

Software Selection – Warehousing & Transport

Anton.Boonstra@simplesolidsolutions.nl, April 2025

Introduction

This article is the third in a series of four on *Warehousing & Transport*. In this part, we focus on selecting the right software to realize your **To-Be** situation. The fourth article will explain how to run a tender process to select the best provider for physical warehousing and transport services.

Software selection always starts with process mapping: both **As-Is** and **To-Be**. Once this is done, you compile a list of requirements and send it to your longlist of potential suppliers. From there, you select 3–4 candidates and invite them to give a focused demo. Eventually, you express a preference for one supplier with whom you'll proceed. Below, we elaborate on this selection process.

The Requirements List

The requirements list is typically a structured Excel sheet, organized in a tabular format. It includes both *requirements* and *wishes*, clustered by topic. The first part contains general information you want from the supplier:

- What kind of products do they offer?
- How large is the company?
- In which countries do they operate?
- And other such qualifying details.

The core of the list, however, is about functional and technical requirements. These are directly derived from your **To-Be**. My preferred approach is to walk through the To-Be, paragraph by paragraph, and immediately write down the associated requirements—ideally grouped by (sub)topic. It's best to do this directly in a table format, which allows you to sort and structure as needed.

As you read each section of the To-Be:

- Write down the requirements that come to mind.
- After each paragraph, check whether you've overlooked anything.
- Once the full document is reviewed, let it rest, then revisit it later with fresh eyes to refine, add, or remove items.
- Afterwards, review the list with stakeholders or colleagues for feedback.

If not already present in your Excel sheet, include a reference to the corresponding To-Be paragraph—very helpful for suppliers seeking context.

The To-Be document should also be shared with longlist suppliers as an introduction, helping them understand how and why each requirement was derived.

In the requirements list, I always use a separate column to classify each item as one of the following:

- **Requirement** – A must-have. Without it, you cannot work efficiently or effectively.
- **Wish** – A nice-to-have. If missing, it may be inconvenient but not a showstopper. A supplier might offer an alternative solution that meets the intent.
- **Knock-Out (KO)** – A strict exclusion criterion. If the supplier doesn't meet this, they're out of the selection process. But be careful with using KO criteria—you might exclude suppliers based on a misunderstanding. Often, I'll mark something internally as a KO, but not communicate it as such to the supplier.

Avoid phrasing too many questions in a closed, yes/no format. This often gives you little meaningful insight, especially if the supplier misunderstood the question or answers strategically. Some suppliers might claim their software can “do everything.” Give them room to provide thoughtful, genuine answers from someone who understands your business and challenges.

In the introductory letter to suppliers, you may want to limit the number of words per response and emphasize that answers should be specific, not copied from brochures. You want clarity and honesty—not marketing fluff.

Personally, I'm not a big fan of hard KOs. They restrict both the supplier and yourself too much. If you do use them, the questions must be carefully worded. A simple “no” ends the conversation, even if the supplier might've had a viable workaround. On the other hand, if you never define KOs, you risk accepting too many compromises. My solution: note internally which items are “critical,” but keep communication flexible.

Also, be deliberate in how many requirements you include. Once you've defined your needs, don't spend too long perfecting the phrasing. Keep it clear and simple—straight from the heart often works best. It's faster and more relatable.

That said, don't try to draft your requirements list in a single afternoon. The process needs careful thought. A strong concept gives you clarity during the selection process, helps you stay critical under pressure, and ensures consistency when others challenge your approach. It also provides a solid foundation for the implementation phase later on.

From my experience, you should plan for at least **three months** to think through the concept, speak with stakeholders, and fine-tune your story. Take your time—it's worth it. After all, if you choose the right software, it could serve you for the next **10 to 15 years**.

The Longlist

How do you build a longlist?

Back in the 1980s and 1990s, when I worked for a consulting firm and was involved in many selection processes, we relied on handbooks listing packages and suppliers. Today, the most effective ways to create a longlist are:

- **Gartner Magic Quadrants**
- **Trade fairs**

- **Good old Google**

Gartner publishes well-known Magic Quadrants covering various software markets. These charts show which solutions are market leaders, which are still developing, and all the nuances in between. They're an excellent starting point to determine which suppliers you should definitely explore further.

Trade fairs are fairly self-explanatory. Suppliers are eager to showcase their solutions. Take it all in and talk to as many people as possible—there's no substitute for real-life impressions.

With **Google**, a few well-chosen keywords will yield a list of relevant solutions. Then, visit supplier websites to understand what they offer—and, equally important, what they don't. Check out their reference customers, the sectors they serve, and their scope. I often supplement this by calling around and asking for input. Initially, the responses may be scattered, but eventually, you'll begin hearing the same names and themes—this is a good sign that you're zeroing in on the right candidates.

I usually work on the longlist in **parallel** with the requirements list. Doing both together helps align the suppliers you're considering with the language and logic you're building into your requirements. You'll often notice recurring terminology among suppliers—if it's relevant to your business, incorporate it into your requirements.

A longlist of around **ten suppliers** is quite typical. A few more or fewer is fine. Keep in mind that if a supplier takes you seriously, they will invest serious time in responding. Larger vendors often have dedicated staff for handling RFPs. It's good to remember that answering these questions can be a world apart from actual software implementation.

Eventually, you'll receive completed questionnaires, which you'll compare. Here's how I typically handle this:

- Assign **weights** in advance to key requirements.
- Separate **requirements** from **wishes** in your evaluation.
- Hold a **workshop** to review the responses with your team.

This workshop is critical. You'll gather both hard and soft feedback—write it down carefully and reflect on what it means. This is why it's essential to determine the weighting beforehand. If not, you risk being influenced too much by the charm or style of a particular supplier's answer.

As a benchmark, I often use a **fit percentage** of 85%. That is, the solution should cover about **85%** of your requirements and wishes **out of the box**—with no major customizations. The remaining 15% may require system adaptation. Before starting an implementation, make sure you understand the **costs** and **implications** of this.

The Shortlist

Once you've narrowed down to a shortlist, you're left with a maximum of **four** candidates—preferably three. Sometimes you might include a fourth if two suppliers score equally and you

need to explore them both further. Ideally, the final three are distinct and scored highly in your evaluation.

A shortlist process takes time—a **lot of time**. If done properly, you organize a full-day **demo** with each supplier. These days are demanding, both in terms of attention and energy. You're expected to dig deep into the supplier's solution and assess how it matches your needs.

Preparation is key. Bring a **case study**—something you know inside and out. Ask the supplier to demonstrate how their system handles it. This puts you in control of the discussion and ensures they don't just show you the shiny, polished parts of their software while avoiding the weak spots.

By steering the demo, you'll see what you want to see—not just what they want to show. Throughout, keep a **scoring sheet** based on your requirements list and the supplier's previous responses. At the end of the day, you'll have a clear picture of how a realistic “happy flow” (with a few exceptions) looks in that specific system.

From the 3–4 candidates, you'll select a **preferred supplier**, and possibly a **runner-up**. The second-ranked supplier is only considered if negotiations with the first one fail. However, it's still smart to keep them in the picture—this strengthens your position during negotiations. To be fair, I've rarely seen the runner-up take over, but you never know. In my view, the runner-up should receive a small fee for being kept on standby.

How do you make the final decision?

Before choosing, hold a wrap-up meeting with everyone who attended the demo sessions. Write the key decision criteria on a flip chart or whiteboard so they're visible to all. Assign weights to each criterion and ask every participant to give a score for each supplier.

Use this session to foster discussion, balance perspectives, and aim for consensus. Make sure to **document the outcome** thoroughly—this provides transparency and a solid rationale for your decision.

Contract Negotiations

Once you've made your choice, **don't announce it immediately**. Ideally, continue negotiations with two preferred candidates. This not only gives you room to negotiate but also keeps both parties engaged and motivated to win the contract. If you reveal your selection too early, the supplier might assume they've won—and relax their efforts.

If possible, play the two suppliers **strategically against each other**, but always in a respectful and transparent way. For example, if Supplier A claims they can deliver a certain functionality, you can check with Supplier B whether they offer the same.

Of course, at some point you'll have to make a final decision. I recommend doing so only once you have a **clear understanding of the implementation costs**. That is, you need insight into what system modifications will mean in terms of both **time** and **money**. Once you enter the modification phase, the **contract must be finalized**. You cannot—and should not—have

two suppliers simultaneously working on system changes. That would be unethical and impractical.

Planning with the Preferred Supplier

Once a preferred supplier is chosen, the next step is to determine what **customizations** are needed. These should be **clearly documented** by the supplier.

Your goal is to get a reliable estimate of:

- The **costs** of modifications
- The **duration** of the implementation

These must be reflected in the contract. Don't forget to include **penalties** for unfulfilled requirements. A complete contract should include:

- What is available **by default**
 - What requires **customization**, along with costs and lead time
 - A detailed **implementation plan**
 - **Conditions and responsibilities** related to the implementation
-

What Should Be in the Contract?

In this section, let's focus purely on **software-specific** contractual elements—not general legal terms. You're about to start the implementation, and if issues arise (as they often do), you need a strong agreement to fall back on. Here's what to include:

• Contract duration and pricing

Define when the contract can be **terminated**, and how long pricing remains valid. A 2–3 year term is common, covering the implementation phase. After that, a new contract is often negotiated. Be aware that due to customizations, you'll likely remain dependent on the same supplier, so pricing stability during this period is important (apart from indexing).

• Customization agreements

Be **very specific** about who delivers what, and when. Reference **functional design documents** where possible. Implementation partners often operate like “time factories”—they bill hours but aren't necessarily committed to delivering results unless clearly agreed. Ideally, you'd have a **results-based agreement**, but that's hard to define. The best approach is to ensure customizations are **well-documented** before implementation begins.

Functional descriptions should include:

- Clear goals
- Screenshots

- Detailed process descriptions
- A well-thought-out **test scenario**
Tests often clarify the intended functionality better than any design document. Take time to write them out—it's worth the effort!

• Copyright

It's common for software suppliers to retain the copyright on customized developments, even if those were based on your ideas and processes. Personally, I find that questionable. You bring business-specific expertise to the table—why shouldn't you share in the value created? Consider negotiating a **royalty clause**—a percentage of proceeds if the customization is resold.

• Updates and release policy

General updates are usually covered under standard maintenance, but **custom updates** may not be. Clarify how custom developments are handled in future releases. Will they be included? Maintained?

• Maintenance

Maintenance fees typically range from **15% to 20%** of the (undiscounted) software price. But what do you actually get?

Clarify whether this includes:

- Access to updates
 - Participation in key-user groups
 - Release planning info
 - Helpdesk support
- Understand exactly what's covered and how it benefits your organization.

Conclusion

This is the **third article** in a series of four on *Warehousing & Transport*. It focuses on **software selection** as a crucial step toward realizing the **To-Be** process model developed in the first two articles.

Key takeaways from this article:

- Start by mapping **As-Is and To-Be** processes.
- Derive a clear and structured **requirements list**.
- Share this list with a longlist of potential suppliers.
- Evaluate written responses and reduce the list to a **shortlist**.
- Organize demo days based on your own case study.
- Choose a **preferred supplier**, and negotiate the contract **only after** implementation costs are known.
- Document customizations and expectations thoroughly in the **contract**.

Best of luck with your selection and implementation process!

-/-/