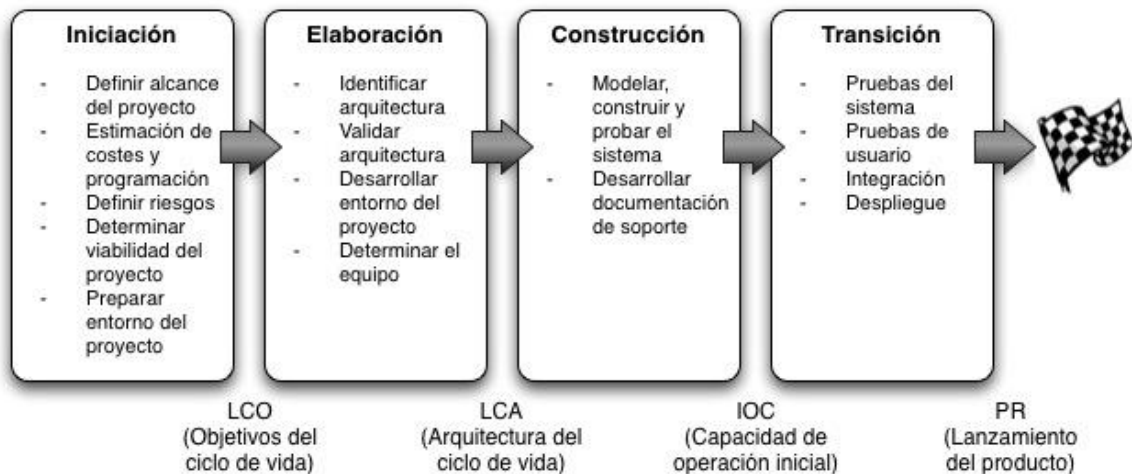


EXPOSICIÓN PARA EL MIÉRCOLES

- Manifestó Ágil..... Sierra
- Open UP..... Alejandro
- **Scrum..... Leslie**
- Extreme Programming..... Ricardo
- Kanban..... Leos
- Scrumban..... Edith
- **LEAN..... Víctor**
- Feature – driven Development... Carlos
- Test-Driven Development..... Kevin

DOCUMENTO



PASO #1

- Definir alcance del proyecto
- Objetivo general:
- Objetivos específicos

PASO #2

Costos

Hosting	Dominio	Materiales

PASO #3

Riesgos

- Falta de comunicación
- Falta de conocimiento de PHP
- Faltas de recursos económicos
- Falta de tiempo
- Falta de energía

PASO #4

Viable Factibilidad

- Técnica
- Operacional
- Económica
- Legal

Unidad 3:

Planificación del proyecto de software:

Estimación de tiempo	Estimación de costos
Gant Pert/CMP	<ul style="list-style-type: none">• decidir cuantas personas son necesarias• controlar el progreso del proyecto

Métodos para la estimación de costos

- Juicio experto
- Analogía
- Parkinson
- modelos algorítmicos

COCOMO

Modelo constructivo de costes (constructive cost model) fue desarrollado por B.W. Bohem a finales de los 70 y principios de los 80, exponiendo directamente su libro “software Engineering Economics”.

Tipos de proyectos que define COCOMO:

- **ORGÁNICO:** proyectos desarrollados en un ambiente familiar y estable. Requiere pocas innovaciones tecnológicas en lo que se refiere a algoritmos, estructuras de datos e integración del hardware
(50,000 LINEAS DE CODIGO)
 - Modelos de negocios
 - Modelos científicos
 - Sistemas operativos de pequeña escala
- **SEMI-ACOPLADO:**
 - sistema de control de producción
 - sistema de procesamiento de transacciones
 - administradores de bases de datos
- **EMPOTRADO:** incluye proyectos de gran envergadura que operan en un ambiente complejo con altas restricciones de hardware, software y procedimientos operacionales tales como los:
 - sistemas de tráfico aéreo

Modelos que define COCOMO:

- modelo básico
E = Esfuerzo (persona x mes)
T = Tiempo de duración del proyecto
P = Personas
 $E = a (KDLC)^b$
 $T = C*(E)^d$
 $P = E/T$

PROYECTO SOFTWARE	A	B	C	D
ORGANICO	3.2	1.05	2.5	0.38
SEMI-ACOPLADO	3.0	1.12	2.5	0.35

$KLDC = (\text{líneas de código} * pf) / 1000$

$(150 * 167) / 1000 = 25.05$

$3.2(25.05)^{1.05} = 94.16$

$2.5 * 94.16^{0.38} = 14.06$

$94.16 / 14.06 = 6.69$

- modelo intermedio

PRACTICA:

Estimar el costo de programación por los métodos de analogía, juicio experto y cocomo

Juicio experto: 11,000 porque?

Analogía: Se compara con un software parecido para saber el precio

Filancer.com

COCOMO:

Líneas de código $(150 \times 167) / 1000 = 25.05$

Esfuerzo $3.2(25.05)^{1.05} = 94.16$

Tiempo $2.5 \times 94.16^{.38} = 14.06$

Personas $94.16 / 14.06 = 6.69$

RIESGOS DEL PROYECTO DE SOFTWARE

Los objetivos de la gestión de riesgos son identificar los riesgos y problemas

Implica dos características

- Perdida
- Incertidumbre

Se puede hacer otra categorización de los riesgos en función de su facilidad de detección :

- Riesgos conocidos
- Riesgos predecibles
- Riesgos impredecibles

Gestión de riesgos:

Registro de riesgos(minimo 10)

riesgos	prioridad	probabilidad	impacto	Causa
No tener tiempo	alta	media	No entregar el software	Otros pendientes

Traer para instalar:

- **XAMPP**
- **Codeigniter**
- **Bootstrap**
- **Jquery**
- **Editor de texto**
- **MySql workbench**

Estudio de factibilidad

-factibilidad operacional

¿Está conforme toda la empresa con las soluciones que se van a obtener mediante el sistema?

Elaboración

Arquitectura de software:

-patrón por capas:

-Patrón cliente-servidor:

- Consiste en un servidor y múltiples clientes

-patrón modelo-vista-controlador

- Divide una aplicación interactiva en tres partes como:

-

En el software se va especificar que se va a utilizar una arquitectura **hibrida** puesto que lleva dos tipos de arquitectura **cliente-servidor, y vista-controlador**

Desarrollar el entorno del proyecto