media/uploads/Innovation\\_1000-2009.pdf](http://www.booz.com/media/uploads/Innovation_1000-2009.pdf)
4. "Why Software Requirements Traceability Remains a Challenge," by Andrew Kannenberg, Garmin International and Dr. Hossein Saiedian, University of Kansas  
<http://www.stsc.hill.af.mil/crossTalk/2009/07/0907KannenbergSaiedian.html>
5. "Requirements Traceability," by Neville Turbit, Project Perfect  
[http://www.projectperfect.com.au/info\\_requirements\\_traceability.php](http://www.projectperfect.com.au/info_requirements_traceability.php)
6. Wikipedia definition of Traceability  
[http://en.wikipedia.org/wiki/Requirements\\_traceability](http://en.wikipedia.org/wiki/Requirements_traceability)
7. "Bidirectional Traceability," by Linda Westfall, the Westfall Team  
<http://www.compaid.com/caiinternet/ezine/westfall-bidirectional.pdf>
8. Traceability Matrix example, Joyce Ludwig.  
[http://www.jiludwig.com/Traceability\\_Matrix\\_Structure.html](http://www.jiludwig.com/Traceability_Matrix_Structure.html)
9. CMMI overview by SEI  
<http://www.sei.cmu.edu/cmmi/>

