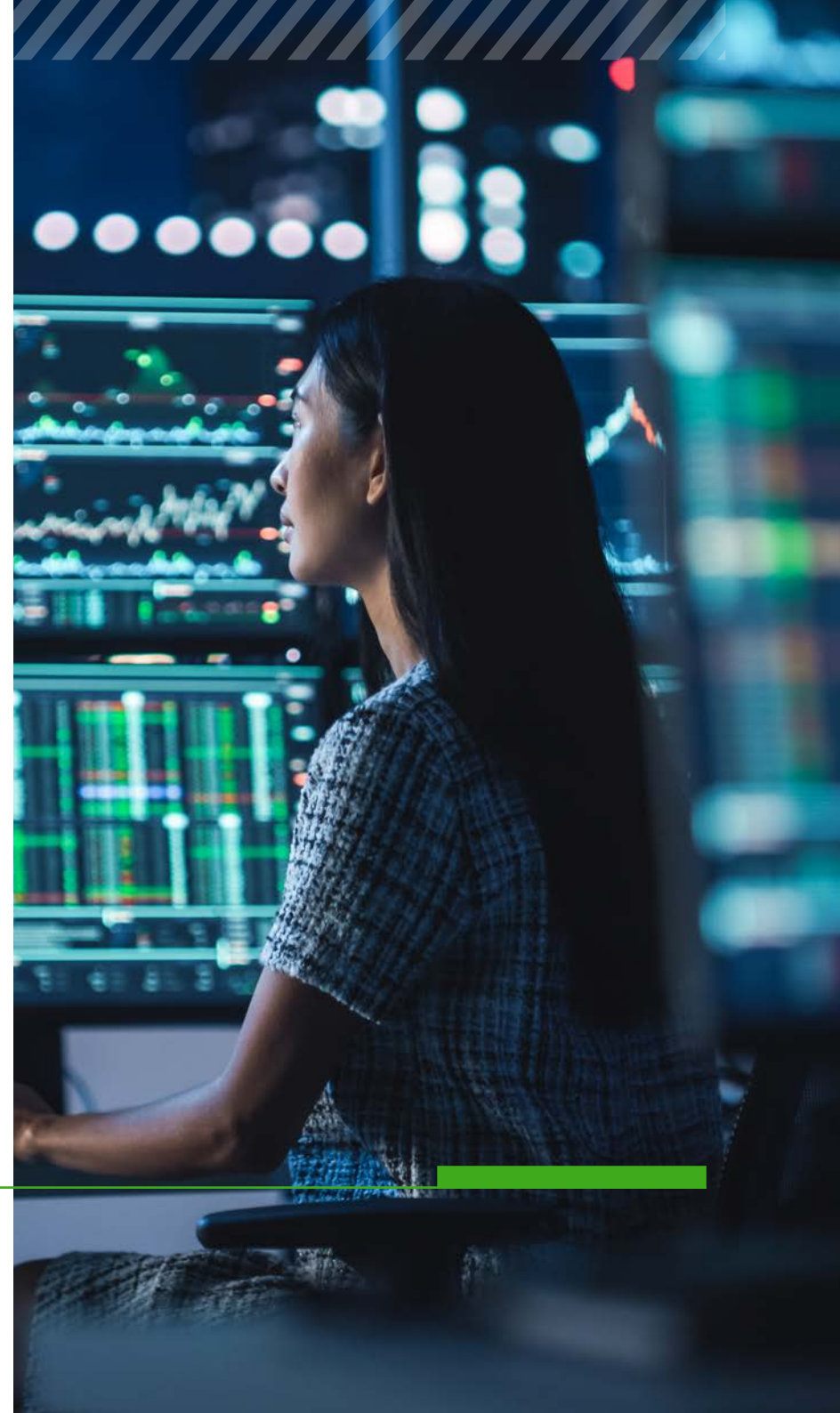




DIGITAL TRANSFORMS PHYSICAL

Requirements Management Challenges & Solutions

**Tips to eliminate your
most frustrating Requirements
Engineering problems**



Introduction

In the age of digitalization, software is everywhere. As software competences are becoming the currency in an increasingly digital market, heavy competition forces product developers to focus on improving their innovation processes, tools, and teams.

Requirements are critical to the success of any software project. In fact, Requirements Management (RM) issues are among the most often-cited causes of failures in technology projects. More and more product developers are realizing that their old ways of managing requirements don't guarantee success in the face of new challenges. In order to avoid failure, they are searching for ways to enhance the efficiency and accuracy of their requirements engineering activities.

This eBook takes a look at the most common RM problem areas that product developers are challenged by, and the strategies that successful teams apply to overcome these issues!



Problem #1

Insufficient Communication and Documentation

Let's start off with a broad group of problems. Overall, the primary reason that most software projects are derailed is the lack of clear and straightforward communication and documentation of requirements.

Product specifications may be "missing, misinterpreted, misunderstood, or mistaken", all of which may be traced back to fundamental issues of communication and collaboration. These problems may result from a lack (or limited nature) of customer participation in the elicitation of requirements. They may be caused by insufficient access to accurate information by any (or all) contributors. And it's possible that requirements are just phrased ambiguously, in a way that may be easily misinterpreted.

Solution:

Whichever is the case, there are several techniques and tools that successful teams employ to tackle these problems.

First, it is crucial to create a shared, central repository of requirements & development information that may be accessed by all contributors and stakeholders at all times.

Second, successful teams rely on team rituals, recurring events, and mature processes to ensure the flow of information between teams and across company boundaries (e.g. channeling feedback from end users to developers). Working in iterations (via any Agile method) and revisiting fundamental requirements during planning sessions help avoid ambiguity in requirements.

Once your team has figured out what strategies, activities, and processes help them clearly define product specifications, all that information will need to be documented and shared with everyone. Again, using the right Requirements Management tooling helps make sure that adequate and up-to-date requirements information is available to all contributors. That's especially important in the case of frequently changing product specifications, which leads us over to another widespread Requirements Engineering issue.

Problem #2

Inadequate Change Management

Requirements change. That's a fact of life. But tracking and following up on all those changes is still quite a challenge for most teams, even if they're taking an iterative and incremental Agile approach. Add to that the widespread issue of delayed input from customers, and what you get is a wasteful process that can easily result in slow and costly product development.

Tracking and documenting changes, and making sure those changes are propagated across the system and the organization can be problematic. You'll need to make sure all stakeholders are notified of changing requirements, and that those changes are understood by all contributors. In addition to that, you'll also need to monitor all the dependencies and manage the impact of those changes on all affected items.

Solution:

Quite naturally, using the right processes is a good first step. For instance, inviting a cross-functional team to regular customer feedback sessions helps make sure that all affected stakeholders can adequately interpret changes in product specifications.

Religiously carrying out impact analysis for any and all changes enables your team to monitor and manage the chain reaction that updating any requirement can set in motion.

However, there's no getting around the fact that change management is practically impossible without the right tooling. All the structured data in your requirements is immensely difficult to keep track of without a flexible and dynamic repository.

Using an integrated platform to manage requirements and their related items (tools such as Application Lifecycle Management platforms can do that), you can automate dependency tracking and notifications. With out-of-the-box or integrated messaging capabilities, these tools also let you @mention stakeholders to accelerate feedback procedures or automate escalation & review processes.

Problem #3

Imprecise Estimates and Scope Creep

Somewhat related to changing requirements, issues of budget and time estimations plague software projects of all shapes and sizes. When planning projects, assessing costs and delivery times is a difficult task in and of itself – and then you'll have to consider any (unknown) future changes to product requirements.

Inadequate planning can also contribute to technical debt which could necessitate the costly rework of certain product components or features. Similarly, weak requirements prioritization can lead to customer dissatisfaction and budget overruns or delays in delivery. Another related issue is scope creep (project changes spiraling out of hand) which is a direct result of inadequate project definition and the insufficient documentation of goals.

Solution:

Project planning & estimation is a tough nut to crack that requires experience and the careful consideration of multiple factors.

Successful teams aim to make data-based decisions to improve planning accuracy. Using a lifecycle management platform that integrates Requirements Management with all the other crucial processes of software development all the way down to product release opens up a goldmine of development data. You can use all those historical insights to enhance the accuracy of your project estimates.

Modern Application Lifecycle Management platforms also let you use baselines which help you make sure changes are fully controlled. Essentially, a standardized requirements baseline will only contain the latest versions of requirements that are approved by relevant stakeholders. This helps avoid scope creep by eliminating the dangers of constantly changing requirements.

Conclusion:

Future-proofing Requirements Management

Most software teams are familiar with the fundamental Requirements Management issues that can cause projects to derail.

Juggling the triple constraint of time, scope, and costs is immensely difficult in a fast-moving environment with constantly changing specifications. High-performing teams understand that making the right tooling decisions can enhance their RM activities going forward.

Investing in and adopting modern Requirements Engineering tools is a strategic business decision that not only helps the product team, but through streamlining innovation, could positively affect the profitability of the company as a whole.

eBook

10 Success Factors of Future-proof Requirements Management

[Read it](#)

eBook

Making the Case for ALM: Simple Tool Update or Strategic Decision?

[Read it](#)

eBook

A Guide to Calculating the ROI of Application Lifecycle Management Tools

[Read it](#)



DIGITAL TRANSFORMS PHYSICAL

121 Seaport Blvd, Boston, MA 02210 : [ptc.com](https://www.ptc.com)

© 2023, PTC Inc. All rights reserved. Information described herein is furnished for informational use only, is subject to change without notice, and should not be taken as a guarantee, commitment, condition or offer by PTC. PTC, the PTC logo, and all other PTC product names and logos are trademarks or registered trademarks of PTC and/or its subsidiaries in the United States and other countries. All other product or company names are property of their respective owners.

006-requirements-management-challenges-&-solutions-02-08