



# ANÁLISIS Y DISEÑO DE SISTEMAS

## Clase II: Producto y Proceso

Primer Cuatrimestre 2013

Mg. María Mercedes Vitturini  
[mvitturi@cs.uns.edu.ar]  
Dpto. Cs. e Ing. de la Computación  
Universidad Nacional del Sur



## Lecturas

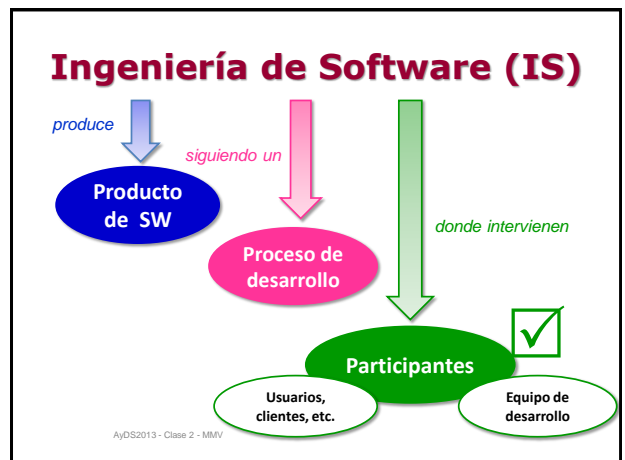
- Constituir en grupos de **dos alumnos**.
- Enviar la conformación de los mismos a [mvitturi@cs.uns.edu](mailto:mvitturi@cs.uns.edu) con los siguientes datos de los integrantes.
  - Legajo, Nombre y apellido, dirección de e-mail
- Objetivo: participar en el Trabajo de Lectura y Exposición de un artículo en idioma Inglés relacionado con temas afines a los contenidos de la materia.

AyDS2013 - Clase 2 - MMV 2

## Ingeniería de Software - Repaso

- Es el área de las ciencias de la computación que estudia la construcción de sistemas de software tan *grandes y complejos* que requieren de un grupo de ingenieros.
- *Características*
  - Es esencialmente un trabajo en *equipo*.
  - Existen *versiones del producto* de software.
  - El producto está *sujeto a cambios*.
  - Se requiere de un *trabajo disciplinado*.

3 AyDS2013 - Clase 2 - MMV



## El producto de software



La IS busca producir productos de software de calidad.

## Producto

**Producto de IS** – el producto de IS es el sistema de software que se distribuye al cliente junto con su documentación.

- Los productos de software se pueden clasificar en:
  - *Software a medida*: software desarrollado para un cliente particular bajo un contrato.
  - *Software genéricos*: desarrollados para ser vendidos a un mercado abierto.

6 AyDS2013 - Clase 2 - MMV

## El producto de Software

### Características del software

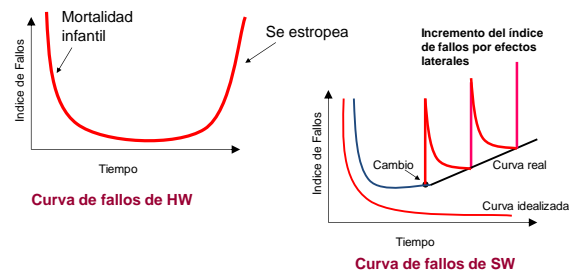
- El SW como producto de ingeniería tiene características particulares que lo distinguen:
  - El software *es lógico* y no físico.
  - El software *es maleable*.
  - El software *se desarrolla* no se fabrica.
    - La industria tiende a ensamblar componentes, pero aún la mayor parte del software se construye a medida.
  - El software *no se estropea* (pero se deteriora).

7

AyDS2013 - Clase 2 - MMV

## SW vs. HW

Diferencias de comportamiento entre el HW y SW

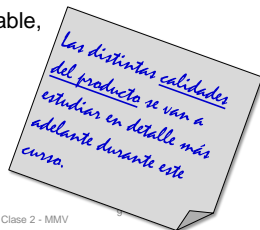


8

AyDS2013 - Clase 2 - MMV

## Productos de calidad

- La IS apunta a la construcción de software como una actividad de ingeniería: *producir productos de calidad*.
- Algunas calidades:
  - Correcto, robusto, confiable,
  - amigable,
  - mantenible, evolutivo,
  - ...



AyDS2013 - Clase 2 - MMV

## Procesos de desarrollo de software



Los procesos de desarrollo colaboran con la producción de productos de calidad

## El proceso

El de **proceso de desarrollo** define *el marco* de trabajo para un conjunto de *tareas claves en la producción de software*. Constituye la base para *el control y gestión de los proyectos de software*.

- Establece el contexto en el cual se aplican:
  - los métodos técnicos,
  - se generan los productos de trabajo (modelos, documentos, datos, código, etc.)
  - se asegura calidad,
  - el cambio se puede manejar.

11

AyDS2013 - Clase 2 - MMV

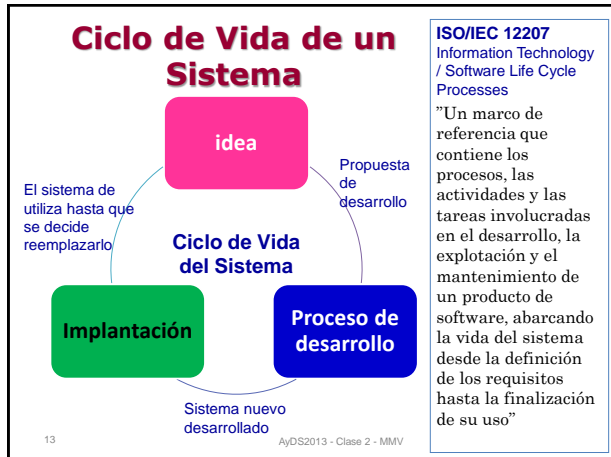
## Ciclo de Vida

**CICLO DE VIDA DE UN SISTEMA** – dado un sistema particular, su *ciclo de vida* comprende desde la primera la idea de desarrollo del software, hasta que es implementado, entregado, y aún después hasta que deje de usarse.

- Hasta obtener el sistema se dice que el mismo pasa por un *ciclo de vida* compuesto de varias etapas.
- Existen diferentes **modelos de procesos**, incluso con algunas variaciones, la mayoría se basa en unos pocos paradigmas.

12

AyDS2013 - Clase 2 - MMV



### Primer "proceso" de desarrollo

#### CODE-FIX

- Fue la primera aproximación de desarrollo.
- No es un modelo de proceso.

**Características**

- El desarrollo de software era una tarea unipersonal.
- El problema era sencillo y bien entendido.
- Los lenguajes de programación bajo nivel.

14 AyDS2013 - Clase 2 - MMV

### Code & Fix

- **Actividades**
  - Escribir código y probar que funciona.
  - Corregirlo (mejorar o agregar funcionalidades).
- **Observaciones**
  - Después de algunas evoluciones se torna no confiable.
  - Posible en aplicaciones cortas y bien entendidas.

15 AyDS2013 - Clase 2 - MMV

### Hitos importantes

- Se separan los roles *programador* y *usuario*.
  - El producto no satisface las necesidades del usuario.
- Necesidad de *aplicaciones más complejas*.
  - Se requiere mayor calidad.
- El *desarrollo* se convierte en una *actividad grupal*.
  - Se necesitan modelos de comunicación.

16 AyDS2013 - Clase 2 - MMV

### Actividades del proceso de desarrollo

- Una forma de **identificar las actividades del proceso según el eje del tiempo** (no excluyentes):
  1. Estudio de factibilidad
  2. Elicitación, análisis y especificación de requerimientos *¿qué?*
  3. Diseño *¿cómo?*
  4. Codificación y testeo de módulos.
  5. Integración y testeo de sistemas.
  6. Distribución, entrega y mantenimiento.
  7. Otras actividades.

17 AyDS2013 - Clase 2 - MMV

### 1- Estudio de Factibilidad

- Una primera actividad de casi cualquier proyecto es hacer un **estudio de factibilidad (EF)**.
  - Se *evalúan costos y beneficios* del desarrollo.
  - Cuanto más conocimiento se tiene del problema mejor. Sin embargo, *no se le puede destinar demasiado tiempo*.
  - Incluye:
    - La definición general del problema.
    - Soluciones alternativas y beneficios esperados.
    - Estimación de recursos, costos, tiempos, fechas de entrega para cada solución alternativa.
- Según el proyecto y su entorno el EF puede ser más o menos formal.

18 AyDS2013 - Clase 2 - MMV

## 2 – Elicitación, Análisis y especificación de requerimientos

- Identificar **qué?** y no **cómo?**. La clave:  
*Separar, abstraer y partir el problema*
- Se identifican las **calidades y servicios** requeridos por la aplicación: funcionales y no funcionales.
- Incluye la descripción de las necesidades a **distintos niveles de abstracción**.
  - Separar en partes analizables.
  - Separar según las visiones del software:
    - Datos - funciones - restricciones - control / comportamiento

19

AyDS2013 - Clase 2 - MMV

## 3 – Diseño de arquitectura y diseño detallado del sistema

- **Diseño de arquitectura**
  - Es de alto nivel.
  - Divide al sistema en subsistemas y módulos.
  - Existen patrones de arquitectura.
- **Diseño de módulos**
  - Descomposición modular.
  - Definición de componentes, interfaces e interconexiones.
  - Existen patrones
- **Documentar!**

20

AyDS2013 - Clase 2 - MMV

## 4 – Codificación y Testeo de Módulos

- Se **escriben los programas** en un lenguaje de programación.
- **Una buena práctica es definir y adoptar convenciones** de programación y testeo.
- Puede incluir revisiones de código (debugging y testeo y chequeos de performance).
- Se realiza el **testeo de unidad**. Esta tarea *puede ser realizada por el programador*.

21

AyDS2013 - Clase 2 - MMV

## 5- Integración y Testeo de Sistema

- Se **ensamblan los componentes** desarrollados y testeados separadamente.
- En esta etapa se pueden **incluir testeos con:**
  - Datos reales.
  - Clientes reales (alpha test).
- Aplicar estándares (*top down* o *bottom-up*).
- **Documentar** los casos de prueba.

22

AyDS2013 - Clase 2 - MMV

## 6 – Distribución, entrega y mantenimiento de sistema

- Ocurre cuando se completa el desarrollo y testeo de la aplicación.
- Se puede entregar primero a un **grupo reducido de clientes** antes de la entrega oficial (**beta testing**).
  - Objetivos: experimentación y retroalimentación.
- Pueden requerirse cambios antes de la entrega oficial.
- El **mantenimiento** incluye a las actividades realizadas una vez que el producto fue entregado al cliente.

23

AyDS2013 - Clase 2 - MMV

## Sobre el Mantenimiento

- El costo de mantenimiento en muchos casos representa más del 60% del costo del producto.
  - Corregir errores encontrados después de la entrega (**mantenimiento correctivo**)
  - Adaptar la aplicación a cambios de entorno (**mantenimiento adaptativo**)
  - Cambiar o agregar funciones o calidades del sistema (**mantenimiento perfecto**)
- El 50% de los costos de mantenimiento se atribuyen a cambios en los requerimientos.

24

AyDS2013 - Clase 2 - MMV

## 7 – Otras actividades

- Según sea el proyecto pueden ser necesarias otras actividades:
  - Documentación.
  - Control de calidad
    - Walk-through.
    - Validación.
    - Verificación.
  - Administración
    - Definición de estándares.
    - Tratamiento con recursos (personas).

25

AyDS2013 - Clase 2 - MMV

## Para discutir

- ¿Cómo afecta al *estudio de factibilidad* la *relación cliente/equipo de desarrollo*?
  - Desarrollos a terceros a medida.
  - Equipo de desarrollo en la empresa.
  - Desarrollos de propósito general.
- ¿Cómo afecta al *testeo* el *sistema a desarrollar*?
  - Un editor de textos.
  - Un sistema para una entidad bancaria.
  - Una sistema de control de vuelos.

26

AyDS2013 - Clase 2 - MMV

## Modelos de Proceso de Desarrollo

Los modelos de proceso definen representaciones abstractas para el proceso de desarrollo



## Modelos de Procesos

- Vamos a estudiar **modelos de procesos clásicos** para la construcción de software.
- Los modelos de procesos *evolucionan* con la *tecnología y los requerimientos a satisfacer*.
- Existen cientos de modelos, la mayoría son variaciones de otros.
- Los modelos clásicos:
  - Cascada.
  - Evolutivos.
  - Transformacionales.

28

AyDS2013 - Clase 2 - MMV

## Modelo CASCADA (waterfall)

- Las etapas de desarrollo se organizan en *cascada*, una después de otra. Las etapas se pueden dividir en subactividades concurrentes.
- Característica: la *salida (output)* de una etapa es la *entrada (input)* a la siguiente etapa.
- Cada etapa genera un documento que requiere ser aprobado antes de pasar a la siguiente etapa.

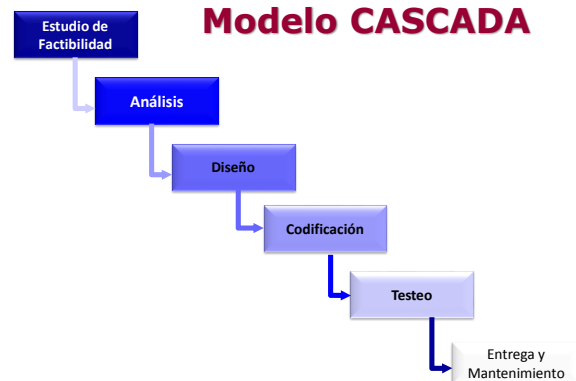
### Origen

- Creado en los '60 para un proyecto de defensa de EEUU.
- Usado en las primeras aplicaciones críticas y complejas.

29

AyDS2013 - Clase 2 - MMV

## Modelo CASCADA



AyDS2013 - Clase 2 - MMV

## Modelo Cascada

### Características:

- Es un *modelo secuencial*. Cada fase debe estar terminada antes de pasar a la siguiente.
- La *definición de las actividades depende de cada proyecto (personalizar el proceso)*.
- Cada actividad *produce un documento de salida* que es el ingreso para la siguiente actividad y que debe ser aprobado y cerrado antes de seguir.
- Las empresas que usan este modelo de proceso definen su estándares.

31

AyDS2013 - Clase 2 - MMV

## Análisis de Cascada

- Es un modelo *conducido por la documentación*.  
Documentos:
  - Especificación de requerimientos.
  - Especificación de diseño.
  - Plan de prueba.
  - Casos de prueba.
- Se deben *definir estándares de documentación*.
- Para pasar de una actividad a la siguiente no deben existir ambigüedades sobre la documentación a entregar.

32

AyDS2013 - Clase 2 - MMV

## MC – Conclusiones

- Fue el primer modelo.
- Es un modelo conducido por la documentación.

### Contribuciones:

- Introduce *disciplina* en el proceso de desarrollo.
- *Postpone la implementación* hasta que los objetivos estén claros.
- Es el paradigma más antiguo y más extensamente usado en ingeniería de software.
- *Impone puntos de control* claros.

33

AyDS2013 - Clase 2 - MMV

## MC – Conclusiones

### Desventajas:

- Es demasiado rígido.
- Retrasa la detección de problemas críticos.
- Difícil la estimación de recursos.
- Es lineal y los proyectos reales raras veces siguen un modelo lineal.
- El usuario interviene al principio y al final del proyecto.
- No provee anticipación al cambio.
- Basado en documentación. Parece burocrático.
- Los costos se trasladan al mantenimiento fácilmente.
- No permite implementaciones parciales.

34

AyDS2013 - Clase 2 - MMV

## El paradigma Evolutivo

**Motivación** – después de varios desarrollos se llegó a la conclusión que las fallas de la primer versión de una aplicación conducen a la necesidad de rehacerla.

**HACERLO DOS VECES.** 

**Problema** – *gap* entre la definición de requerimientos y la distribución de la aplicación.

### Estrategia –

- Entregar algo y medir el valor agregado.
- Ajustar diseño y objetivos.

• **Iterar.**

AyDS2013 - Clase 2 - MMV

## Modelos Evolutivos

### Modelos de Proceso Evolutivos

- Prototipo
- Iterativo incremental.
- Espiral



36

AyDS2013 - Clase 2 - MMV

## Modelo PROTOTIPO (MP)

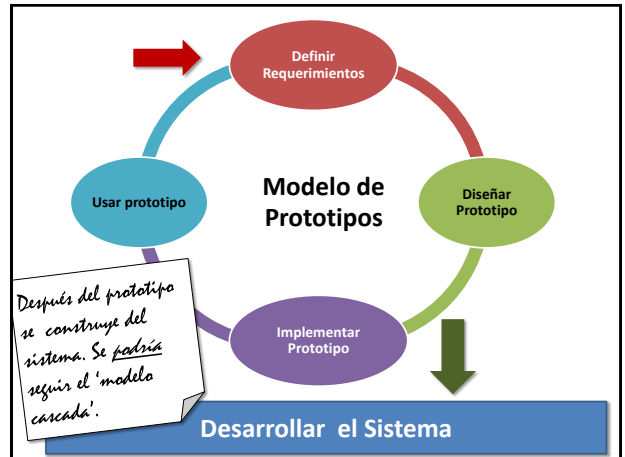
Primer desarrollo: se elabora *un prototipo para descartar*. El prototipo sirve para validar los requerimientos y resolver aspectos críticos del diseño.

### Objetivos

- Investigación repetida de requerimientos y diseño.
- Reducir el riesgo de que la solución no se ajuste a las necesidades del cliente.
- La revisión del prototipo puede resultar en revisar los requerimientos anteriores.

37

AyDS2013 - Clase 2 - MMV



## Modelo de Prototipos

### Actividades

- *Recolección requerimientos del sistema*: desarrollador y cliente definen los objetivos globales.
- Se *elabora un diseño rápido*, centrado en los aspectos del software que son *visibles a los usuarios* (enfoques de entrada y formatos de salida).

### Desventajas

- El cliente ve *el prototipo y lo confunde con el sistema real*.
- Se toman *decisiones rápidas* para poder construir el prototipo, que *son difíciles de revertir* (por ejemplo el lenguaje de programación).
- El prototipo para descartar *nunca se descarta*.

39

AyDS2013 - Clase 2 - MMV

## Modelo Iterativo Incremental

**Iterativo** – hacemos varias veces lo mismo.

**Incremental** – el producto se “incrementa” a medida que se avanza. También son llamados “evolutivos”.

- Consisten en de *incrementos expandidos de un producto de software operativo*, conducidos por la evolución determinada según la experiencia operativa.
- Los incrementos se distribuyen a medida que se completan.
- *Incremento integrado*: es una unidad de software auto-contenida que realiza algún propósito útil.

40

AyDS2013 - Clase 2 - MMV

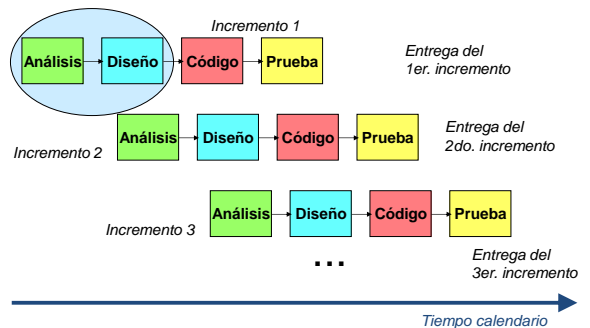
## Ejemplo

- Un ejemplo de desarrollo siguiendo el modelo iterativo incremental para un procesador de textos:
  - **Incremento #1**: administración de archivos básicos y producción de documentos.
  - **Incremento #2**: funciones de edición más sofisticadas (gráficos, tablas, etc.)
  - **Incremento #3**: corrección gramatical y ortográfica.
- Al primer incremento se lo *denomina producto esencial*.

41

AyDS2013 - Clase 2 - MMV

## Modelo Iterativo Incremental



AyDS2013 - Clase 2 - MMV

42



## Modelo Transformacional

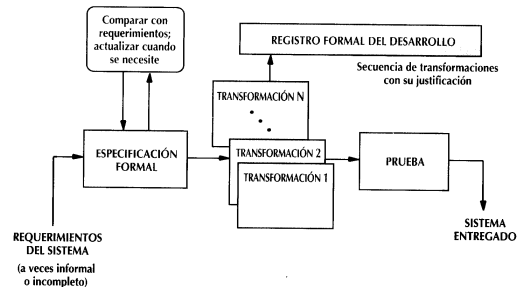
**Modelo Transformacional** –basado en *especificaciones formales*. Ve al desarrollo de software como una secuencia que *transforma* gradualmente la especificación en implementación.

- Requiere que se transformen los requerimientos informales en especificaciones formales: *métodos de especificación formal*.
- Pretende reducir errores automatizando pasos del desarrollo.

49

AyDS2013 - Clase 2 - MMV

## Modelo Transformacional



50

AyDS2013 - Clase 2 - MMV

## Modelo Transformacional

### Ventajas

- Valora reusar y construir componentes reusables.
- Los cambios se hacen en la especificación. Todo queda documentado.
- Garantiza resultados correctos.

### Desventajas

- El desarrollo de modelos formales es caro y lleva tiempo.
- Se requiere de estudios detallados y personal capacitado.
- Es difícil utilizar el modelo como medio de comunicación con el cliente.

51

AyDS2013 - Clase 2 - MMV

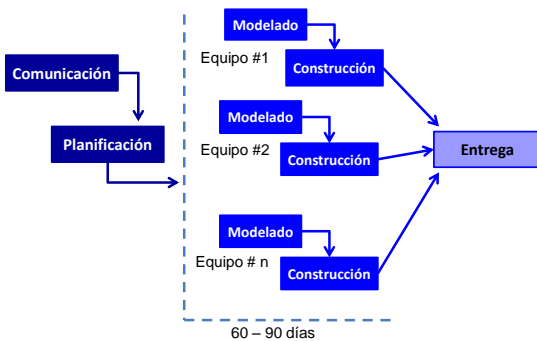
## Desarrollo Rápido de Aplicaciones (DRA)

- Es un modelo de proceso que apunta a proyectos rápidos que finalizan en **60-90 días**.
- Utiliza un enfoque de construcción **basado en componentes**.
- Requiere muchos recursos humanos como para crear varios **equipos DRA**.
- Desarrolladores y clientes deben estar **comprometidos** en las actividades necesarias para completar un sistema en un **marco de tiempo abreviado**.

52

AyDS2013 - Clase 2 - MMV

## Desarrollo Rápido de Aplicaciones



53

AyDS2013 - Clase 2 - MMV

## DRA

### Desventajas

- Tiene requerimientos fuertes de personal de desarrollo.
- No es un modelo adecuado para sistemas que requieren de muchas actividades de mantenimiento.
- No es adecuado cuando los riesgos técnicos son altos.
- No funciona si no existe un compromiso real del cliente.

54

AyDS2013 - Clase 2 - MMV

## Generalidades

¿Cuál es la importancia de los modelos de proceso?

- Organizan la **tarea de desarrollo** de software.
- Dan el marco para la **planificación y control del desarrollo**.
- Permiten **asegurar calidad** en el producto.
- Dan **visibilidad** al proceso de desarrollo

55

AyDS2013 - Clase 2 - MMV

## Madurez en Desarrollo de SW

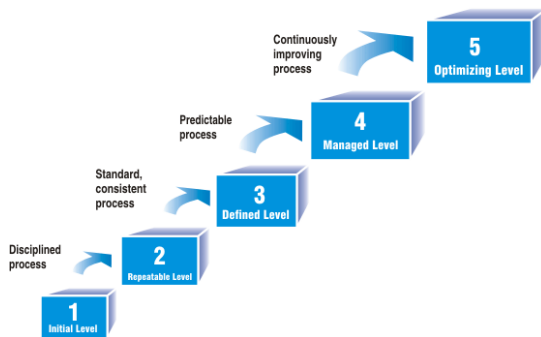
### CMMI– Modelo integrado de Capacidad y Madurez

- El desarrollo de sistemas en una organización madura **disminuye costos y tiempo de producción** y **aumenta la productividad y la calidad del producto**.
- El **Modelo de Integrado de Capacidad y Madurez (CMMI)**, Desarrollados por Software Engineering Institute (SEI) provee un estándar de referencia para evaluar el nivel de madurez del desarrollo de sistema de una organización.

56

AyDS2013 - Clase 2 - MMV

## Nivel de Madurez



57

AyDS2013 - Clase 2 - MMV

## Nivel de Madurez

Nivel	Características	Resultados
Inicial	<ul style="list-style-type: none"> <li>- Ausencia de gestión de proyectos.</li> <li>- El proceso de software es cambiante e irregular:</li> <li>- Los planes, estimaciones y calidad son impredecibles.</li> <li>- El rendimiento depende de la capacidad individual de los miembros del grupo.</li> <li>- Se establecen programas de formación del personal de desarrollo y mantenimiento.</li> </ul>	Productividad y calidad escasa. Riesgo máximo
Repetible	<ul style="list-style-type: none"> <li>- Los procesos de software son estables y repetibles.</li> <li>- La organización establece políticas de gerencia de proyectos y procesos.</li> <li>- La planificación se basa en proyectos similares.</li> <li>- Existen estándares definidos y exigidos.</li> <li>- El proceso se enmarca en un sistema de gerencia de proyectos basado en experiencias pasadas.</li> </ul>	Productividad y calidad baja. Riesgo alto.

## Nivel de Madurez

Nivel	Características	Resultados
Definido	<ul style="list-style-type: none"> <li>- Los procesos son definidos: estandarizados, documentados e institucionalizados.</li> <li>- Los procesos de ingeniería y gerencia son estables y se integran en uno sólo.</li> <li>- Existe un entendimiento común de los procesos, funciones y responsabilidades.</li> <li>- La organización mantiene un grupo dedicado a la definición, mejoramiento y difusión del proceso de Ingeniería de Software.</li> </ul>	Productividad y calidad media. Riesgo medio.
Gestionado	<ul style="list-style-type: none"> <li>- Los procesos son medibles o cuantificables</li> <li>- La productividad y la calidad se miden y registran para cada proyecto de la organización.</li> <li>- Se fijan metas cuantitativas de la calidad del software.</li> <li>- Mediante el uso de métricas de software, se crea una base cuantitativa para la evaluación y estimación en proyectos futuros.</li> </ul>	Productividad y calidad alta. Riesgo mínimo.
Optimizado	<ul style="list-style-type: none"> <li>- Los procesos se mejoran continuamente.</li> <li>- La organización busca lograr el nivel máximo de capacidad.</li> <li>- Se incorporan nuevas tecnologías y métodos para mejorar los procesos.</li> </ul>	Productividad y calidad total. Riesgo nulo.

## Temas de la Clase de hoy

- El producto
- El proceso
  - Modelos de procesos (Ciclos de vida).

### Bibliografía

- *Fundamentals of Software Engineering*– C. Ghezzy. Capítulo 7.
- *Ingeniería del Software – Un enfoque Práctico*. R. Pressman. Capítulo 3.
- Otras lecturas sugeridas.
  - *Cualquier libro de IS incluye modelos de proceso*

60

AyDS2013 - Clase 2 - MMV