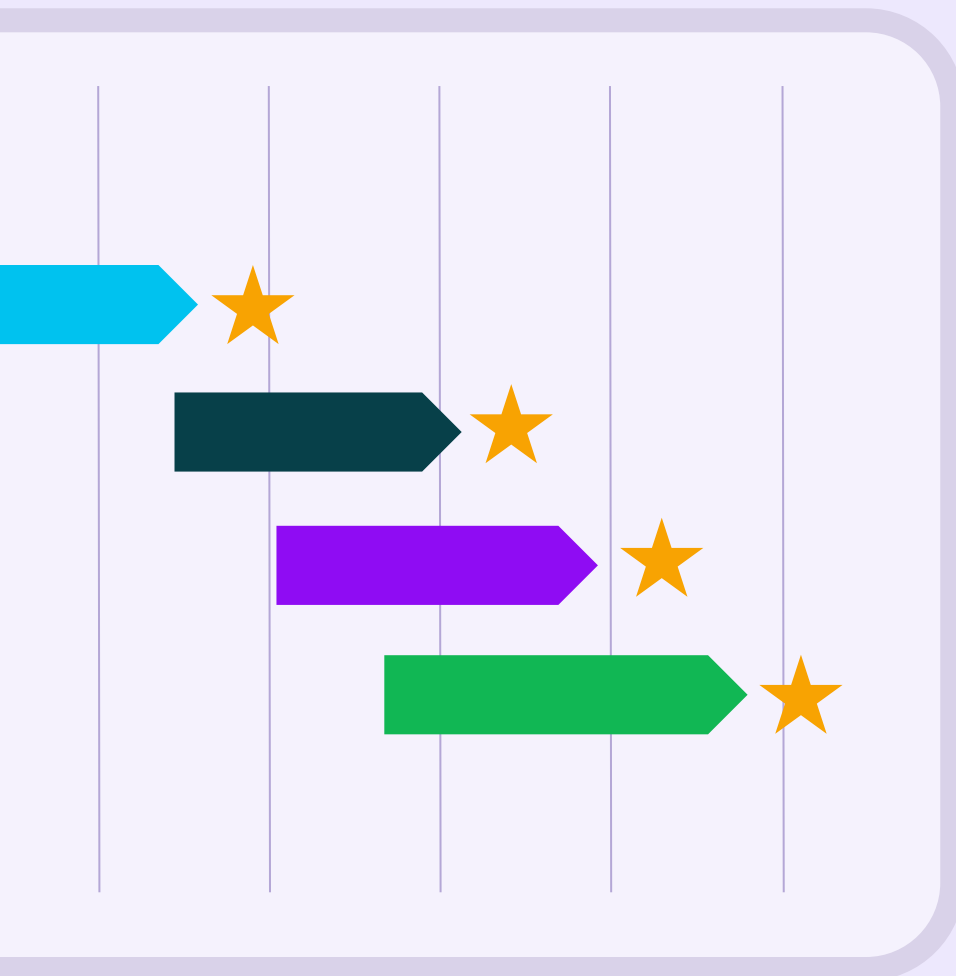




OMS PROJECTS

3 key ways to reduce risk and ensure success

Steps you can take to get real value from your Order Management System (OMS) project. Fast.

What's your most
important
strategic driver?

Increase Sales
or Reduce Costs?



Enterprise software projects have a bad reputation. Scope creep. Budget overruns. Too many project dependencies. Delays. They can take a long time to provide net new value. And when a return on investment takes too long, executives get nervous. The result?

Canceled projects. Or worse, a failed project can be a career killer. Luckily, there are some tried and true ways to reduce risk and avoid them. So let's look at the top three reasons Order Management System (OMS) projects get delayed, and steps you can take now to ensure future project success.

1. Get alignment across different business units early

Just like an ERP, an OMS touches many departments. Which means there are lots of stakeholders. Each with their own goals, Key Performance Indicators (KPIs), and strategic objectives. Unfortunately, when you have a large stakeholder group, it can result in conflicting priorities. Let's take a closer look at three common examples in OMS projects.

Digital / Ecommerce vs Supply Chain

Here you're talking about two teams with fundamentally different objectives.

The primary goal of a **digital / ecommerce front end team** is to **increase sales**. They want to **increase conversions** and **increase Average Order Value (AOV)**. So their OMS project priorities will include how a new solution will help them:

- **Extend the range** of products offered online, for example by exposing more stock across all locations online, and offering 'online only' items which may be shipped from third party Drop Ship Vendors (DSVs) or suppliers to increase both orders and AOV.
- **Enhance the customer promise**, for example, with more accurate Estimated Delivery Dates (EDDs) or faster pickup options, to build trust and increase conversions.
- **Rollout new fulfillment offerings** quickly—for example, offering preorders or backorders—also designed to increase sales and conversions.

Front end teams really don't care where the stock is coming from. They just want to increase availability so they can drive



growth. This can put them at odds with their efficiency focused supply chain colleagues.

The primary goal of the **supply chain backend team** is to **increase margins and reduce costs**. This group is responsible for inventory management, ERPs, Warehouse Management Systems (WMSs) and Transportation Management Systems (TMSs), and overall fulfillment operations. Efficiency is key. So they care about how an OMS can help them:

- **Optimize sourcing** logic so they can:
 - Increase inventory turns and hit sell through targets both in-stores and online
 - Increase their On-Time in Full (OTIF) rate—the rate at which orders are delivered on-time, in full, on the first attempt
 - Reduce delivery costs, and
- **Increase operational efficiency** which includes store fulfillment operations.

How do you make sure these competing priorities don't delay your project?

When it comes to prioritizing which OMS use cases to implement first, you need to get executive alignment on **which objective is most strategically important for the business**. Should you focus on cost cutting (back end) first? Or revenue growth (front end)? Then ensure you have a strong project leader who is empowered to apply this guiding principle for all project decisions, and push back when necessary. Not only will it help you limit initial scope, but also deliver real business value faster.

Customer Service teams feel ignored

One of the most overlooked teams on an OMS project is Customer Service. Why? Because they're typically not part of the digital / ecommerce front end team and they're not part of the supply chain or store operations teams involved in last mile logistics. Yet the decisions and initiatives of both of these teams can have a huge impact on customer service operations. When this team is brought into either the OMS buying cycle or project discovery late, it can derail the implementation process. And ultimately cause missed deadlines. Why?



Because an OMS project can have a big impact on key metrics like **average call resolution time** and **Where Is My Order (WISMO)** call rate. For example, if they're currently getting, say 3.2 WISMO inquiries per order, how can an order management system help them get it down to less than one. What will they want to understand?

- **What does this mean for day-to-day operations and the employee experience?**
 - Will staff have to go to an additional system to answer customer questions? Or can the OMS be integrated (using a headless approach) with their existing customer service solutions to reduce the need for 'tool hopping'? The latter will help decrease call resolution time, so it's important you're clear on an approach early.
 - Will they be able to see more order status details so they can provide customers with a higher level of service?
 - Can the OMS trigger more frequent and granular customer notifications to reduce WISMO calls?
 - Can the OMS provide more order status details to the commerce platform to reduce WISMO calls?



The irony is, often customer service teams can obtain some of the greatest benefits from an OMS project. It can **reduce call volume** and help them service customers faster so they can **reduce call resolution time**. Given the cost of call center labor, this can have a significant impact on the overall project ROI. But to get this benefit as quickly as possible it's important the customer service team's needs are addressed early in the process to avoid later delays.

IT vs Business

This one comes up a lot. Especially when the business owns all or most of the budget. Often it results in bloated scope, suboptimal architectural decisions. And a solution that's more difficult to implement and expensive to support. How does it happen? There are two key reasons:

- **The business expects all 'out-of-the-box' front end features to work from DAY ONE**

Makes sense right? If you can see functionality in a system, surely it must 'just work' from day one? The challenge is, for every front end feature in an OMS, there are many back-end system interactions, integrations and processes that have to be mapped and implemented.

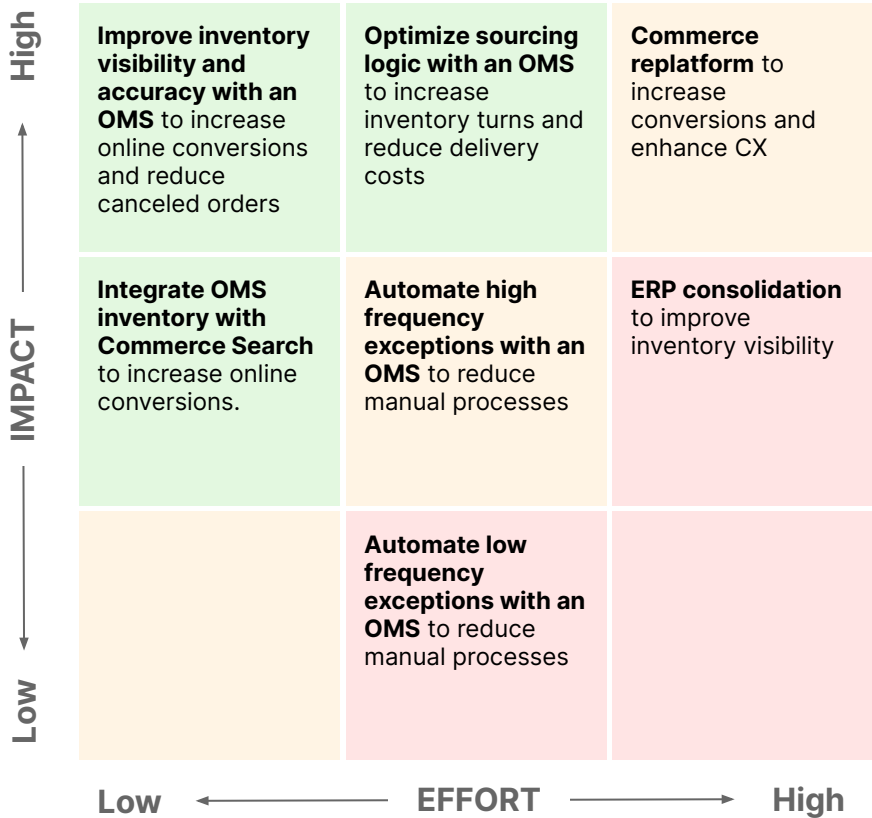


Far more than a simple ‘point solution’. And if you try to deliver them all at once you’ll get stuck in very long project cycles. What’s more, usually during implementation the business finds that those ‘out of the box’ features don’t quite work the way they want. They need to be enhanced. So rollout drags on even longer. And costs increase.

- **Not enough focus on ‘digital agility’**
When business features and functions are prioritized at the expense of IT requirements during an implementation, it can result in longer project timelines, both now and in the future. What’s the value of being able to make a system change quickly in the future? What KPIs are associated with this value? It’s important to factor in digital agility requirements too.

When it comes to prioritizing what’s going to be delivered during a project, it’s essential to be ruthless and KPI driven. What features and functions, whether for the business or IT, are going to have the greatest short-term impact on the business?

To help with prioritization, plot each requirement on an effort vs. impact matrix. That way it will be easy to see where you can get the most value quickly.



IN SCOPE



OUT OF
SCOPE

With modern API driven OMS solutions, you shouldn't have to roll out everything on day one. So start with the processes and capabilities that will deliver value fastest. Then add more over time.

2. Take proactive steps to minimize scope creep

Scope creep is the enemy of every project. It will have the biggest impact on how quickly you can achieve a return on any tech project, but particularly an OMS. Why?

Because an OMS sits at the heart of your inventory availability and fulfillment operations. As a result, it needs to connect to many other systems in order to do its job. This means every use case can involve multiple integrations. So to avoid scope creep it's essential to keep each project phase small. Yet this can be difficult when you have a large stakeholder group. Why?

Because each business unit wants to have its needs met first. Let's take a look at the biggest contributors to scope creep and how you can avoid them.

Scope Creep Contributors

What drives scope creep?

- **Too many use cases in each phase of the project.** To get fast value you need to keep each phase small. There will always be a few 'surprise' requirements in each phase, but if there are too many use cases to begin with, your 'time to value' will be far too long.
- **The business expects that all use cases you support today will be included in phase 1.** This is the biggest contributor to having too many use cases in each project phase.
- **Vague or ambiguous requirements.** When people make assumptions about what is in scope, it can lead to project delays later. This is especially important where there is limited or no documentation of current solutions and processes as 'extra requirements' often come up during user testing which can blow out project timelines.
- **Not enough due diligence during vendor selection.** During vendor demos, it's easy to make assumptions (and not ask deeper questions) about use cases a solution can support. When people see something 'very similar to what they need' they often assume that a few



Let's move that
to Phase 2

small changes to a process or screen should be 'no problem'. This is not always the case. And can have a huge, unanticipated impact on implementation budgets. Many projects have been canceled as a result.

- **Wanting to solve for every exception from day one.** There is a lot of buzz around process automation. And reducing manual labor can have great cost efficiencies. But if you try to solve for every exception in phase one of your project, you'll add a lot of scope for limited return. What's more, often the way people think they want exceptions handled changes once a solution has been rolled out. Which means additional rework in future phases.

So what can you do?

How to avoid scope creep

Here are some strategies to help you keep scope in check.

- **Take a 'Minimum Viable Project' approach.** Only include what is absolutely necessary to deliver initial business value. That way you can demonstrate success early. Keep your CFO from getting nervous. And ensure funding for future phases.
- Use your **top-level strategic objective** (revenue growth or cost cutting) to drive prioritization of what's in scope and out of scope for each phase. Start with your biggest pain point so you get value fast.
- **Attach business KPIs to every requirement** to drive lower-level prioritization of what should be included in each phase.
- **Set clear in scope / out of scope expectations** for each phase of the project with all teams, so everyone is clear and there are no surprises. Then stick with them.
- **Conduct workshops during the vendor selection process** to make sure all your critical processes and requirements can be supported. Pick a few use cases where your requirements are different to the 'standard demo' and ask vendors to show you how they can be supported. Then ask the level of effort required to make those changes. Ironically, this is especially important if the solution offers robust capabilities in a particular area that may be very difficult and expensive, or impossible, to change.
- **Question every requirement.** Empower a strong Business Analyst function. And don't include requirements just because you 'do them today'.



What KPIs are driving this requirement?

The best business analysts are “problem solvers, not order takers.” They will ensure you run efficiently, and are critical in the battle against scope creep.

- **Insist on clear functional requirements** so there is no ambiguity around the details. Misaligned expectations on what will be delivered results in scope creep.
- **Have a single scope owner.** Define clear criteria for how scope decisions will be made. Then empower them to make the final decision on what will be included in each project phase. And also empower them to cut scope if project timelines will be impacted.
- **Start with the ‘happy path’.** You don’t need to automate how every exception is handled from day one. In fact, it’s important not to. Once people start using the new OMS the way they want to process exceptions may change. So don’t automate them too early. Make sure you have a clear escalation process, but be ok with handling exceptions manually at first, then automate them over time. Even better, to prioritize this work, collect metrics on how often each type of exception occurs, and the business impact of each (e.g., manual effort, lost sales, etc.)

Now you have a good plan for avoiding scope creep, let’s look at another contributor to OMS project delays.

3. Rethink IT project dependencies

OMS projects are often part of a larger ‘digital transformation’ initiative. At a high level, digital transformation is very strategic and important. Unfortunately though, when planning the rollout of multiple solutions, the project plan for each one tends to be viewed in isolation.

No one maps the timeline for each phase of each project against dependencies. Which means strategic decisions around the order in which each phase of each project should be deployed, can’t be made. What’s the impact?

You may be ready to roll out phase one of your OMS project, but your ERP or Commerce project isn’t ready. Which delays getting value from your OMS investment. Let’s take a closer look.

What will add the
most value, fast?



What causes ERP Project Delays?

ERPs touch many functions and departments. And they're not replaced very often, which means they accumulate a lot of functionality over time. Often as tribal knowledge. And if you've grown through acquisition, you probably have more than one. So the biggest delays are usually:

- **Complexity of ERP consolidation projects**
Integrating one ERP with another means mapping a lot of historical data. Integrating processes. And managing a lot of stakeholders, not just across functional departments but divisions of an organization as well. As a result these projects often take longer than expected.
- **Legacy tech not well documented**
Often there's a 'spaghetti mess' of custom code, and there is no documentation of the functionality it provides. This leads to scope creep and project delays.

And the issue of legacy tech not being well documented isn't just an ERP issue.

What causes Commerce Project Delays?

Why do commerce projects get delayed? Here are three common reasons.

- **Promotions and Payments** - Complexity around commerce components like promotions/coupons, multiple payment methods (e.g., gift cards, credit cards, Paypal, etc.) and how rounding errors will be handled.
- **Legacy tech not well documented** - Unforeseen requirements and scope creep because the legacy commerce platform wasn't well documented.
- **Composable Commerce Integrations** - When a company is transforming their whole front-end stack to a composable commerce approach, making all the components work together—Commerce platform, Content Management System (CMS), Marketing automation platform, Product Information Management (PIM), commerce search, etc. —can take longer than expected.

But there's actually an opportunity here. You can use an OMS to mitigate this risk. How?

**With an OMS you
can extend the
life of your
existing systems**



Take an OMS-first approach to digital transformation

This may seem counter intuitive. Why would you start with a solution that sits in the middle of your front end and back-end stack? Because it can extend the life, and let you extract more value from, your existing systems. Let's look at some examples:

- **Delay the need for ERP consolidation** – You can use your OMS to simplify the order funnel and use rules in the order workflow to route orders to the appropriate ERP. This approach can deliver a unified experience in months rather than the years it could take to consolidate several ERPs. And it can sit on top of your existing infrastructure, to reduce risk.

What's more, an OMS will let you fine-tune your sourcing logic to reduce delivery costs and increase inventory turns. So the supply chain team can improve their KPIs earlier.

- **Increase commerce conversions before you roll out a new commerce platform** – Using your OMS to show customers an accurate view of inventory across all digital channels will take much less time than a Commerce replatform. It will also let you show them more accurate delivery and pickup estimates as well. Not only on the Product Details page and during checkout, but across search as well.

And it can respond to those inventory availability requests at scale. So it's a faster way to provide customers with a better experience, and let the digital/commerce team hit their increased conversions and sales KPIs, earlier in the digital transformation.

So rather than lock yourself into a multi-year upgrade cycle without getting any incremental business value, take an OMS first approach. Use your OMS to extend the life of your existing systems, and mitigate project delays.

Nice work!

oms nerd.

Summary

While enterprise software projects are prone to scope creep and budget overruns, with an OMS project, it doesn't have to be that way. With the right preparation and structure in place, you can reduce the risk of both timeline and budget overruns, and achieve real business value. Fast. The key is preparation.

So what should you be doing now?

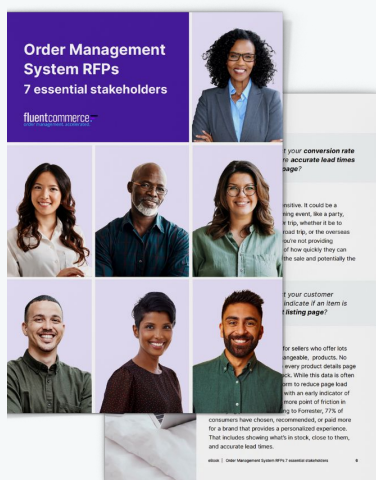
Set clear goals. Get business units and stakeholders aligned. Define internal roles, responsibilities and deliverables. And the key KPIs that will be used to measure success. Start small. Take a phased rollout approach to reduce risk. And don't block yourself from digital agility. Make sure you can deliver incremental value over time. That way, you'll ensure success. Both for your project, and your business.

You might also like...



Order Management System selection: 43 smart questions to ask a vendor

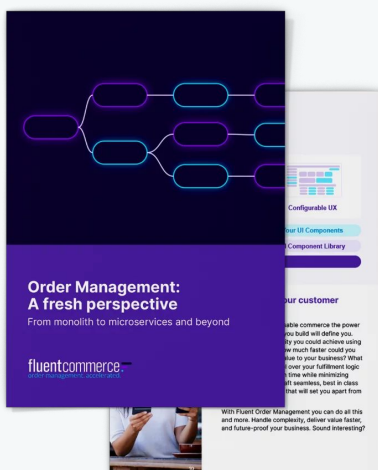
[Read now](#)



Order Management System RFPs

7 essential stakeholders

[Read now](#)



Order Management: A fresh perspective

From monolith to microservices and beyond

[Read now](#)

**Have a question about
phased OMS rollouts?**

**Or how to reduce risk and
costs on your project?**

Contact us today!

