

# Ingeniería de Software

Organismo académico:

Facultad de Contaduría y Administración  
De la UAEM

Programa educativos en los que se imparte:

Licenciatura en Informática Administrativa presencial y a distancia

Prerrequisitos:

Conocer lenguajes de programación

Unidad de Aprendizaje Antecedente:

Programación Orientada a Objetos

Unidad de Aprendizaje Consecuentes:

Administración de Bases de Datos

Horas de teoría:	Horas de práctica:	Créditos:
2	4	10
Núcleo de formación	Carácter de la Unidad de Aprendizaje	
Sustantivo	Obligatoria	

Desarrollo Ágil y Programación  
Extrema  
para  
Ingeniería de Software

Dr. En A. Dulce Ma. Morán Linares

Fecha de elaboración:

Septiembre del 2015

# Bibliografía

- Pressman Roger S., Ingeniería de Software. Un enfoque práctico, 5ta. Edición, McGraw Hill, 2004.
- Pressman Roger S. Ingeniería de Software, Un enfoque practico, Mac Graw Hill, 5ª ed.,2002, España
- Ian Sommerville, Ingeniería de Software, Pearson Adisson Wesley, 7ª ed., 2004, España.
- Bernard Bruegge, Ingeniería de Software. Orientada a Objetos, 1ra Edición, Prentice Hall, 2002
- Richard E Fairley. Ingeniería de sw . McGraw Hill, 1988.

# Guión explicativo

Este material puede ser empleado en la materia de Ingeniería de Software para introducir a los alumnos en los conceptos de paradigmas de la Ingeniería de software en :

## **UNIDAD DE COMPETENCIA I**

Introducción, Arquitecturas y metodologías para la construcción del Sw. Conocerá las diferentes metodologías de desarrollo de software para distinguir en que proyectos pueden aplicarse.

# DESARROLLO ÁGIL

Capacidad de realizar una tarea o acción de la manera más rápida y eficiente, o sea empleando la menor cantidad de energía posible.



La ingeniería de software ágil combina una filosofía con un conjunto de lineamientos de desarrollo.  
Pone énfasis en

- La satisfacción del cliente y en la entrega rápida de sw incremental.
- Los equipos pequeños y muy motivados para efectuar el proyecto.
- Los productos del trabajo con mínima ingeniería del sw y la sencillez general en el desarrollo
- Los lineamientos de desarrollo enfatizan la entrega sobre el análisis y el diseño y la comunicación activa y continua entre desarrolladores y cliente.



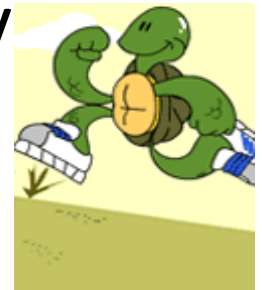
# ¿Quién lo hace?

Los ingenieros de sw y otros participantes del proyecto (gerentes, clientes, usuarios finales, etc.) trabajan juntos en un proyecto ágil, formando un equipo con organización propia y que controla su propio destino. Un equipo ágil controla la comunicación y colaboración entre aquellos a quien sirve.



# ¿Por qué es importante?

El ambiente moderno de negocios genera sistemas basados en computadores y productos de software evoluciona rápida y constantemente. La ingeniería de software ágil representa una alternativa razonable a la ingeniería del sw convencional para ciertas clases de sw y en algunos tipos de proyectos. Asimismo, se ha demostrado que concluye rapidez sistemas exitosos.



# ¿Cuáles son los pasos?

Un nombre más apropiado para el desarrollo ágil sería “ingeniería del sw ligero”.

Permanecen las condiciones estructurales fundamentales: comunicación, planeación, modelado, construcción y despliegue. Pero se transforman en un conjunto mínimo de tareas que lleva al equipo del proyecto hacia la construcción y entrega.



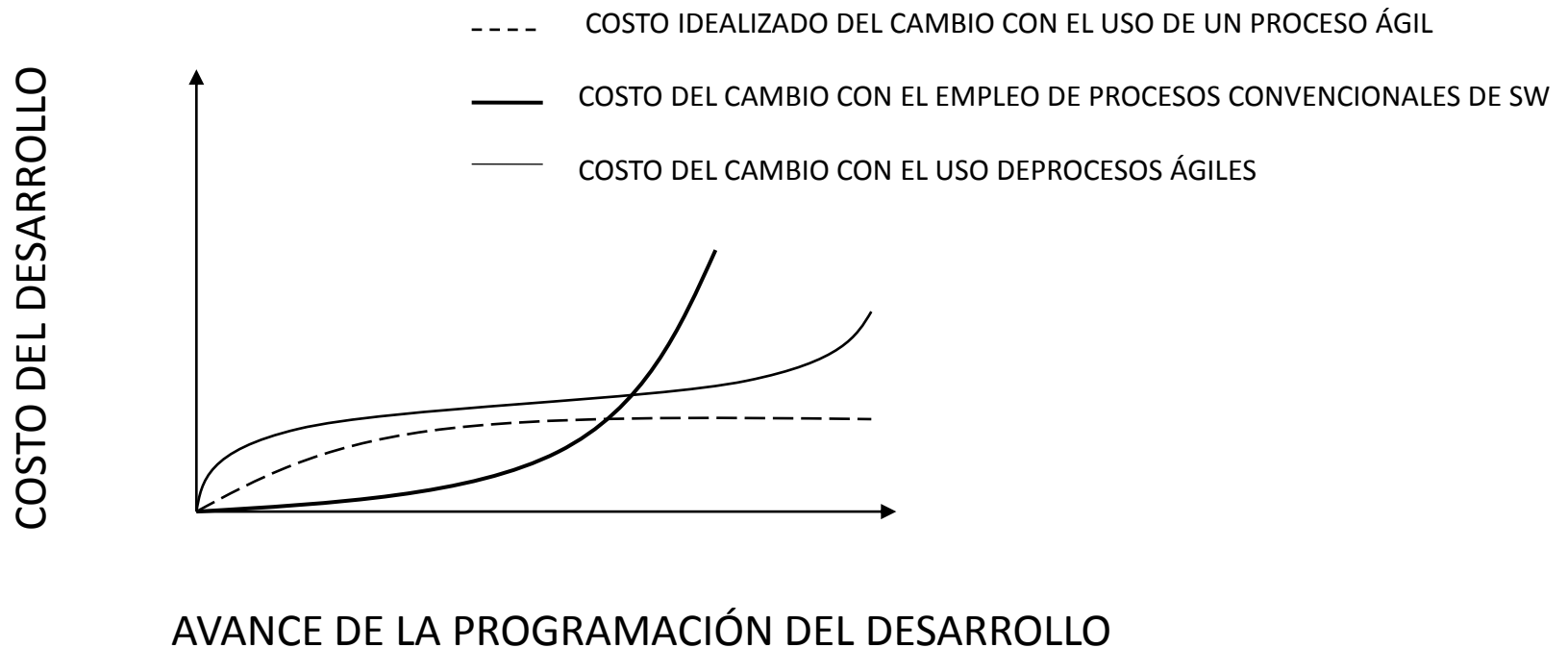
# Agilidad en el contexto del trabajo de ingeniería de sw

La agilidad puede aplicarse a cualquier proceso del sw, Para lograrlo es esencial:

- Que este se diseñe en forma que permita al equipo del proyecto adaptar las tareas y hacerlas directas.
- Ejecutar la planeación de manera que entienda la fluidez de un enfoque ágil del desarrollo.
- Poner el énfasis en una estrategia de entrega incremental que haga trabajar al software tan rápido como sea posible para el cliente, según el tipo de producto y el ambiente de operación.

# LA AGILIDAD Y EL COSTO DEL CAMBIO

Los defensores de la agilidad afirman que un proceso ágil bien diseñado “aplana” el costo de la curva de cambio.



# ¿Qué es un proceso Ágil?

Cualquier proceso del sw ágil se caracteriza por la forma en la que aborda cierto número de suposiciones clave acerca de la mayoría de los proyectos de sw.



## ¿Qué es un proceso Ágil?

- Es difícil predecir que requerimientos del sw persistirán y cuales cambiarán. También es difícil pronosticar cómo cambiarán las prioridades del cliente a medida que avanza el proyecto.
- Para muchos tipos de sw, el diseño y la construcción están imbricados. Es decir, ambas actividades deben ejecutarse en forma simultánea, de modo que los modelos de diseño se aprueben a medida que se crean. Es difícil predecir cuanto diseño se necesita antes de que se use la construcción para probar el diseño.
- El análisis, el diseño, la construcción y las pruebas no son tan predecibles como nos gustaría.

## ¿Qué es un proceso Ágil?

¿Cómo crear un proceso que pueda manejar lo impredecible?

- La respuesta está en la adaptabilidad del proceso.
- Por tanto un proceso ágil debe de ser adaptable.
- un procesos de sw ágil debe de adaptarse incrementalmente, para lograrlo se necesita la retroalimentación del cliente.
- Deben entregarse incrementos de sw (prototipos ejecutables o porciones de un sistema operativo) en periodos cortos de tiempo, de modo que la adaptación vaya a ritmo con el cambio impredecible



# Principios de agilidad

Se definen algunos principios de la agilidad, los principales son:

- La prioridad más alta de satisfacer al cliente a través de la entrega pronta y continua de software valioso.
- Son bienvenidos los requerimientos cambiantes, aún en una etapa avanzada del desarrollo. Los procesos ágiles dominan el cambio para provecho de la ventaja competitiva del cliente.
- Entregar con frecuencia software que funcione, de dos semanas a un par de meses, de preferencia lo más pronto que se pueda.



- Hay que desarrollar los proyectos con individuos motivados. Debe darse a éstos el ambiente y el apoyo que necesiten, y confiar en que harán el trabajo.
- El método más eficiente y eficaz para transmitir información a los integrantes de un equipo de desarrollo , y entre estos, es la conversación cara a cara.
- La atención continua a la excelencia técnica y el buen diseño, mejora la agilidad.

## **FACTORES HUMANOS:**

“El desarrollo ágil se centra en los talentos y habilidades de los individuos y adaptan el proceso a personas y equipos específicos”

Los miembros del equipo de sw deben tener ciertas características como:



- **COMPETENCIA**
- **ENFOQUE COMUN**
- **COLABORACION**
- **HABILIDAD PARA TOMAR DESICIONES**
- **CAPACIDAD PARA RESOLVER PROBLEMAS DIFUSOS**
- **CONFIANZA Y RESPETO MUTUOS**
- **ORGANIZACIÓN PROPIA**



# Programación Extrema

- El enfoque mas utilizado del desarrollo del sw ágil.
- Desarrollada a finales de 1980, escrito fundamentalmente por Kent Beck tiene como objetivo el proceso ágil para ser usado en organizaciones grandes

- VALORES XP: Beck lo define como: conjunto de 5 valores que establecen el fundamento para todo trabajo realizado como parte de XP:
- **Comunicación**
- **Simplicidad**
- **Retroalimentación**
  - sw implementado
  - prueba unitaria
  - historia de usuario o casos de uso
- **Valentía**
- **Respeto.**



# EL PROCESO XP

- La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y practicas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas.

**PLANEACIÓN:** lleva a la creación de “historias”

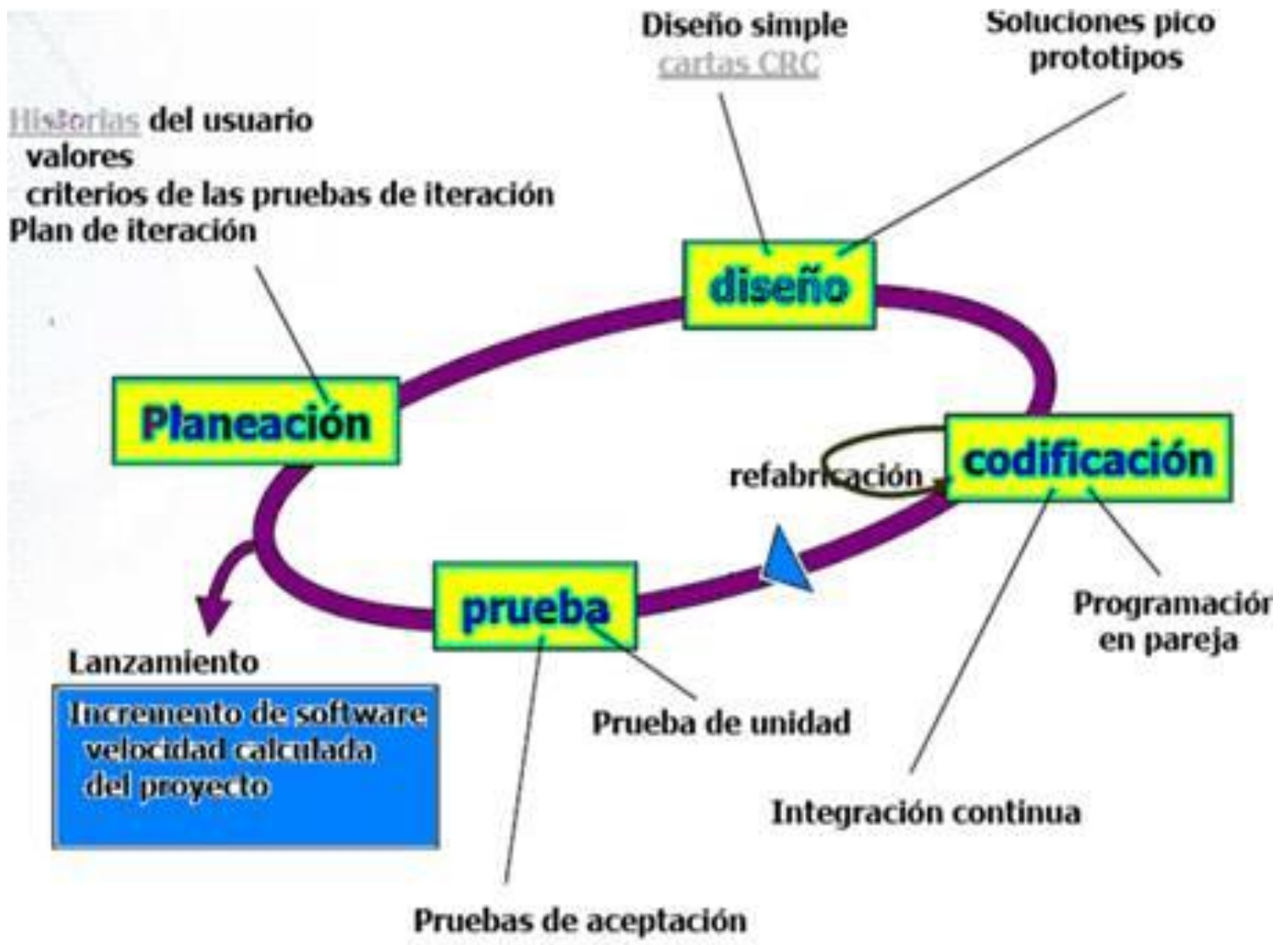
Las historias se ordenan de acuerdo a la valoración del cliente y se clasifican en :

- 1) todas las historias implementas de inmediato
- 2) las de mas valor se implementan primero
- 3) las historias mas riesgosas forman parte de la programación y se implementan primero.

- **DISEÑO:**
- sigue el principio MS (mantenlo sencillo)
- Estimula el uso de tarjetas CRC(clase-responsabilidad-colaborador)
- **CODIFICACION:** Una vez que el código este terminado, se le aplica de inmediato una prueba unitaria, con lo que se obtiene retroalimentación instantánea para los desarrolladores
- **PRUEBAS:**







*XP Para un desarrollo  
Industrial*

# Programación Extrema Industrial

## IXP

•Joshua Kerjevsky Describe a la IXP de la Forma Siguiete:

*“Es la evolución Orgánica de XP. Esta imbuida del espíritu minimalista, centrado en el cliente y orientado hacia las pruebas que tiene XP. IXP difiere sobre todo de la XP Original en su mayor incisión de la gerencia, el papel mas amplio de los clientes y en sus practicas técnicas actualizadas.”*

La IXP incorpora **“6 Practicas Nuevas”**, Diseñadas para ayudar a Garantizar que un proyecto XP Funciona con éxito para proyectos Significativos dentro de una Organización Grande.

# 1.- Evaluación de la Factibilidad.

***\*La Organización debe Efectuar una “Evaluación de Factibilidad” Antes de iniciar un proyecto IXP.***

- a) Existe un ambiente apropiado de Desarrollo que acepte IXP.
- b) El equipo estará constituido por los participantes adecuados
- c) La organización tiene un programa de calidad distintivo y apoya la mejora continua.
- d) La cultura organizacional apoyará los nuevos valores de un equipo ágil.
- e) La comunidad extendida del proyecto estará constituida de modo apropiado.

## 2.- Comunidad del Proyecto

\*En IXP, los miembros de la comunidad y sus papeles deben definirse de modo explícito, así como establecer los mecanismos para la comunicación y coordinación entre los integrantes de la comunidad.



# 3.- Calificación del Proyecto

El equipo de IXP evalúa el proyecto para determinar si existe una Justificación apropiada de Negocios y si el proyecto cumplirá las metas y objetivos generales de la organización.

La calificación también analiza el contexto del proyecto a fin de determinar cómo complementa, extiende o reemplaza sistemas o procesos existentes.

## 4.- Administración orientada a Pruebas

- Un proyecto IXP requiere criterios medibles para evaluar el estado del proyecto y el avance realizado.
- La Administración orientada a pruebas establece una serie de “destinos” medibles y luego define los mecanismos para determinar si se han alcanzado o no estos.

# 5.- Retrospectivas

**Después de entregar un incremento del Software , el equipo XP realiza una revisión técnica especializada que se llama “Retrospectiva” el cual:**

**-Examina los temas, eventos y lecciones aprendidas**

**Todo esto a lo largo del incremento de Software y/o de la liberación de todo el software.**

**“El objetivo es mejorar el proceso IXP”**



## 6.- Aprendizaje Continuo

- Los Miembros de XP son invitados (y tal vez incentivados ) a aprender nuevos métodos y técnicas que conduzcan a una calidad mas alta del producto.



Además de las 6 nuevas practicas analizadas, IXP modifica algunas existentes en XP :

**DIH.-** Desarrollo impulsado por la historia insiste en que las historias de las pruebas de aceptación se escriban antes de generar una sola línea de código.

**DID.-** Diseño Impulsado por el Dominio, Es una mejora sobre el concepto de la “Metáfora del Sistema” usado en XP. Sugiere la creación evolutiva de un modelo de dominio que.- Represente con exactitud como piensan los expertos del dominio en su materia.

**Formación de Parejas.-** Amplia el concepto de programación en pareja para que incluya a los gerentes y a otros participantes. El objetivo es compartir conocimientos entre los integrantes que no estén involucrados en el desarrollo técnico.

**Usabilidad Iterativa.-** Estimula un diseño que evoluciona a medida que se liberan los incrementos del software y que estudia la interacción de los usuarios con el Software.

## 3.5 Otros Modelos Ágiles de Proceso

La historia de la Ingeniería de Software está salpicada de decenas de descriptores y de metodologías de proceso, métodos de modelado, etc.

Cada uno tuvo notoriedad y luego fue eclipsado por algo nuevo y (supuestamente ) mejor.

Con la introducción de una alta variedad de modelos ágiles del proceso – cada uno en lucha por la aceptación de la comunidad de desarrollo de software- el movimiento ágil esta siguiendo la misma ruta histórica.

Como se ha dicho, el mas usado de todos los modelos ágiles de proceso es la Programación Extrema (XP) .

Pero se han propuesto muchos otros y están en uso en toda la industria. Entre ellos están los Siguietes:

\*Desarrollo Adaptativo de Software (DAS)

\*Scrum

\*Método de Desarrollo de Sistemas Dinámicos (MDSD)

\*Cristal

\*Desarrollo Impulsado por las Características (DIC)

\*Desarrollo Esbelto de Software (DES)

\*Modelado Ágil (MA)

\*Proceso Unificado Ágil (PUA)

Es importante notar que todos los modelos mencionados, se apegan al “Manifiesto para el Desarrollo Ágil de Software” y sus principios.

# Desarrollo Adaptativo de Software (DAS)

Propuesta por “Jim Highsmith” como una técnica para elaborar software y sistemas complejos. Los fundamentos filosóficos del DAS se centran en la colaboración humana y en la organización propia del equipo.

Highsmith argumenta que un enfoque de desarrollo adaptativo basado en la colaboración es “tanto una fuente de orden en nuestras complejas interacciones, como de disciplina e ingeniería”.

El Define un Ciclo de Vida del “DAS” que incorpora 3 Fases :

- \* Especulación
- \* Colaboración
- \* Aprendizaje

**Planeación adaptativa del ciclo *enunciado de la misión* restricciones de proyecto *requerimientos básicos*, plan de entrega en el tiempo.**

**Recabar requerimientos JAD miniespecificaciones**

**Especulación**

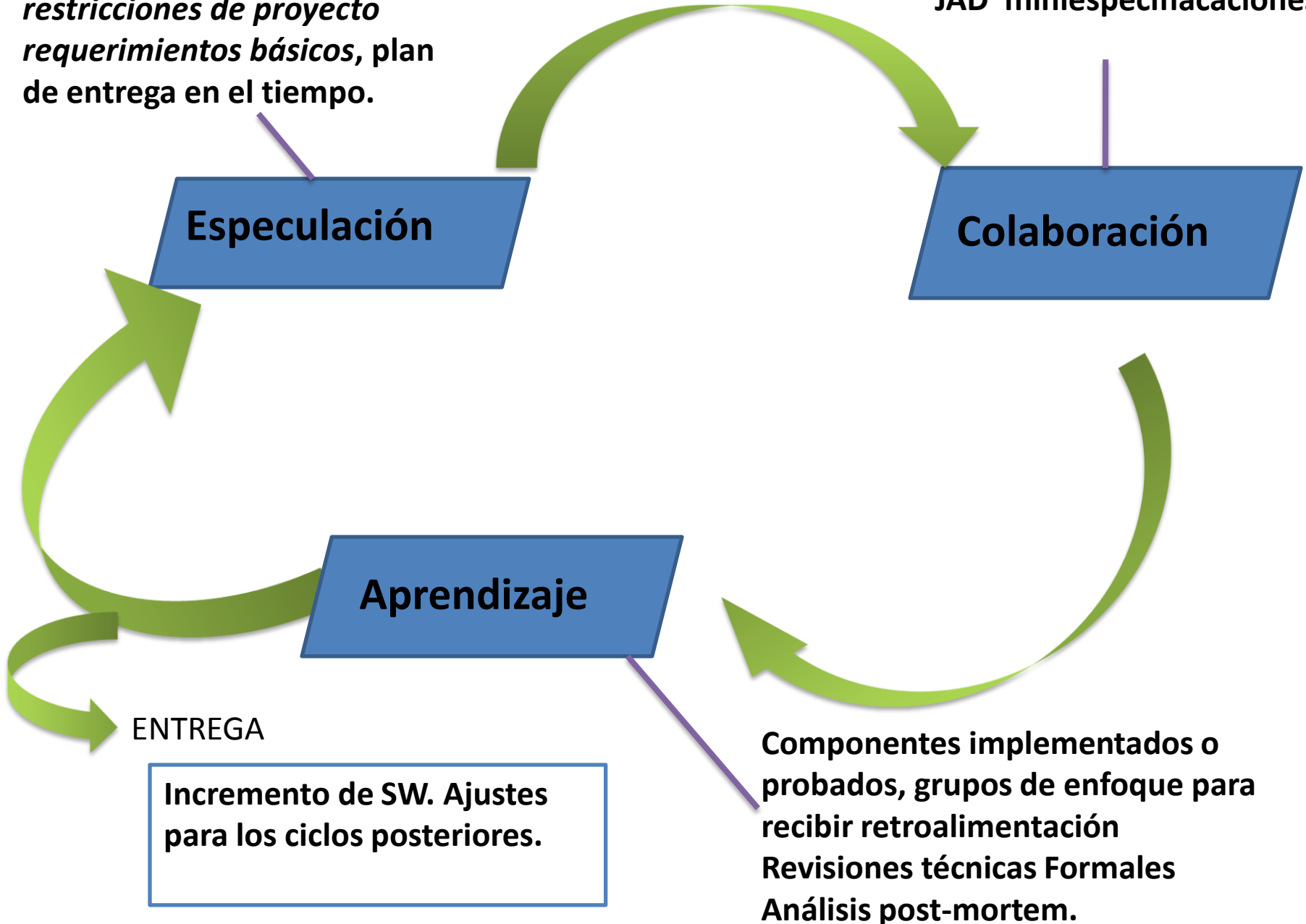
**Colaboración**

**Aprendizaje**

**ENTREGA**

**Incremento de SW. Ajustes para los ciclos posteriores.**

**Componentes implementados o probados, grupos de enfoque para recibir retroalimentación  
Revisiones técnicas Formales  
Análisis post-mortem.**





# Gracias por su atención!



# PREGUNTAS

- Menciona 2 lineamientos de ingeniería del software
- Características de los miembros del equipo de SW
- Actividades estructurales del proceso XP
- Definición de IXP