

CC51A – Ingeniería de Software

El Estándar de Ingeniería de Software de la “European Space Agency (ESA)”

Sergio Ochoa D.

Estructura de la Presentación

- Fases, actividades e hitos establecidos por el Estándar.
- Conclusiones

Fases, Actividades e Hitos

- ◆ El ciclo de vida del software se inicia cuando el producto de software se concibe, y termina cuando ya no está disponible para su uso (el producto es retirado).
- ◆ Un modelo de ciclo de vida estructura las actividades de un proyecto en fases, y define las actividades y los objetivos de cada fase.

Fases, Actividades e Hitos

◆ Fases del Ciclo de Vida:

- UR: Definición de Requisitos de Usuarios.
- SR: Definición de Requisitos de Software.
- AD: Definición del Diseño Arquitectónico.
- DD: Diseño Detallado y producción del código.
- TR: TRansferencia del software a operaciones.
- OM: Operación y Mantenimiento.

Fases, Actividades e Hitos

PHASES ITEMS	UR User Requeriments Definition	UR/R	SR Software Requeriments Definition	SR/R	AD Achitectural Design	AD/R	DD Detailed Design and Production	DD/R	TR Transfer	OM Operations and Manteinance		
MAJOR ACTIVITIES	<ul style="list-style-type: none"> determination of operational environment identification of users requirements 		<ul style="list-style-type: none"> construcción of logical model identification of software requirements 		<ul style="list-style-type: none"> construcción of phisical model definition of major components 		<ul style="list-style-type: none"> module design coding unit test integration test system test 		<ul style="list-style-type: none"> installation provisional acceptance tests 	<ul style="list-style-type: none"> final acceptance test operations maintenance of code and documentation 		
DELIVERABLE ITEMS	User Requirements Document pág. 72	URD →	Software Requirements Document pág. 73	SRD →	Architectural Design Document pág. 74	ADD →	Detailed Design Document págs. 75/76 Code SUM Software User Manual	DDD → Code → SUM →	Software Transfer Document pág. 76	STD →	Project History Document pág. 77	PHD
REVIEWS (See Checklist Apendix D)		tech. review pág.82	tech. walkthroughs Inspections pág. 83	tech. review pág.83	tech. walkthroughs Inspections pág. 84	tech. review pág.84	tech. walkthroughs Inspections pág. 85	tech. review pág.85	pág. 86		pág. 86	
MAJOR MILESTONES		URD approved	SRD approved		ADD approved		Code/DDD/SUM approved		STD delivered Provisional Acceptance		PHD delivered Final Acceptance	

Fases, Actividades e Hitos

Activity Plan	USER REQUIREMENTS REVIEW		SOFTWARE REQUIREMENTS DEFINITION		ARCHITECTURAL DESIGN		DETAILED DESIGN AND PRODUCTION	
	Activity	Output	Activity	Output	Activity	Output	Activity	Output
Software Project Management	Estimate project cost Plan SR phase WBS & staffing Outline plan for whole project	SPMP/SR Pág 77 y 47	Estimate project cost to 30% accuracy Plan AD phase WBS & staffing	SPMP/AD Pág 77 y 47	Estimate project cost to 10% accuracy Plan DD phase WBS & staffing	SPMP/DD Pág 77 y 47	Detail DD phase WBS Plan TR phase WBS & staffing	SPMP/Dd updates SPMP/TR Pág 77 y 47
Software Configuration Management	Define SR phase procedures for: -Documents -Case tools prod. -Prototype code	SCMP/SR Pág 78 y 51	Define AD phase procedures for: -Documents -Case tools prod. -Prototype code	SCMP/AD Pág 78 y 51	Define DD phase procedures for: -Documents -Case tools prod. -Prototype code	SCMP/DD Pág 78 y 51	Define operational environment procedures for: -Documents -Deliverable code	SCMP/TR Pág 78 y 51
Software Verification & Validation	Define SR phase review and traceability procedures Plan acceptance tests	SVVP/SR Pág 79 SVVP/AT Pág 57	Define AD phase review and traceability procedures Plan system tests	SVVP/AD Pág 79 SVVP/ST Pág 57	Define DD phase review and traceability procedures Plan integration tests	SVVP/DD Pág 79 SVVP/IT Pág 57	Define follow tests: Acceptance, System, Integration Plan and define unit tests	SVVP/AT updates SVVP/ST updates SVVP/IT updates SVVP/UT updates Pág 79/57
Software Quality Assurance	Plan SR phase monitoring activities Outline plan for whole project	SQAP/SR Pág 81 y 62	Plan AD phase monitoring activities	SQAP/AD Pág 81 y 62	Plan DD phase monitoring activities	SQAP/DD Pág 81 y 62	Plan TR phase monitoring activities	SQAP/TR Pág 81 y 62

Fases, Actividades e Hitos

- Hitos principales de ESA:
 - Aprobación del Documento de Requisitos de Usuario (URD);
 - Aprobación del Documento de Requisitos de Software (SRD);
 - Aprobación del Documento de Diseño Arquitectónico (ADD);
 - Aprobación del Documento de Diseño Detallado (DDD), el Manual de Software de Usuario (SUM), el código, y la declaración de terminación para el testeo de aceptación provisional;
 - Declaración de aceptación provisional y la confección del Documento de Transferencia de Software (STD);
 - Declaración de aceptación final y entrega del Documento de Historia del Proyecto (PHD).

Fase de Requisitos de Usuarios

- ◆ Fase de definición del problema.
- ◆ Los requisitos del usuario deben ser “capturados”.
- ◆ Entrevistas, encuestas, formularios, prototipos.
- ◆ Se debe generar un “Documento de Requisitos de Usuario” (URD).

Fase de Requisitos de Usuarios

- ◆ El apoyo de los desarrolladores en esta etapa varía según la familiaridad de los usuarios con el software.
- ◆ Siempre debe producirse un URD. La revisión (UR/R) es hecha por los usuarios, los ingenieros de software y hardware, y por el administrador del proyecto.

Fase de Requisitos de Usuarios

- ◆ Antes de completar la UR/R debe construirse un Plan de Administración del Proyecto de Software que muestre todas las fases siguientes, con estimación de recursos.

Fase de Requisitos de Software

- ◆ Fase de análisis del proyecto.
- ◆ Parte vital de la descripción del modelo, que especifica lo que debe hacer el software, y no cómo debe hacerlo.
- ◆ Puede ser necesario escribir prototipos para clarificar requisitos de software y completar requisitos de usuarios.

Fase de Requisitos de Software

- ◆ El entregable principal es el “Documento de Requisitos de Software” (SRD).
- ◆ Cada proyecto de software debe tener este documento.
- ◆ Debe omitirse terminología de implementación.
- ◆ Revisado por los usuarios, los ingenieros de software y hardware, y por el administrador del proyecto, durante la “Revisión de Requisitos de Software” (SR/R).

Fase de Requisitos de Software

- ◆ En esta fase se debe revisar el Plan de Administración del Proyecto de Software.
- ◆ Mejores esfuerzos para tener estimaciones con errores no mayores al 30%.
- ◆ Debe construirse un plan detallado para la fase AD.

Fase de Diseño Arquitectónico

- ◆ El propósito es el de definir la estructura del software. El modelo construido en la fase SR es el punto de partida.
- ◆ Este modelo es transformado en diseño arquitectónico asignando funciones a componentes de software y definiendo el control y flujo de datos entre ellos.

Fase de Diseño Arquitectónico

- ◆ Esta fase puede considerar varias iteraciones del diseño.
- ◆ Deben ser identificadas las dificultades técnicas o partes críticas del diseño.
- ◆ Puede ser necesario realizar prototipos del software para confirmar las suposiciones básicas del diseño.

Fase de Diseño Arquitectónico

- ◆ El ítem entregable que constituye la salida formal de esta fase es el Documento de Diseño Arquitectónico (ADD).
- ◆ El ADD debe ser producido para cada proyecto de software.
- ◆ El ADD debe ser formalmente revisado por los ingenieros de hardware y de software, por los usuarios, y por el administrador del proyecto, durante el proceso de Revisión del Diseño Arquitectónico (AD/R).

Fase de Diseño Arquitectónico

- ◆ Durante la fase AD, debe construirse un Plan de Administración de Proyecto de Software, que describa el resto del proyecto.
- ◆ Este plan debe contener una estimación del costo del proyecto, apuntando a un margen de error no mayor al 10%.
- ◆ También se debe producir planes detallados para la fase DD.

Fase de Diseño Detallado y Producción

- ◆ El propósito de esta fase es el de detallar el diseño del software, codificarlo, documentarlo y testearlo.
- ◆ En forma concurrente con la codificación y testeo, se produce el Documento de Diseño Detallado (DDD) y el Manual de Software de Usuario (SUM).
- ◆ Inicialmente, el DDD y SUM contienen las secciones correspondientes a los niveles superiores del sistema. A medida que el diseño progresa a niveles más bajos, se añaden subsecciones relacionadas.

Fase de Diseño Detallado y Producción

- ◆ Al final de la fase, los documentos están completos y, junto con el código, constituyen los ítems entregables de esta fase.
- ◆ Durante esta fase, debe realizarse las actividades de testeo unitario, de integración y de sistema de acuerdo con los planes de verificación establecidos en las fases SR y AD.
- ◆ También debe verificarse la calidad del software.

Fase de Diseño Detallado y Producción

- ◆ Los 3 entregables (código, DDD y SUM), deben ser revisados formalmente por los ingenieros de software y el administrador, durante el proceso de Revisión del Diseño Detallado (DD/R).
- ◆ Al final del proceso de revisión, el software puede considerarse como listo para el testeo de aceptación provisional.

Fase de Transferencia

- ◆ El propósito de esta fase es el de establecer que el software cumple con los requisitos especificados en el URD.
- ◆ Esto es hecho instalando el software y realizando tests de aceptación.
- ◆ Si el software demuestra que provee las capacidades requeridas, el software puede ser aceptado provisionalmente y con ello, puede empezar la operación.

Fase de Transferencia

- ◆ Durante la fase TR, debe producirse el Documento de Transferencia de Software (STD), que documente el proceso de transferencia del software al equipo de operaciones.

Fase de Operación y Mantenimiento

- ◆ Una vez que el software ha entrado en operación, debe ser monitoreado cuidadosamente para confirmar que cumple con los requisitos definidos en el URD.
- ◆ Algunos requisitos, tales como los referentes a disponibilidad, puede tomar algún tiempo validarlos. Cuando el software ha pasado todos los tests de aceptación, entonces recién puede ser finalmente aceptado.

Fase de Operación y Mantenimiento

- ◆ El Documento de Historia del Proyecto (PHD) resume la información administrativa de importancia, la cual se ha acumulado en el transcurso del proyecto.
- ◆ Este documento debe ser generado después de la aceptación final.
- ◆ Debe ser re-hecho al final del ciclo de vida, con información acumulada durante la fase OM.

Fase de Operación y Mantenimiento

- ◆ Después de la aceptación final, el software puede ser modificado para corregir errores no detectados durante fases anteriores, o porque aparecen nuevos requisitos. A esto se le denomina mantenimiento.
- ◆ Durante todo el período de operación, se debe hacer un esfuerzo especial para mantener la documentación al día.
- ◆ Información sobre fallas y caídas debe ser registrada para establecer datos para el establecimiento de métricas de calidad de software para proyectos siguientes.

Modelo del Proceso (cont.)

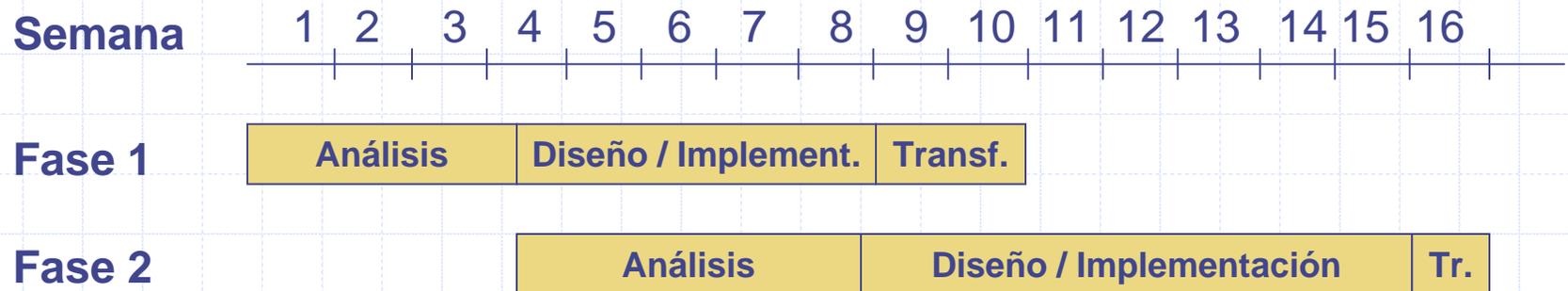
El modelo de proceso a utilizar en el curso es una adaptación del modelo propuesto por la ESA.

La adaptación apunta a:

- Simplificar el modelo original, haciéndolo más aplicable a proyectos chicos.
- Realizar un desarrollo incremental del producto final.
- Aumentar el paralelismo entre las tareas que realizan los miembros del equipo de trabajo.

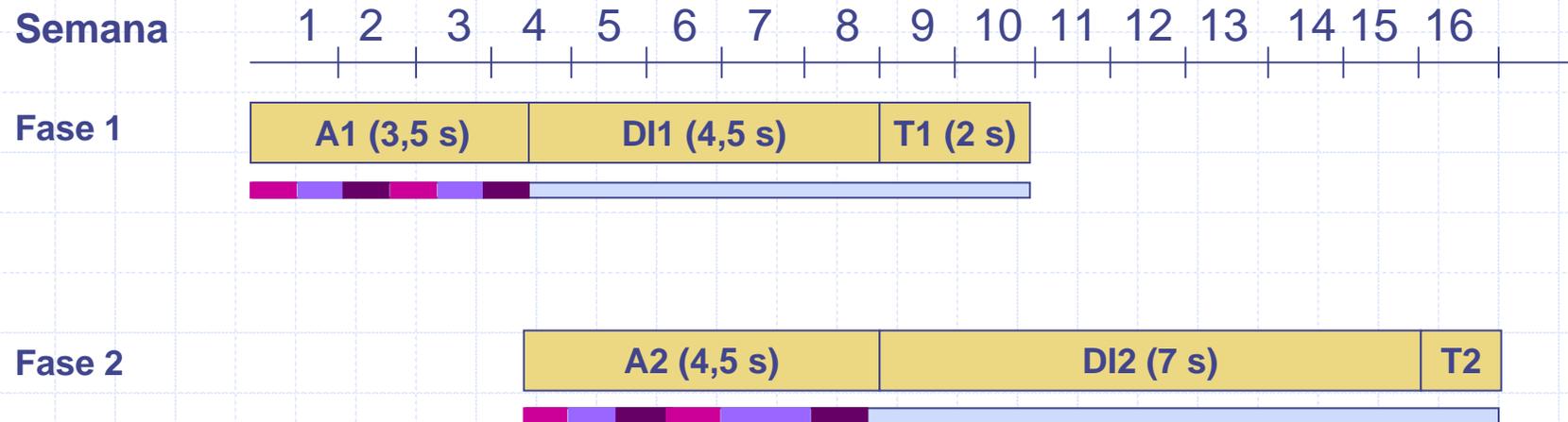
A continuación se muestra la dinámica del modelo de proceso adaptado.

Modelo del Proceso (cont.)



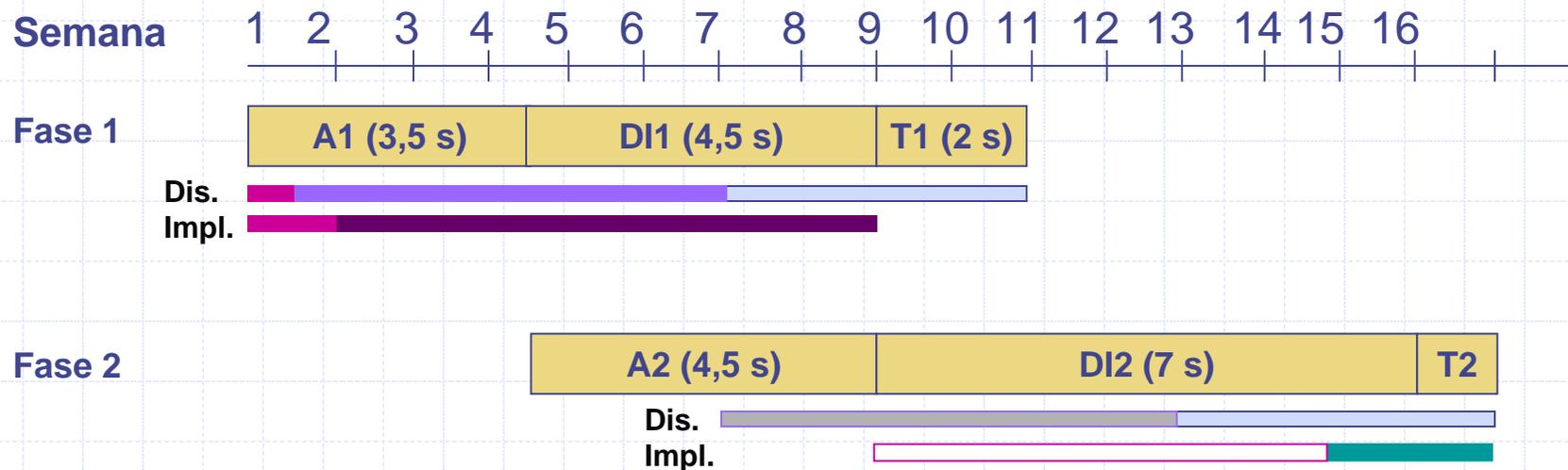
Modelo del Proceso (cont.)

Analistas



Modelo del Proceso (cont.)

Diseño / Implementación

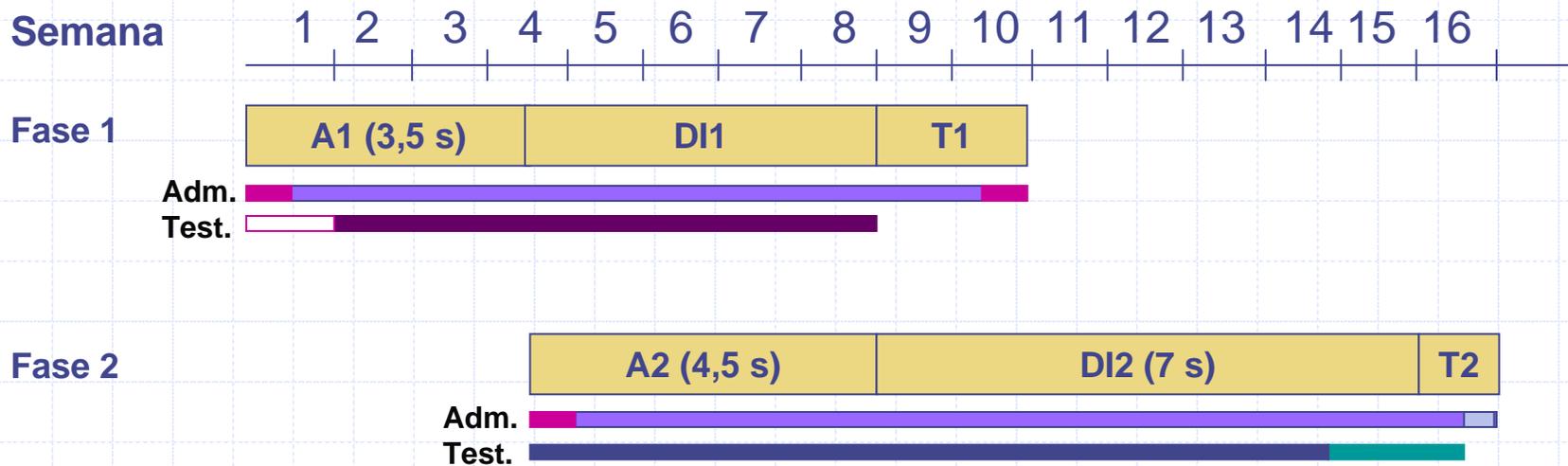


Referencias:

- Preparación de Herramientas y Definición de Reglas de Funcionamiento
- Diseño del Prototipo
- Implementación del Prototipo
- Control de Cumplimiento del Diseño
- Diseño del Producto
- Implementación del Producto
- Corrección y Ajuste del Producto

Modelo del Proceso (cont.)

Administración / Testing (~ SQA)



Referencias:

- Interacción con el Cliente
- Monitoreo, Redefinición y Coordinación de Tareas (Guiar el Barco)
- Entrega del Producto
- Planificación de Controles, y Definición del Tipo de Revisión.
- Control de Productos y Emisión de Alertas
- Prueba del Producto

Conclusiones ...

- ◆ El estándar de la ESA propone un conjunto de actividades para atacar el “ciclo de vida” de un proyecto de software.
- ◆ Puede ser adaptado a la realidad de una empresa específica.
- ◆ Puede ser aplicado en forma parcial, según la naturaleza del problema a resolver.
- ◆ Es simple, por lo tanto, con poca práctica se puede hacer un buen uso de él.
- ◆ Nos podría ayudar a certificar ISO 9000.

Bibliografía y Casos

- ◆ I. Sommerville, P. Sawyer. Requirements Engineering, A Good Practice Guide. Wiley, 1999.
- ◆ S. Robertson, J. Robertson. Mastering the Requirements Process. Addison-Wesley, 1999.
- ◆ ESA, SOFTWARE ENGINEERING STANDARDS, ISSUE 2. Preparado por: ESA Board for Software Standardisation and Control (BSSC). URL: <http://www.estec.esa.nl/wmwww/WME/index.html>