Enterprise resource planning (ERP)

**Enterprise resource planning** (**ERP**) is an enterprise-wide information system designed to coordinate all the resources, information, and activities needed to complete business processes such as order fulfillment or billing.

An ERP system supports most of the business system that maintains in a single database the data needed for a variety of business functions such as Manufacturing, Supply Chain Management, Financials, Projects, Human Resources and Customer Relationship Management.

An ERP system is based on a common database and a modular software design. The common database can allow every department of a business to store and retrieve information in real-time. The information should be reliable, accessible, and easily shared. The modular software design should mean a business can select the modules they need, mix and match modules from different vendors, and add new modules of their own to improve business performance.

Ideally, the data for the various business functions are integrated. In practice the ERP system may comprise a set of discrete applications, each maintaining a discrete data store within one physical database.

In order for a software system to be considered ERP, it must provide an organization with functionality for two or more systems. While some ERP packages exist that only cover two functions for an organization (QuickBooks: Payroll & Accounting), most ERP systems cover several functions.

## The Ideal ERP System

An ideal ERP system is when a single database is utilized and contains all data for various software modules. These software modules can include:

**Manufacturing:** Some of the functions include; engineering, capacity, workflow management, quality control, bills of material, manufacturing process, etc.

**Financials:** Accounts payable, accounts receivable, fixed assets, general ledger and cash management, etc.

**Human Resources:** Benefits, training, payroll, time and attendance, etc

**Supply Chain Management:** Inventory, supply chain planning, supplier scheduling, claim processing, order entry, purchasing, etc.

**Projects:** Costing, billing, activity management, time and expense, etc.

**Customer relationship management** (**CRM**): CRM is a term applied to processes implemented by a company to handle its contact with its customers.

**Data Warehouse:** Usually this is a module that can be accessed by an organizations customers, suppliers and employees.Data warehouse is a repository of an organization’s electronically stored data.

## Advantages of ERP Systems

There are many advantages of implementing an EPR system; here are a few of them:

* A totally integrated system
* The ability to streamline different processes and workflows
* The ability to easily share data across various departments in an organization
* Improved efficiency and productivity levels
* Better tracking and forecasting
* Lower costs
* Improved customer service

## Disadvantages of ERP Systems

While advantages usually outweigh disadvantages for most organizations implementing an ERP system, here are some of the most common obstacles experienced:

Usually many obstacles can be prevented if adequate investment is made and adequate training is involved, however, success does depend on skills and the experience of the workforce to quickly adapt to the new system.

* Customization in many situations is limited
* The need to reengineer business processes
* ERP systems can be cost prohibitive to install and run
* Technical support can be shoddy
* ERP’s may be too rigid for specific organizations that are either new or want to move in a new direction in the near future.

# **SAP ERP**

**SAP ERP**[[2]](https://en.wikipedia.org/wiki/SAP_ERP" \l "cite_note-2) is [enterprise resource planning](https://en.wikipedia.org/wiki/Enterprise_resource_planning)software developed by the German company [SAP SE](https://en.wikipedia.org/wiki/SAP_SE). SAP ERP incorporates the key business functions of an organization. The latest version (SAP ERP 6.0) was made available in 2006. The most recent Enhancement Package (EHP8) for SAP ERP 6.0 was released in 2016.[[3](https://en.wikipedia.org/wiki/SAP_ERP#cite_note-3)

Business Processes included in SAP ERP are Operations ([Sales](https://en.wikipedia.org/wiki/Sales) & [Distribution](https://en.wikipedia.org/wiki/Distribution_(business)),

SAP ERP consists of several modules, including Financial Accounting (FI), Controlling (CO), Asset Accounting (AA), Sales & Distribution (SD), Material Management (MM), Product Planning (PP), Quality Management (QM), Project System (PS), Plant Maintenance (PM), Human Resources (HR).[[6]](https://en.wikipedia.org/wiki/SAP_ERP#cite_note-6) SAP ERP collects and combines data from the separate modules to provide the company or organization with enterprise resource planning.

ERP advantages and disadvantages[[edit](https://en.wikipedia.org/w/index.php?title=SAP_ERP&action=edit&section=5" \o "Edit section: ERP advantages and disadvantages)]

**Advantages**

* Allows easier global integration (barriers of currency exchange rates, language, and culture can be bridged automatically)
* Updates only need to be done once to be implemented company-wide
* Provides real-time information, reducing the possibility of redundancy errors
* May create a more efficient work environment for employees[[9]](https://en.wikipedia.org/wiki/SAP_ERP#cite_note-Monk_Wagner_2009_p23-34-9)
* Vendors have past knowledge and expertise on how to best build and implement a system
* [User interface](https://en.wikipedia.org/wiki/User_interface) is completely customizable allowing end users to dictate the operational structure of the product

**Disadvantages**

* Locked into relationship by contract and manageability with vendor - a contract can hold a company to the vendor until it expires and it can be unprofitable to switch vendors if [switching costs](https://en.wikipedia.org/wiki/Switching_barriers) are too high
* Inflexibility - vendor packages may not fit a company's business model well and customization can be expensive
* [Return on Investment](https://en.wikipedia.org/wiki/Rate_of_return) may take too long to be profitable
* Implementations have a risk of project failure[[9]](https://en.wikipedia.org/wiki/SAP_ERP#cite_note-Monk_Wagner_2009_p23-34-9)

**Advantages Of Linux:**

Linux was one of the first open-source technologies, but many programmers have contributed and added software that’s completely open-source for any user. This means that you can download the source code and change it any way you like. Some developers have restrictions on how you can distribute the code. For instance, some developers allow you to change the code, but you cannot distribute it for money.

**Low cost:**

You don’t need to spend time and money to obtain licenses since Linux and much of its software come with the GNU General Public License. You can start to work immediately without worrying that your software may stop working anytime because the free trial version expires. Additionally, there are large repositories from which you can freely download high quality software for almost any task you can think of.

**Stability:**

Linux doesn’t need to be rebooted periodically to maintain performance levels. It doesn’t freeze up or slow down over time due to memory leaks and such. Continuous up-times of hundreds of days (up to a year or more) are not uncommon.

**Performance:**

Linux provides persistent high performance on workstations and on networks. It can handle unusually large numbers of users simultaneously, and can make old computers sufficiently responsive to be useful again.

**Network friendliness:**

Linux was developed by a group of programmers over the Internet and has therefore strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks such as network backups faster and more reliably than alternative systems.

**Flexibility:**

Linux can be used for high performance server applications, desktop applications, and embedded systems. You can save disk space by only installing the components needed for a particular use. You can restrict the use of specific computers by installing for example only selected office applications instead of the whole suite.

**Compatibility:**

It runs all common Unix software packages and can process all common file formats.

**Choice**:

The large number of Linux distributions gives you a choice. Each distribution is developed and supported by a different organization. You can pick the one you like best; the core functionalities are the same; most software runs on most distributions. Fast and easy installation: Most Linux distributions come with user-friendly installation and setup programs. Popular Linux distributions come with tools that make installation of additional software very user friendly as well.

**Full use of hard disk**:

Linux continues work well even when the hard disk is almost full.

**Multitasking:**

Linux is designed to do many things at the same time; e.g., a large printing job in the background won’t slow down your other work.

**Security:**

Linux is one of the most secure operating systems. “Walls” and flexible file access permission systems prevent access by unwanted visitors or viruses. Linux users have to option to select and safely download software, free of charge, from online repositories containing thousands of high quality packages. No purchase transactions requiring credit card numbers or other sensitive personal information are necessary.

**Open Source:**

If you develop software that requires knowledge or modification of the operating system code, Linux’s source code is at your fingertips. Most Linux applications are Open Source as well.

**Disadvantages of Linux**

There’s no standard edition of Linux. Whereas Microsoft offers several different editions of each version of Windows, there are countless variations of Linux. For a new user it can be confusing to work out which is best for you.

Linux has patchier support for drivers (the software which coordinates your hardware and your operating system). This means you’ll sometimes find it trickier to get a new device set up.

Linux is, for new users at least, not as easy to use as Windows. That’s largely because Linux gives you more control, but does mean you’ll have to spend some time getting used to the way it works.

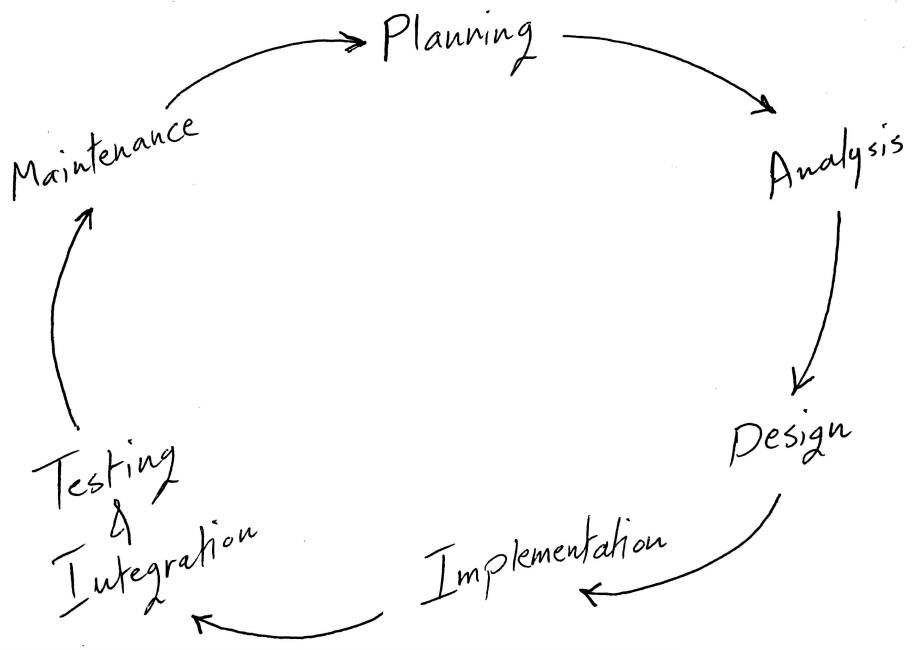
Because Linux is neither as popular as Windows, nor a commercial product, support works in a different way. You may have to look harder to find the answer to a problem and, while Linux supporters are more likely to offer help, it may not always match your own level of technical understanding.

# **System Development Life Cycle**

System Development Life Cycle (SDLC) is a series of six main phases to create a hardware system only, a software system only or a combination of both to meet or exceed customer’s expectations.

**System** is a broad and a general term, and as per to Wikipedia; “A **system** is a set of interacting or interdependent components forming an integrated whole” it’s a term that can be used in different industries, therefore [Software Development Life Cycle](https://airbrake.io/blog/insight/what-is-the-software-development-life-cycle) is a limited term that explains the phases of creating a software component that integrates with other software components to create the whole system.

# **System Development Life Cycle Phases:**



 1- System Planning

The Planning phase is the most crucial step in creating a successful system, during this phase you decide exactly what you want to do and the problems you’re trying to solve, by:

* Defining the problems, the objectives and the resources such as personnel and costs.
* Studying the ability of proposing alternative solutions after meeting with clients, suppliers, consultants and employees.
* Studying how to make your product better than your competitors’.

After analyzing this data you will have three choices: develop a new system, improve the current system or leave the system as it is.

## 2- System Analysis

The end-user’s requirements should be determined and documented, what their expectations are for the system, and how it will perform. A feasibility study will be made for the project as well, involving determining whether it’s organizationally, economically, socially, technologically feasible. it’s very important to maintain strong communication level with the clients to make sure you have a clear vision of the finished product and its function.

## 3- System Design

The design phase comes after a good understanding of customer’s requirements, this phase defines the elements of a system, the components, the security level, modules, architecture and the different interfaces and type of data that goes through the system.

A general system design can be done with a pen and a piece of paper to determine how the system will look like and how it will function, and then a detailed and expanded system design is produced, and it will meet all functional and technical requirements, logically and physically.

## 4- Implementation and Deployment

This phase comes after a complete understanding of system requirements and specifications, it’s the actual construction process after having a complete and illustrated design for the requested system.

In the Software Development Life Cycle, the actual code is written here, and if the system contains hardware, then the implementation phase will contain configuration and fine-tuning for the hardware to meet certain requirements and functions.

In this phase, the system is ready to be deployed and installed in customer’s premises, ready to become running, live and productive, training may be required for end users to make sure they know how to use the system and to get familiar with it, the implementation phase may take a long time and that depends on the complexity of the system and the solution it presents.

## 5- System Testing and Integration

Bringing different components and subsystems together to create the whole integrated system, and then Introducing the system to different inputs to obtain and analyze its outputs and behavior and the way it functions. Testing is becoming more and more important to ensure customer’s satisfaction, and it requires no knowledge in coding, hardware configuration or design.

Testing can be performed by real users, or by a team of specialized personnel, it can also be systematic and automated to ensure that the actual outcomes are compared and equal to the predicted and desired outcomes.

## 6-  System Maintenance

In this phase, periodic maintenance for the system will be carried out to make sure that the system won’t become obsolete, this will include replacing the old hardware and continuously evaluating system’s performance, it also includes providing latest updates for certain components to make sure it meets the right standards and the latest technologies to face current security threats.

These are the main six phases of the System Development Life Cycle, and it’s an iterative process for each project. It’s important to mention that excellent communication level should be maintained with the customer, and Prototypes are very important and helpful when it comes to meeting the requirements. By building the system in short iterations; we can guarantee meeting the customer’s requirements before we build the whole system.