



# Implementation Guidelines

---

**We are happy you've decided to implement MRPeasy, a powerful MRP/ERP system for your manufacturing or distribution operation! This guide will discuss best practices for ensuring a successful implementation.**

---

**There are two main ways to implement production software.**

**1.**

The classic "all-in" approach involves allocating considerable resources and staffing from day one to roll out maximum functionality in minimum time. This usually entails intense project management and parallel staff training sessions.

**2.**

The other option is a phased implementation. This involves rolling out essential functionality first and adding more advanced functionalities as your use case expands and staff have become accustomed to using the software.

MRPeasy can be successfully implemented using either method. Most clients are functional within a few weeks to a few months after committing. We recommend a phased implementation for everyone except seasoned professionals with extensive experience implementing production software.

## Implementation resources

MRPeasy is a self-service ERP specifically designed for easy implementation. Our [Resources](#) page offers many demo videos, how-tos, and a complete user manual.

In addition, [MRPeasy.com](https://www.mrpeasy.com) is equipped with an AI chatbot, Mr.Peasy, which can be a great additional resource in your implementation journey.

We also have a skilled Customer Success team and a highly praised Support team.

If you wish to opt for an assisted implementation, MRPeasy also has several [Authorized Consultants](#).

# The foundations for success

Three key aspects lay the foundation for a successful implementation:

## 01

### **Taking complete ownership of the implementation process :**

Only the people appointed to manage the project can be responsible for the implementation process and its results. It would be best if you had an internal implementation team that decides the best way to set up operations and execute the implementation. This applies even if you hire an external consultant.

## 02

### **Prioritizing the steps of the implementation :**

Even if you decide to implement advanced functionality early, prioritize resources to the most impactful requirements first. Avoid “nice-to-haves,” “shiny add-ons,” and “this could be useful down the line,” at least until the core system is up and running.

## 03

### **Understanding the reasons and goals for the implementation.**

The goals of the implementation need to be clear beforehand.

- What are the primary reasons for implementing?
- What key issues are we facing that the software can solve?
- What measurable benefits (usually KPIs) can the software bring?

**Let's look at some examples:**

### **a) What are the primary reasons for implementing?**

- Accurate operation and material planning
- Easier quoting
- Better communication and visibility
- Real-time and accurate inventory overview
- Reduction of manual or duplicate data entry
- Automated stock balance calculation and product costing
- Meeting regulatory requirements
- Achieving quality

## b) What key issues are we facing that the software can solve?

- Difficult and laborious to get information
- Difficult to estimate lead times and costs for making quotations
- Difficult to schedule production and load machines
- Ineffective or slow communication
- Reoccurring mistakes and human error
- Low stock visibility, frequent stockouts, or excess inventory
- Difficult regulatory requirements for tracability
- Inventory valuation incorrect or missing

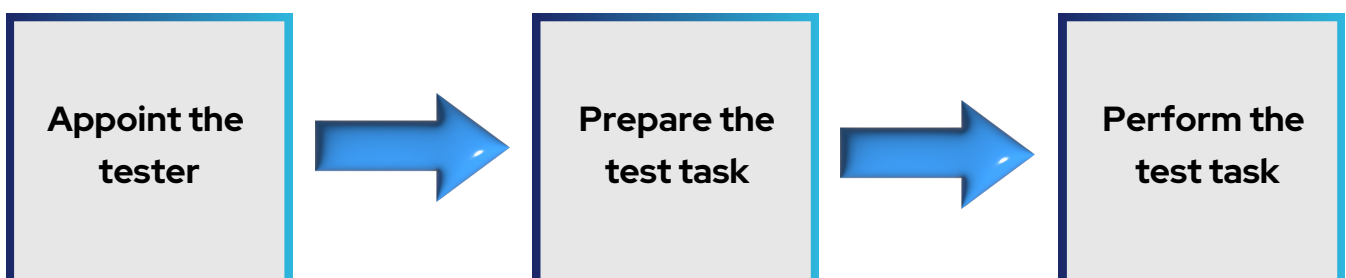
## c) What are the Key Performance Indicators (KPIs) to measure?

- On-time delivery
- Customer satisfaction
- Lead time reduction
- Loading/Effectiveness increases
- Decreases in overheads
- Smaller inventory
- DIFOTIS

# Project outline

Once you're clear about your requirements for the software, we recommend dividing the implementation project into 2 phases – acceptance testing and implementation. In acceptance testing, you will perform tests on the software to decide whether to commit to implementation.

## I. Acceptance testing



## II. Implementation



# Acceptance testing

## 1 Appointing the tester

Your first task is to assign a project lead to the testing process. Depending on the scope of your operations, this can be an employee, a team of employees, or a dedicated software analyst.

The lead tester or team should have a clear understanding of your current business processes and at least some relevant experience with production software.

## 2 Preparing the test task

The tester should then outline your main usage scenarios and requirements for the software. Here's a suggested workflow for how to go about this:

### 1. Describe and map the current processes of the production operation by importance.

For example: When an order comes in, the production manager emails the stock clerk with the required items and awaits their reply about stock levels and required purchases.

### 2. Map and prioritize the functionalities of the software and how they can improve the above processes.

For example: We want the software to automate material requirements calculations and provide the production manager a real-time view of material availability when orders come in. This is our top priority.

### 3. Compile a simplified testing dataset or datasets.

The dataset should be as simple as possible to test the requirements. You can also have different datasets to test different functionalities.

### 4. List use cases and scenarios to test, including expected outcomes.

We recommend starting with core functionalities to assure essential compatibility.

## 3 Performing the test task

Next, import the test data into the system and run the required tests within the software. Analyze the results to gain clear answers to the following questions:

1. Did the test(s) achieve the required outcome(s) ?
2. Is the software easy enough to use to scale to production ?
3. Is the software easy enough to implement and commit to ?
4. What are the measurable benefits of implementation, if any ?

For more specific information on testing the software, check out our Testing Manual on the [Resources](#) page.

# Implementation

If the acceptance testing was successful and you decide to commit to implementation, here is a simplified list for getting started.

## 1. Appoint the implementation team

The implementation process should be led by a dedicated project manager (and assistants, if applicable) who should be granted sufficient time and resources to run it effectively.

The implementation team should include or at least involve the heads of all relevant departments that will be using the system, as well as general management and company board representatives.

**We strongly recommend not assigning the project manager role to an employee with large existing responsibilities in the company.** Setting up production software is a systematic and high-priority process that lays the groundwork for the whole company's production effort. The project lead should be able to dedicate themselves to the process and not have to juggle other responsibilities until core functionality is set up.

## 2. Prepare the implementation plan.

The implementation plan sets out the following:

- Strategy for implementing the software.
- Deadline for the planned roll-out.
- Timeframe of each implementation stage.
- Detailed description of each stage.
- Necessary preparatory activities.
- List of appointed roles along with their assigned tasks.

If the acceptance testing was successful and you decide to commit to implementation, here is a simplified list for getting started.

**Figure 1. Activity list example.**

Activity number	Activity description	Activity description	Appointed to	Deadline	Appointees consent
1	Analysis of requirements	Overview of the core issues that need solving	<b>Person</b>	Date 1	<b>Signature</b>
2	Preparation of test data	Test dataset and clear scenarios	<b>Person</b>	Date 2	<b>Signature</b>
3	Testing	Clear overview of software capabilities	<b>Person</b>	Date 3	<b>Signature</b>
4	Implementation planning	Detailed implementation plan	<b>Person</b>	Date 4	<b>Signature</b>
5	Data preparation	Seed data prepared	<b>Person</b>	Date 5	<b>Signature</b>
6	Implementing critical functions	Limited functionality or departments implemented	<b>Person</b>	Date 6	<b>Signature</b>
7	Implementing secondary functions	All functions or departments implemented	<b>Person</b>	Date 7	<b>Signature</b>
8	Developing integrations and customizations	Integrations to accounting software, e-commerce, etc.	<b>Person</b>	Date 8	<b>Signature</b>
9	Performance review	Overview of realized benefits, project closed	<b>Person</b>	Date 9	<b>Signature</b>

### 3. Fulfill the implementation plan.

Once the plan is developed, the next step is to set the implementation plan in motion. To prevent the process from dwindling, we recommend not leaving a time gap.

Fulfilling the implementation process involves intensive project management. For example :

- **Daily tasks** like individual or one-on-one work to complete certain tasks.
- **Weekly tasks** like inter-departmental and managerial meetings to assess progress and remove obstacles;
- **Monthly tasks** like performance reviews or all-hands meetings to keep everyone involved and informed.

As every company workflow is different, there is no “one right way” to implement production software. However, we strongly recommend acquainting yourself with the following implementation tips and recommendations.

## Implementation best practices

Here’s a comprehensive list of implementation best practices based on years of cooperation with manufacturing experts, the development of MRPeasy, and communication with clients.

### Company size and requirements matter

Not every company is ready to implement an ERP/MRP system. This can be especially true for micro-companies, which may lack the resources or competence to install and operate one.

It’s also important to weigh whether software is needed at all. For example, tiny distributors with simple inventories might end up with an insufficient return on investment for the software’s implementation cost and subscription fee.

Still, modern SaaS (Software as a Service) manufacturing software can be very flexible in accommodating different use cases. MRPeasy’s flexible pricing and tiered functionality plans make it a viable option for many micro-manufacturers.

### Avoid micro-tasking and prioritize tasks

If set up right, an ERP/MRP system can solve major issues associated with manufacturing. Avoid wasting time and resources on fine-tuning tasks that look nice but have a minor impact. Focusing on micro-automation can often lead to increased and unnecessary complexity within the implementation project.

Instead, set a few major implementation goals closely related to your business processes and the problems you want to solve. Implementation goals should serve feasible business requirements, not try to achieve higher levels of automation at any cost.

### Implement step-by-step

Even if you have opted for a comprehensive implementation with advanced functionalities, we strongly recommend implementing the system one step at a time. Attempting to implement different functionalities simultaneously will often introduce redundancies and confusion later.

Prioritize the functionalities according to your requirements and systematically bring core systems online. Depending on your company's size and structure, you can implement the system iteratively by module, functionality, production line, or personnel.

### Data classification

Thoroughly think through data classification and categorization. Changing the data classification after you go live is nearly impossible, so plan accordingly.

Name and code the stock items logically and intuitively and create logical product groups. Also, structure or group your workstations and designate the bills of materials and routings.

### Keep testing and implementation separate

Can you make it foolproof to distinguish between using test data and actual data? Using actual data during testing can complicate the process and often lengthens the testing phase.

Here are some tips for testing:

- Data should be simplified for initial functionality testing.
- More data points can lengthen the testing phase exponentially

- Different functionalities should be tested both separately and together
- Implementation should start from an empty sheet – if actual data is mixed with test data, cleaning the database might prove problematic later.

### Customization is not the only way

Finding a standard ERP/MRP solution that suits all your needs and perfectly aligns with existing processes is practically impossible. In most cases, you must either adapt or customize. Customization might seem the obvious solution, but it's not always the best option.

Adapting your business processes and documentation to a sound MRP system can often be more effective and deliver better results in the long run. MRP systems have been specifically developed to provide simple and effective solutions to workflows and common hurdles. After all, many of your existing workflows might result from a lack of automation in resource planning, which an MRP system is designed to provide

### Users need training

Every person interacting with the software needs clear training to understand how to use it. They should also know their role in using the software and what everyone else does. It's a best practice for initially trained users to pass their knowledge on to others. Another option would be software vendor training.

We suggest approaching employee training systematically and developing a training plan that outlines employee roles in using the new system and the software's overall benefits to the company

### Keep future users informed about the goals and project schedule

There will often be at least some pushback from employees when changing company procedures. Experience shows that employees are more likely to accept procedural change when they understand its reason and benefits.

Ensure that all users interacting with the new system understand the implementation goals and schedule. This means holding a general meeting at the beginning of the implementation process and following up with periodic communication.

## Prior experience with ERP/MRP systems helps a lot

The implementation project team leader should have prior experience using and/or implementing ERP/MRP systems in your industry. If your company doesn't employ such a person, finding a partner consultant with this expertise is a good idea.

## Don't rush – test, and test again

Test intensively before using an ERP/MRP system in production mode. It is much easier to fix errors and change procedures during testing than when you're operating with actual data and customer satisfaction is on the line.

Prolonging the launch by a few weeks or even months is often a more cost-effective way to smooth out rough edges than running a brand-new system and addressing fundamental issues simultaneously. Remember, even in the best cases, the first weeks of using the new system will be a time of change and adaptation.

## Keep the old system working

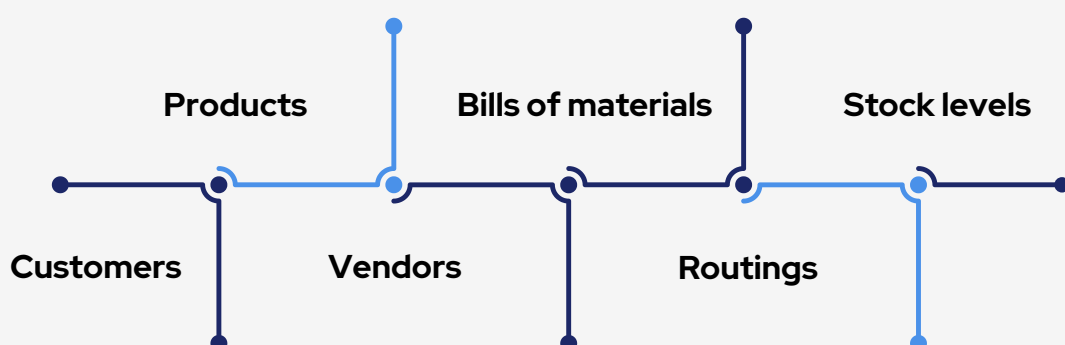
We recommend using the old system in parallel with the new one for at least 1-2 months after implementing the new software, even though it adds workload. This helps ensure that the new solution has been configured and used properly; for example, the reports in the old and new systems are identical.

Parallel use is also a good way to compound the new system's benefits for employees and management. And in case of any major issues with the new system, you can roll back to the old one.

## Only new orders in the new system

When migrating the old system, we strongly recommend keeping the new system clean of old orders. Instead, finalize old orders in your old system and set a deadline for processing the last order.

### Migrate these into your new system :



## Implementation insights from MRPeasy clients

### Among our client feedback, a few recurring recommendations stand out:

- While implementing MRP software may feel overwhelming at first, the effort is almost always worth the initial time and resources you put into setting up. A functional MRP system's level of automation shaves minutes from every hour of production operations.
- Always get a clear idea of your requirements for the software and particular use cases, and prioritize these. Bringing a few core functionalities online can already hugely benefit the company. More advanced functionalities are ready for when it's time to scale operations.
- The implementation process can go smoother if you learn the basics of MRP beforehand. This can be achieved through consulting with experts, taking an online course, or requesting a product demo.
- Take extra time to set up critical data like part numbers, item descriptions, vendors, etc., before going live. Importing large amounts of data later is much more work.
- If it's clear that your company would directly benefit from a sound MRP system, commit to implementing it sooner rather than later. The more legacy systems and workarounds you have, the more complicated the required change management will likely be.
- Communicate the new system's benefits to employees and consider developing an internal training plan. First, bring supervisors up to speed, then introduce the changes to the shop floor.

**We wish you the best of luck with implementing MRPeasy !**

Remember, a detailed user manual is always available within the app or online at <https://www.mrpeasy.com/resources/>

With questions, you can always turn to our Support team at <https://app.mrpeasy.com/support>

If you decide to involve external assistance, check out our authorized consultants at <https://www.mrpeasy.com/consultants/consultant/>