



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
Facultad de Ingeniería

Universidad De San Carlos de Guatemala

Análisis y Diseño de Sistemas
Conceptos Principales de Sistemas (Tema 1)



Conceptos Generales

- Sistema
 - Conjunto de entidades u objetos relacionados entre si, conformando una estructura con un fin común
- Sistema de Información – SI
 - Sistema cuya finalidad es procesar datos de entrada para producir como salida información oportuna, relevante y confiable .

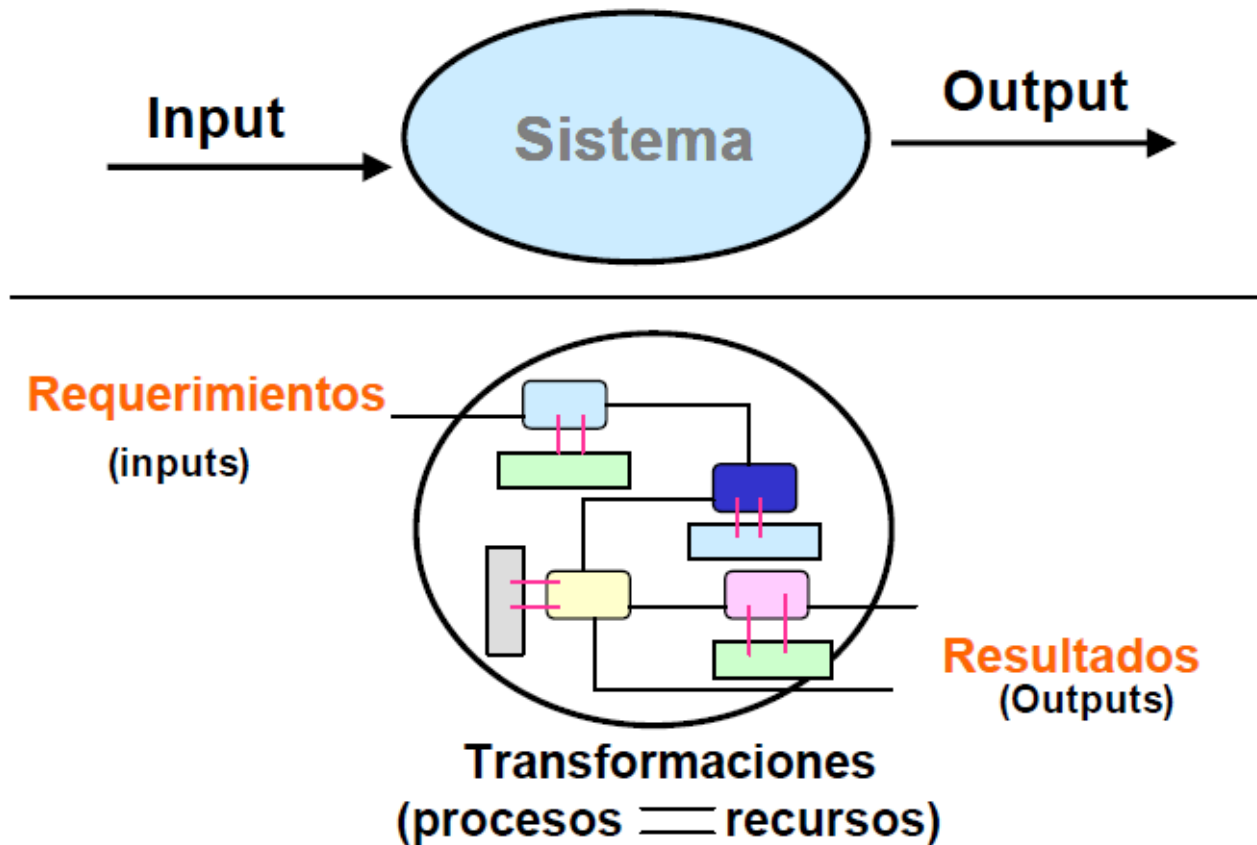


Conceptos Generales

- Análisis de SI
 - Análisis de las reglas del negocio (actividades) y de las necesidades de soporte de información.
- Diseño de SI
 - Especificación de los procedimientos de trabajo y de la plataforma tecnológica sobre la que se implementará el SI.



Esquema General de los Sistemas



Ejercicio (5 minutos)

- Mencione 5 ejemplos de sistemas mencionando:
 - Entrada
 - Proceso
 - Salida
 - Componentes principales
- Mencione 3 ejemplos de sistemas de información mencionando:
 - Entrada
 - Proceso
 - Salida
 - Componentes Principales



Descripción de los Sistemas de Información

- Alimentación : captura o entrada de datos
- Procesamiento : proceso para organizar los datos y realizado en una computadora
- Almacenamiento : disco o base de datos
- Comunicación : medio de transmisión de datos
- Salida : consultas o reportes impresos





Tipos de Sistemas de Información





División de los Sistemas

- Sistemas on-line
- Sistemas de tiempo real
- Sistemas de soporte a las decisiones–DSS
- Sistemas basados en el conocimiento



Datos

- Son los hechos que describen sucesos y entidades. "Datos" es una palabra en plural que se refiere a más de un hecho. A un hecho simple se le denomina "data-ítem" o elemento de dato.
- Los datos son comunicados por varios tipos de símbolos tales como las letras del alfabeto, números, movimientos de labios, puntos y rayas, señales con la mano, dibujos, etc.



Información

- Es una colección de hechos significativos y pertinentes, para el organismo u organización que los percibe.
- Es un conjunto de datos significativos y pertinentes que describan sucesos o entidades.



Datos e Información

- Los DATOS se transforman en INFORMACION al momento de ser presentada y revisada por los directivos, gerentes u operadores de la organización, quienes interpretan la forma de presentación de los datos y toman las decisiones de su responsabilidad.



Base de Datos

- Colección de datos organizados para dar servicio a muchas aplicaciones al mismo tiempo al combinar los datos de manera que parezcan estar en una sola ubicación.



Usuarios, Beneficiarios o Stakeholders

- Son aquellos que utilizan y se benefician directamente del sistema o para quienes se construye el sistema.
- Usuario es la persona a la que se tendrá que entrevistar, a menudo con gran detalle ("sonsacándoles" la información), a fin de conocer sus requerimientos para el nuevo sistema.
- También se les suele denominar clientes internos. Pueden clasificarse por su categoría de trabajo en: usuarios operacionales, usuarios supervisores y usuarios ejecutivos.



Administradores

- Son los que tienen que ver con la asignación de recursos al proyecto y su supervisión.
- Suelen conformar un comité directivo que velará por que el proyecto se desarrolle dentro de los márgenes y los requerimientos establecidos en la documentación aprobada.



Analista de Sistemas

- Es el papel o rol que jugará usted. Es responsable de determinar y especificar los requerimientos del sistema a construir desde los usuarios.
- A veces representa a un arqueólogo o a un mediador o a un innovador o a un jefe de proyecto.
- Esto significa que como analista de sistemas se necesita más que una habilidad para dibujar diagramas. Se requiere facilidad para el manejo de personas durante las entrevistas, mediar en desacuerdos comunes en todo proyecto, poseer conocimientos de computación y una mente lógica y organizada que le permita analizar los sistemas desde distintas perspectivas.



Diseñador de Sistemas (Arquitecto)

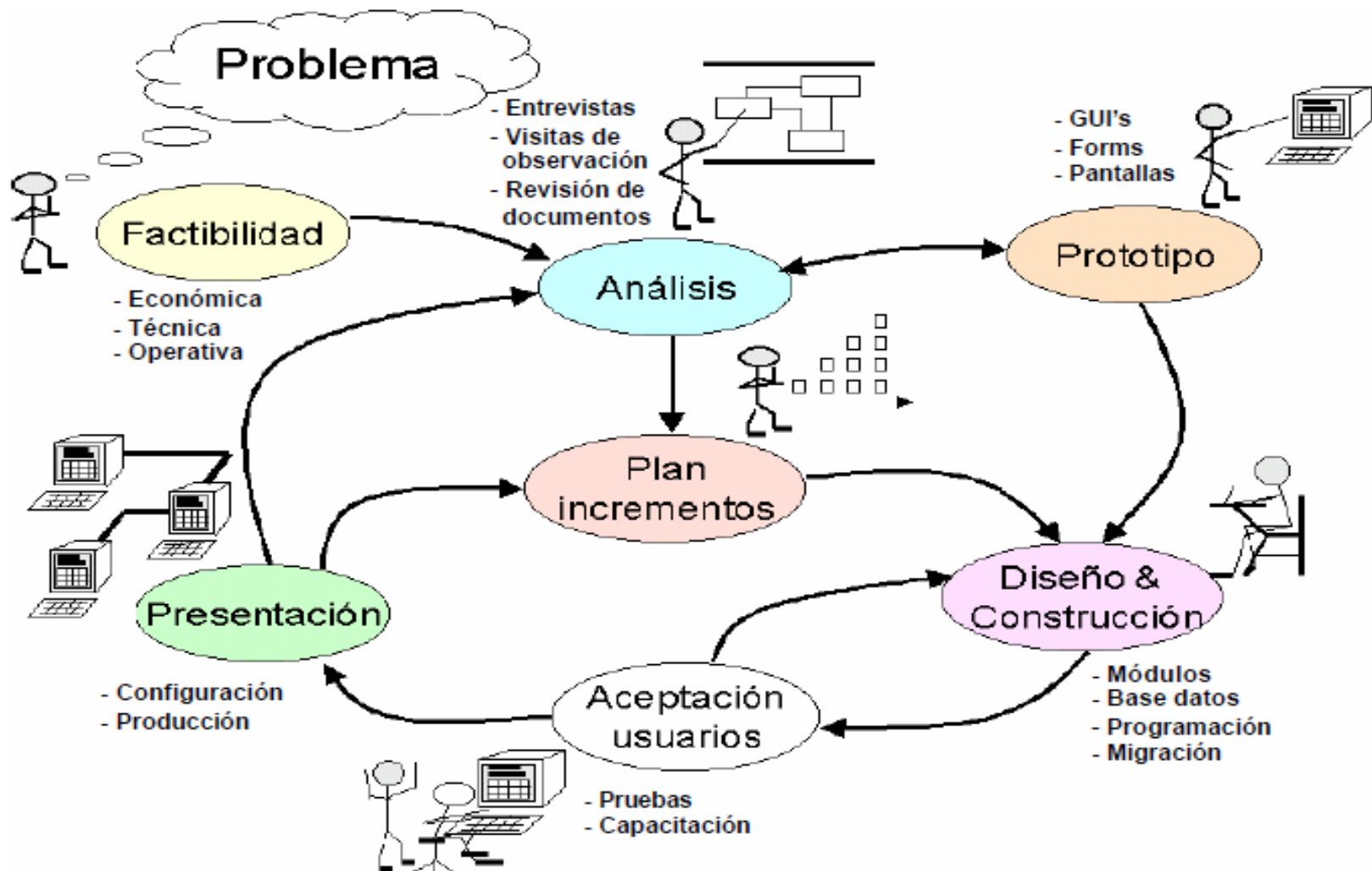
- Diseñador es el responsable de especificar las características de la arquitectura del sistema (n-capas, web, cliente servidor, etc) y que servirá de base para el trabajo de los programadores. En muchos casos, el analista y el diseñador son la misma persona.
- En las metodologías actuales se habla también del rol de arquitecto, el cual equivale al rol de diseñador



Programador

- Programador es la persona responsable de pasar a un lenguaje de programación de aplicaciones las características de diseño del sistema especificadas por el diseñador. A menudo es el que descubre errores y ambigüedades en la propuesta de requerimientos entregada por el analista.

Ejemplo del Proceso



Fuentes de Solicitudes de Información

- Existen cuatro fuentes principales de solicitudes de proyectos, estos solicitantes pueden ser:
 - Jefes de departamento.
 - Altos ejecutivos.
 - Analistas de sistemas.
 - Grupos externos.



Tarea

- Formar Grupos e Investigar cada grupo sobre las metodologías de software (Tema 2)
 - RUP
 - Ciclo de Vida Clásico
 - PSP
 - CMMI
 - Espiral
 - XP



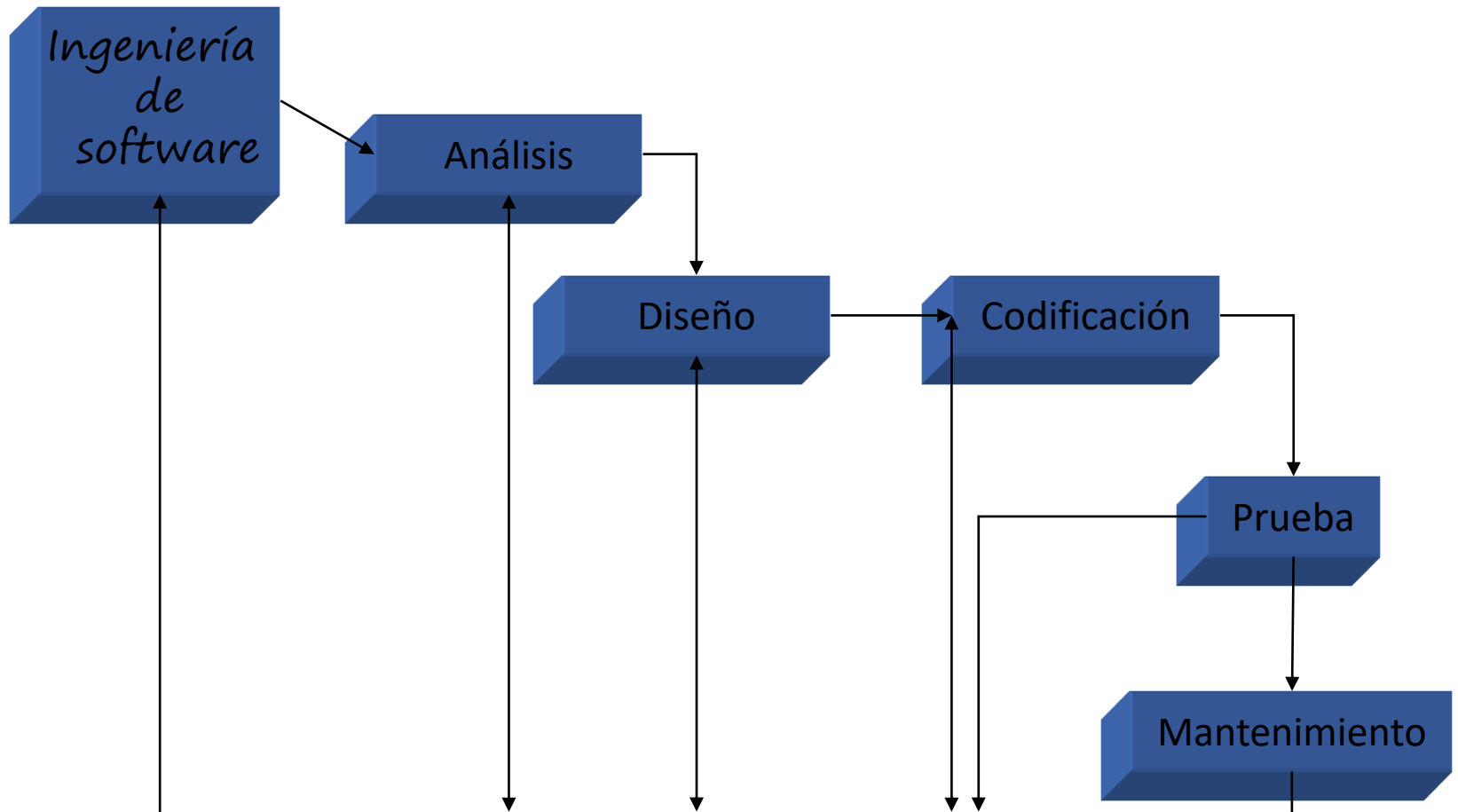
Modelo de ciclo de vida clásico o en cascada

Ciclo de vida clásico

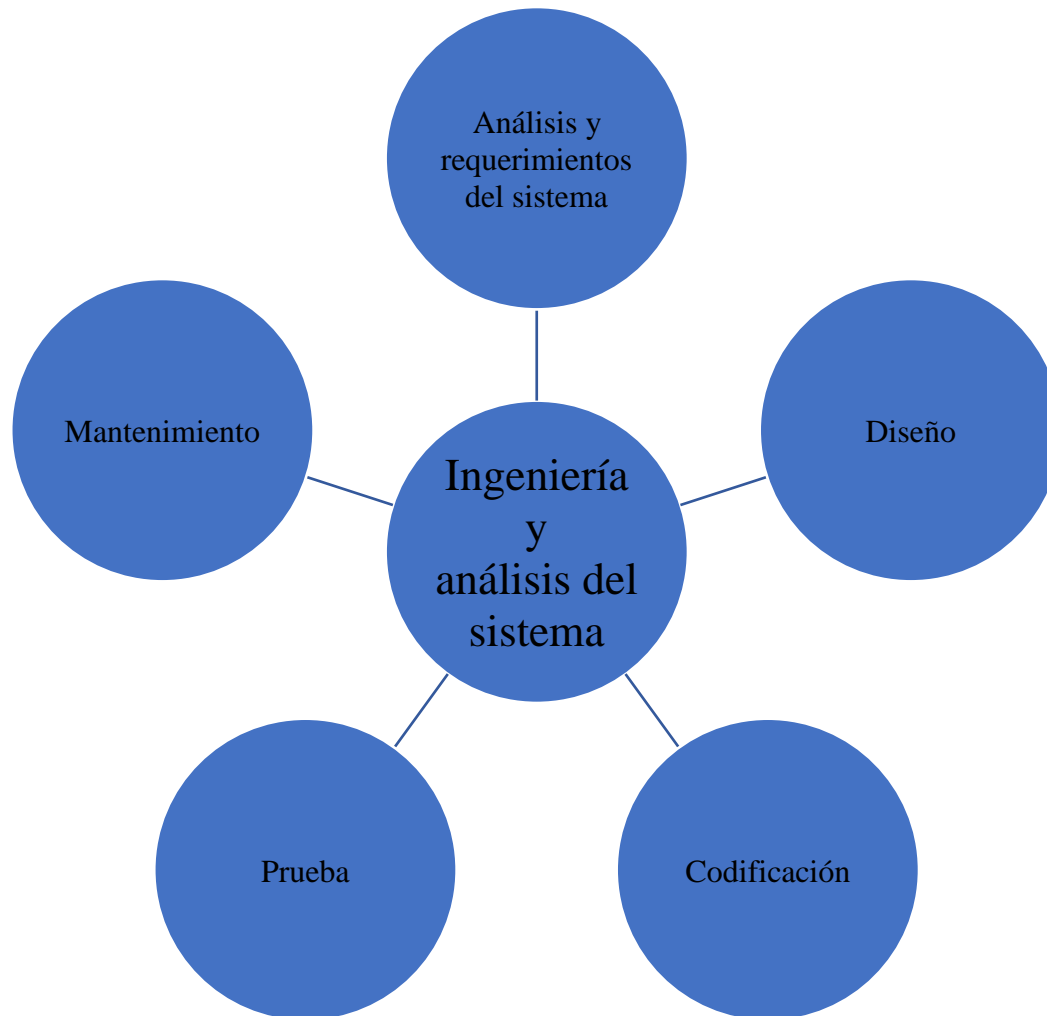
PARADIGMA CICLO DE VIDA DEL SOFTWARE

- Este fue el modelo inicial planteado para organizar el proceso de desarrollo, aunque antiguo, tiene vigencia en algunos proyectos o como parte de otros modelos, da la medida de los pasos tradicionales de cualquier modelo: análisis, diseño, codificación, prueba y mantenimiento.

PARADIGMA CICLO DE VIDA DEL SOFTWARE



Ciclo de vida clásico



Ingeniería y análisis del sistema



Es esencial cuando el software debe interrelacionarse con otros elementos tales como hardware, personas y bases de datos.

Ingeniería y análisis del sistema



La ingeniería y análisis del sistema abarcan los requerimientos globales a un nivel de sistema con una pequeña cantidad de análisis y diseño a nivel superior. Además de un análisis costo beneficio del sistema es decir si toda la inversión que se hará para el sistema conviene a los beneficios que traerá el mismo.



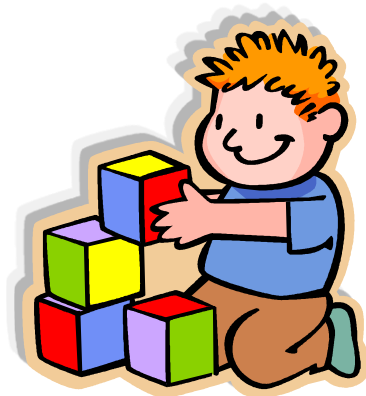
Análisis de los requerimientos del sistema

En esta etapa los
requerimientos del
sistema se documentan
y se analizan con el
cliente.

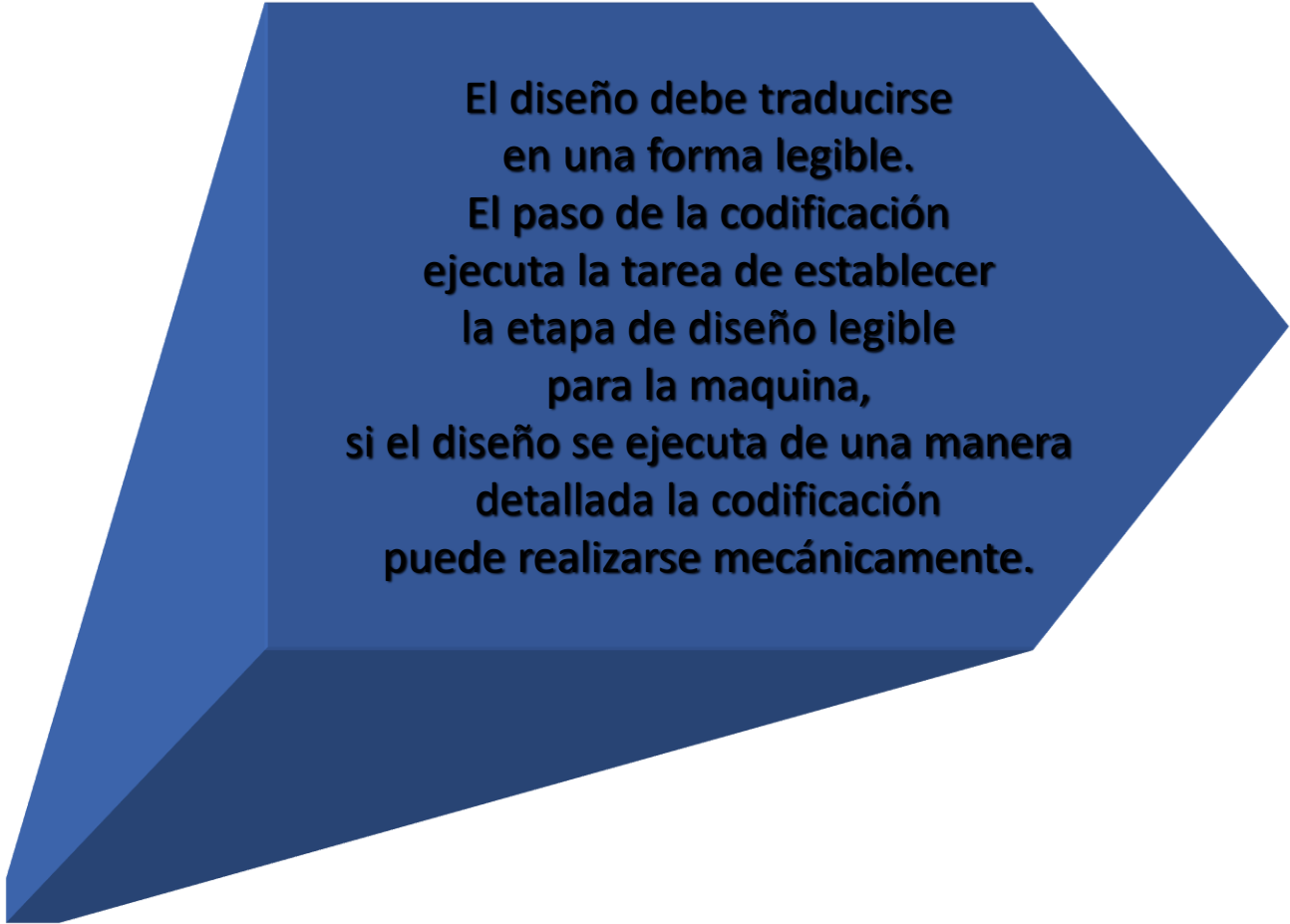


Diseño

- ♦estructura de datos
- ♦arquitectura de software
- ♦detalle procedimental

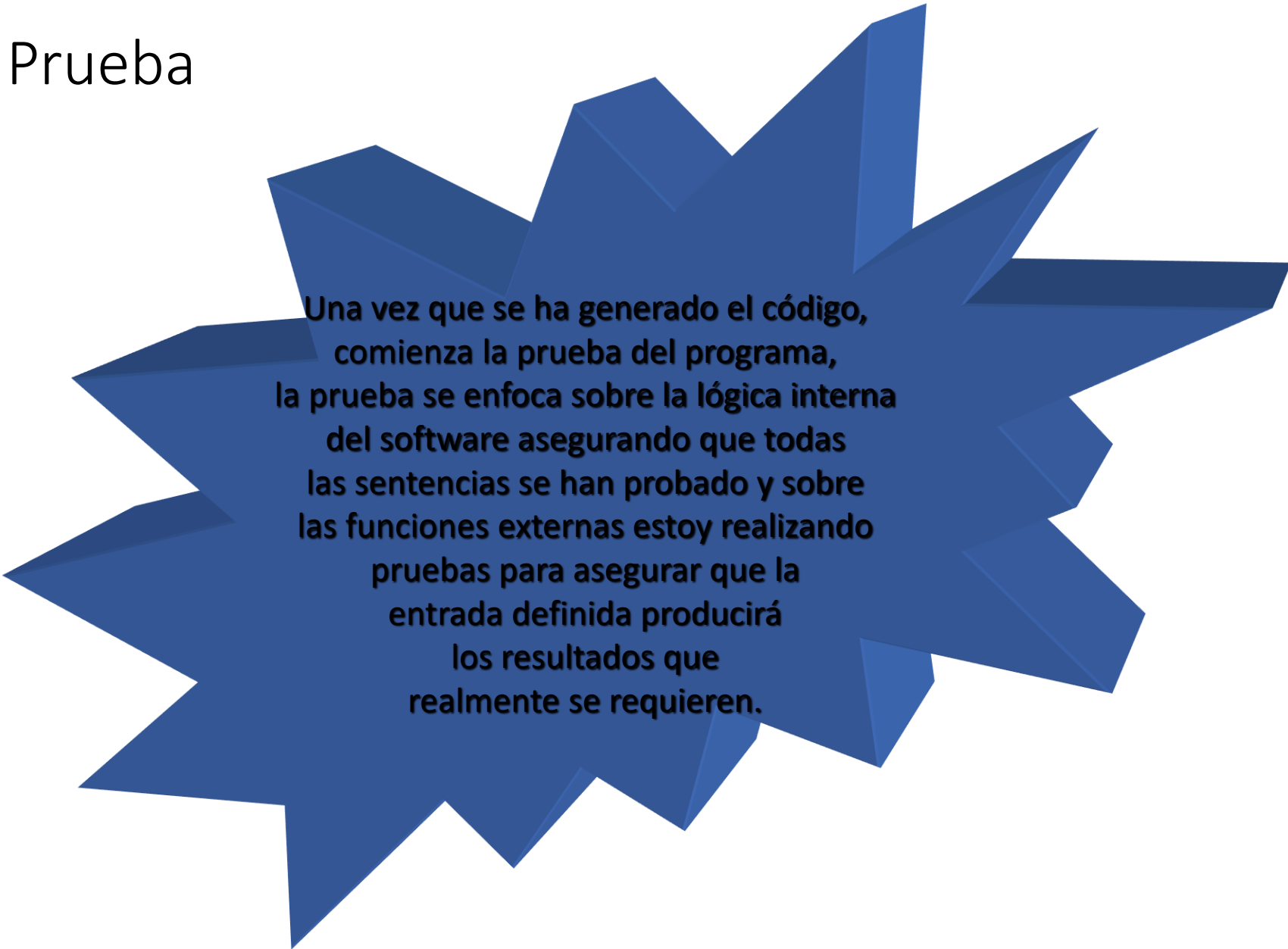


Codificación



**El diseño debe traducirse
en una forma legible.
El paso de la codificación
ejecuta la tarea de establecer
la etapa de diseño legible
para la maquina,
si el diseño se ejecuta de una manera
detallada la codificación
puede realizarse mecánicamente.**

Prueba



**Una vez que se ha generado el código,
comienza la prueba del programa,
la prueba se enfoca sobre la lógica interna
del software asegurando que todas
las sentencias se han probado y sobre
las funciones externas estoy realizando
pruebas para asegurar que la
entrada definida producirá
los resultados que
realmente se requieren.**

Mantenimiento

El mantenimiento del software se aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en lugar de uno nuevo.



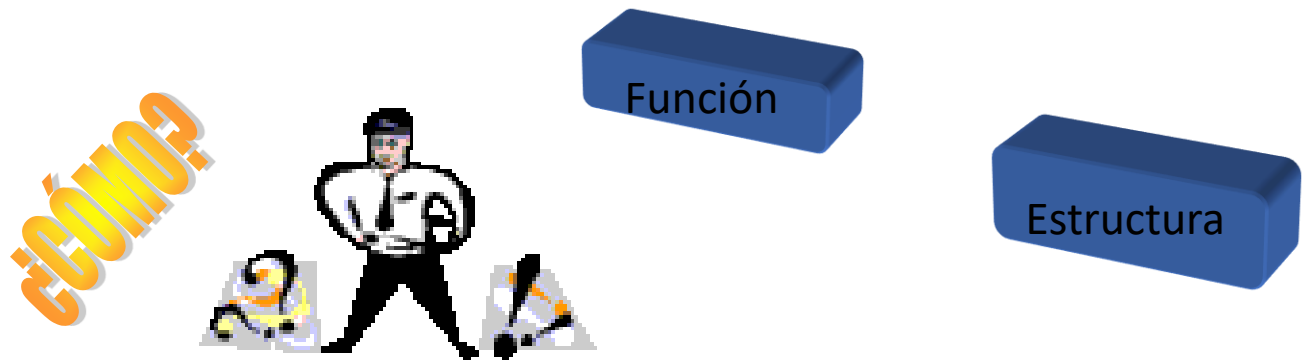


Ingeniería de requerimientos



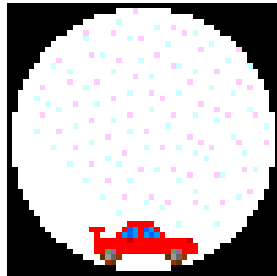
- ♦ Identificar y documentar los requerimientos exactos del sistema según las necesidades de los usuarios finales.
- ♦ 'Cualidades del sistema.
- ♦ 'Funcionales, no-funcionales, del proceso y del mantenimiento

Diseño - Especificación



- Dividir el sistema en partes y establecer las relaciones entre ellas.
- Arquitectura y diseño detallado.
- Establecer *qué* hará exactamente cada parte.
- En esta fase se crea un modelo funcional – estructural de los requerimientos.
- El diseño debe permitir implementaciones que verifiquen los requerimientos.

Verificación



- **Comprobar que los distintos productos del ciclo de vida del software verifican las propiedades y funciones establecidas en los requerimientos.**
- **Testing vs. análisis**
- **Testing funcional y estructural**
- **Ejecución simbólica**
- **Verificación & Validación**

Universidad de San Carlos de Guatemala
Facultad de Ingeniería en Sistemas
Análisis de Sistemas

Construcción de prototipos de software

- Por lo general los clientes y usuarios finales del software encuentran muy difícil expresar sus requerimientos reales.
- Es casi imposible predecir la manera en que un sistema afectará el trabajo diario, como interactuará con otros sistemas y qué operaciones del usuario se deberían automatizar.

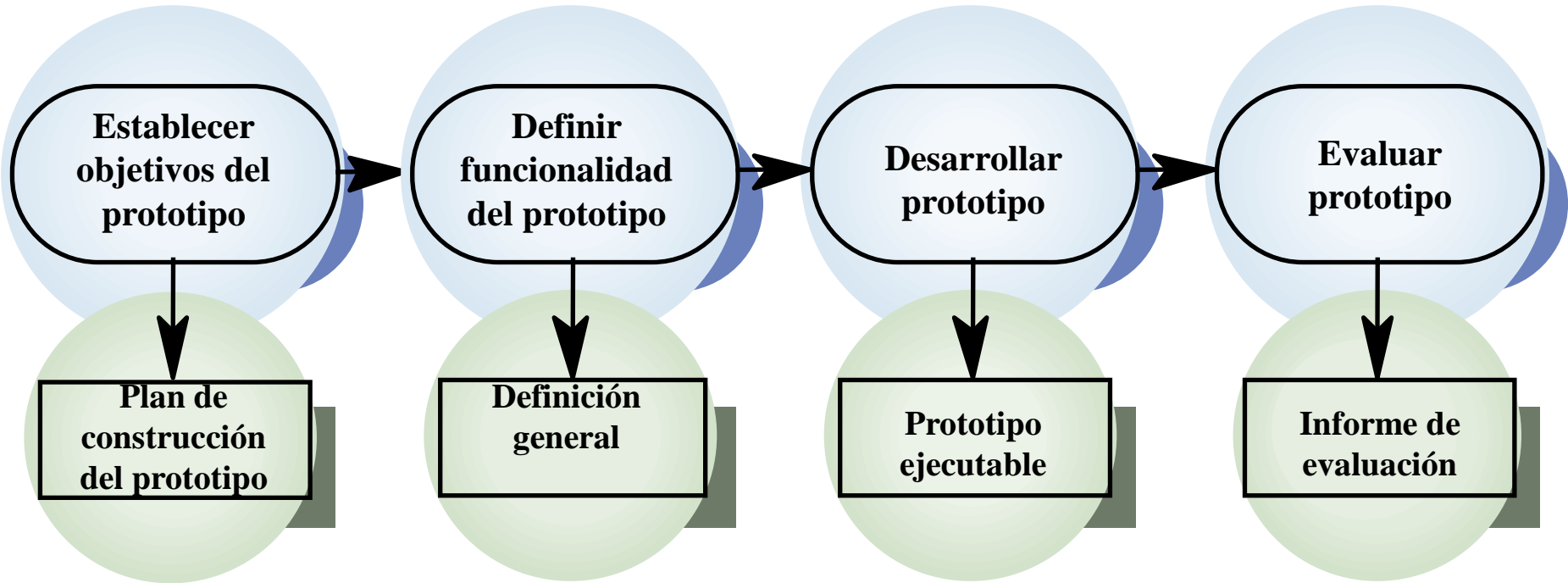
Construcción de prototipos de software

- Sin embargo es posible probar el sistema si está disponible un prototipo de él.
- Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el “problema” y su solución.

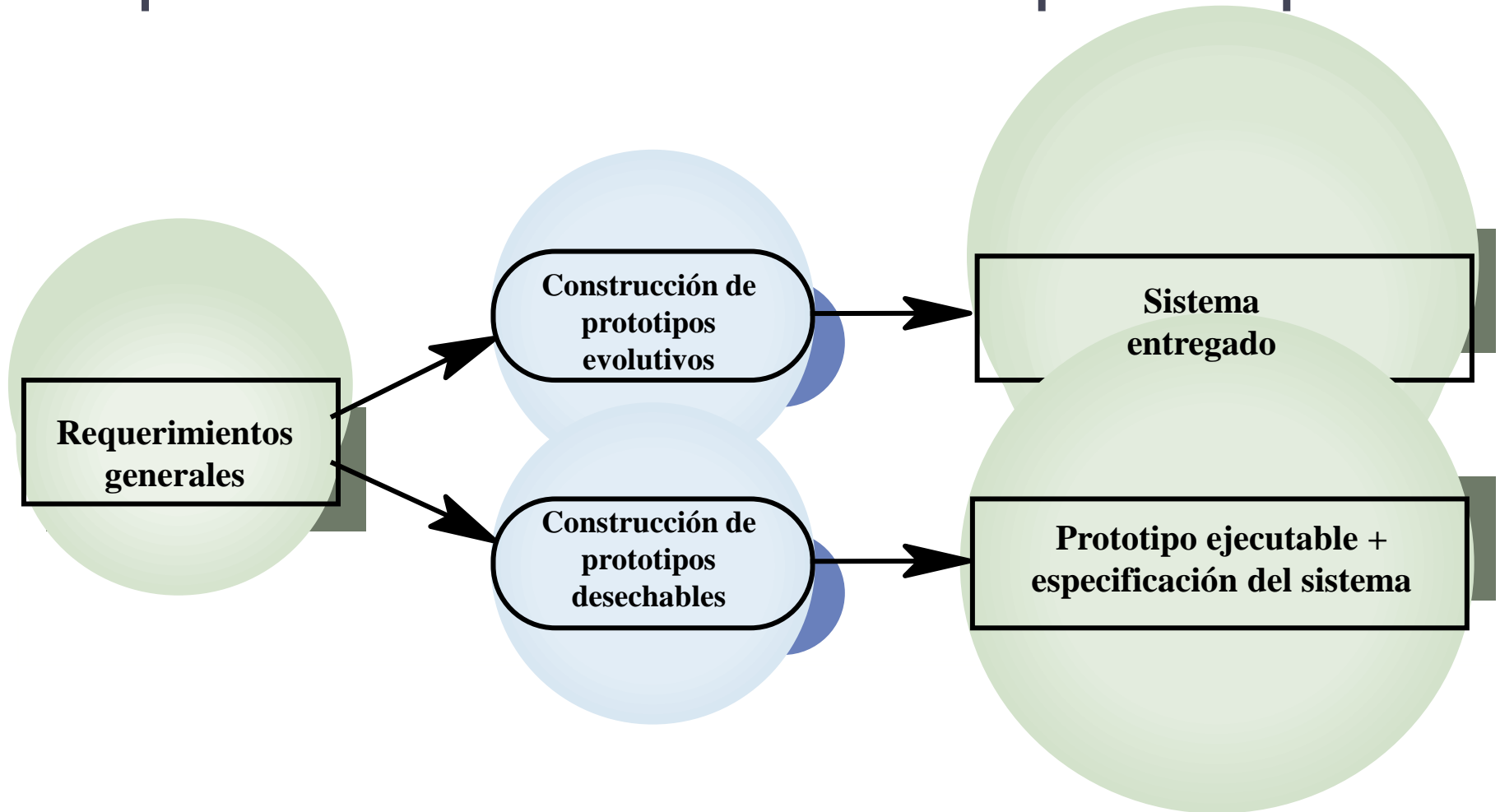
Construcción de prototipos de software

- Un prototipo de software apoya dos actividades del proceso de ingeniería de requerimientos:
 1. Obtención de requerimientos: los usuarios experimentan como el sistema ayudará su trabajo.
 2. Validación de requerimientos: el prototipo puede revelar errores u omisiones en los requerimientos propuestos.

El proceso de desarrollo de prototipos



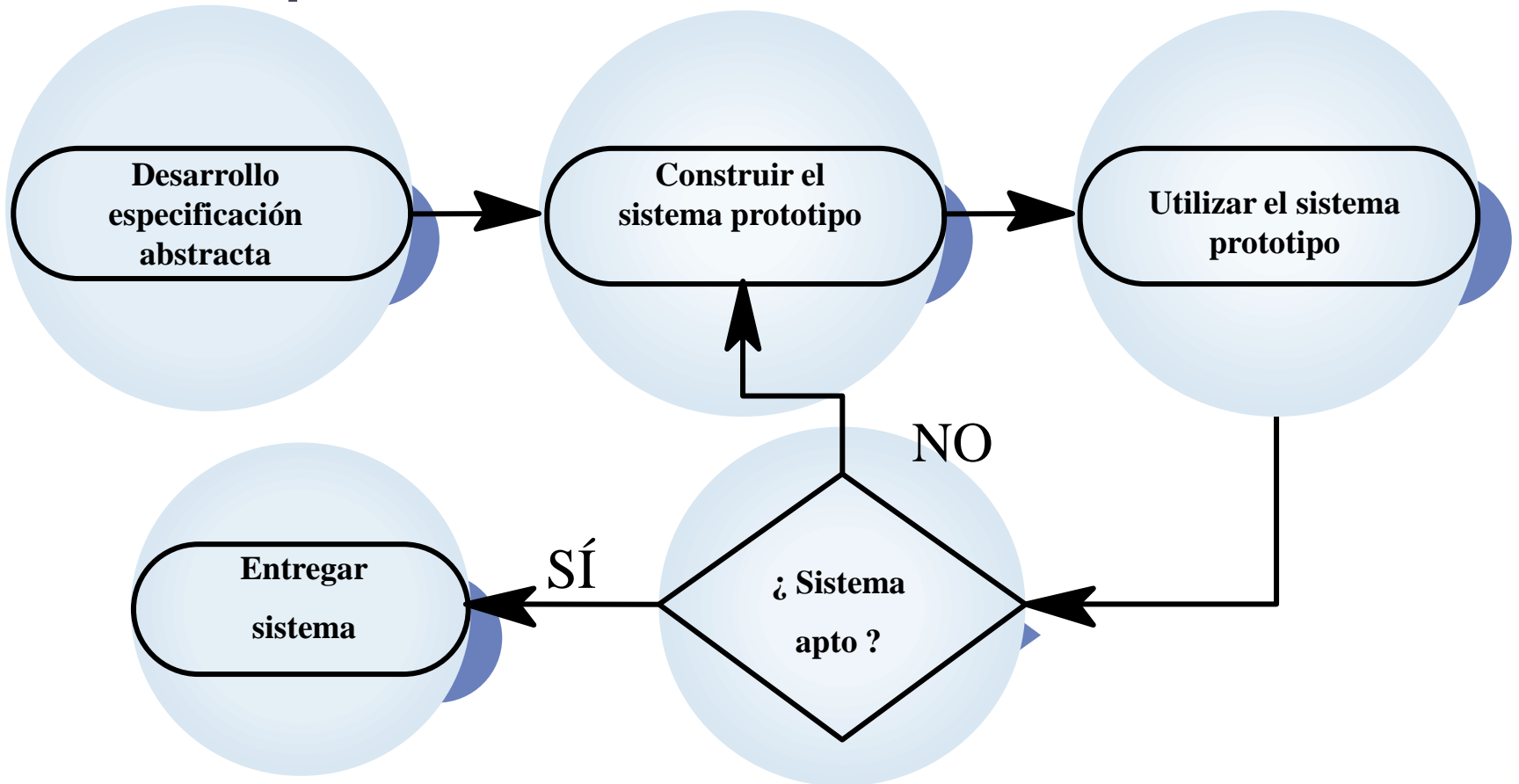
El proceso de desarrollo de prototipos



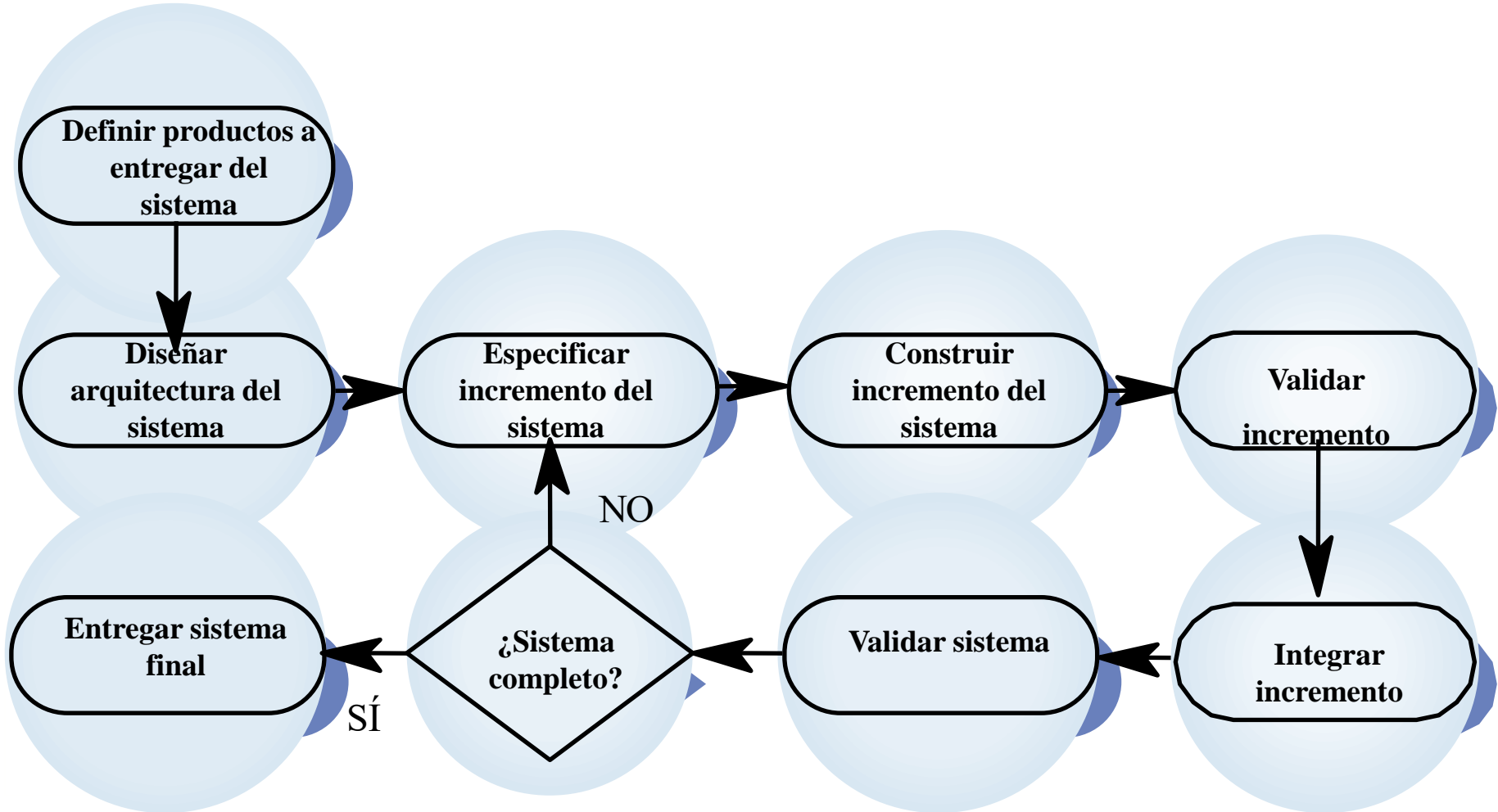
El proceso de desarrollo de prototipos

- Existe gran diferencia entre los dos tipos de prototipos:
 1. El prototipo evolutivo entrega a los usuarios finales un sistema funcionando. Se usa con los requerimientos que mejor se comprenden.
 2. El prototipo desechable valida o deriva los requerimientos del sistema. Se usa con los requerimientos que no se conocen bien.
Período de vida corto.

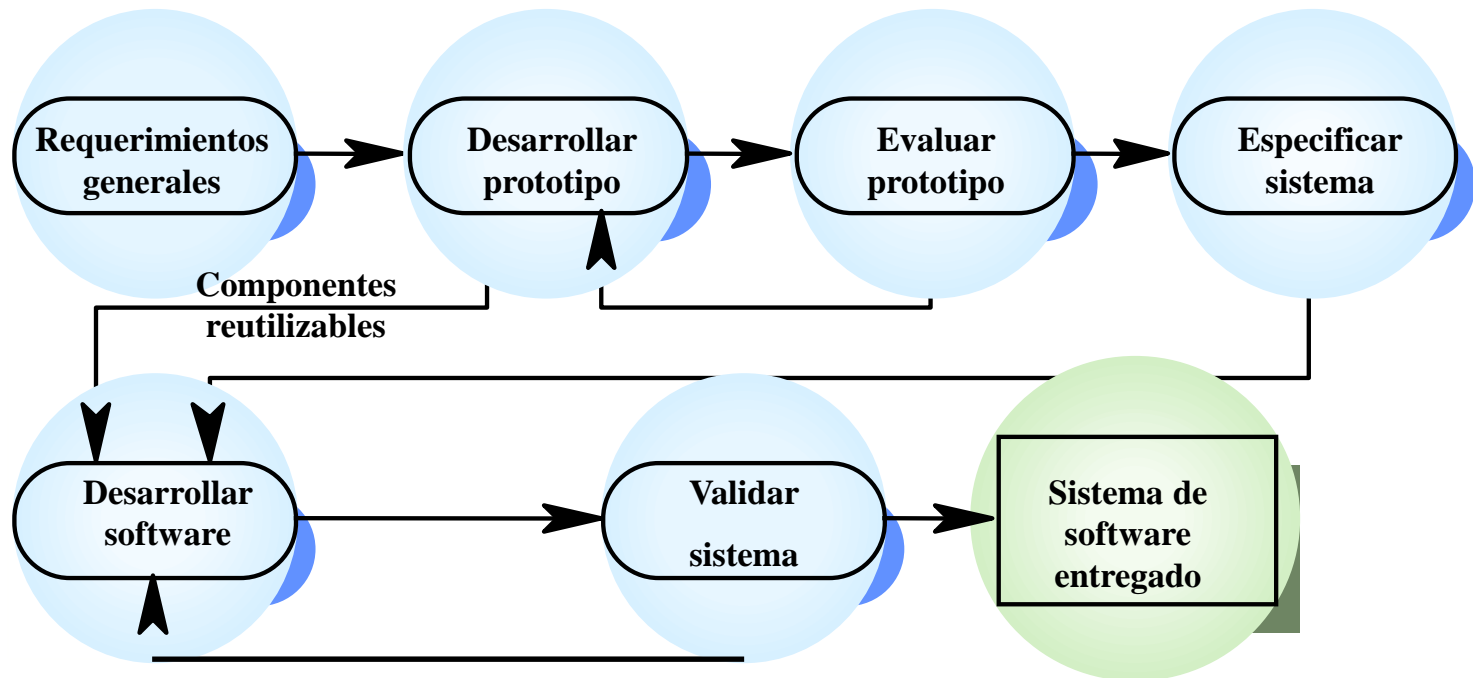
Prototipo evolutivo



Proceso de desarrollo incremental



Prototipos desechables



EXtreme Programming

Universidad de San Carlos
Ingeniería en Sistemas
Análisis de Sistemas

Historia

- Nació en 1996.
- Ken Beck fundador de la metodología de XP.
- Proyecto C3 de DaimlerChrysler.
- Identificó que era lo simple y lo difícil al momento de programar.

¿Qué es Extreme Programming

- ¿En qué consiste XP?
- La Programación Extrema es una metodología ligera de desarrollo de software que se basa en:
 - la simplicidad,
 - la comunicación
 - la retroalimentación
 - reutilización del código desarrollado

Objetivos perseguidos

- Obtención del producto, software funcionando, y con la satisfacción del cliente.
- Minimización del riesgo actuando sobre:
 - Variables del proyecto :
 - Coste
 - Tiempo
 - Calidad
 - Alcance

Valores que inspiran XP

SIMPLICIDAD

FEEDBACK

CORAJE

COMUNICACIÓN

Comunicación

XP pone en comunicación directa y continua a clientes y desarrolladores. El cliente se integra en el equipo para establecer prioridades y resolver dudas. De esta forma ve el avance día a día, y es posible ajustar la agenda y las funcionalidades de forma consecuente

Feedback rápido y continuo

Una metodología basada en el desarrollo incremental de pequeñas partes, con entregas y pruebas frecuentes y continuas, proporciona un flujo de retro-información valioso para detectar los problemas o desviaciones.

- De esta forma fallos se localizan muy pronto.
- La planificación no puede evitar algunos errores, que sólo se evidencian al desarrollar el sistema.
- La retro-información es la herramienta que permite reajustar la agenda y los planes.

Simplicidad

La simplicidad consiste en desarrollar sólo el sistema que realmente se necesita. Implica resolver en cada momento sólo las necesidades actuales.

Los costes y la complejidad de predecir el futuro son muy elevados, y la mejor forma de acertar es esperar al futuro.

Con este principio de simplicidad, junto con la comunicación y el feedback resulta más fácil conocer las necesidades reales

Coraje

- Disciplina en la aplicación de XP
- Parar cuando se está cansado
- Permitir que el usuario tome las decisiones de negocio
- Permitir que el desarrollador tome las decisiones técnicas
- Descartar código si es necesario
- Introducir cambios cuando las cosas no funcionan

Roles de personas en XP

- **Programador**

- Pieza básica en desarrollos XP
- Más responsabilidad que en otros modos de desarrollo
- Responsable sobre el código
- Responsable sobre el diseño (refactorización, simplicidad)
- Responsable sobre la integridad del sistema (pruebas)
- Capacidad de comunicación (pair-programming)
- Acepta críticas (código colectivo)

- **Cliente**

- Pieza básica en desarrollos XP
- Define especificaciones (user stories)
- Influye sin controlar
- Confía en el grupo de desarrollo
- Define pruebas funcionales

- **Encargado de Pruebas**

- Apoya al cliente en la preparación/realización de las pruebas funcionales
- Ejecuta las pruebas funcionales y publica los resultados

- **Encargado de Seguimiento(Tracker)**
 - Recoge, analiza y publica información sobre la marcha del proyecto sin afectar demasiado el proceso
 - Supervisa el cumplimiento de la estimaciones en cada iteración
 - Informa sobre la marcha de la iteración en curso
 - Controla la marcha de las pruebas funcionales, de los errores reportados, de las responsabilidades aceptadas y de las prueba añadidas por los errores encontrados
- **Entrenador (Coach)**
 - Experto en XP
 - Responsable del proceso en su conjunto
 - Identifica las desviaciones y reclama atención sobre las mismas
 - Guía al grupo de forma indirecta (sin dañar su seguridad ni confianza)
 - Interviene directamente si es necesario
 - Atajar rápidamente el problema
- **Consultor**
 - Apoya al equipo XP en cuestiones puntuales
- **Jefe del Proyecto**
 - Favorece la relación entre usuarios y desarrolladores
 - Confía en el equipo XP
 - Cubre las necesidades del equipo XP
 - Asegura que alcanza sus objetivos

Captura de Requisitos en XP

- **Historias del Usuario (*User-Stories*)**
 - Establecen los requisitos del cliente
 - Trozos de funcionalidad que aportan valor
 - Se les asignan tareas de programación con un número de horas de desarrollo
 - Las establece el cliente
 - Son la base para las pruebas funcionales

Captura de Requisitos en XP

Una ficha de User-Story

Customer Story and Task Card Blw Development / COLA

DATE: 3/19/98 TYPE OF ACTIVITY: NEW: X FIX: ENHANCE: FUNC. TEST:

STORY NUMBER: ~~1275~~ 1275 PRIORITY: USER: TECH:

PRIOR REFERENCE: RISK: TECH ESTIMATE:

TASK DESCRIPTION:
 SPLIT COLA: When the COLA rate chgs in the middle of the Blw Pay Period, we will want to pay the 1st week of the pay period at the OLD COLA rate and the 2nd week of the Pay Period at the NEW COLA rate. Should occur automatically based on system design.

NOTES:
 For the OT, we will run a m/f frame program that will pay or calc the COLA on the 2nd week of OT. The plant currently retransmits the hours data for the 2nd week exclusively so that we can calc COLA. This will come into the Model as a "2744" COLA.

TASK TRACKING: Gross Pay Adjustment. Create RM Boundary and Place in DE Ent Excess COLA

Date	Status	To Do	Comments

FIGURE 6. A story card

Planificación en XP

- Planificación por entregas (*releases*)
- Se priorizan aquellas user-stories que el cliente selecciona porque son más importantes para el negocio
- Entregas:
 - Son lo más pequeñas posibles
 - Se dividen en iteraciones (iteración = 2 o 3 semanas)
 - Están compuestas por historias
- A cada programador se le asigna una tarea de la user-story

Programación en XP

- La programación de tareas se realiza por parejas
- La pareja diseña, prueba, implementa e integra el código de la tarea
- Código dirigido por las pruebas
- Código modular, intentando refactorizar siempre que se pueda

Programación en XP

Una ficha de Tarea

Engineering Task Card			
DATE: <u>3/17/98</u>		BIN <u>Smalltalk/Future</u> Based on Conversation w/REBAMA NEW	
STORY NUMBER: <u>X923</u>		SOFTWARE ENGINEER: _____	TASK ESTIMATE: _____
TASK DESCRIPTION: Composite Bin - Regular Base Needs to Be Displayed on GUI. We have the hidden bin for Regular Base (last time) to display <u>NOT</u> the auto gen bin but the BIN that composites the Auto Pay: the <u>Last Time</u> . There is SOFTWARE ENGINEER'S NOTES: a separate composite bin started that needs to be completed??			
TASK TRACKING:			
Date	Done	To Do	Comments

FIGURE 7. A task card

➤ **Tarjetas CRC** (Clase - Responsabilidad – Colaborador).

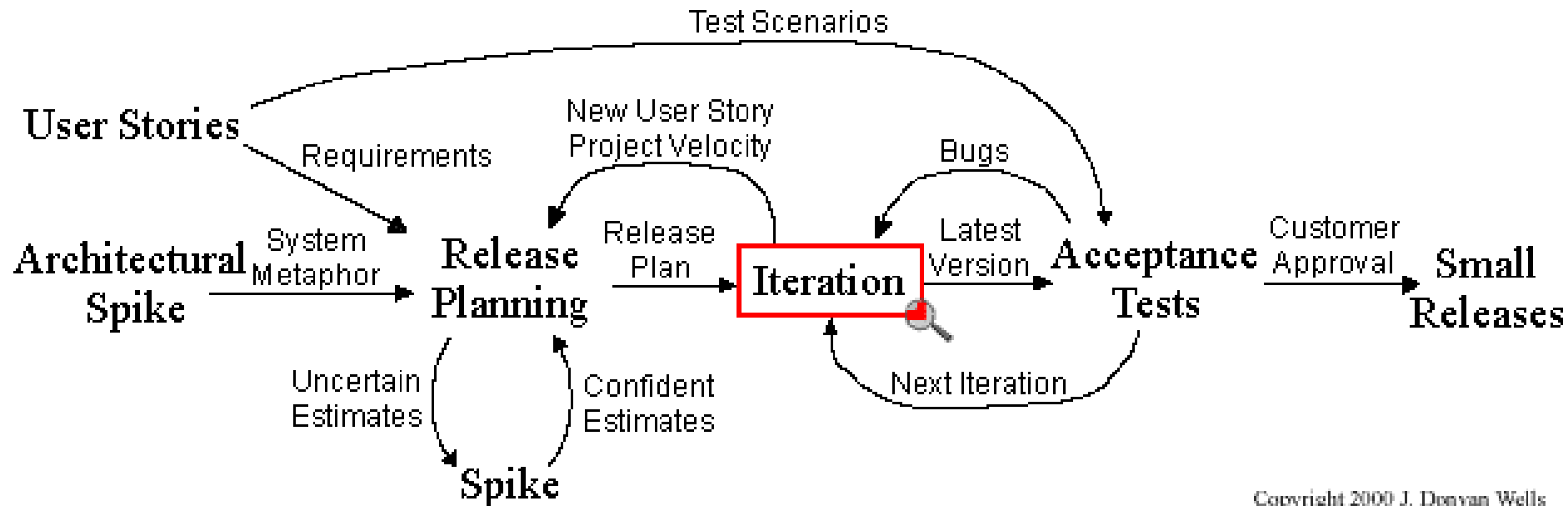
Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realiza, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

Nombre de la clase.	
Responsabilidades	Colaboradores

Modelo de un Proyecto XP

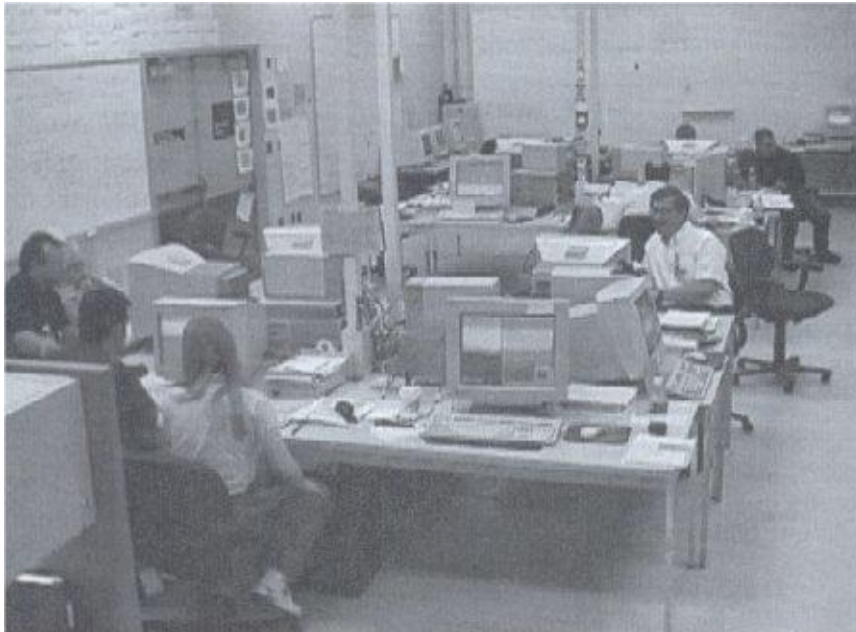


Extreme Programming Project



Espacio de trabajo XP

- Espacio abierto
- Mesas centrales
- Cubículos en el espacio exterior



Espacio de trabajo
del proyecto C3 de
DaimlerChrysler

Ventajas y desventajas de Extreme Programming

Ventajas:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

Desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.

Prácticas de XP

PRÁCTICAS DE CODIFICACIÓN

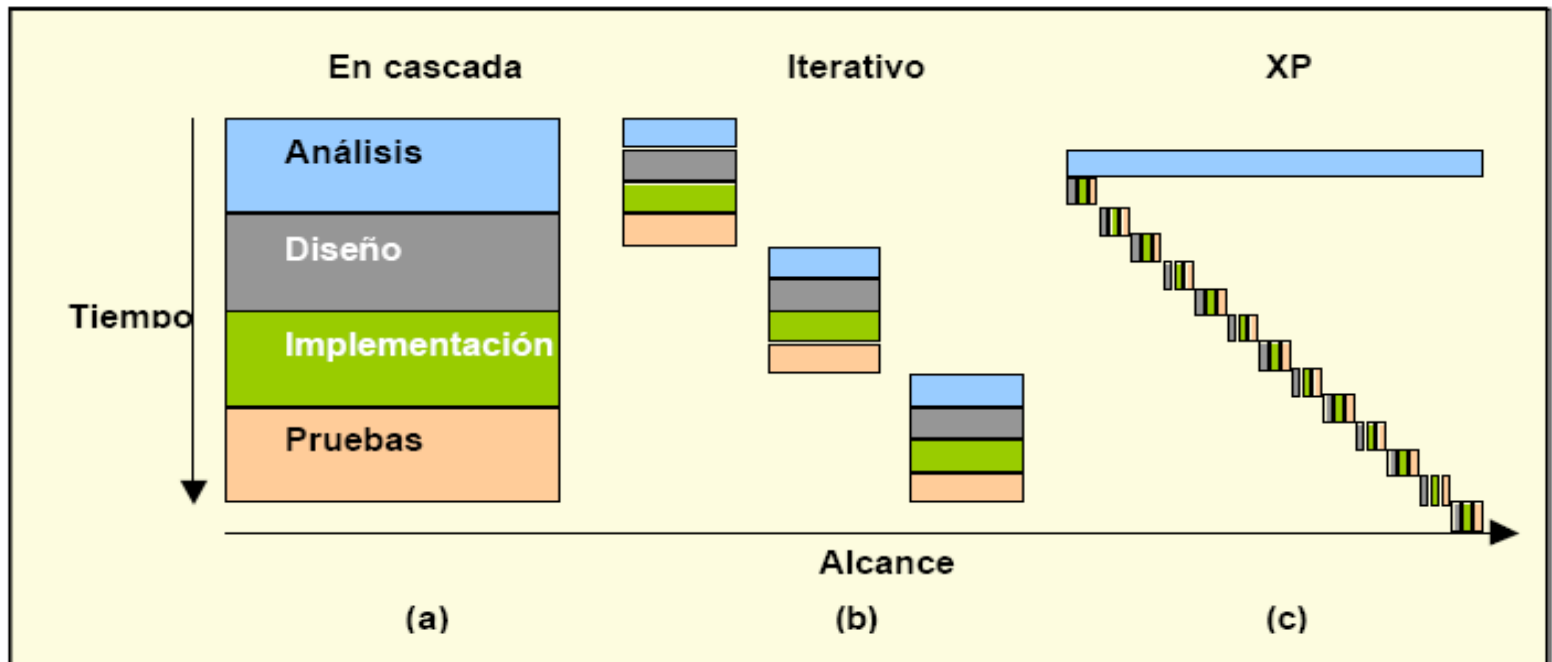
- 1.- Simplicidad de código y de diseño para producir software fácil de modificar.
- 2.- Reingeniería continua para lograr que el código tenga un diseño óptimo.
- 3.- Desarrollar estándares de codificación, para comunicar ideas con claridad a través del código.
- 4.- Desarrollar un vocabulario común, para comunicar las ideas sobre el código con claridad.

PRÁCTICAS DE DESARROLLO

- 1.- Adoptar un método de desarrollo basado en las pruebas para asegurar que el código se comporta según lo esperado.
- 2.- Programación por parejas, para incrementar el conocimiento, la experiencia y las ideas.
- 3.- Asumir la propiedad colectiva del código, para que todo el equipo sea responsable de él.
- 4.- Integración continua, para reducir el impacto de la incorporación de nuevas funcionalidades.

PRÁCTICAS DE NEGOCIO

- 1.- Integración de un representante del cliente en el equipo, para encauzar las cuestiones de negocio del sistema de forma directa, sin retrasos o pérdidas por intermediación.
- 2.- Adoptar el juego de la planificación para centrar en la agenda el trabajo más importante.
- 3.- Entregas regulares y frecuentes para satisfacer la inversión del cliente.
- 4.- Ritmo de trabajo sostenible, para terminar la jornada cansado pero no agotado.



Universidad Mariano Gálvez

Análisis de Sistemas

Conceptos Principales de Sistemas (Tema 1)

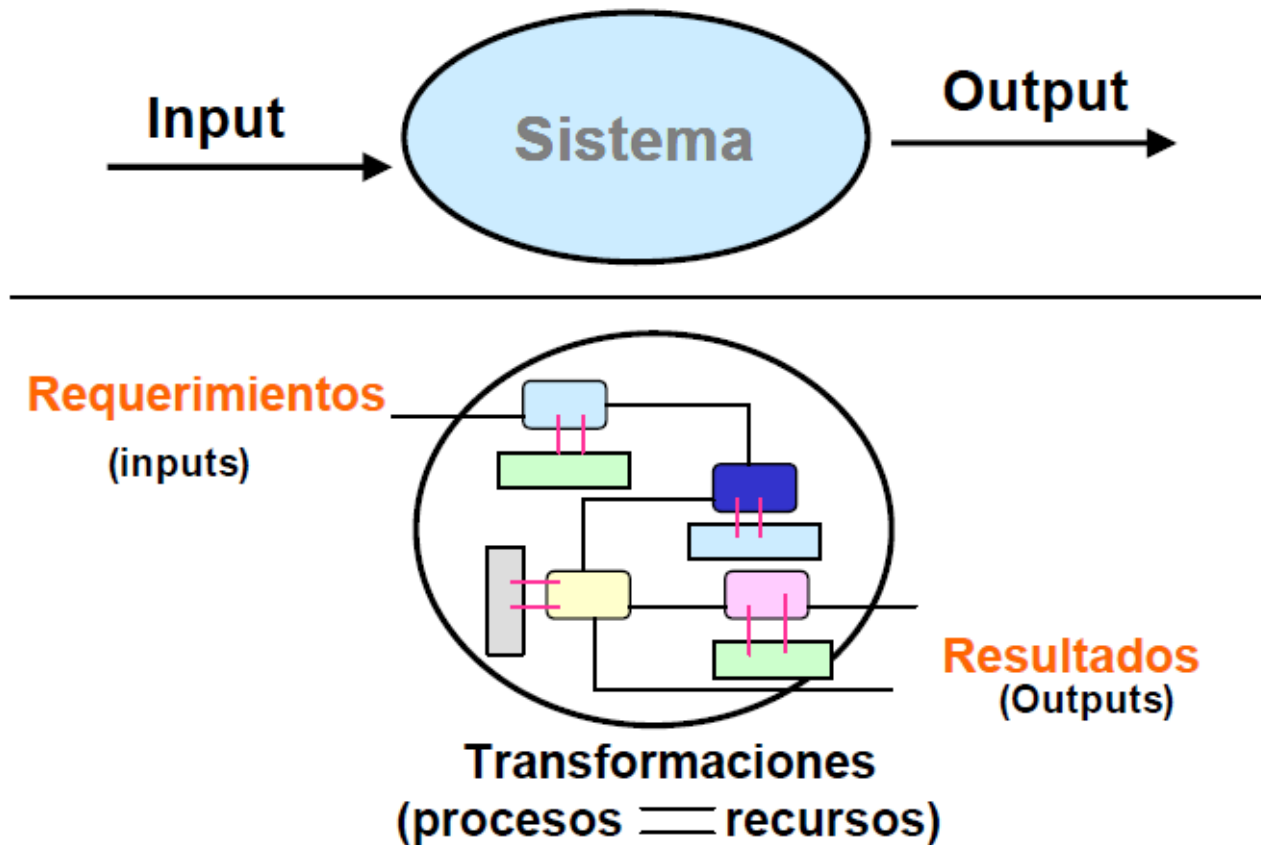
Conceptos Generales

- Sistema
 - Conjunto de entidades u objetos relacionados entre si, conformando una estructura con un fin común
- Sistema de Información – SI
 - Sistema cuya finalidad es procesar datos de entrada para producir como salida información oportuna, relevante y confiable .

Conceptos Generales

- Análisis de SI
 - Análisis de las reglas del negocio (actividades) y de las necesidades de soporte de información.
- Diseño de SI
 - Especificación de los procedimientos de trabajo y de la plataforma tecnológica sobre la que se implementará el SI.

Esquema General de los Sistemas



Ejercicio (5 minutos)

- Mencione 5 ejemplos de sistemas mencionando:
 - Entrada
 - Proceso
 - Salida
 - Componentes principales
- Mencione 3 ejemplos de sistemas de información mencionando:
 - Entrada
 - Proceso
 - Salida
 - Componentes Principales

Descripción de los Sistemas de Información

- Alimentación : captura o entrada de datos
- Procesamiento : proceso para organizar los datos y realizado en una computadora
- Almacenamiento : disco o base de datos
- Comunicación : medio de transmisión de datos
- Salida : consultas o reportes impresos

Tipos de Sistemas de Información



División de los Sistemas

- Sistemas on-line
- Sistemas de tiempo real
- Sistemas de soporte a las decisiones–DSS
- Sistemas basados en el conocimiento

Datos

- Son los hechos que describen sucesos y entidades. "Datos" es una palabra en plural que se refiere a más de un hecho. A un hecho simple se le denomina "data-ítem" o elemento de dato.
- Los datos son comunicados por varios tipos de símbolos tales como las letras del alfabeto, números, movimientos de labios, puntos y rayas, señales con la mano, dibujos, etc.

Información

- Es una colección de hechos significativos y pertinentes, para el organismo u organización que los percibe.
- Es un conjunto de datos significativos y pertinentes que describan sucesos o entidades.

Datos e Información

- Los DATOS se transforman en INFORMACION al momento de ser presentada y revisada por los directivos, gerentes u operadores de la organización, quienes interpretan la forma de presentación de los datos y toman las decisiones de su responsabilidad.

Base de Datos

- Colección de datos organizados para dar servicio a muchas aplicaciones al mismo tiempo al combinar los datos de manera que parezcan estar en una sola ubicación.

Usuarios, Beneficiarios o Stakeholders

- Son aquellos que utilizan y se benefician directamente del sistema o para quienes se construye el sistema.
- Usuario es la persona a la que se tendrá que entrevistar, a menudo con gran detalle ("son sacándoles" la información), a fin de conocer sus requerimientos para el nuevo sistema.
- También se les suele denominar clientes internos. Pueden clasificarse por su categoría de trabajo en: usuarios operacionales, usuarios supervisores y usuarios ejecutivos.

Administradores

- Son los que tienen que ver con la asignación de recursos al proyecto y su supervisión.
- Suelen conformar un comité directivo que velará por que el proyecto se desarrolle dentro de los márgenes y los requerimientos establecidos en la documentación aprobada.

Analista de Sistemas

- Es el papel o rol que jugará usted. Es responsable de determinar y especificar los requerimientos del sistema a construir desde los usuarios.
- A veces representa a un arqueólogo o a un mediador o a un innovador o a un jefe de proyecto.
- Esto significa que como analista de sistemas se necesita más que una habilidad para dibujar diagramas. Se requiere facilidad para el manejo de personas durante las entrevistas, mediar en desacuerdos comunes en todo proyecto, poseer conocimientos de computación y una mente lógica y organizada que le permita analizar los sistemas desde distintas perspectivas.

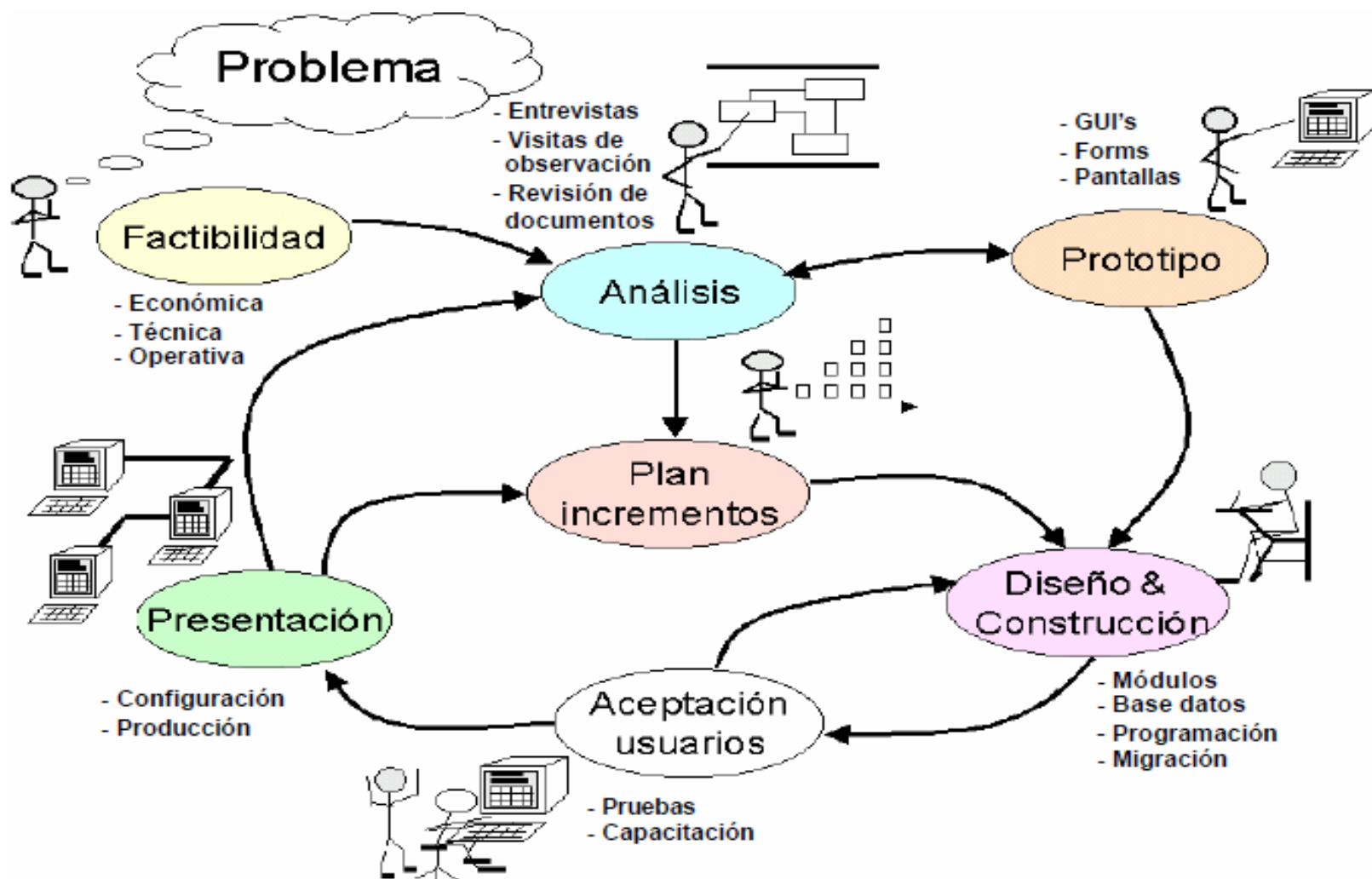
Diseñador de Sistemas (Arquitecto)

- Diseñador es el responsable de especificar las características de la arquitectura del sistema (n-capas, web, cliente servidor, etc) y que servirá de base para el trabajo de los programadores. En muchos casos, el analista y el diseñador son la misma persona.
- En las metodologías actuales se habla también del rol de arquitecto, el cual equivale al rol de diseñador

Programador

- Programador es la persona responsable de pasar a un lenguaje de programación de aplicaciones las características de diseño del sistema especificadas por el diseñador. A menudo es el que descubre errores y ambigüedades en la propuesta de requerimientos entregada por el analista.

Ejemplo del Proceso



Fuentes de Solicitudes de Información

- Existen cuatro fuentes principales de solicitudes de proyectos, estos solicitantes pueden ser:
 - Jefes de departamento.
 - Altos ejecutivos.
 - Analistas de sistemas.
 - Grupos externos.

Tarea

- Formar Grupos e Investigar cada grupo sobre las metodologías de software (Tema 2)
 - RUP
 - Ciclo de Vida Clásico
 - PSP
 - CMMI
 - Espiral
 - XP