IES Maestre de Calatrava

Maestre de Calatrava
INSTITUTO DE EDUCACION SECUNDARIA

UNIT2. Installation and Configuration of ERP/CRM systems

Rosa María Zapata Calle

Index

- Software license
- Propietary Software licenses
- Open License Program (OLP)
- Free and open-source software licenses
- General Public License (GPL)
- PERMISSIVE FREE SOFTWARE LICENSES
- Free Software Foundation
- Free Software
- Definition and the Four Freedoms
- Open Source Initiative
- Free Software foundation vs open source initiative
- GNU Licenses

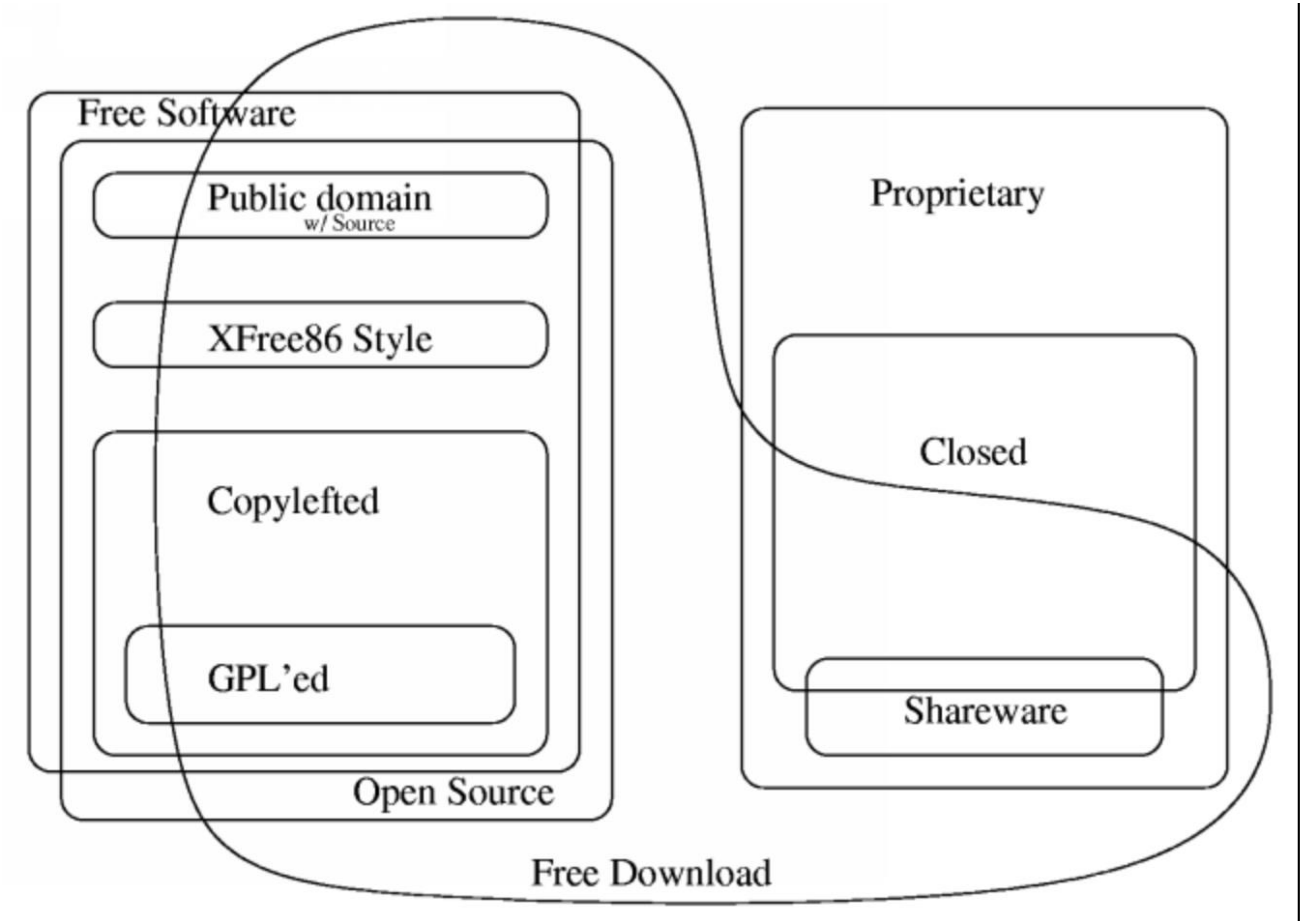
Index

- System Infrastructure
- On-Premise ERP System
- HOSTED/SaaS ERP System
- Cloud ERP System
- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- CLOUD ERP System
- ERP System Installing
- ODOO Structure
- Dynamic Nav
- SCRUM

Software Licenses

- A **software license** is a legal instrument (usually by way of contract law, with or without printed material) governing the use or redistribution of software.
- Software licenses can generally be fit into the following categories:
  - proprietary licenses
  - free and open source.

Software Licenses



Propietary Software licenses

- The software publisher grants the use of one or more copies of software under the end-user license agreement, but ownership of those copies remains with the software publisher (hence use of the term "proprietary").
- This feature of proprietary software licenses means that certain rights regarding the software are reserved by the software publisher. Therefore, it is typical of them to include terms which define the uses of the software, such as the number of installations allowed or the terms of distribution.

Propietary Software licenses

- The most significant effect of this form of licensing is that, **if ownership of the software remains with the software publisher, then the end-user *must* accept the software license**. In other words, without acceptance of the license, the end-user may not use the software at all.
  - One example of such a proprietary software license is the license for Microsoft Windows

Propietary Software license

- These open volume license programs are typically called:
  - Open License Program (OLP)
  - Transactional License Program (TLP)
  - Volume License Program (VLP) etc.

and are contrary to the **Contractual License Program (CLP)**, where the customer commits to purchase a certain amount of licenses over a fixed period (mostly two years).

Open License Program (OLP)

It is a Microsoft service that allows corporate, academic, charitable, or government organizations to obtain volume licenses for Microsoft products. It is ideally suited for companies with between 2 – 250 personal computers, but can accommodate organizations with up to 750 computers.

Free and open-source software licenses

- Free and open-source licenses generally fall under two categories:
  - Those with the aim to have minimal requirements about how the software can be redistributed (permissive licenses), and those that aim to preserve the freedoms that are given to the users by ensuring that all subsequent users receive those rights (copyleft Licenses).

General Public License (GPL)

- An example of a **copyleft free software license** is the **GNU General Public License (GPL)**.
- This license is aimed at giving all user unlimited freedom to use, study, and privately modify the software, and if the user adheres to the terms and conditions of GPL, freedom to redistribute the software or any modifications to it.
- For instance, any modifications made and redistributed by the end-user must include the source code for these, and the license of any derivative work must not put any additional restrictions beyond what GPL allows.

PERMISSIVE FREE SOFTWARE LICENSES

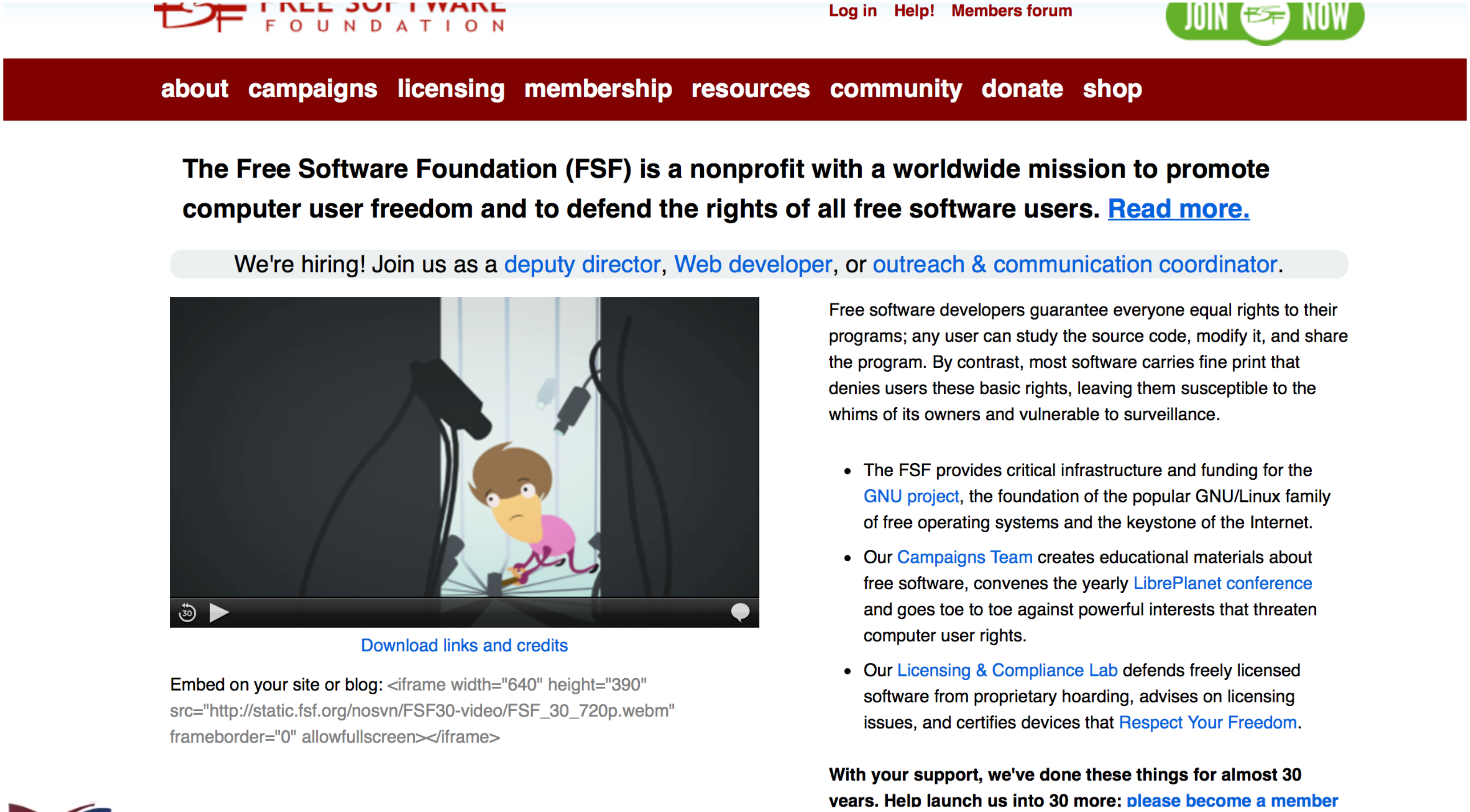
- Examples of permissive free software licenses are:
  - the BSD license
  - the MIT license

  which give unlimited permission to use, study, and privately modify the software, and includes only minimal requirements on redistribution. This gives a user the permission to take the code and use it as part of closed-source software or software released under a proprietary software license.

Free Software Foundation

- **Free Software Foundation**, the group that maintains The Free Software Definition, maintains a non-exhaustive list of free software licenses.
- The list distinguishes between free software licenses that are compatible or incompatible with the FSF license of choice, the GNU General Public License, which is a copyleft license.
- The list also contains licenses which the FSF considers non-free for various reasons, but which are sometimes mistaken as being free.

Free Software Foundation



The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom and to defend the rights of all free software users. Read more.

We're hiring! Join us as a deputy director, Web developer, or outreach & communication coordinator.

Embed on your site or blog: <iframe width="640" height="390" src="http://static.fsf.org/nosvn/FSF30-video/FSF_30_720p.webm" frameborder="0" allowfullscreen></iframe>

Free software developers guarantee everyone equal rights to their programs; any user can study the source code, modify it, and share the program. By contrast, most software carries fine print that denies users these basic rights, leaving them susceptible to the whims of its owners and vulnerable to surveillance.

- The FSF provides critical infrastructure and funding for the GNU project, the foundation of the popular GNU/Linux family of free operating systems and the keystone of the Internet.
- Our Campaigns Team creates educational materials about free software, convenes the yearly LibrePlanet conference and goes toe to toe against powerful interests that threaten computer user rights.
- Our Licensing & Compliance Lab defends freely licensed software from proprietary hoarding, advises on licensing issues, and certifies devices that Respect Your Freedom.

With your support, we've done these things for almost 30 years. Help launch us into 30 more: please become a member

Free Software

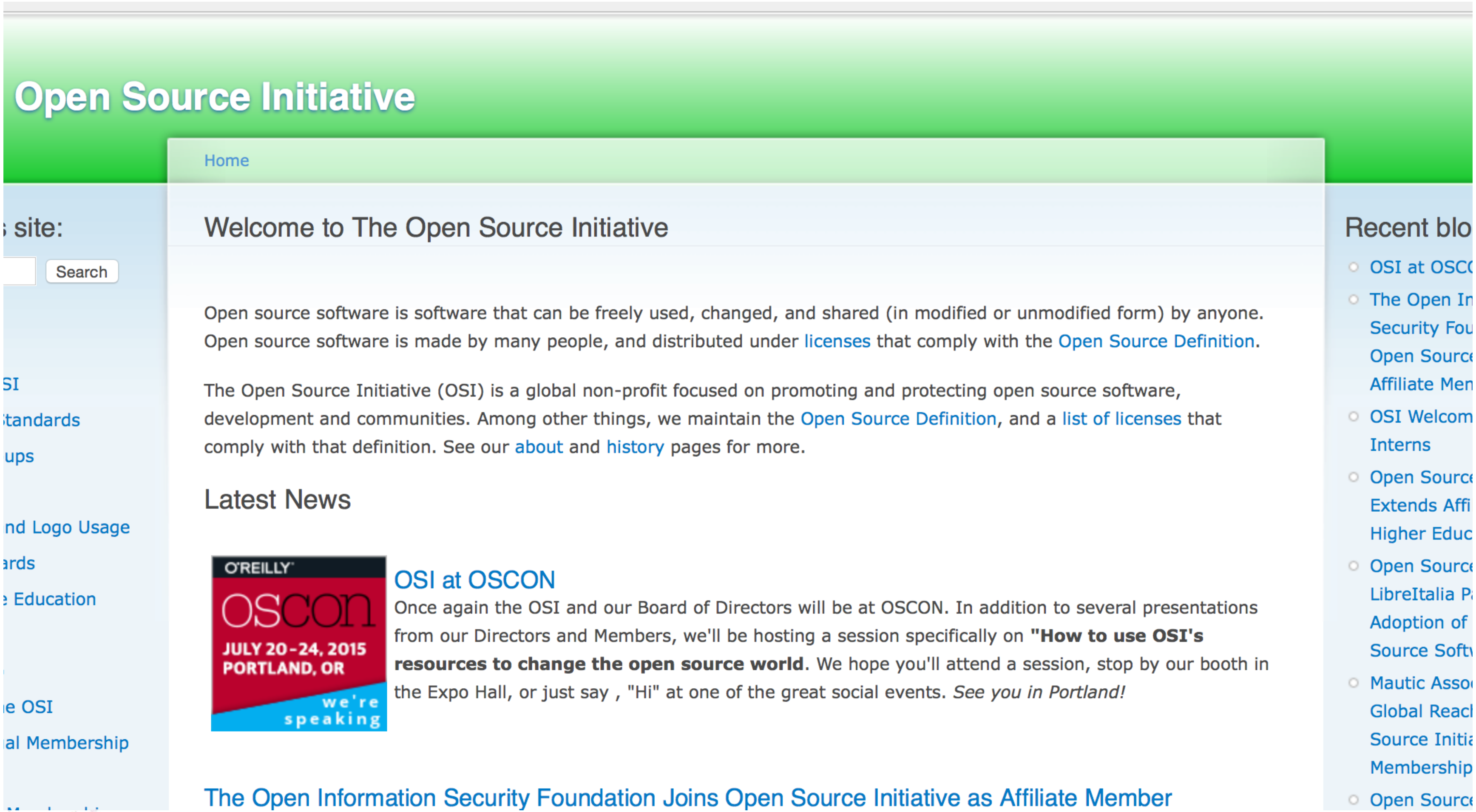
- **Free software** is computer software that gives users the freedom to run the software for any purpose as well as to study, modify, and distribute the original software and the adapted versions. The rights to study and modify free software imply unfettered access to its source code.

Definition and the Four Freedoms

- **Freedom 0:** The freedom to **run** the program for any purpose.
- **Freedom 1:** The freedom to **study** how the program works, and change it to make it do what you wish.
- **Freedom 2:** The freedom to **redistribute** copies so you can help your neighbor.
- **Freedom 3:** The freedom to **improve** the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

- Freedoms 1 and 3 require source code to be available because studying and modifying software without its source code can range from highly impractical to nearly impossible.

Open Source Initiative



Open Source Initiative

Home

site:

Search

SI

tandards

ups

nd Logo Usage

ards

Education

e OSI

al Membership

Welcome to The Open Source Initiative

Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.

The Open Source Initiative (OSI) is a global non-profit focused on promoting and protecting open source software, development and communities. Among other things, we maintain the Open Source Definition, and a list of licenses that comply with that definition. See our about and history pages for more.

Latest News

OSI at OSCON

Once again the OSI and our Board of Directors will be at OSCON. In addition to several presentations from our Directors and Members, we'll be hosting a session specifically on **"How to use OSI's resources to change the open source world**. We hope you'll attend a session, stop by our booth in the Expo Hall, or just say , "Hi" at one of the great social events. *See you in Portland!*

The Open Information Security Foundation Joins Open Source Initiative as Affiliate Member

Recent blo

- OSI at OSC
- The Open In Security Fou Open Source Affiliate Mem
- OSI Welcom Interns
- Open Source Extends Affi Higher Educ
- Open Source LibreItalia Pa Adoption of Source Softw
- Mautic Asso Global Reach Source Initia Membership
- Open Source

Open Source Initiative

http://opensource.org

- Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition
- The Open Source Initiative (OSI) is a global non-profit focused on promoting and protecting open source software, development and communities. Among other things, we maintain the Open Source Definition, and a list of licenses that comply with that definition.

Open Source Definition

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

Open Source Definition

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

Open Source Definition

5. **No Discrimination Against Persons or Groups**

The license must not discriminate against any person or group of persons.

6. **No Discrimination Against Fields of Endeavor**

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. **Distribution of License**

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

Open Source Definition

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

Free Software foundation vs open source initiative

What´s the different between Free Software foundation and open source initiative?

Read the following article, which was written by Richard Stallman:

http://www.gnu.org/philosophy/open-source-misses-the-point.en.html

Find out about it and give us the differences and misunderstanding between both of them.

GNU Licenses

- http://www.gnu.org/licenses/gpl-faq.en.html

Know the most important questions about it is the best way to learn about the GNU licenses.

Exercise

- Find out about the different licenses of the ERP software systems, we learnt one unit before.

(you ought to use a table in order to be clearer)

System Infrastructure

- The three primary options are:
  - on-premise ERP.
  - hosted/managed service ERP
  - cloud ERP solutions.

System infrastructure

- Traditionally, on-premise and hosted ERP solutions provide the most flexibility to tailor packages to meet organizational needs, but they're also costly.
- Feature-rich cloud ERP solutions with robust development tools are rapidly emerging , But in a cloud-based solution model, you ought to make sure it meets your needs and that your data will be safe (properly backed up and secured)

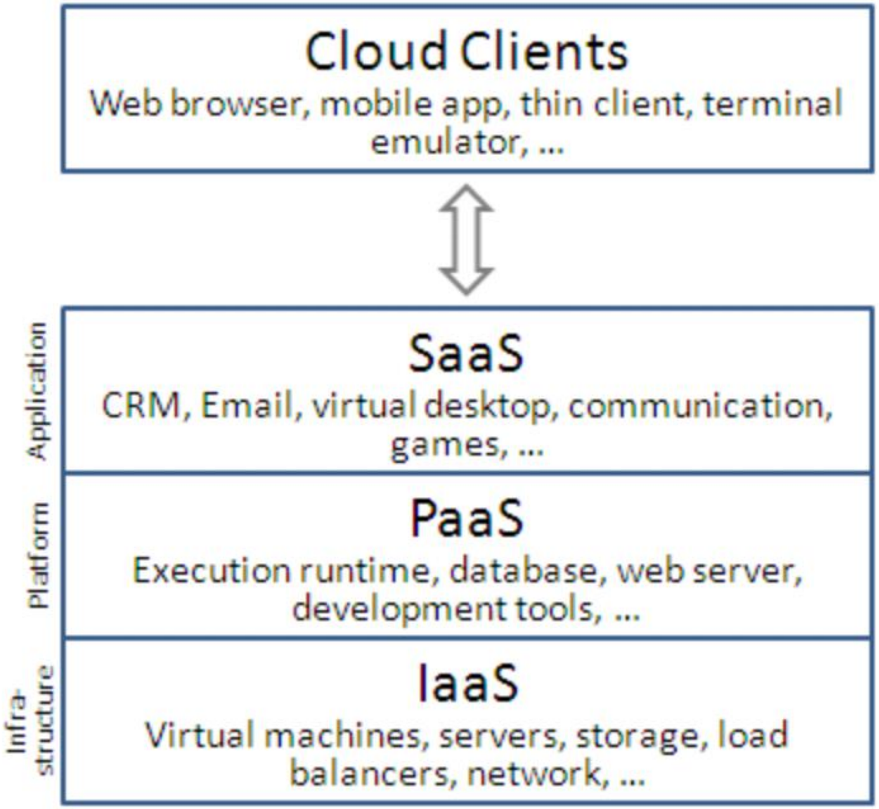
On-Premise ERP System

- **On-premises software** is installed and runs on computers on the premises (in the building) of the person or organization using the software.

HOSTED/SaaS ERP System

- **Software as a service** (**SaaS**) is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted.
- It is sometimes referred to as "on-demand software”. SaaS is typically accessed by users using a thin client via a web browser.

Cloud ERP System



Cloud Clients

Web browser, mobile app, thin client, terminal emulator, ...

Application

SaaS

CRM, Email, virtual desktop, communication, games, ...

Platform

PaaS

Execution runtime, database, web server, development tools, ...

Infra-structure

IaaS

Virtual machines, servers, storage, load balancers, network, ...

Infrastructure as a service (IaaS)

In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources.

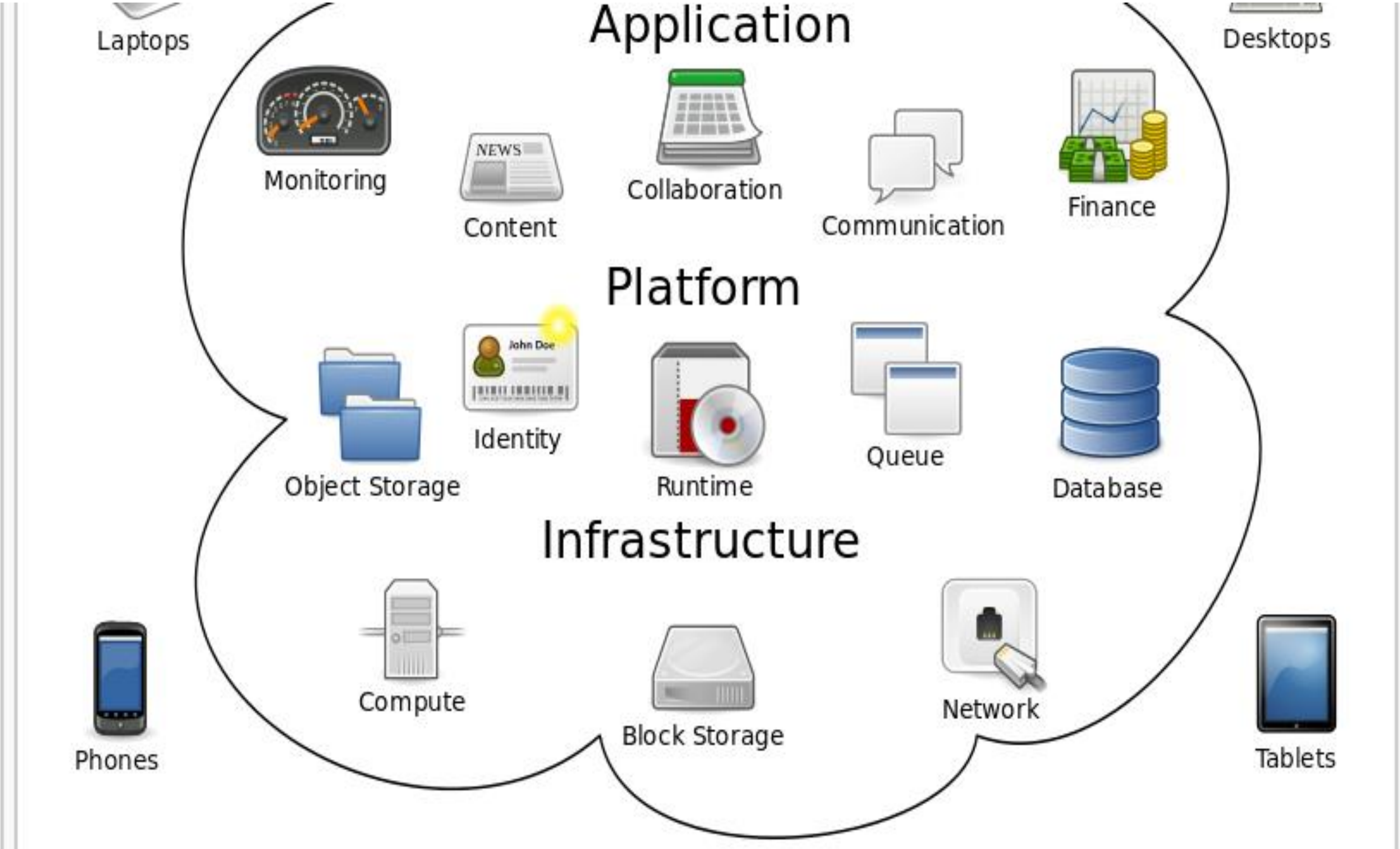
Platform as a service (PaaS)

In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server.

Software as a service (SaaS)

- In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis or using a subscription fee

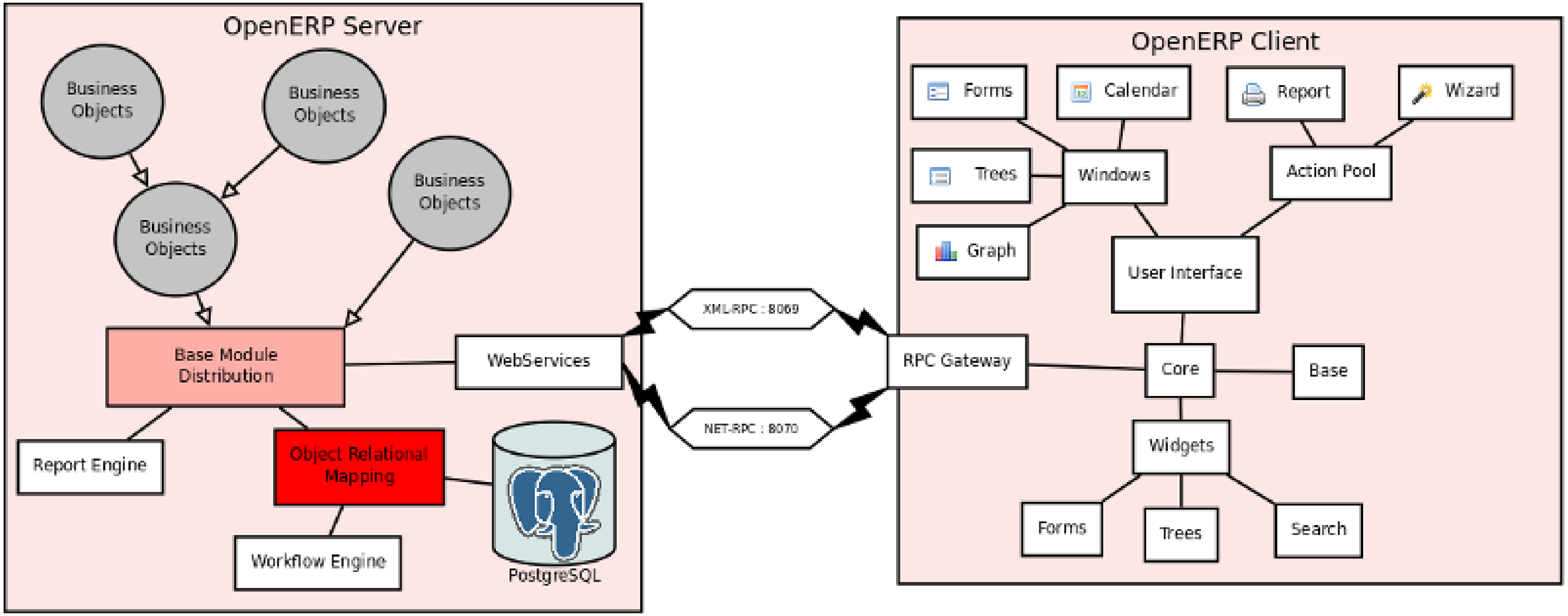
CLOUD ERP System



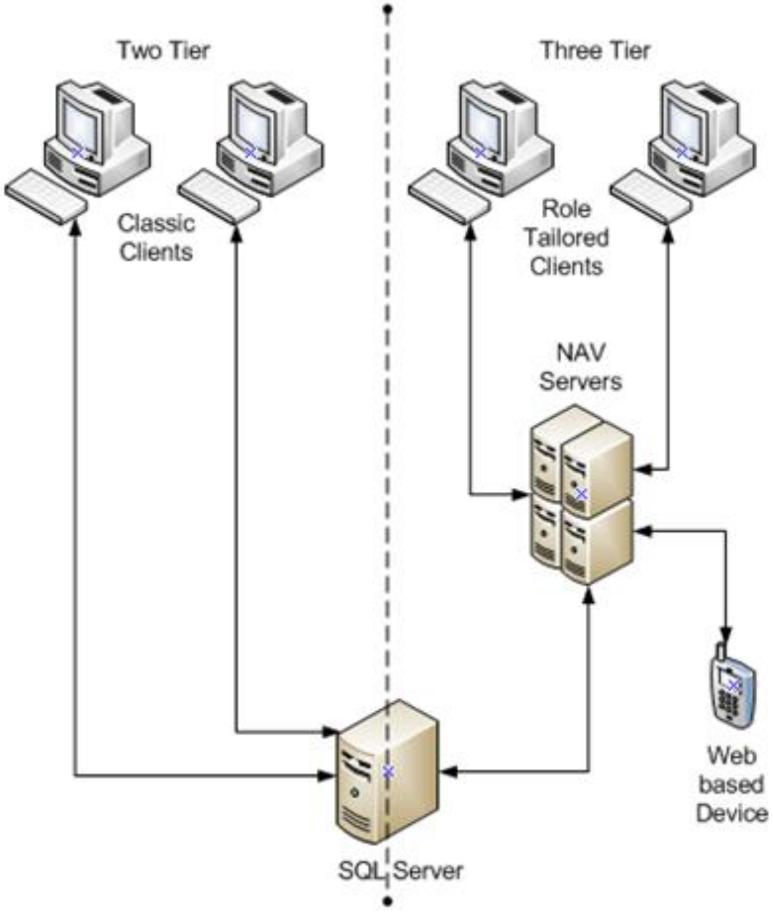
ERP System Installing

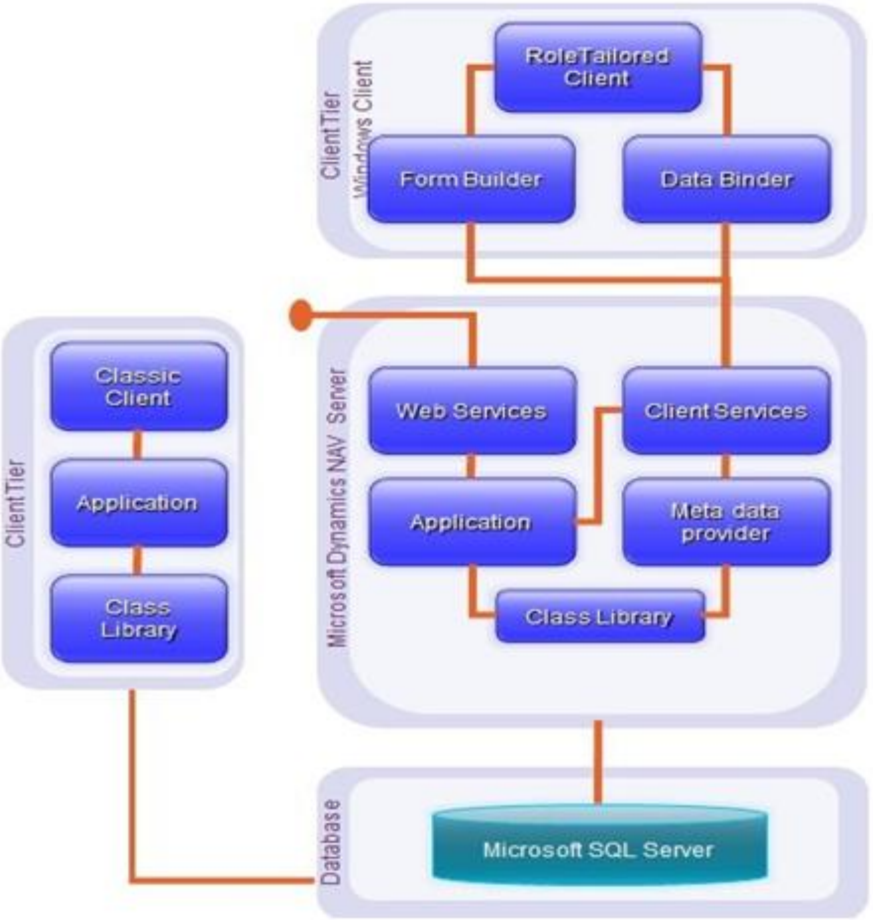
- Now, we are going to install a SAGE system.

ODOO Structure

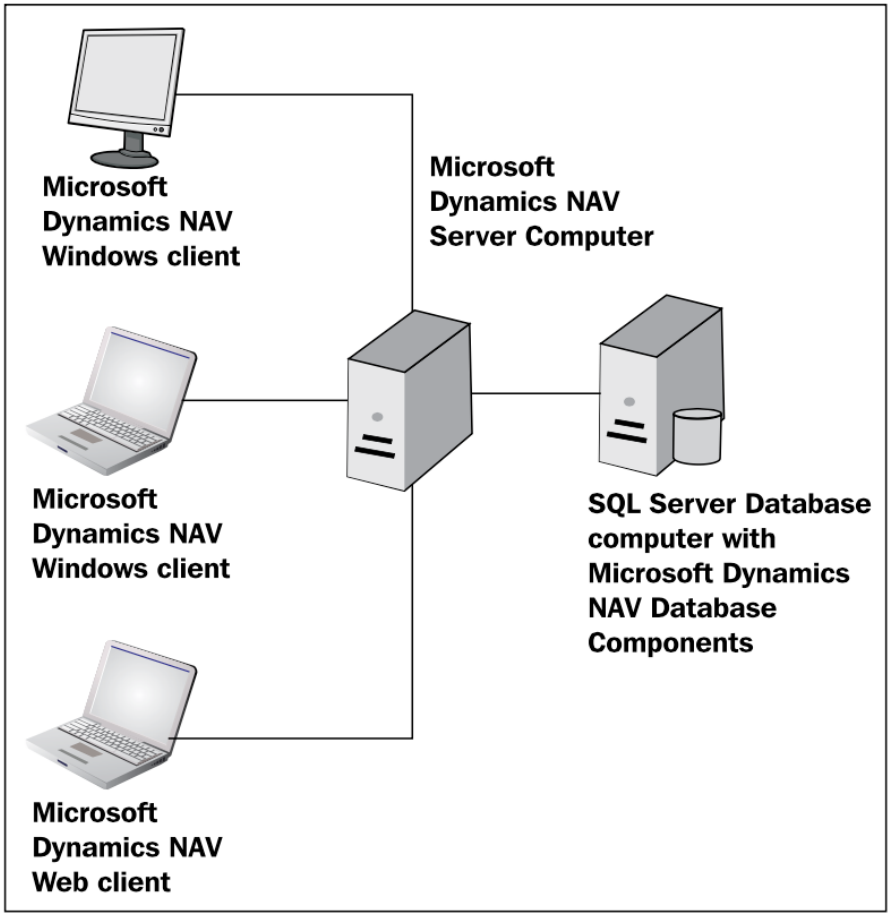


Dynamic Nav


Two Tier
Three Tier
Classic Clients
Role Tailored Clients
NAV Servers
Web based Device
SQL Server


Client Tier
Windows Client
RoleTailored Client
Form Builder
Data Binder
Client Tier
Classic Client
Application
Class Library
Microsoft Dynamics NAV Server
Web Services
Client Services
Application
Meta data provider
Class Library
Database
Microsoft SQL Server

Dynamic Nav


Microsoft Dynamics NAV Windows client
Microsoft Dynamics NAV Server Computer
Microsoft Dynamics NAV Windows client
SQL Server Database computer with Microsoft Dynamics NAV Database Components
Microsoft Dynamics NAV Web client

Free Softwarefoundationvs opensource initiationsgo

EXERCISE 2:

What is the difference between the Free Software Foundation and the Open Source Initiative?

Read the following article, written by RichardStallman:

https://www.gnu.org/philosophy/open-source-misses-the-point.es.html

Research the topic and describe the differences and conceptual errors between the two, according to its author.

INDEX

- 1. PROJECT MANAGER
  - Agile methodologies
  - MethodologiesAgile versus traditional methodologies
  - Scrum

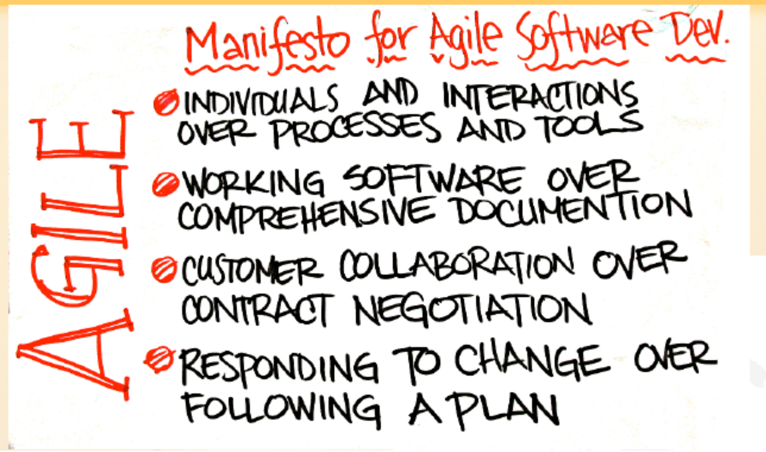
- 2. IMPLEMENTATION METHODOLOGY

1. PROJECT MANAGER

The preliminary step to any project development, and especially in software projects, is the selection of the project manager that the development team, together with the project manager, will use for the complete development of the project.
Previously, the development of software projects was based on rigid methodologies, especially in the face of changes.
To improve these aspects, agile methodologies emerged. Among the most important is the SCRUM project manager.

Search for information about**SCRUM**

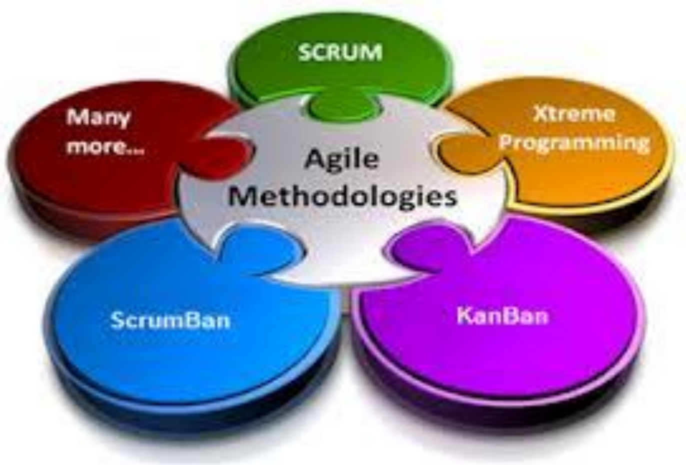
METHODOLOGIESAGILE IN THE IMPLEMENTATION OF AN EMS



AGILE

Manifesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN



Agile Methodologies

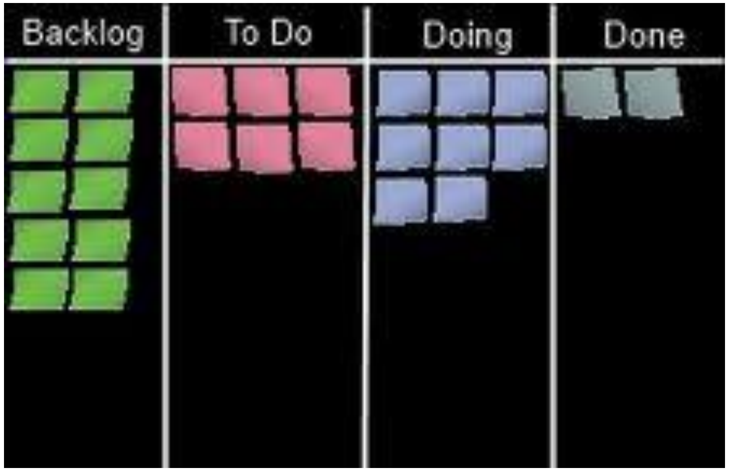
SCRUM

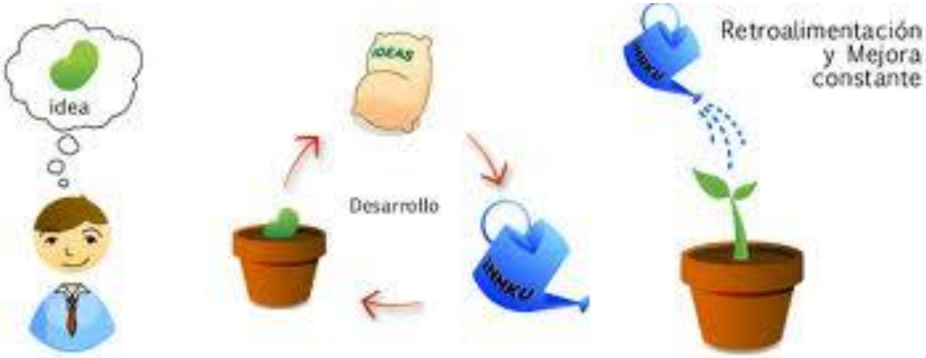
Xtreme Programming

KanBan

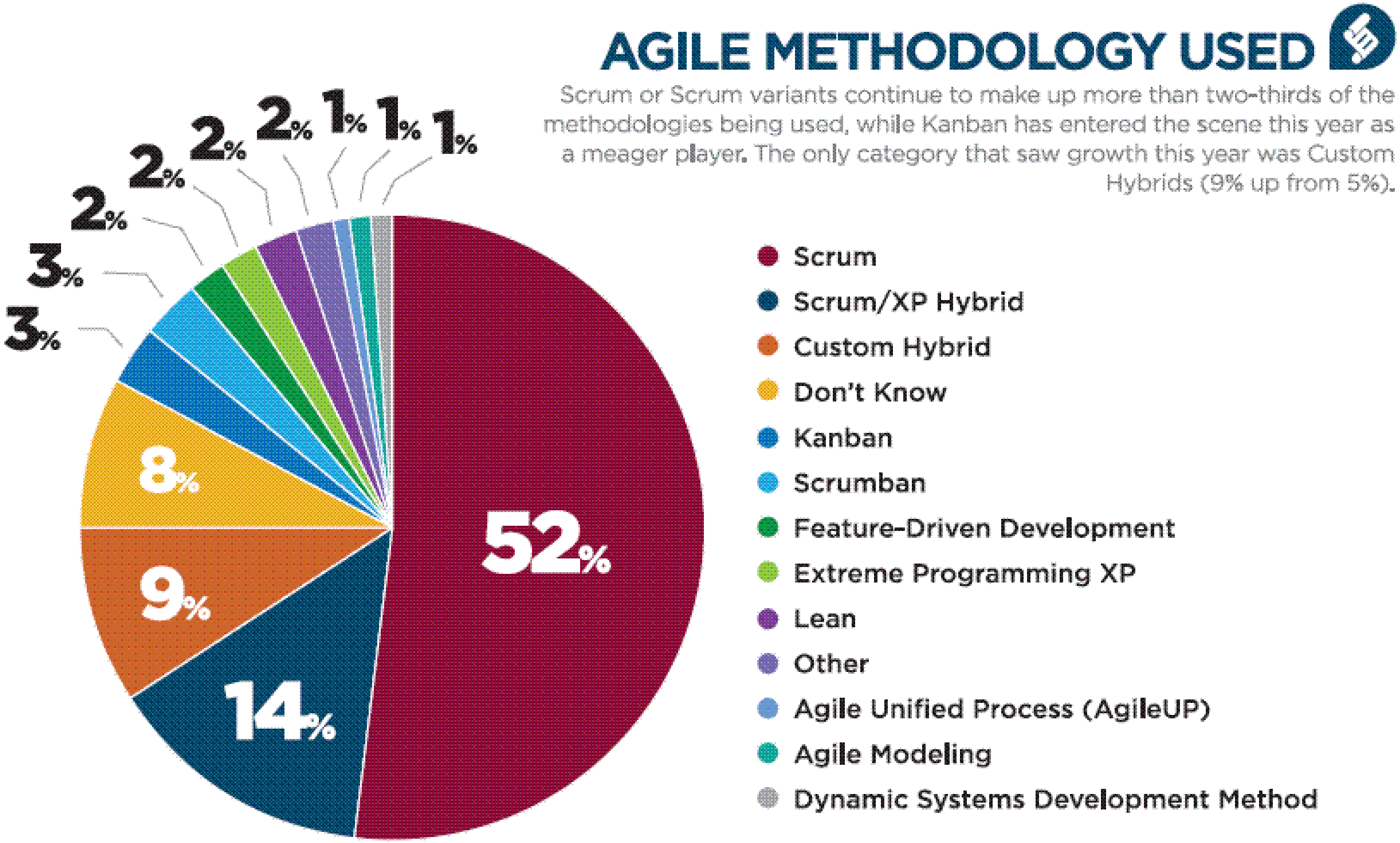
ScrumBan

Many more...





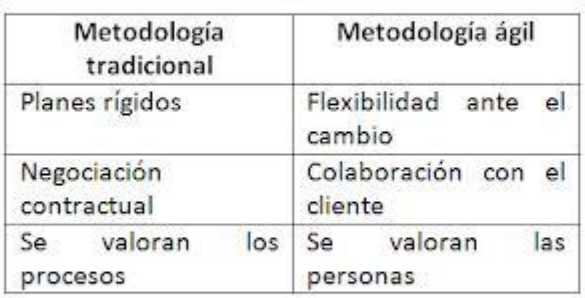
METHODOLOGIESAGILE IN THE IMPLEMENTATION OF AN EMS



AGILE METHODOLOGY USED

Scrum or Scrum variants continue to make up more than two-thirds of the methodologies being used, while Kanban has entered the scene this year as a meager player. The only category that saw growth this year was Custom Hybrids (9% up from 5%).

MethodologiesAgile versus traditional methodologies

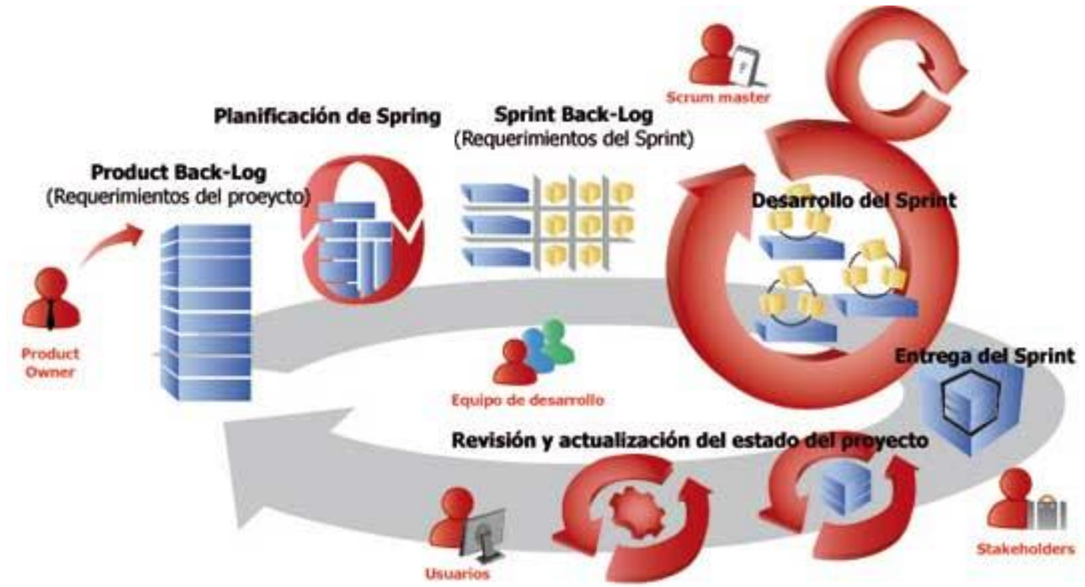


Metodología tradicional	Metodología ágil
Planes rígidos	Flexibilidad ante el cambio
Negociación contractual	Colaboración con el cliente
Se valoran los procesos	Se valoran las personas

METHODOLOGIESAGILE IN THE IMPLEMENTATION OF AN EMS

- Key “agile” principles for business management systems
  - Assume simplicity
  - Welcoming changes naturally
  - Make changes incrementally
  - Assess the efforts of project participants
  - Setting objectives in management tasks
  - Accept multiple views of the project and agree on solutions
  - Find ways to make feedback agile
  - Keep in mind the philosophy ofworking progress

METHODOLOGIESAGILE IN THE IMPLEMENTATION OF AN EMS



SCRUM

Introduction to the model

Scrum it is a very simple development methodology, which requires hard work because it is not based on following a plan, but on

continuous adaptation to the circumstances of the project's evolution.

ScrumIt is an agile methodology, and as such:

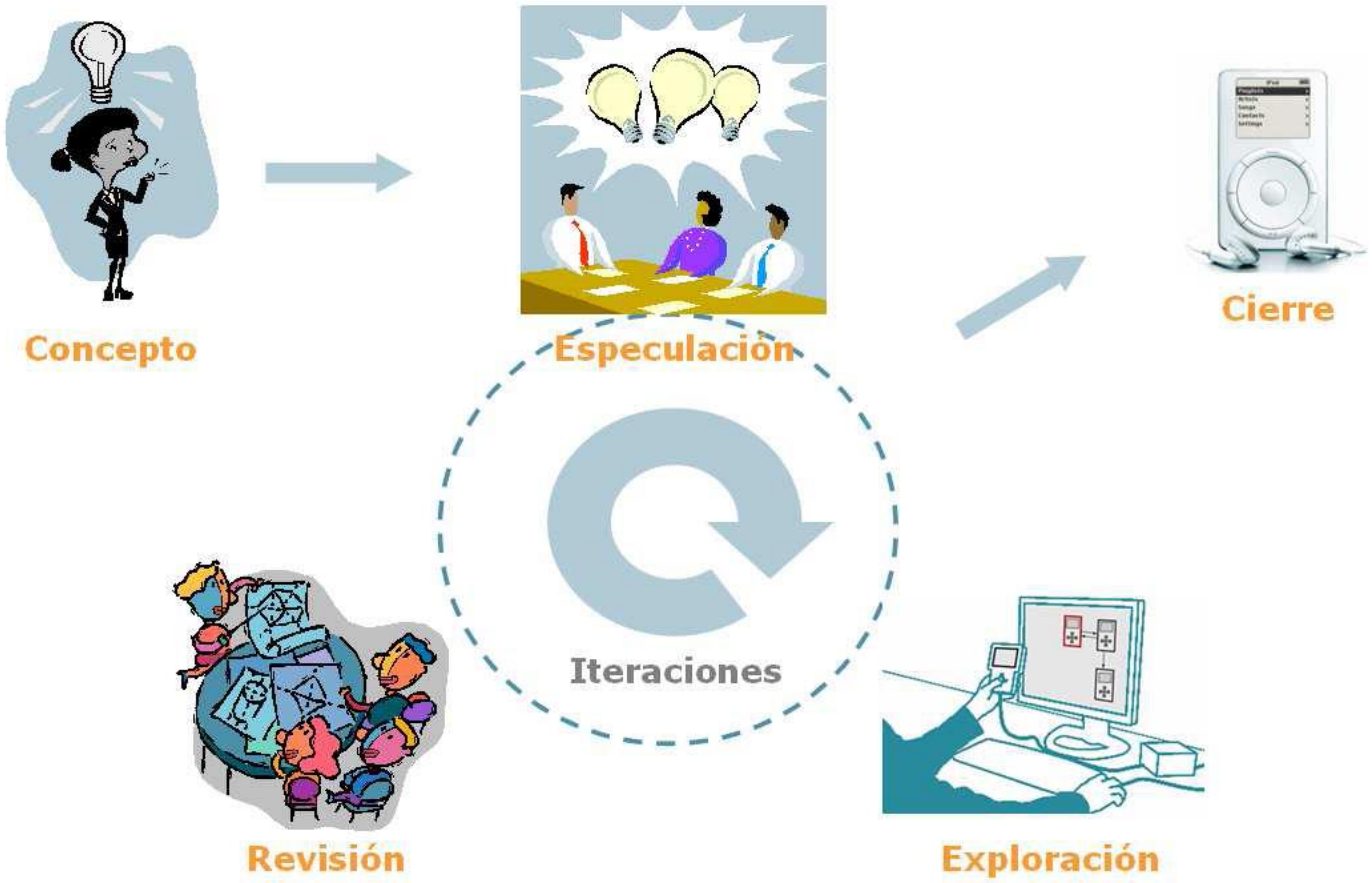
It is a development mode that is adaptive rather than predictive.

People-oriented rather than process-oriented.

Uses the agile development framework:

incremental based on iterations and revisions.

Introduction to the model



Introduction to the model

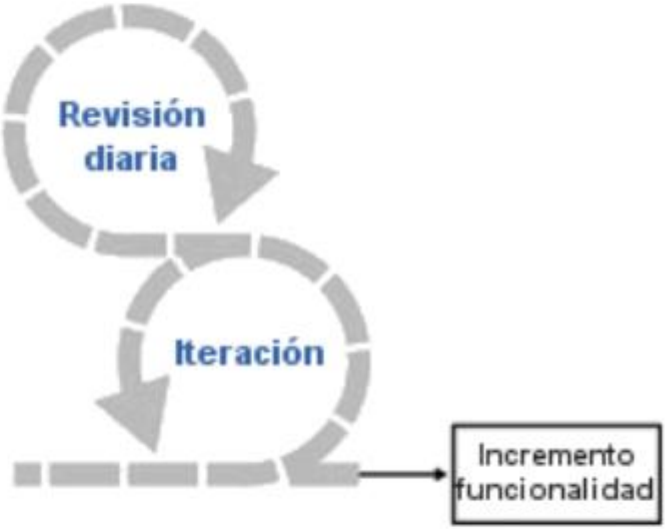
It starts with the product overview, specifying and giving detail to the functionalities or parts that have the highest development priority and that can be carried out in a short period of time (normally 30 days).

Each of these periods of development **It is a iteration that ends with the production of a operational product increase**.

Introduction to the model

These iterations are the basis of agile development, and Scrum manages its evolution through

Daily short meetings where everyone reviews the work done the previous day and the work planned for the following day.



Monitoring project progress

Scrum empirically and adaptively controls the evolution of the project, using the following agile management practices:

Review of iterations

- Incremental development
- Evolutionary development
- Self-organization
- Collaboration

Agile Management Practices

- Review of iterations

At the end of each iteration (usually 30 days)

A review is carried out with all the

people involved in the project(Retrospective). This is the maximum period of time it takes to redirect a deviation in the project or in the circumstances of the product.

Agile Management Practices

- **Incremental development**

During the project, the people involved did not

They work with designs or abstractions.

Incremental development implies that at the end of Each iteration has a portion of the operational product that can be inspected and

assess.

Agile Management Practices

Evolutionary development

Agile management models are used to work in environments of uncertainty and instability of requirements.

Trying to predict in the early stages what the final product will be like, and then developing the product design and architecture based on that prediction, is not realistic, because circumstances will force it to be remodeled many times.

Agile Management Practices

Why predict the final states of architecture or design if they are going to be changing?Scrum instability is taken as a premise, and working techniques are adopted to allow this evolution without degrading the quality of the architecture that will be generated during development.

The developmentScrumis generating the design and thearchitectureend in an evolutionary way throughoutthe project. They are not considered as products to be carried out in the first “phase” of the project.

(Agile development is not phased development)

Agile Management Practices

Self-organization

During the development of a project there are many the unpredictable factors that arise in all areas and levels. Predictive management entrusts the responsibility for its resolution to the manager of projects.

InScrumThe teams are self-organized (not self-directed), with sufficient decision-making room

to make the decisions they consider timely.

Agile Management Practices

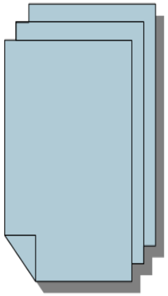
Collaboration

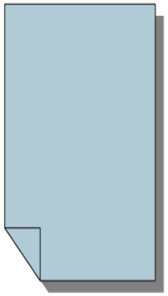
Agile practices and work environments facilitate team collaboration. This is necessary because for self-organization to function as an effective control, each team member must collaborate openly with others, according to their abilities and not according to their role or position.

Elements of development ofScrum

The elements ofScrumare:

- **Product stack:**list of user requirements that originates with the initial vision of the product and grows and evolves during development.
- **Sprint Backlog:**List of work that the team must do during the sprint to generate the expected increase.
- **Increase:**Result of each sprin





Elements of development ofScrum

The roles

Scrum classifies all the people involved or interested in the development of the project into:

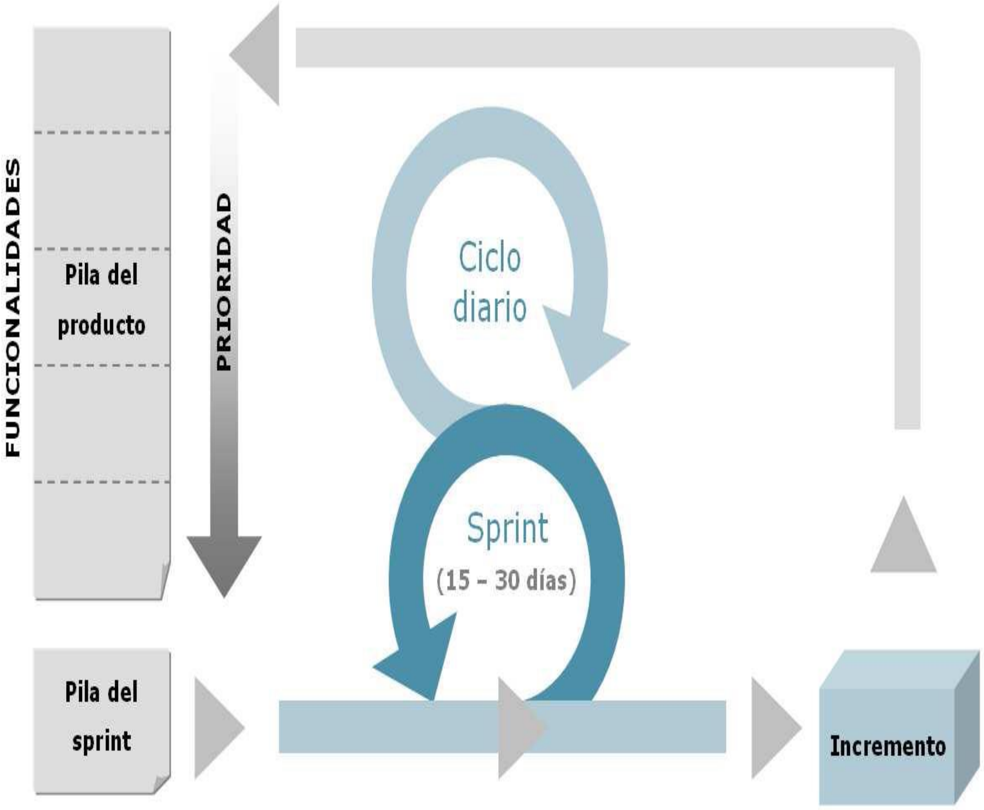
product owner, team,manager ofScrum(also Scrum Manager or Scrum Master) and “other interested parties”.

The first three groups (owner, team and manager) are responsible for the project.

while the rest of the interested parties.

Process overview

- Scrum calls each development iteration a "sprint" and recommends carrying them out with durations of 30 days, for example.
- The sprint is therefore the central core that provides the basis for iterative and incremental development.



Elements of development ofScrum

- The elements that make up development
- Scrumare:
- The meetings.
- The elements
- The roles

Elements of development ofScrum

The meetings

Sprint Planning:

Work session prior to the start of each sprint in which the work to be done and the objectives to be met in that iteration are determined.

Daily meeting:

Brief team review of work done to date and forecast for the following day.

Sprint Review:

Analysis and review of the increase generated.

Elements of development ofScrum

Pigs and chickens.

This metaphor graphically illustrates the difference in involvement in the project between the two groups:

A hen and a pig were walking along the road.

The hen said to the pig: “Do you want to open a restaurant with me?”

The pig considered the proposal and replied: “Yes, I would like to. And what shall we call it?”

The hen replied, “Bacon and eggs.” The pig stopped, paused, and replied,

“On second thought, I guess I won’t open a restaurant with you. I’d be really committed, while you’d just be involved.”

Elements of development ofScrum

COMMITTED (pigs)	INVOLVED(chickens)
Owner of the product Equipment ScrumManager	Other interested parties (General management Business address Marketing Users, etc)

Product Owner: The person responsible for obtaining the greatest product value for customers, users and other stakeholders.

Development team: group or working groups that develop the product.

ScrumManager:team manager who is responsible for the operation of the methodologyScrumand the productivity of the development team.

VALUES

ScrumIt is a “body work” for giving shape to agile principles. It is an aid to organizing people and the workflow; as can be other proposals for agile ways of working:

Crystal, DSDM, etc.

A body without an engine, without the values that give meaning to agile development, does not work.

Delegation of powers (*empowerment) to the*team so that it can self-organize and make decisions about development.

Respect among people. Team members must trust each other and respect each other's knowledge and abilities.

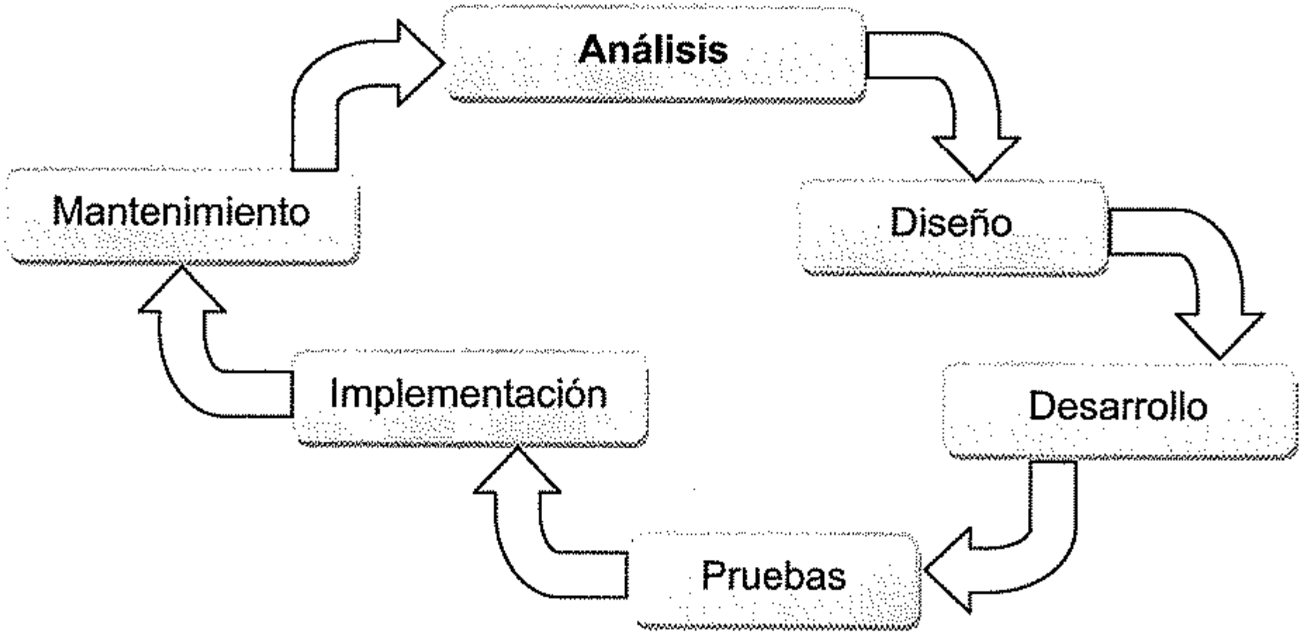
Responsibility and self-discipline (not imposed discipline).

Work focused on the development of what was promised. Information, transparency and visibility of the project development

2. IMPLEMENTATION OF AN ERP

- Once we have decided on a framework, which could be SCRUM, due to the benefits it brings us, we move on to learn about the methodology for implementing an EMS.
- An implementation methodology is a structured and methodological process to successfully carry out development.
- One proposed implementation methodology is one adapted to the classic software development life cycle.

ERP IMPLEMENTATION

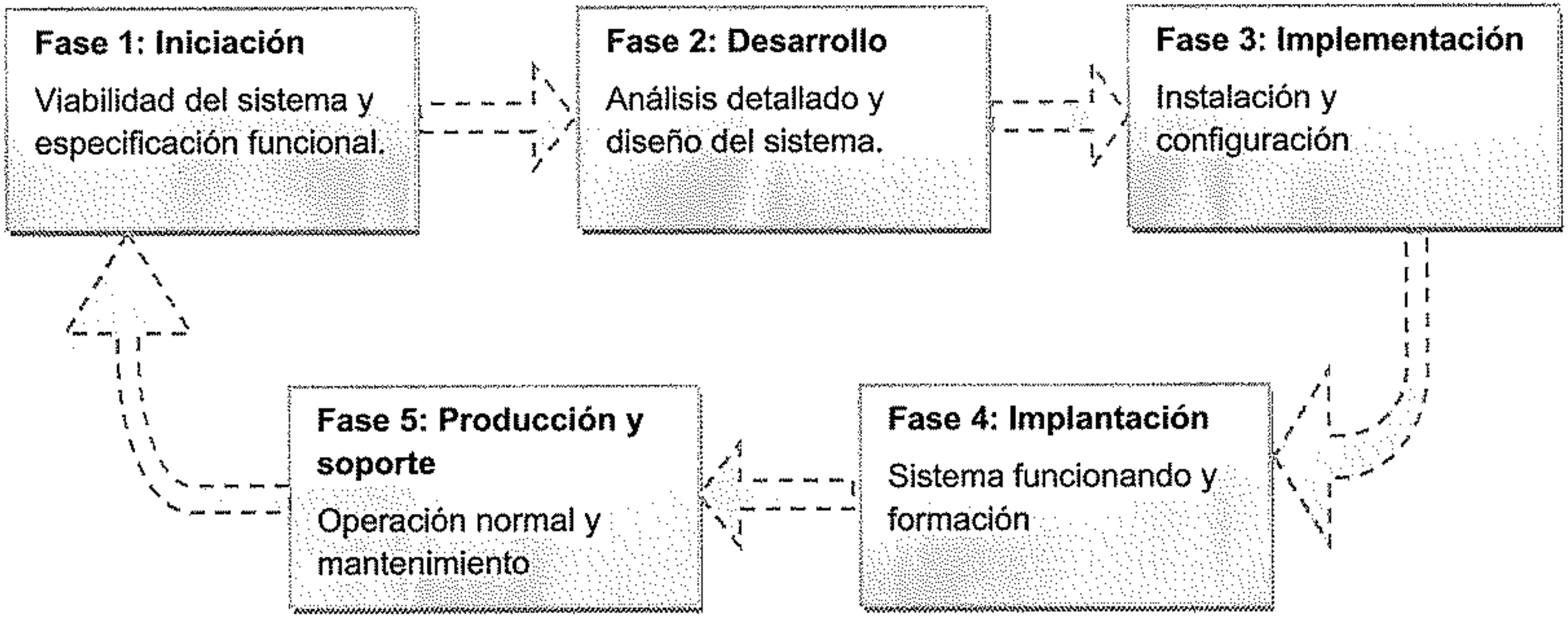


Ciclo de vida clásico

Classic life cycle of a Software project

ERP IMPLEMENTATION

Metodología general: Fases



Fase 1: Iniciación

Viabilidad del sistema y especificación funcional.

Fase 2: Desarrollo

Análisis detallado y diseño del sistema.

Fase 3: Implementación

Instalación y configuración

Fase 4: Implantación

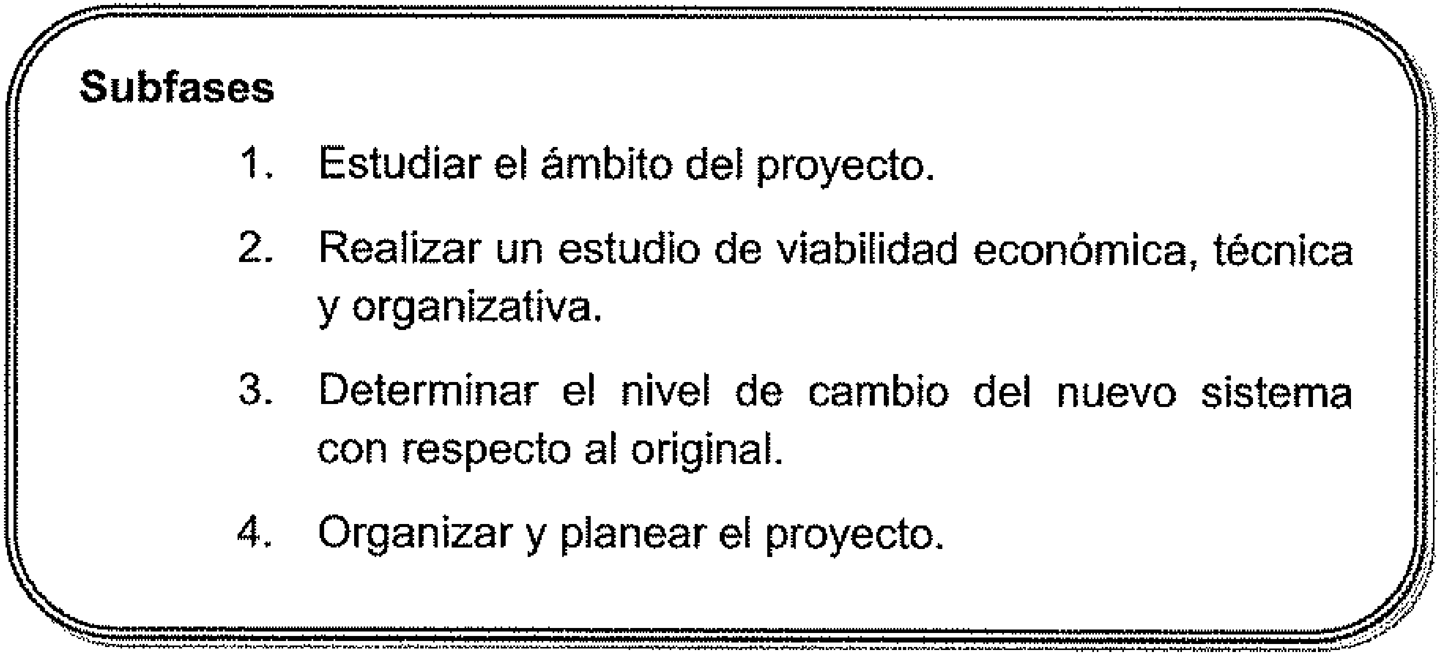
Sistema funcionando y formación

Fase 5: Producción y soporte

Operación normal y mantenimiento

ERP IMPLEMENTATION

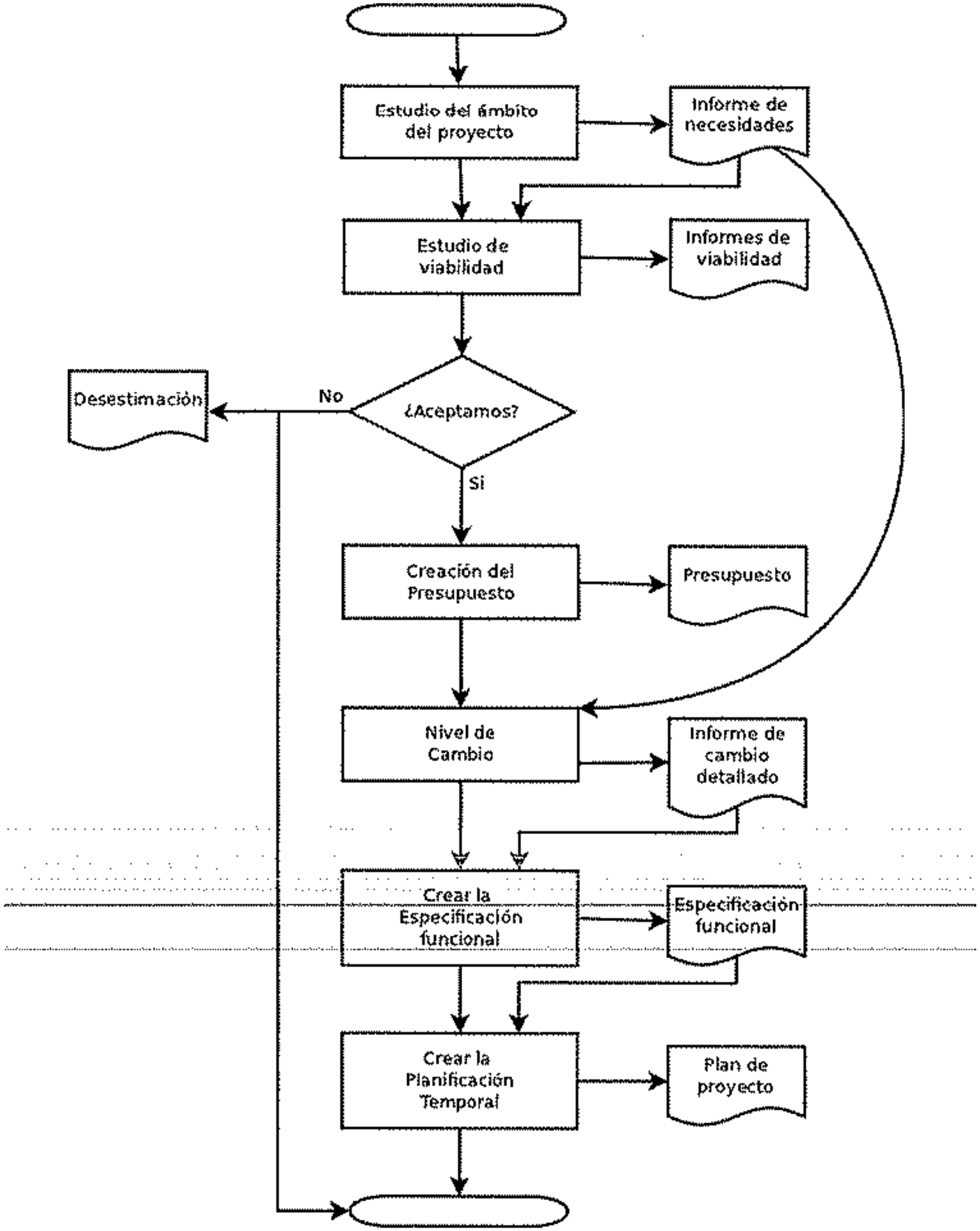
Fase 1: Iniciación



Subfases

1. Estudiar el ámbito del proyecto.
2. Realizar un estudio de viabilidad económica, técnica y organizativa.
3. Determinar el nivel de cambio del nuevo sistema con respecto al original.
4. Organizar y planear el proyecto.

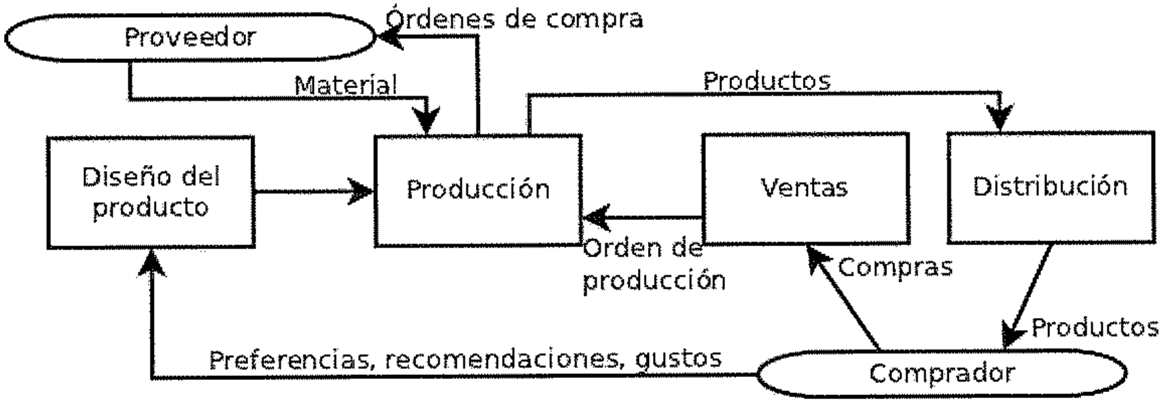
SUBPHASE	ENTRANCE	EXIT
1. Study the scope of the project		Report needs expressed bythe client and the project completion date, a list of components ofthecompanywho will be affected and in what wayquantity, all information deemed important regarding changes to be introduced in the production system and its operation.
2. Feasibility study	Previous report	Acceptance proposal(budget) or rejection of the project.
3.Determine the level of change of the new system with respect to the original (if it exists)	Needs Report	Detailed change report.
4.Organize and plan the project		Specificationsystem functional and project plan.



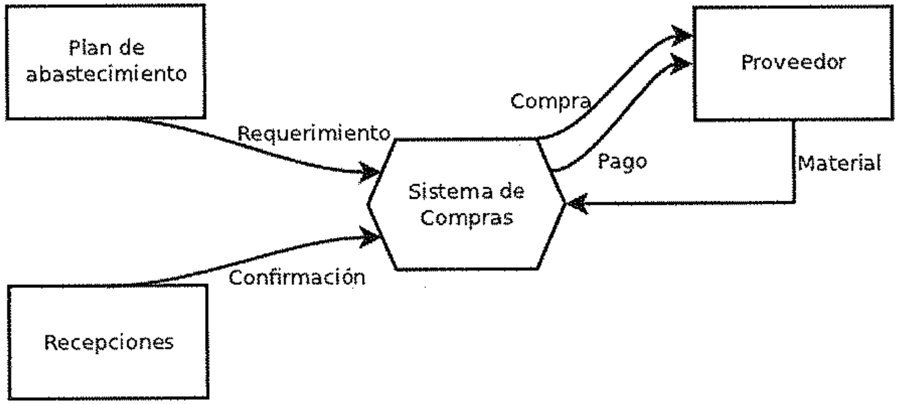
DFD resumen de la fase Uno

DFD Examples (Diagrams for Analysis)

Ejemplos de DFDs

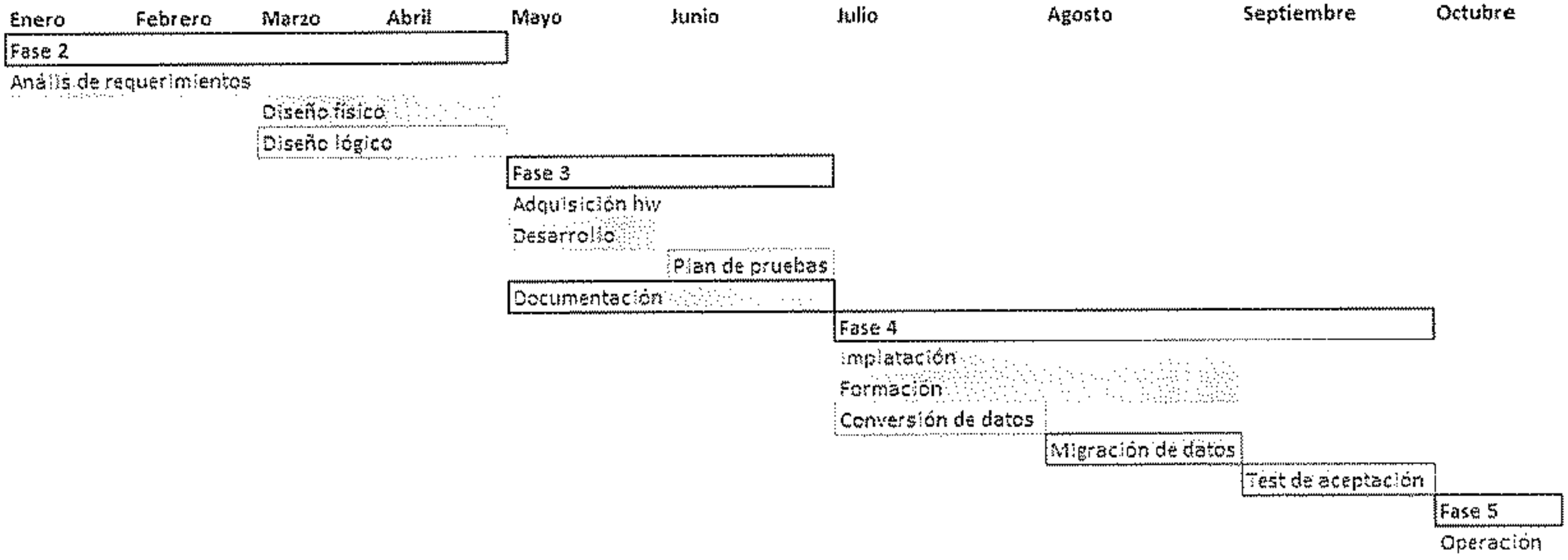


DFD general de una empresa (subfase 1)



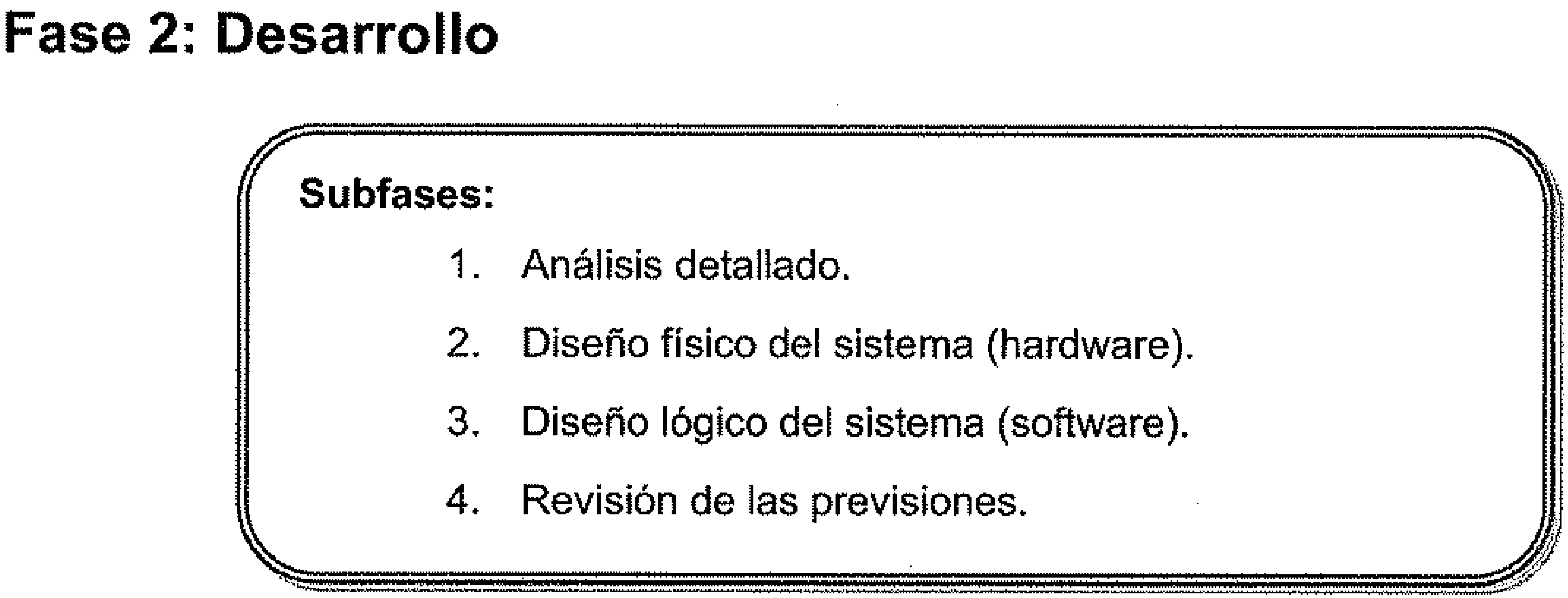
DFD del sistema de compras (subfase 4)

Diagram ofGannt-Project planning



Plan de proyecto simplificado (subfase 4)

ERP IMPLEMENTATION

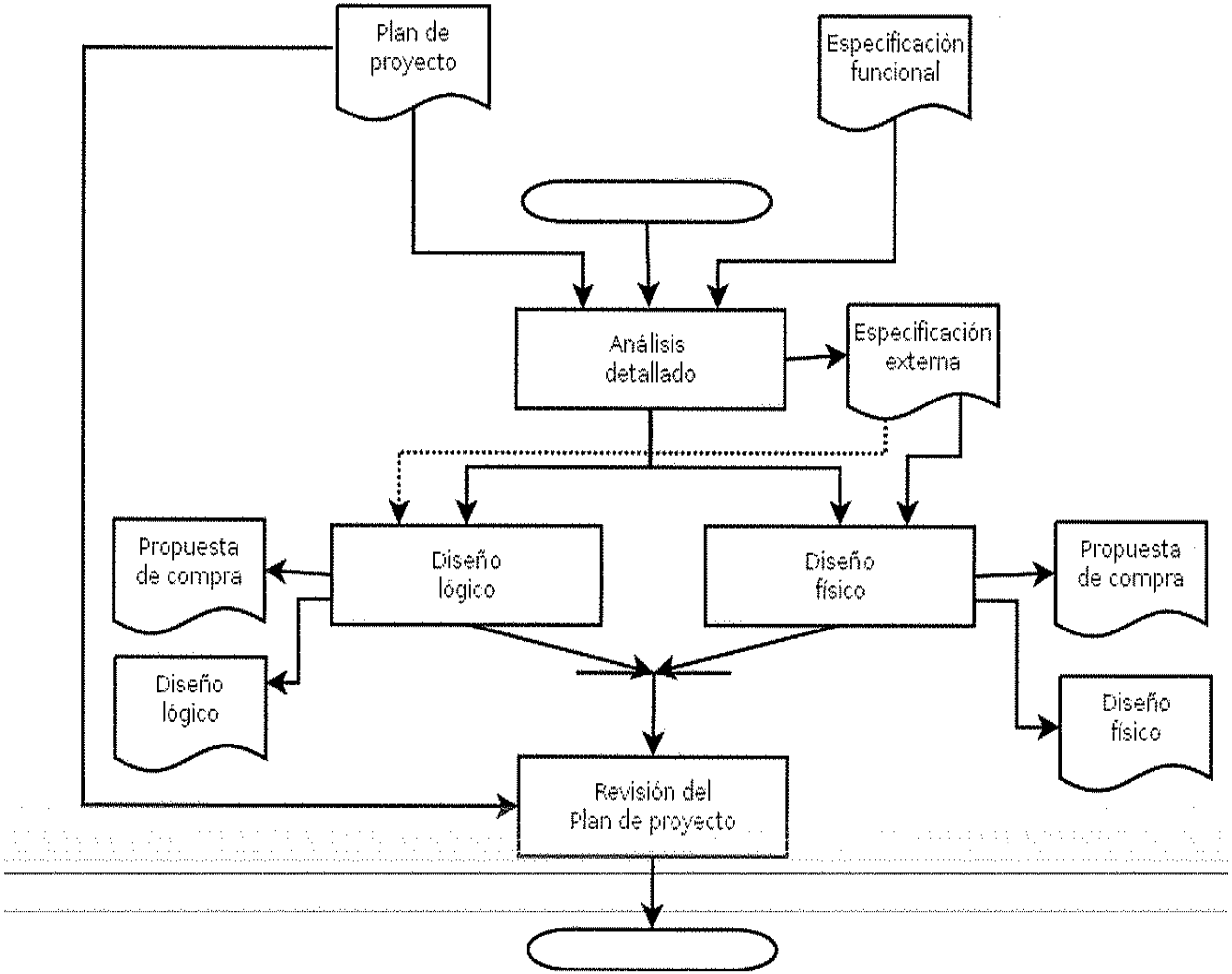


Fase 2: Desarrollo

Subfases:

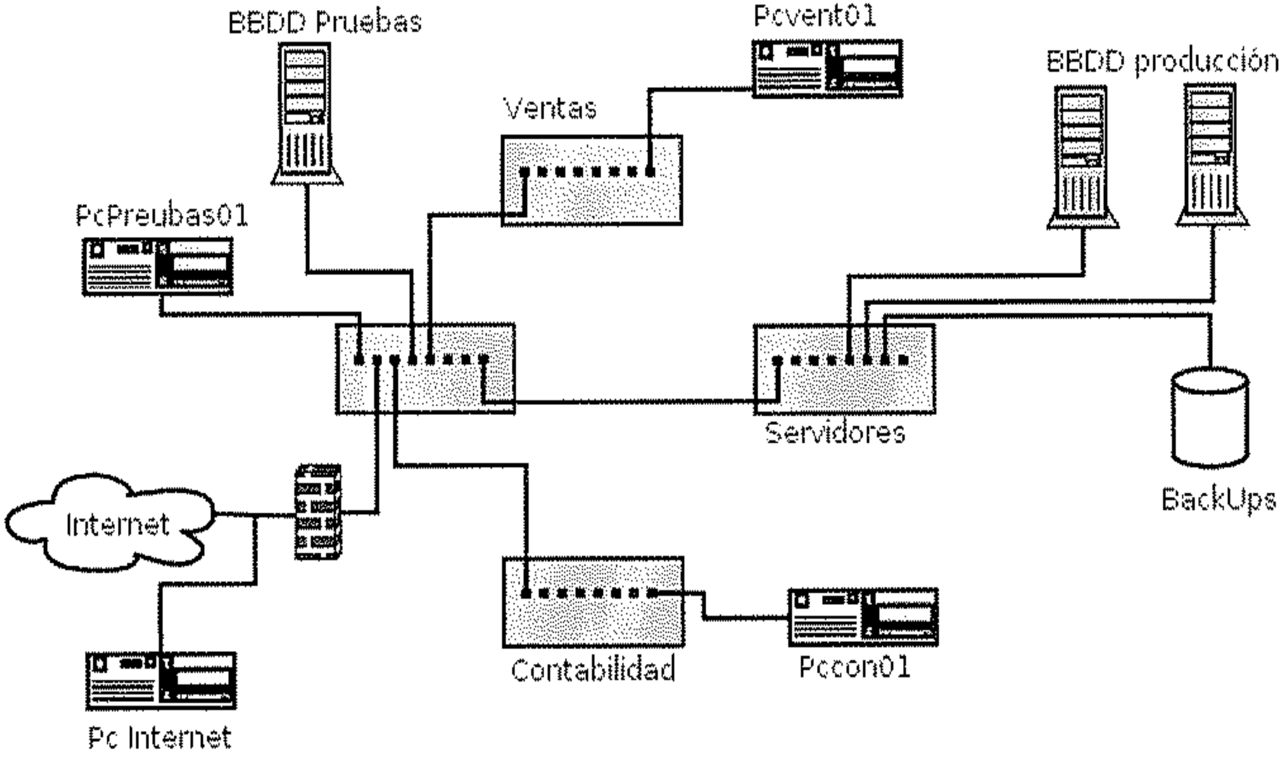
1. Análisis detallado.
2. Diseño físico del sistema (hardware).
3. Diseño lógico del sistema (software).
4. Revisión de las previsiones.

SUBPHASE	ENTRANCE	EXIT
1. AnalysisDetailed	Specificationfunctional and project plan	External specification.
2.Designsystem physicist	Specificationexternal	Physical design and hardware purchase proposal
3.Designsystem logic	Exsecificationexternal	Logical design and software purchase proposal
4.Reviewof the forecasts	Project plan	Revised project plan



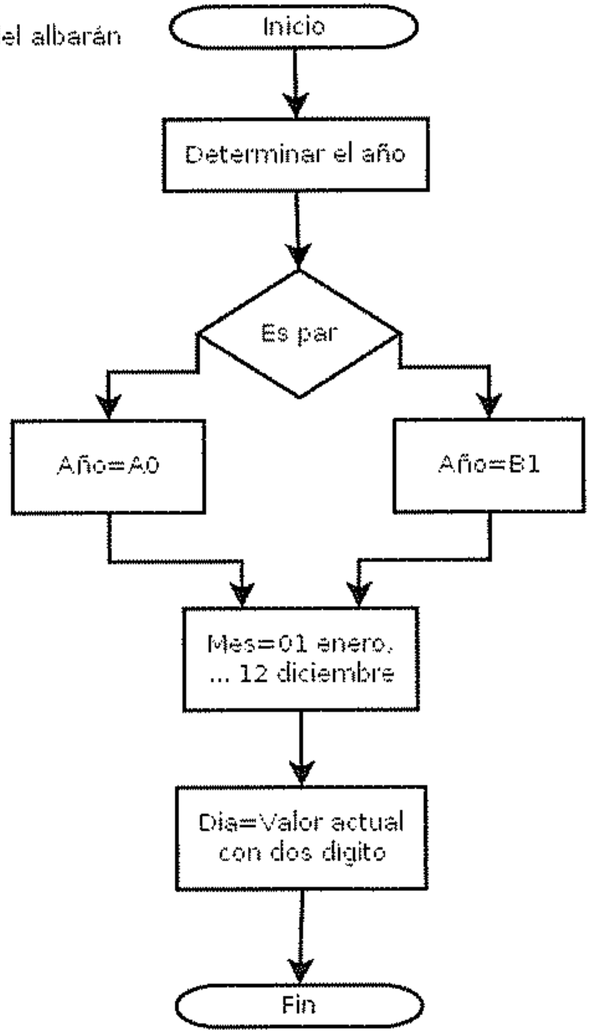
DFD resumen de la fase Dos

Ejemplos de DFDs



Diseño físico esquema general (subfase 2)

Cálculo del número del albarán
año par: A0MMDD
apo impar:B1MMDD



Inicio

Determinar el año

Es par

Año=A0

Año=B1

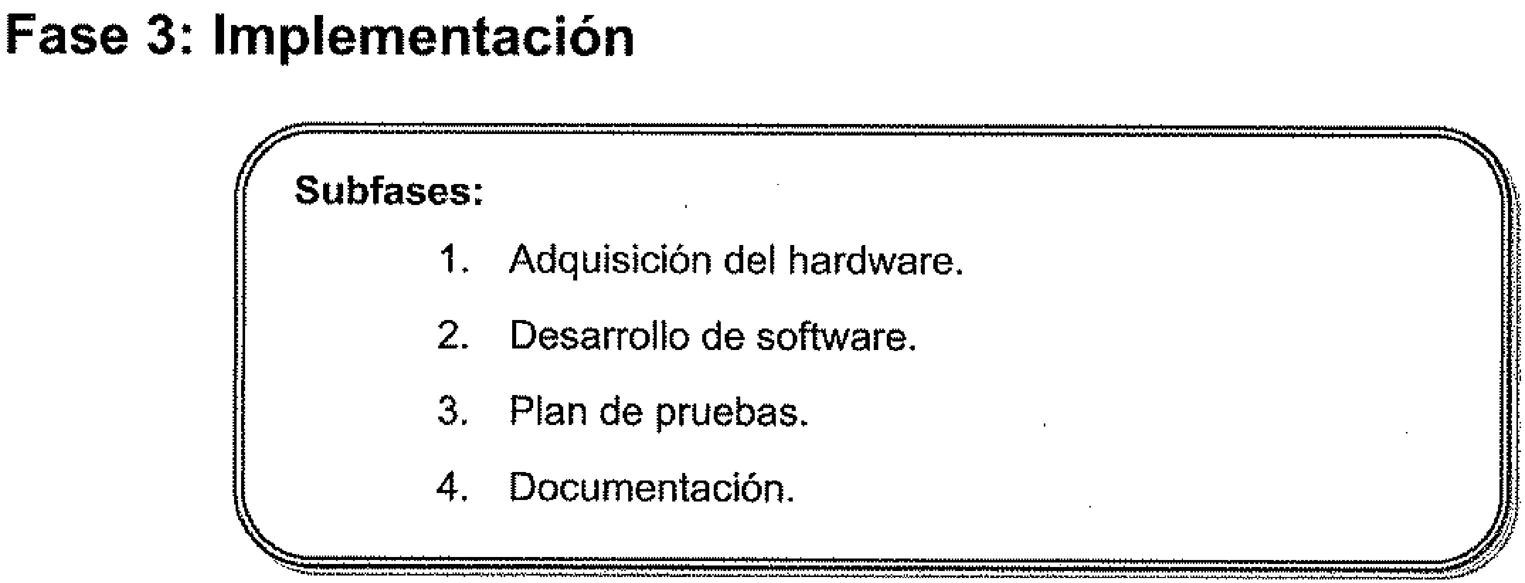
Mes=01 enero, ... 12 diciembre

Dia=Valor actual con dos digito

Fin

Cálculo del número del albarán (subfase 4)

ERP IMPLEMENTATION

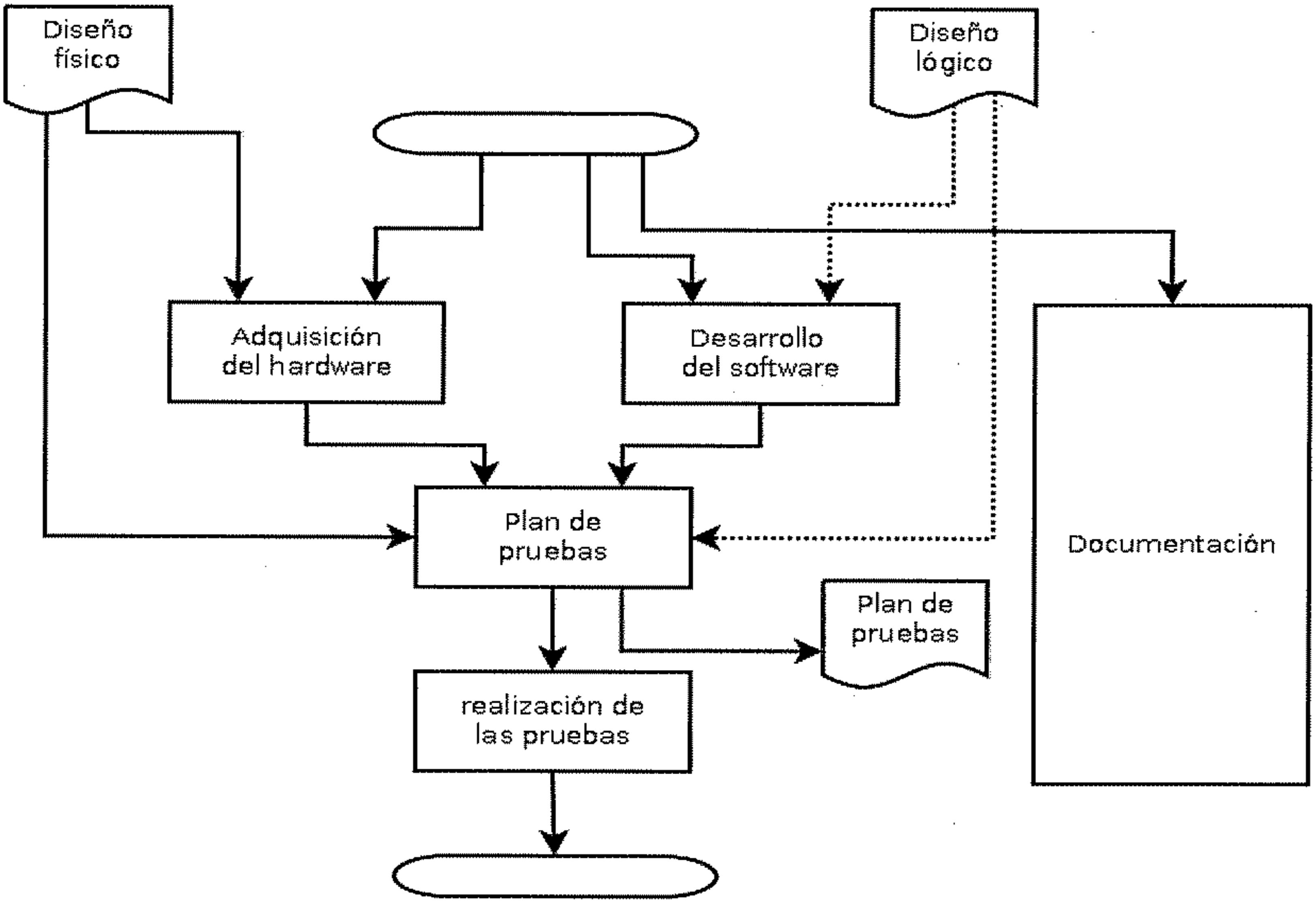


Fase 3: Implementación

Subfases:

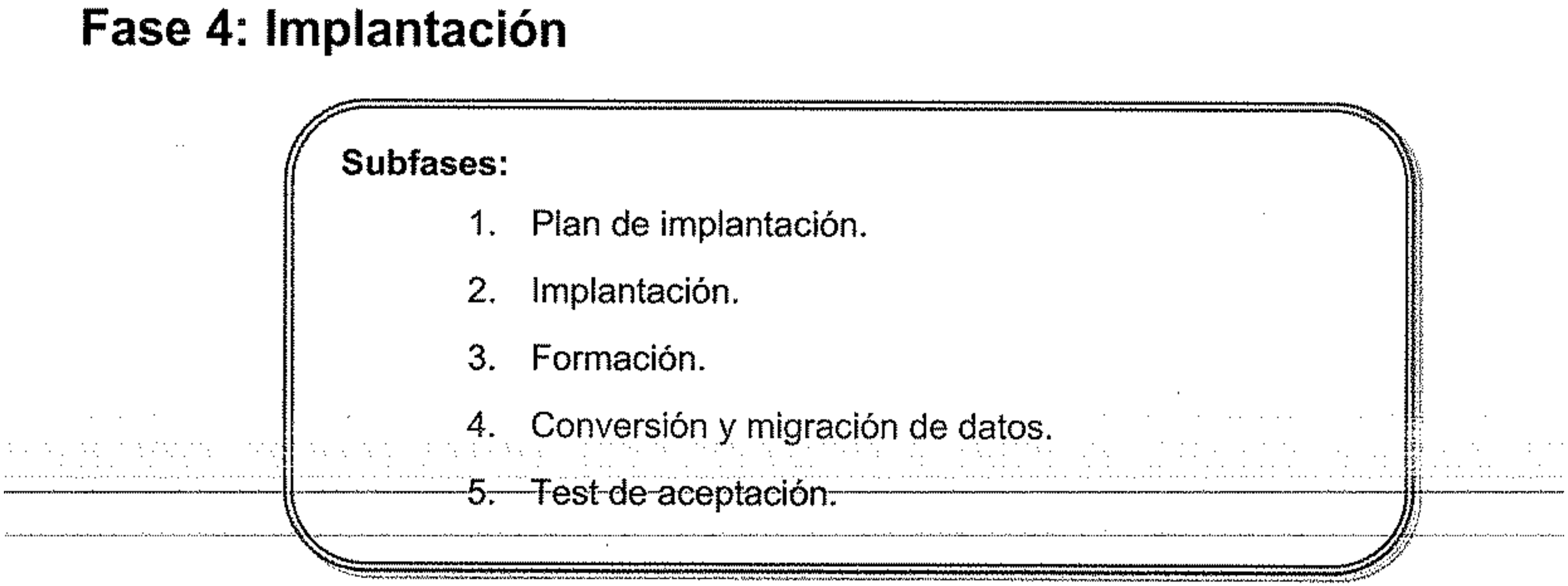
1. Adquisición del hardware.
2. Desarrollo de software.
3. Plan de pruebas.
4. Documentación.

SUBPHASE	ENTRANCE	EXIT
1.Hardware Acquisition.	Designphysical and purchase proposal	none
Software Development	Logical design and purchase proposal	none
Test plan	Physical design and logical design	Test plan andtested system document
Documentation.	Physical design, designlogical, planoftests,allprevious documentation	Technical documentationof the system



DFD resumen de la fase Tres

ERP IMPLEMENTATION

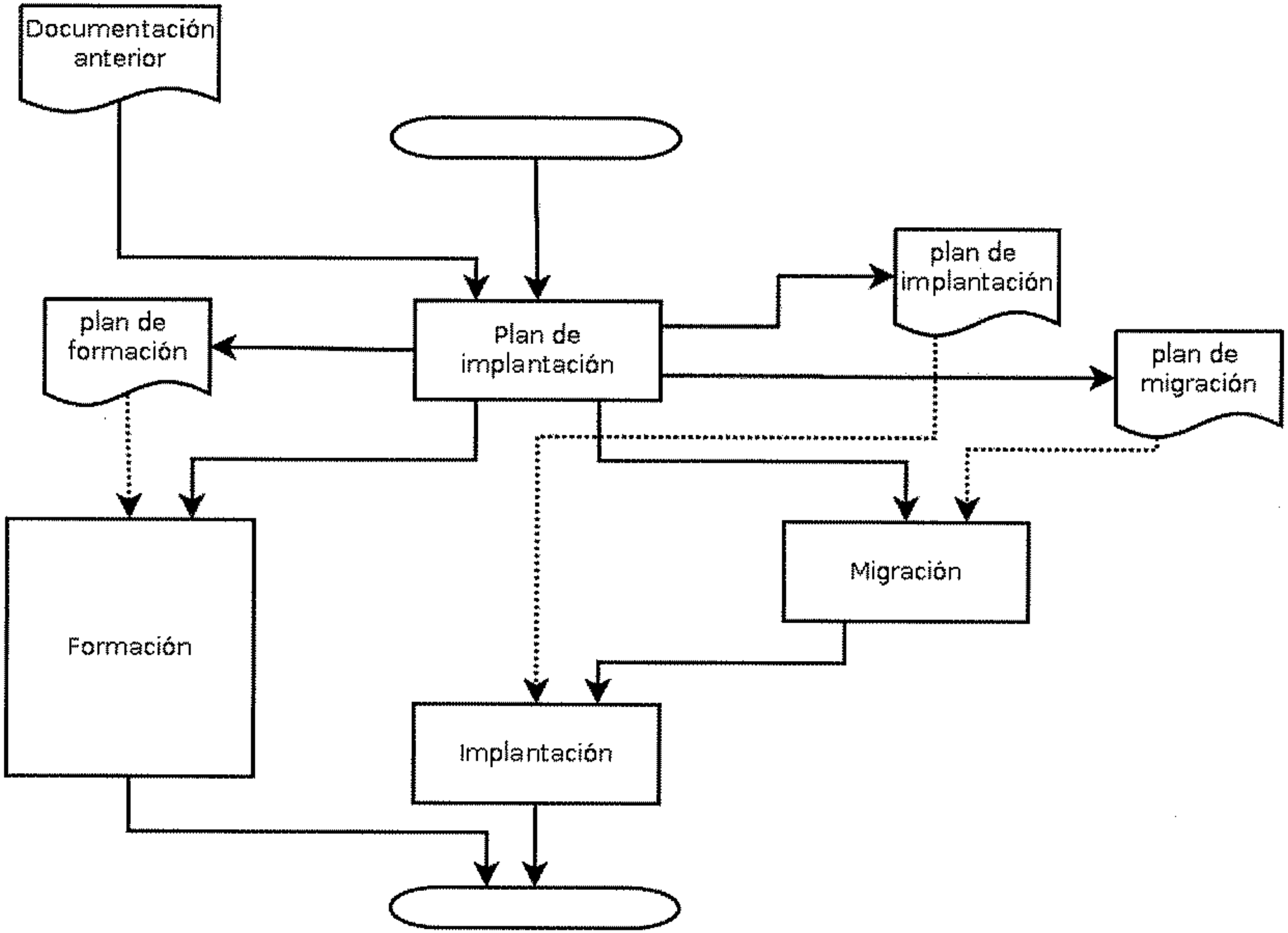


Fase 4: Implantación

Subfases:

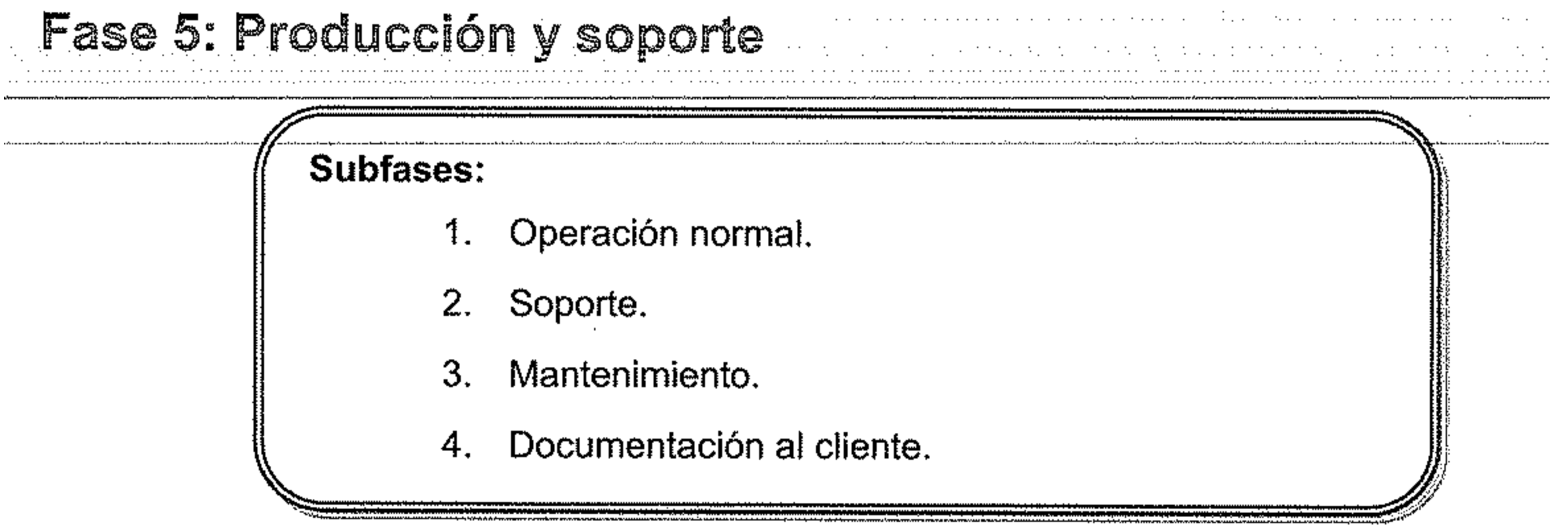
1. Plan de implantación.
2. Implantación.
3. Formación.
4. Conversión y migración de datos.
5. Test de aceptación.

SUBPHASE	ENTRANCE	EXIT
1. Implementation plan	Documentationformer	Planimplementation plan,training, planof migration and conversion.
2. Implementation	Planof implementation	none
3.Training	Plantraining	none
4.Conversionand data migration.	Plan ofmigration,conversion.	none
5.Acceptance test	Tests to be performed	System acceptance document



DFD resumen de la fase Cuatro

ERP IMPLEMENTATION



Fase 5: Producción y soporte

Subfases:

1. Operación normal.
2. Soporte.
3. Mantenimiento.
4. Documentación al cliente.

EXAMPLE

- SEE EXAMPLE DOCUMENT PHASE 1 (pages 195 to 199)