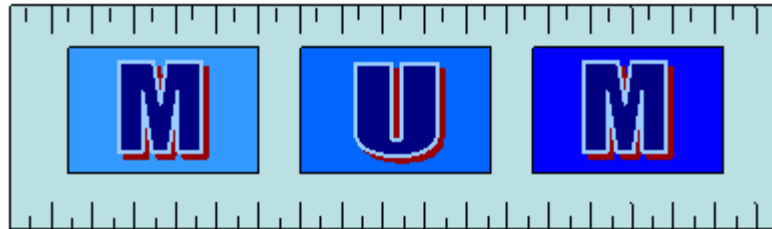


# Modelo de Desarrollo de Software Modularizado, Unificado y Medible



**Grupo de Ingeniería de Software (GrIS)  
INCO - Facultad de Ingeniería  
Universidad de la República  
Montevideo, Uruguay**

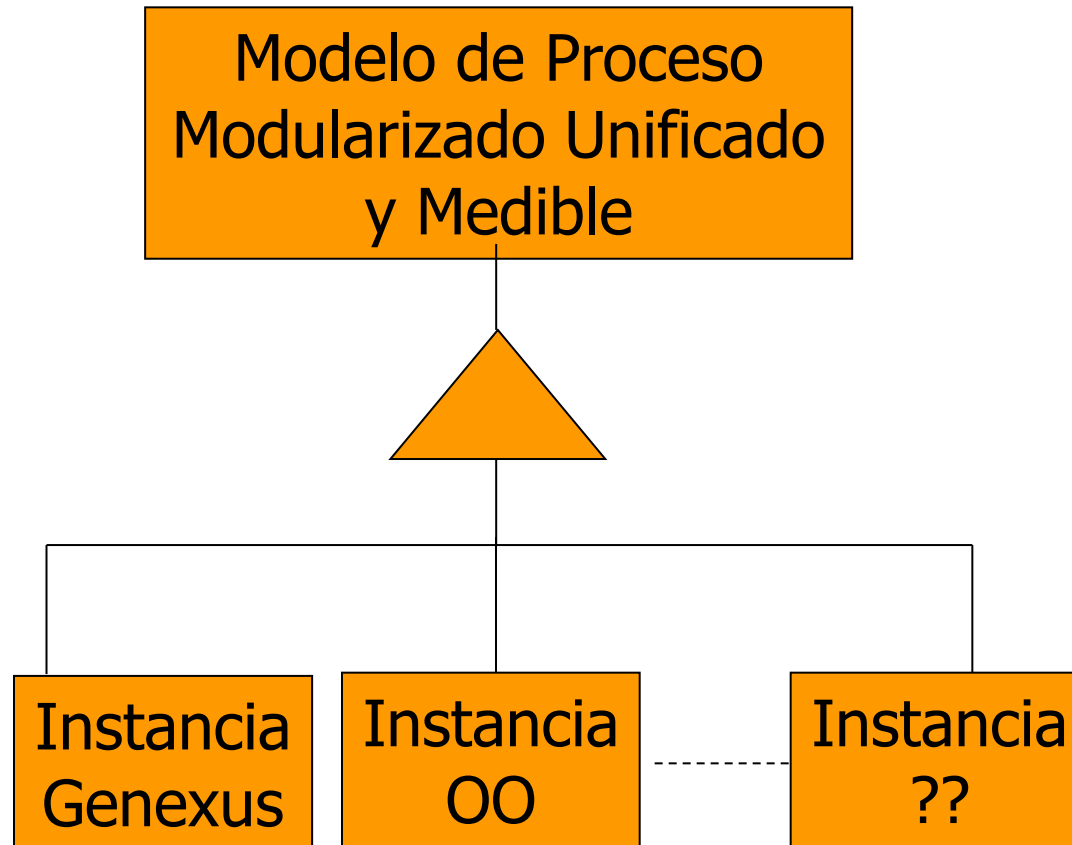


# TEMARIO

- Principales Características
- Dimensión del Tiempo
  - Fases, iteraciones, objetivos
- Dimensión del Modelo
  - Disciplinas, Actividades, Entregables, Roles



# El proceso...



# Principales características

- Iterativo incremental
  - Desarrollo del sistema en refinamientos sucesivos incrementando la solución definida
  - Mitigación de riesgos en cada iteración
- Basado en Casos de Uso
  - Captura de requerimientos mediante Casos de Uso que guían el diseño, implementación y verificación del software en desarrollo
- Centrado en la Arquitectura
  - la arquitectura prioriza los Casos de Uso más significativos y especifica la estructura del Sistema
  - Definición y construcción temprana del esqueleto base de la solución definida
- Modelado en UML



# Beneficios del Enfoque Iterativo

- Los riesgos son identificados y mitigados
  - Los principales riesgos que ayuda a mitigar:
    - Construir el sistema equivocado
    - Problemas en Integración
    - Arquitectura
- Permite planificar el cambio en la próxima iteración
- Alto nivel de reutilización
  - Se identifican partes comunes ya implementadas



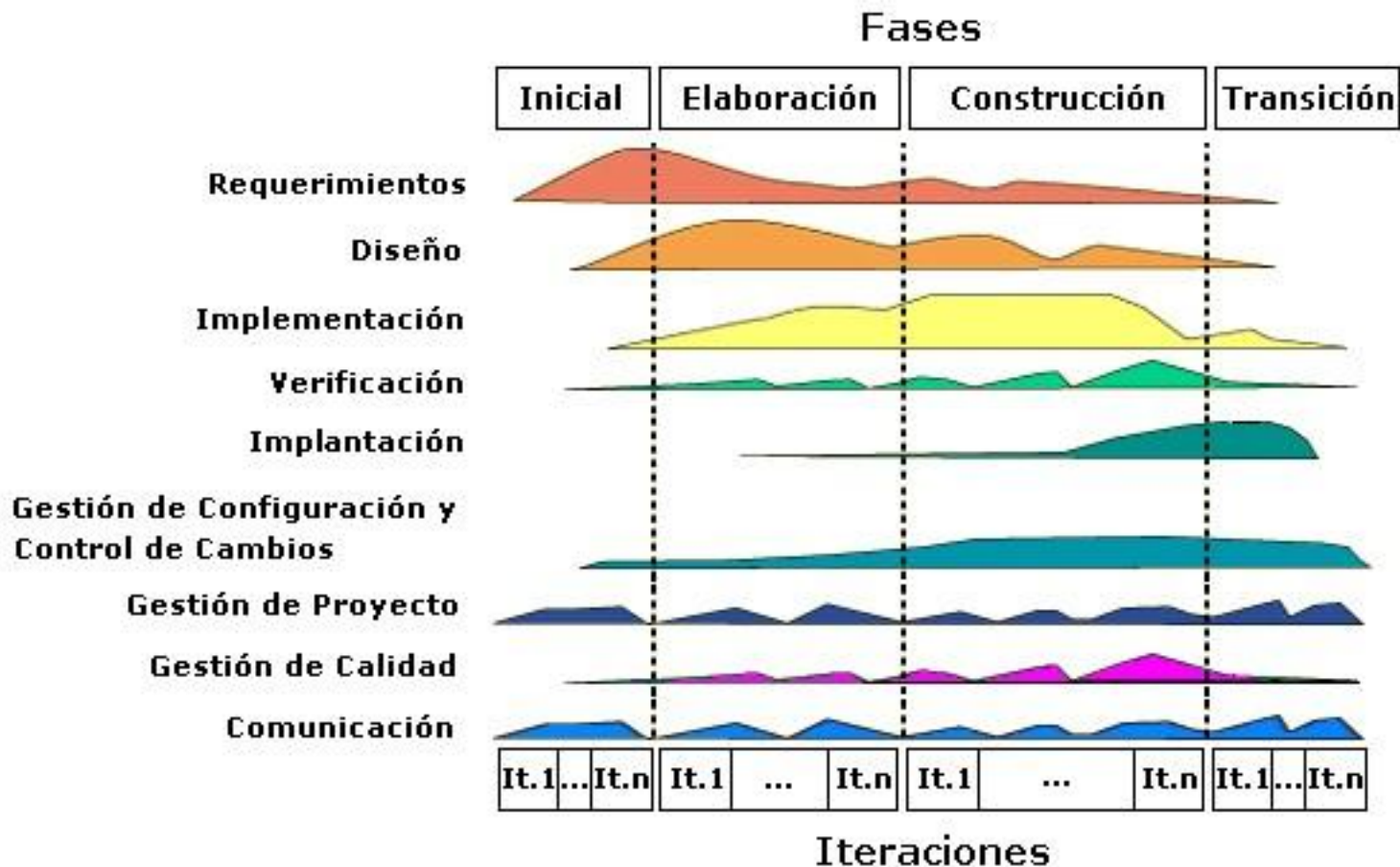
# Beneficios del Enfoque Iterativo

- El equipo aprende a lo largo del proyecto
  - Los distintos roles comienzan su trabajo con el proyecto
  - Los verificadores prueban desde el principio, etc
  - El proceso puede ser mejorado sobre la marcha
- El producto logra una mejor calidad global
  - Los requerimientos están relacionados con las necesidades de los usuarios
  - El producto fue verificado varias veces



# Dimensiones del Proceso

- Tiempo: Fases, iteraciones, objetivos
- Modelo: Disciplinas, roles, actividades, entregables



# Dimensión del Tiempo

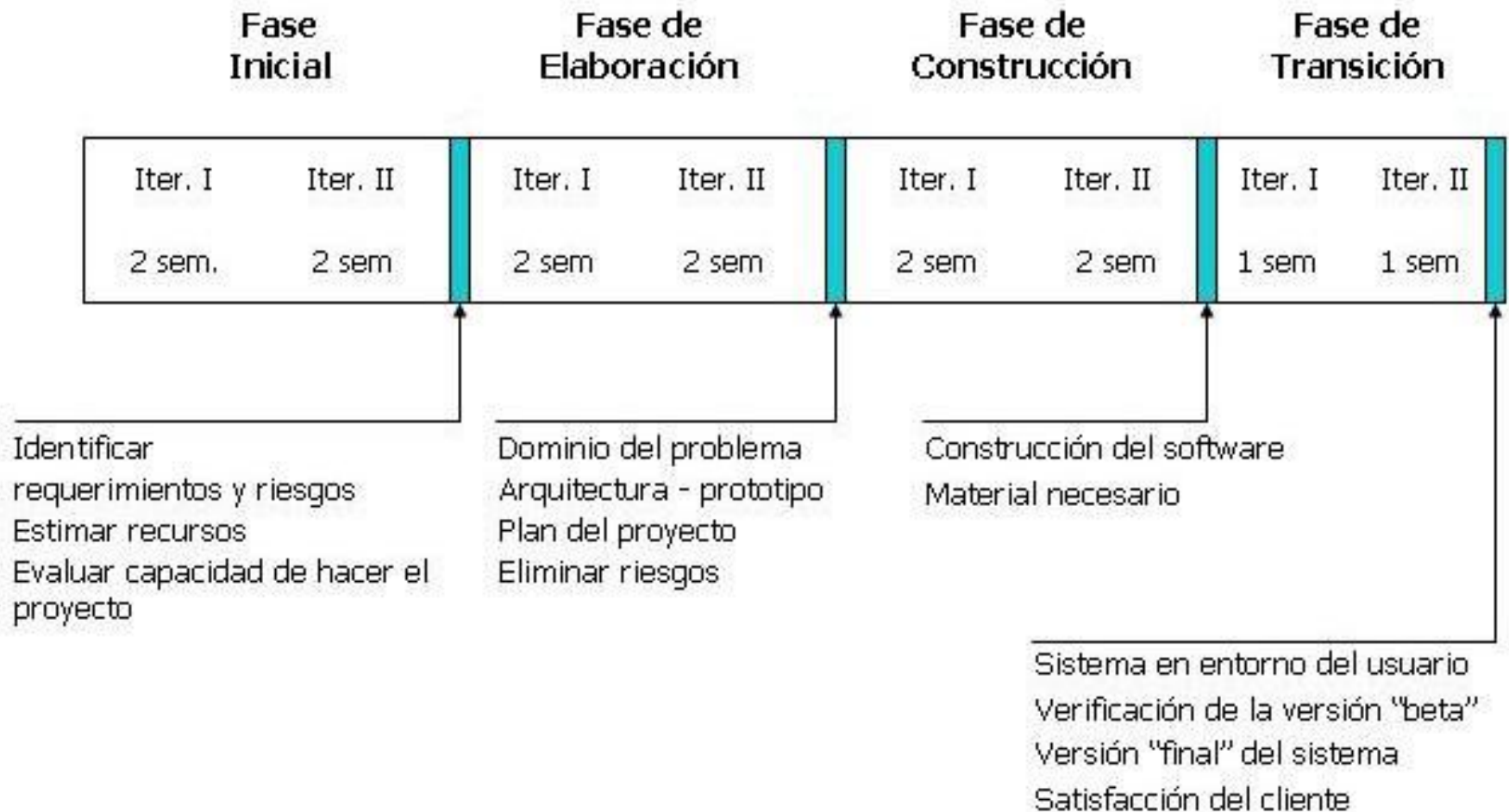
- El desarrollo se divide en cuatro fases: Inicial, Elaboración, Construcción y Transición
- Cada fase se divide en iteraciones y tiene objetivos definidos que se alcanzan según entregables que se deben obtener

Fases	Inicial				Elaboración				Construcción				Transición	
Iteraciones	Iter. I		Iter. II		Iter. I		Iter. II		Iter. I		Iter. II		Iter. I	
Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14

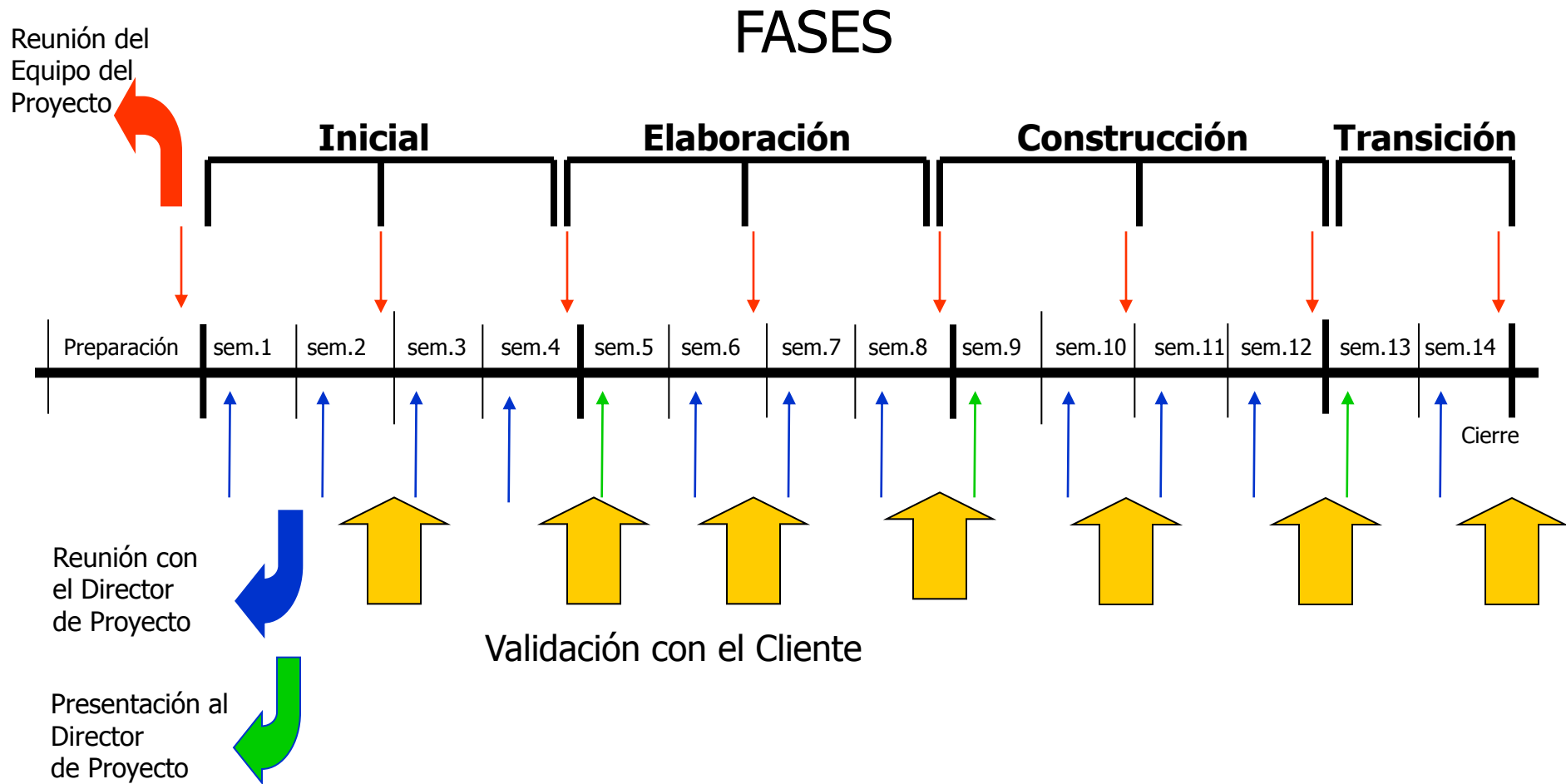




# Fases y Objetivos



# Agenda de Reuniones a tener en cuenta



# Seguimiento del Curso

- **Revisión con el Director:**
  - monitoreo semanal del grupo (roles seleccionados) por docente del curso, revisión de entregables
  - reunión de coordinación de directores y encargado del curso para poner en común y nivelar
- **Auditorias del proceso:** revisión del avance y estado de los grupos por docentes del curso y estudiantes que definieron el proceso
- **Encuestas de satisfacción:** cuestionarios finales sobre el proceso a estudiantes, directores y clientes
- **Memoria Organizacional:** sitio web con los procesos utilizados en cada año y los entregables generados por los grupos



# Reunión de Equipo

En esta reunión se debe:

- Evaluar el cumplimiento del Plan de la iteración (pasada)
  - productos obtenidos (alcance y calidad)
  - carga de trabajo
- Identificar causas de desvíos y problemas
- Analizar los riesgos que enfrenta el proyecto
- Definir acciones a tomar
- Ajustar el Plan de la iteración (nueva)



# Plan de la Iteración

- Cada actividad se identifica por un código, el que sirve para registrar el trabajo realizado en la planilla de Registro de Actividad.
- Se debe indicar además la descripción, las fechas de inicio y fin planificadas, el esfuerzo (en horas/persona) planificado, los participantes previstos y el responsable.
- Las actividades son:
  - Las que surgen del modelo de proceso (agrupadas de forma adecuada para la planificación y el control)
  - y actividades necesarias que no figuran en el modelo (estudio del dominio, instalación de herramientas, etc.)



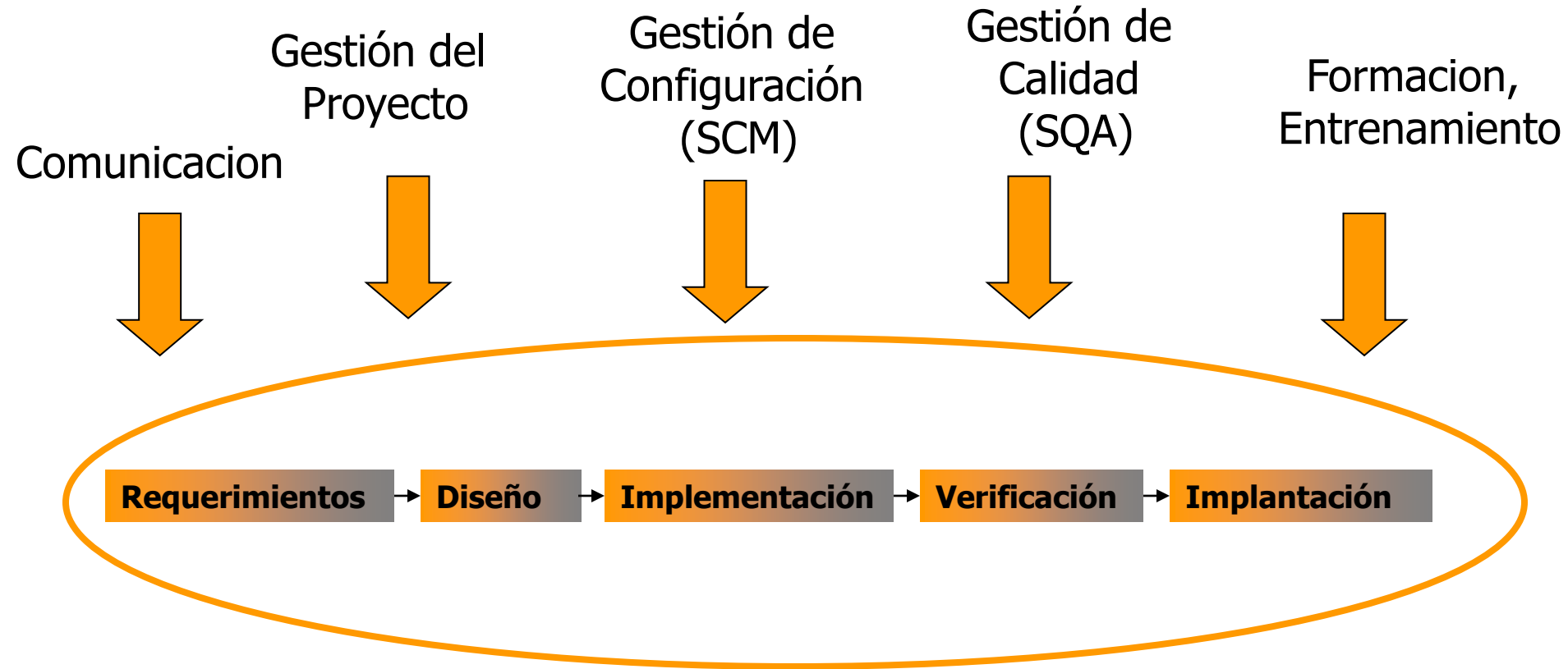
# Dimensión del Modelo

- Cuatro elementos para mostrar quién está haciendo qué, de qué forma (cómo) y en qué momento (cuándo)
  - **Disciplinas**: describen cuándo, agrupan actividades en forma lógica
  - **Roles**: describen quién, responsabilidades
  - **Actividades**: describen cómo, las acciones
  - **Entregables**: describen qué, artefactos



# Disciplinas

visión en cada iteración



# Actividades y Entregables

- Actividades
  - Conjunto de acciones en una disciplina para crear o actualizar uno o varios entregables
    - Pre-condiciones: entregables de entrada
    - Participantes: roles asociados
    - Descripción: tareas a realizar
    - Post-condiciones: entregables de salida
- Entregables
  - Productos tangibles del proyecto que son entrada y salida de las actividades
    - Contenido: Plantilla que sirve como guía
    - Responsable: rol asociado







# Requerimientos

- Establecer y mantener un acuerdo con los clientes e involucrados sobre qué debe hacer el sistema.
- Proporcionar a los desarrolladores un mejor entendimiento de los requerimientos del sistema.
- Definir el alcance del sistema.
- Proporcionar las bases para la planificación del contenido técnico de las iteraciones.
- Proporcionar las bases para estimar costo y tiempo para desarrollar el sistema.
- Definir la interfaz de usuario del sistema, enfocando en las necesidades y objetivos de los usuarios
- Para lograr estos objetivos, es importante entender la definición y alcance del problema que trata de resolver este sistema.





# Diseño

- Transformar los requerimientos en el diseño de lo que debe ser el sistema.
- Desarrollar una arquitectura robusta para el sistema.
- Adaptar el diseño para que se corresponda con el entorno de implementación, diseñando de forma que la implementación tenga buena performance.
- Para lograr estos objetivos es necesario conocer el alcance del sistema, las herramientas de desarrollo y los recursos disponibles





# Implementación

- Definir la organización del código en términos de implementación de subsistemas organizados en capas.
- Implementar clases y objetos en términos de componentes (archivos fuentes, binarios, ejecutables y otros).
- Verificar los componentes desarrollados como unidades.
- Integrar los resultados producidos por cada implementador (o por equipos), en un único sistema ejecutable.
- En esta disciplina se implementan los distintos subsistemas y luego se integran para formar un sistema





# Verificación

- Verificar la interacción entre componentes.
- Verificar la correcta integración de todos los componentes del sistema.
- Verificar que todos los requerimientos han sido correctamente implementados.
- Identificar y asegurar que los defectos sean corregidos antes de la liberación de la versión del producto de software.





# Implantación

- Algunas formas:
  - Instalación personalizada
  - Ofrecimiento del producto empaquetado
  - Acceso al software por internet
- Si bien la mayoría de las actividades de Implantación se realizan en la Fase Transición ...
- Algunas comienzan en fases anteriores,
  - Ejemplo: la Planificación de la Implantación





# Gestión de Proyecto

- Proveer un ambiente para manejar proyectos de software intensivos.
- Proveer una guía práctica para la planificación, gestión de recursos humanos, ejecución y monitoreo de los proyectos.
- Proveer un ambiente para la Gestión de riesgos.
- Realizar un seguimiento del avance del proyecto.
- Realizar estimaciones y mediciones: de esfuerzo (en horas de trabajo), tamaño del producto en desarrollo y factibilidad del proyecto.





# Gestión de Configuración (SCM)

- Identificar los elementos del proyecto que deben estar bajo configuración
- Restringir los cambios a dichos elementos
- Auditar los cambios a estos elementos
- Definir y gestionar la configuración de estos elementos





# Gestión de Calidad (SQA)

- Identificar las propiedades de calidad
- Realizar el seguimiento y control de la calidad del sistema en desarrollo.
- Realizar el seguimiento y control de la calidad del proceso, las métricas y los procedimientos seguidos en el proyecto.
- Indicar qué acciones correctivas deben tomarse en caso de encontrar inconsistencias o incumplimientos de la calidad.
- Asegurar que el sistema desarrollado cumple con determinadas propiedades de calidad preestablecidas.







# Comunicación

- Definir los métodos de comunicación y herramientas utilizadas para la comunicación entre los integrantes del equipo.
- Definir los métodos de comunicación y herramientas utilizadas para la comunicación de los integrantes del equipo con entidades externas al equipo (Director de proyecto, docentes, Cliente, usuarios, etc...).
- Convocar reuniones informativas donde se traten temas de interés para los participantes.
- Elaborar documentos informativos sobre temas de interés para los integrantes del equipo.
- Realizar el seguimiento de la Satisfacción del Cliente.





# Formación y Entrenamiento

- Proporcionar al equipo de trabajo la formación necesaria así como también el entrenamiento y la capacitación para lograr una correcta comprensión del modelo y sus actividades.



# Roles y combinación de roles

- Un rol define el comportamiento y las responsabilidades de la/s persona/s
  - Comportamiento --> actividades en que participa
  - Responsabilidades --> entregables asociados
- Combinación de roles según carga de trabajo en el tiempo, compatibilidad de tareas.



# Roles

- Analista
  - Requerimientos, arquitectura, diseño
- Arquitecto
  - Visión global, coordinador de pasaje a desarrollo
- Especialista Técnico
  - Es la referencia en las herramientas utilizadas
- Implementador
  - Desarrollo, pruebas unitarias
- Resp. de Verificación
  - Gestión de las pruebas de integración, funcionales y del sistema
  - Diseño de pruebas



# Roles

- Administrador
  - Gestión del proyecto, comunicación
  - Planificación del proyecto
  - Seguimiento y control
  - Estimaciones y mediciones
- Resp. de SQA
  - Identificar y asegurar que se cumplan la propiedades de calidad del producto
  - Asegurar que se siga el proceso
  - Asegurar la calidad de los productos intermedios



# Roles

- Resp. de SCM
  - Definir productos bajo gestión de configuración
  - Definir ambiente controlado y establecer los distintos procedimientos y formas de trabajo para utilizarlo por parte del equipo
  - Controlar las versiones y los cambios y auditar la línea base del software
  - Crear ejecutables al final de cada iteración
- Documentador de Usuario
  - Diseño de interfase, manual de usuario
- Asistente de Verificación
  - Diseño y ejecución de las pruebas
  - Reporte de incidentes





# Roles para Grupos Java/.Net /Ruby

<b>Combinación de Roles personas</b>	<b>Cantidad de</b>	15
Administrador-Asistente de Verificación-Responsable de la Comunicación	1	1
Analista-Documentador de Usuario-Asistente de Verificación	1	1
Analista-Implementador	4	4
Responsable de SQA – Asistente de Verificación	1	1
Analista-Diseñador de Interfaz de Usuario-Implementador	1	1
Responsable de Verificación - Asistente de SQA	1	1
Arquitecto - Asistente de Verificación - Coordinador de Desarrollo *	1	1
Especialista Técnico - Implementador -Responsable de Integración +	4	4
Responsable de SCM - Especialista Técnico - Implementador	1	1

**Grupo de Ingeniería de Software (GrIS)**



# Roles para Grupos Genexus



Combinación de Roles	Cantidad de personas
Administrador-Asistente de Verificación-Responsable de la Comunicación	15
Analista-Documentador de Usuario-Asistente de Verificación	1
Analista-Implementador	1
Analista-Implementador	5
Analista-Implementador - Responsable del Núcleo	1
Responsable de SQA – Asistente de Verificación	1
Analista-Diseñador de Interfaz de Usuario-Implementador	1
Responsable de Verificación - Asistente de SQA	1
Arquitecto - Asistente de Verificación - Coordinador de Desarrollo *	1
Especialista Técnico GeneXus y Base de Datos-Implementador	1
Especialista Técnico - Implementador - Responsable de Consolidado +	1
Responsable de SCM - Implementador-Especialista Técnico del Lenguaje y Configuración	1

**Grupo de Ingeniería de Software (GrIS)**



# ¿Preguntas?

- <http://www.fing.edu.uy/inco/cursos/pis/>

