

Usabilidad en el proceso de desarrollo de SCRUM

Aplicación del marco de usabilidad desarrollado por el profesor Xavier Ferrè al proceso de desarrollo SCRUM

Danilo Spada

Curso: Interfaces de Usuario Avanzadas

Programa Oficial de Posgrado en Ingeniería Informática y de Telecomunicación -Master y Doctorado

Contenidos:

1. Resumen.....	3
2. Introducción	3
2.1 Objetivos del trabajo y estructura lógica de le memoria.....	4
3. Descripción del Proceso de Desarrollo Software de SCRUM.....	4
3.1 Visión global de SCRUM	5
3.2.1 Técnicas usadas en SCRUM	5
3.2.2 Fases de SCRUM.....	7
4. Marco de Integración de la Usabilidad	8
5. Instanciación del Marco de Integración	9
5.1 Técnicas seleccionadas	9
5.1.1 Personas	9
5.1.2 Prototipos de Papel.....	10
5.1.3 Especificaciones de Usabilidad.....	10
5.1.4 Inspecciones	10
5.1.5 Pensar en Voz Alta.....	10
5.1.6 Retroalimentación del Usuario.....	11
6. Actividades y ubicación de Técnicas seleccionadas en Scrum en los tres momentos de vida:	11
ventajas e inconvenientes.....	16
7. Conclusiones	16
8. Referencias:.....	17
9 Bibliografía Consultada	17

1. Resumen

En este trabajo se pretende ofrecer la descripción de un proceso de desarrollo ágil conocido con el nombre de Scrum y aplicar a este proceso el Marco de Integración de la Usabilidad desarrollado por el profesor Xavier Ferré en su tesis doctoral^[1]. Se presentan entonces las técnicas y las actividades de IPO elegidas y se indican los momentos de aplicación en un posible caso concreto.

2. Introducción

Según el ISO/IEC 9126, la calidad de un producto software depende de seis factores: funcionalidad, fiabilidad, eficiencia, usabilidad, mantenibilidad y portabilidad. Sin embargo, aunque la mayoría de estos factores se tienen en cuenta a la hora de plantear el desarrollo de un sistema software, la usabilidad muchas veces no tiene un papel adecuado. Pero ¿que se entiende por usabilidad de un sistema software? Pues es la medida de cuanto el sistema sea sencillo de aprender y utilizar, de la baja probabilidad de cometer errores durante su uso, incluyendo además la satisfacción y la eficiencia de los usuarios finales.

Durante los últimos años, sobre todo debido a la gran afluencia de una multitud de usuarios con diferentes perfiles a muchas aplicaciones a través Internet, el tema de la usabilidad está cobrando una importancia cada día mas relevante. La usabilidad determina cuanto eficientemente y cómodamente un usuario alcanza su objetivo. Si nos fijamos que los sistemas software a menudo pretenden intervenir en la realización de alguna tarea más o menos compleja, ayudando y optimizando el esfuerzo del usuario, pues es indispensable que éste pueda aprovechar la ayuda sin sufrir. Interfaces demasiado complejas o procedimientos largos y poco intuitivos, comportamientos incomprensibles del sistema, etc. pueden volver estresante y agotante tareas que deberían resultar más sencillas y controladas.

En el desarrollo de una nueva aplicación es imprescindible tener en cuenta que un producto, hasta muy sofisticado, pero no usable, puede ser un fracaso total; aquí vale la regla: si el usuario no encuentra o no entiende una funcionalidad, ¡la funcionalidad no existe! Aun más: puede que sea necesario afrontar un coste muy alto para arreglar faltas en usabilidad: se tome como ejemplo lo que pasó en EEUU en el 2000 cuando hubo que resolver el problema de la dificultad de comprensión y utilización de papeletas electoral para votar el nuevo presidente!

Craig Larman sostiene que “no hay probablemente ninguna otra técnica con mayor disparidad entre su importancia para el éxito del desarrollo de software y la falta de una atención y educación formal que la ingeniería de usabilidad y el diseño de las interfaces de usuario”^[2]. Muchos esfuerzos se están haciendo para demostrar la oportunidad de invertir en usabilidad y hay trabajos que muestran como para cada \$ invertido en usabilidad, se obtienen de \$10 hasta 100\$^[3].

Está claro entonces, que los ingenieros deberían tener más en cuenta la usabilidad en sus proyectos, y eso durante todas las fases del desarrollo, desde el diseño hasta la evaluación. Sin embargo en los estándares mas utilizados para desarrollar proyectos software (IEEE Std 1074-1997^[4], Modelo en cascada, Modelo incremental^[5], Extreme Programming^[6], SCRUM^[7]) no está muy claro que medidas tomar para mejorar este aspecto a lo largo de todo el ciclo de vida y hasta ahora no han sido añadido "etapas" vueltas precisamente a obtener productos usables. De alguna manera se puede pensar a este problema como a una independencia todavía "impermeable" entre IPO (Interacción Persona-Ordenador) y IS (Ingeniería del software)^{[8][9][10]}.

Sin embargo la industria del software está creciendo cada día más y las técnicas IPO empiezan a ser maduras y está naciendo la necesidad de integrar los conocimientos de estos dos campos para que

los productos finales, los sistemas software, sus usuarios, pero también los ingenieros que los desarrollan, puedan aprovechar los conocimientos madurados. El profesor Xavier Ferrè en su tesis doctoral^[1] nos ofrece un resultado importante en este sentido: nos indica una metodología simple, clara y flexible para integrar las técnicas de usabilidad en un cualquier proceso de desarrollo software incremental. Como él mismo dice, "el único requisito que debe cumplir un proceso para poder integrar técnicas de usabilidad con dicho objetivo es estar basado en un enfoque iterativo, pues tal característica es clave para desarrollar productos con un buen nivel de usabilidad" .

2.1 Objetivos del trabajo y estructura lógica de la memoria

El objetivo de este trabajo es presentar un proceso de desarrollo de software entre los más novedosos y actuales, el proceso software de SCRUM e indicar una posible aplicación del marco de usabilidad desarrollado por el profesor Xavier Ferrè a este proceso.

Esta memoria se estructurará presentando el proceso de desarrollo SCRUM, describiendo el marco de integración de la usabilidad utilizado y, por último, indicando una posible instanciación de este marco, junto con sus lecciones aprendidas. En la sección 3 se describe entonces el Proceso de Desarrollo Software de SCRUM, en la sección 4 se introduce el Marco de Integración de la Usabilidad de Ferré, en la sesión 5 se presenta una Instanciación del Marco de Integración, en la sección 6 se ofrece la Representación esquemática de dicha instanciación y finalmente en la sección 7 se dan unas conclusiones.

3. Descripción del Proceso de Desarrollo Software de SCRUM

Scrum es una Metodología Ágil^[14] para desarrollar el software en equipos. La filosofía en la que se basa es que el proceso de desarrollo software es muy complejo y imprevisible y que la única manera de manejarlo e intentar controlarlo es tratarlo como una caja negra y no como un proceso lineal definido en todas sus etapas. Las motivaciones más importantes en las que se enfoca esta idea son las que todos los ingenieros del software conocen: es muy difícil entender bien los requisitos desde las primeras fases del proyecto, los requisitos pueden cambiar durante el desarrollo y la evolución del sistema es imprevisible cuando se necesite cambiar o añadir tecnologías y herramientas en marcha.

En los modelos clásicos de ciclo de vida del software (Modelo en cascada, Modelo incremental, Modelo de prototipo, Modelo en espiral, ..) las etapas de desarrollo consisten básicamente en cuatro actividades principales:

- Análisis;
- Diseño;
- Desarrollo;
- Evaluación;

Estas actividades se llevan a cabo con un orden definido que suele vincular el proceso de desarrollo. Sin embargo, si se opta por seguir la metodología SCRUM, el ciclo de vida no se trata como un flujo lineal en el que se puedan distinguir exactamente estas actividades; además no se precisa seguir un orden preciso en el proceso de desarrollo. El proyecto puede empezar con cualquier actividad y se puede pasar de una actividad a otra en cualquier momento maximizando la flexibilidad y la productividad del equipo. Hay que trabajar en equipos pequeños para reducir problemas de comunicación y coordinamiento y, a la vez, maximizar la cooperación; se precisa finalmente que todas las actividades y las entregas, tienen que ser flexibles. Data su flexibilidad, esta metodología se puede utilizar incluso como *wrapper* para modelos de desarrollo ya existentes.

3.1 Visión global de SCRUM

El nombre SCRUM deriva de una analogía con el Rugby: en este deporte se llama SCRUM una manera de reiniciar rápidamente el juego cuando se haya interrumpido por alguna infracción accidental. Otras analogías con el Rugby son que el entorno (recursos, requisitos, tecnología,..) determina la evolución y la terminación del proceso de desarrollo; como el Rugby ha nacido violando las reglas del fútbol, así SCRUM se ha desarrollado anteponiendo las prioridades de unos proyectos concretos a las reglas de la Ingeniería del Software. Pero la analogía mas representativa es que el primer objetivo del equipo, bien en SCRUM como en el Rugby, es siempre lo de adelantar la pelota (el proyecto!).

3.2.1 Técnicas usadas en SCRUM

Para aplicar SCRUM y aprovechar su flexibilidad, hay que utilizar las siguientes técnicas: Creación del Equipo, Definición de Los *Backlog*, Segmentación del Proyecto, SCRUM *Meetings*.

Creación del Equipo: un equipos de desarrollo tiene que estar compuesto al maximo por 8 personas; hay un facilitator del equipo (Scrum master) que orquesta los miembros del equipo a lo largo del proceso, actua como interfaz con las influencias externas y intenta resolver los problemas encontrados por los miembros del equipo. El equipo define las practicas, la terminología, la evolución del producto coralmente y el facilitator del equipo se preocupa que se éstos respecten.

Definición de Los *Backlogs*: los Backlogs definen los trabajos que hay que hacer. Hay 3 diferentes tipologias: los Backlogs del producto, que son los requisitos a cumplir en una dada version del producto; los Backlogs de la *release* que son los requisitos elejidos para cumplir en una dada *release* del producto y finalmente los Sprint Backlogs que son los requisitos a realizar en una unidad de trabajo de máximo 4 semanas; mas cada elemento atomico del Sprint Backlog necesita 8-16 horas de trabajo. Estos tipos de Backlogs se muestran en la Figura 1.

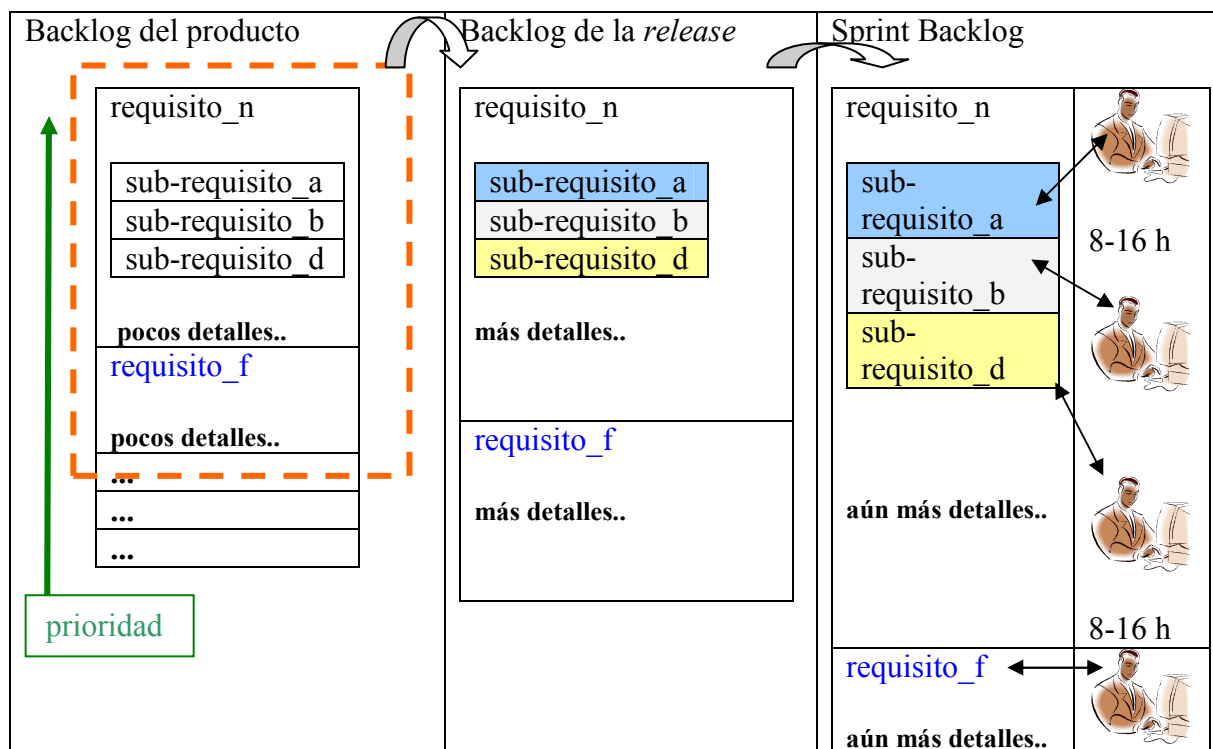


Figura 1. Backlogs

Segmentación del Proyecto: el proyecto se divide en periodos (Sprint) de máximo 4 semanas, como se muestra en la Figura 2:

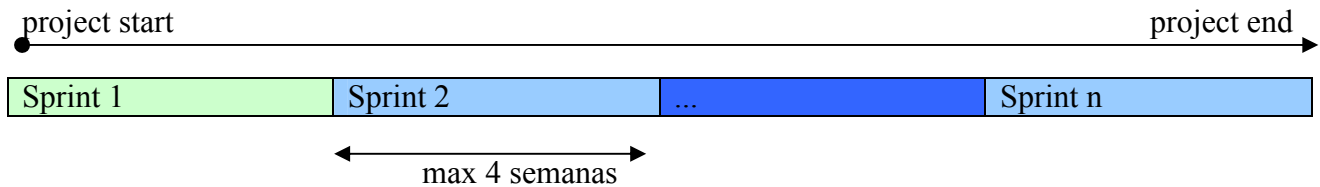


Figura 2. Segmentación del Proyecto en Sprints

SCRUM Meetings: durante los Sprints los *meetings*, es decir las reuniones, se tienen cada día y duran una media hora, siempre en el mismo lugar y a la misma hora (para no tener que meterse de acuerdo cada vez). El SCRUM master hace tres preguntas a cada componente del equipo:

¿que has hecho desde la última reunión?

¿que ha retrasado o impedido tu trabajo?

¿que planteas hacer entre hoy y la próxima reunión?

si se manifiestan problemas el SCRUM master tiene que tomar decisiones muy rápidamente para solventarlas y puede fijar reuniones para profundizar con los interesados durante el resto del día.

En la Figura 3 se muestran, de manera esquemática, las fases temporales en las que se divide un proyecto:

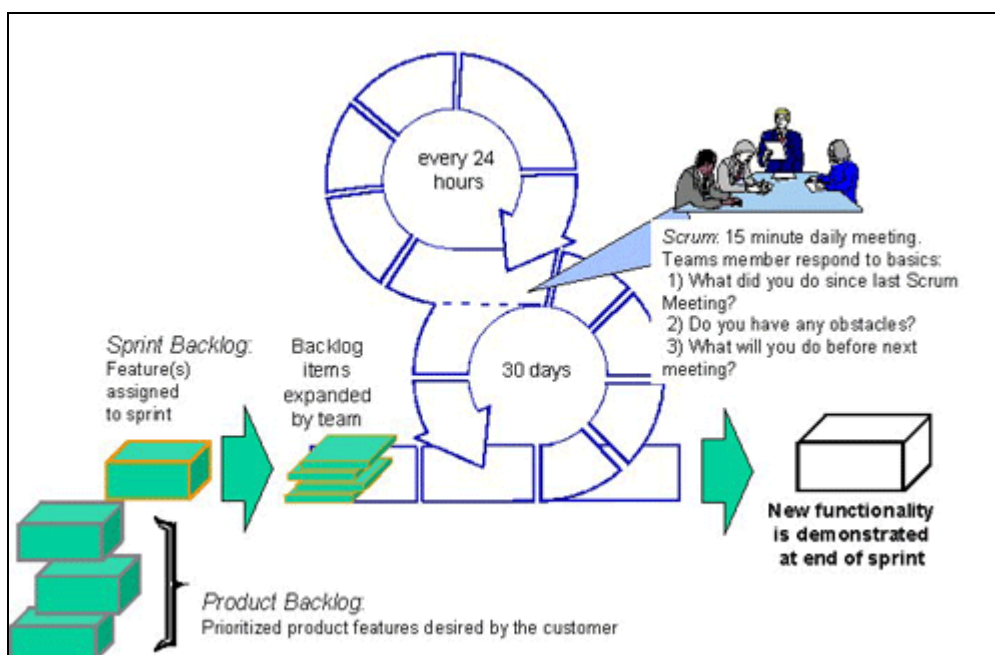
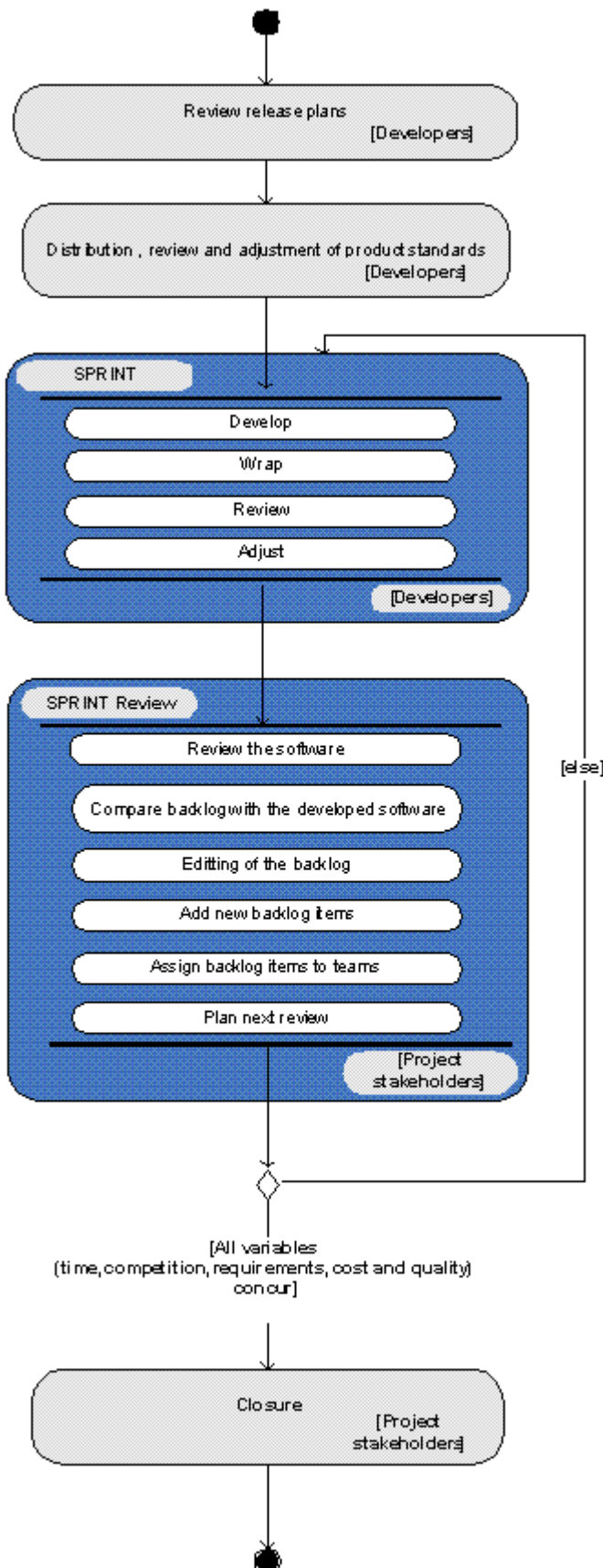


Figura 3. Sprint Backlogs y Fases de SCRUM

3.2.2 Fases de SCRUM

En el proceso de desarrollo con SCRUM se distinguen cinco fases (Figura 4):



Review release plans y Distribution , review and adjustment of product standards:

se efectúa en esta fase una revisión de lo que hay que hacer y los detalles de la distribución actual (tecnologías, estándares,..).

SPRINT:

es la fase de desarrollo, iterativa. Las sub-actividades no tienen que tener el orden de la imagen, la fase de desarrollo (*develop*) incluye la análisis, la implementación, el testing; la fase de paquetizar (*wrap*) sirve para generar paquetes, ejecutables, etc., en la fase de revisión (*review*) se resuelven problemas y se añaden nuevos *items* y en la fase de ajuste (*adjust*) se utilizan las mejoras y ajustes encontrados para mejorar el producto (código, documentación, etc..).

SPRINT REVIEW:

después de cada Sprint hay una revisión (*Sprint Review*) con el SCRUM Master, en la que se revisa el producto hallado durante el último Sprint y se pueden añadir backlog nuevos. Esta fase puede admitir la participación de los clientes, los ejecutivos, etc..

CLOSURE:

en esta fase se encuentran las típicas actividades de fin de proyecto actas a obtener una versión distribuible, como el testing, el debugging, la promoción, el marketing.

Figura 4. Fases de SCRUM en detalle

Como se puede imaginar es muy útil desarrollar con SCRUM cuando los requisitos cambian rápidamente a lo largo del desarrollo, puesto que esta metodología pretende controlar el caos producido por conflictos de intereses y de necesidades. En un proyecto en el que se quiera dar el justo papel a la usabilidad hay que prever que las indicaciones ofertas por expertos y usuarios para mejorar la usabilidad afecten el entero proceso de desarrollo. En este caso, utilizar una metodología ágil como esta, puede ser una buena idea!

4. Marco de Integración de la Usabilidad

El marco de integración de la usabilidad que se pretende utilizar en este trabajo es el marco propuesto por el profesor Xavier Ferré en su tesis doctoral^[1].

Para cada actividad de un cualquier proceso de desarrollo software (Análisis de usuario, Desarrollo, Prototipado, Especificación de Requisitos, Validación de Requisitos, Diseño, Evaluación, Evaluación de la usabilidad, etc..), el Marco indica unas técnicas de IPO aplicables.

Cada técnica IPO es explicada, se indica la referencia básica y se le dan unos índices que valúan su aplicabilidad, su representatividad, la aportación frente al esfuerzo y otras medidas que ayudan al proyectista a seleccionar que actividades IPO quiere aplicar en su proyecto en función de sus recursos, de sus objetivos y del momento de vida del proyecto mismo.

En la Tabla 1 (obtenida del trabajo del profesor Xavier Ferré^[1]) se muestran las Técnicas IPO por Actividad que se utilizará en la sección de Instanciación para integrar el Marco en el proceso SCRUM. Se nota como la terminología relativa a las actividades acopladas a las técnicas IPO, es la terminología típica de la Ingeniería del Software. Eso es porque este Marco pretende, de alguna manera, realizar un “puente” entre IPO y IS.

Tipo de Actividad	Técnica	P. U	Necesidad de Formación	Aplicabilidad	Cercanía a la IS	Aportación / Esfuerzo	Rep.	Val. Total	Momentos de Aplicación			Referencia Básica	
									C. Iniciales	C. Centrales	C. Evolución		
Análisis (Ing. de Requisitos)	<i>Card Sorting</i>	sí	bajo	alto	medio	alto	3	muy útil	neutra	neutra	neutra	[Robertson, 01]	
	Análisis Competitivo	no	medio	alto	medio	alto	1	útil	esp. apropiada	neutra	neutra	[Nielsen, 93]	
	Diagramas de Afinidad	sí	bajo	alto	medio	alto	1	útil	esp. apropiada	no es habitual	no es habitual	[Beyer, 98]	
	Investigación Contextual	sí	alto	medio	medio	alto	3	útil	esp. apropiada	no es habitual	no es habitual	[Beyer, 98]	
	JEM	sí	medio	medio	alto	medio	1	útil	neutra	neutra	neutra	[Constantine, 99]	
	Observación Etnográfica	no	alto	medio	medio	medio	2	útil	esp. apropiada	no es habitual	no es habitual	[Wixon, 96]	
	Personas	no	medio	medio	medio	alto	3	muy útil	esp. apropiada	no es habitual	no es habitual	[Cooper, 03a]	
	Mapa de Roles de Usuario	no	bajo	medio	alto	medio	1	útil	esp. apropiada	neutra	neutra	[Constantine, 99]	
	Perfiles de Usuario	no	alto	alto	alto	alto	5	útil	esp. apropiada	neutra	neutra	[Mayhew, 99]	
	Casos de Uso Esenciales	no	medio	alto	alto	alto	1	muy útil	neutra	neutra	neutra	[Constantine, 99]	
	Escenarios de Tareas	sí	medio	medio	medio	alto	1	útil	neutra	neutra	neutra	[Mayhew, 99]	
	HTA	no	medio	medio	alto	medio	1	útil	esp. apropiada	no es habitual	no es habitual	[Annett, 04]	
	Escenarios y storyboards	sí	medio	medio	bajo	alto	3	muy útil	esp. apropiada	no es habitual	no es habitual	[Carroll, 97]	
	Tormenta de Ideas Visual	sí	bajo	alto	bajo	alto	1	útil	esp. apropiada	no es habitual	no es habitual	[Preece, 94]	
	Prototipado	Prototipos de Papel	sí	bajo	alto	alto	alto	3	muy útil	esp. apropiada	neutra	neutra	[Snyder, 03]
	Especificación de Requisitos	Especificaciones de Usabilidad	no	medio	medio	medio	alto	4	muy útil	neutra	neutra	neutra	[Hix, 93]
	Validación de Requisitos	Inspecciones	no	medio	alto	medio	alto	4	muy útil	neutra	neutra	neutra	[Nielsen, 94]
		Evaluación Heurística	no	alto	alto	bajo	alto	6	útil	neutra	neutra	neutra	[Nielsen, 93]
		Inspecciones Colaborativas	sí	medio	medio	medio	medio	1	útil	esp. apropiada	neutra	neutra	[Constantine, 99]
		Recorrido Cognitivo	no	alto	medio	medio	medio	4	útil	neutra	neutra	neutra	[Lewis, 97]
	Recorrido Pluralístico	sí	bajo	medio	medio	medio	4	útil	esp. apropiada	neutra	neutra	[Bias, 94]	

Tabla 1: Técnicas IPO propuestas Organizadas por Actividad, Ferré, 2005

Tipo de Actividad	Técnica	P. U	Necesidad de Formación	Aplicabilidad	Cercanía a la IS	Aportación / Esfuerzo	Rep.	Val. Total	Momentos de Aplicación			Referencia Básica	
									C. Iniciales	C. Centrales	C. Evolución		
Diseño	Diseño de la Interacción	Árboles de Menús	no	bajo	medio	alto	alto	1	muy útil	neutra	neutra	neutra	[Shneiderman, 98]
		Diagramas de Transición de Estados de la Interfaz	no	bajo	alto	alto	medio	2	útil	neutra	neutra	neutra	[Wasseman, 85]
		Guía de Estilo del Producto	no	alto	medio	medio	medio	1	útil	no es habitual	esp. apropiada	neutra	[Mayhew, 99]
		Mapa de Navegación	no	medio	alto	alto	medio	1	útil	neutra	neutra	neutra	[Constantine, 99]
		Modelo del Contenido de la Interfaz	no	medio	alto	medio	medio	1	útil	neutra	esp. apropiada	neutra	[Constantine, 99]
		Análisis de Impacto	no	medio	medio	alto	medio	3	útil	no es habitual	neutra	neutra	[Hix, 93]
Evaluación	Evaluación por Expertos	Organización de la Ayuda según Casos de Uso	no	medio	medio	alto	medio	1	útil	neutra	neutra	neutra	[Constantine, 99]
		Inspecciones	no	medio	alto	medio	alto	4	muy útil	neutra	neutra	neutra	[Nielsen, 94]
		Evaluación Heurística	no	alto	alto	bajo	alto	6	útil	neutra	neutra	neutra	[Nielsen, 93]
		Inspecciones Colaborativas	sí	medio	medio	medio	medio	1	útil	esp. apropiada	neutra	neutra	[Constantine, 99]
		Recorrido Cognitivo	no	alto	medio	medio	medio	4	útil	neutra	neutra	neutra	[Lewis, 97]
	Test de Usabilidad	Recorrido Pluralístico	sí	bajo	medio	medio	medio	4	útil	esp. apropiada	neutra	neutra	[Bias, 94]
		Pensar en Voz Alta	sí	medio	alto	bajo	alto	5	muy útil	no es habitual	neutra	neutra	[Nielsen, 93]
		Información Post-Test	sí	medio	alto	medio	alto	1	útil	no es habitual	neutra	neutra	[Constantine, 99]
		Medición del Rendimiento	sí	medio	medio	medio	medio	3	útil	no es habitual	neutra	neutra	[Dumas, 99]
		Test de Usabilidad en Laboratorio	sí	medio	medio	medio	medio	4	útil	no es habitual	neutra	neutra	[Dumas, 99]
	Estudios de Seguimiento de Sistemas Instalados	Retroalimentación del Usuario	sí	bajo	alto	alto	alto	3	muy útil	no es habitual	no es habitual	esp. apropiada	[Shneiderman, 98]
		Cuestionarios, Entrevistas y Encuestas	sí	medio	alto	medio	medio	3	útil	no es habitual	neutra	esp. apropiada	[Mayhew, 99]
Registro del Uso		no	alto	medio	alto	medio	5	útil	no es habitual	neutra	esp. apropiada	[Shneiderman, 98]	

Tabla 1: Técnicas IPO propuestas Organizadas por Actividad, Ferré, 2005

5. Instanciación del Marco de Integración

Como se ha explicado, el marco de usabilidad indica un conjunto de actividades y técnicas que el ingeniero del proceso puede escoger según sus necesidades y preferencias con respecto a la realización de un proyecto concreto. Se supone aquí indicar una posible instanciación del proceso de desarrollo SCRUM.

5.1 Técnicas seleccionadas

Se han elegido solo unas técnicas IPO entre todas las indicadas por el profesor Xavier Ferré: las que se supone sean más representativas y que tengan mayor impacto para la usabilidad sin ser demasiado costosas.

A continuación se indican dichas actividades con la definición que aparece en el trabajo de Xavier Ferré.

5.1.1 Personas

“Esta técnica ayuda a sintetizar todos los datos de que se dispongan sobre los usuarios previstos del sistema, en unos usuarios arquetípicos que puedan usarse para alcanzar consenso en el equipo de desarrollo y para centrar las discusiones de diseño. Ayuda también a determinar qué es lo que el producto debe hacer, relacionado con las necesidades a satisfacer, por lo que puede contribuir a todo el proceso de análisis y negociación de requisitos. Al proveer un lenguaje común para referirse a los usuarios concretos del sistema (concreto, frente al término genérico y a menudo equívoco "usuarios del sistema"), ayuda a alcanzar consenso en el equipo de desarrollo. En concreto, evita un problema típico entre un cierto número de desarrolladores: el identificar las capacidades y querencias de los futuros usuarios del sistema con las propias (del desarrollador), conduciendo este tipo de actitud a la producción de sistemas software que únicamente usuarios con marcado perfil tecnológico pueden usar.”^[1]

5.1.2 Prototipos de Papel

“Los prototipos permiten al desarrollador comunicarse con mayor efectividad con los usuarios y clientes. Necesitamos construir prototipos porque las especificaciones técnicas abstractas no son un buen medio de comunicación cuando queremos implicar a los usuarios en el proceso de desarrollo.

El prototipado está estrechamente ligado al desarrollo iterativo. Los prototipos pueden usarse para probar ideas de diseño con los usuarios para obtener su retroalimentación. Con este fin, los prototipos menos costosos (de papel), son suficientes. Los prototipos de papel son útiles para obtener opiniones del usuario mejor que los prototipos que se asemejan al sistema final, pues al confrontarse con un trabajo con un buen acabado, muchos usuarios pueden convencerse erróneamente de que esa es la única o la mejor solución al problema planteado. Los prototipos de papel cuentan con la ventaja de que son mucho más baratos de elaborar que los prototipos software, a no ser que se disponga de una herramienta de prototipado rápido.”^[1]

5.1.3 Especificaciones de Usabilidad

“Son metas cuantitativas de usabilidad, que se usan como guía para saber cuando una interfaz es suficientemente buena. Las especificaciones pueden ser basadas en medidas objetivas o subjetivas. Las medidas objetivas se asocian normalmente a tareas de benchmark, mientras que las subjetivas se asocian normalmente con cuestionarios de usuario.”^[1]

5.1.4 Inspecciones

“En el desarrollo de software el objetivo de una inspección es encontrar defectos. Las inspecciones de usabilidad tienen por objetivo descubrir defectos de usabilidad. Se trata de un proceso sistemático, frente a la evaluación heurística que no tiene un proceso establecido. Hay dos tipos de inspecciones de usabilidad:

- Inspecciones de conformidad (con una guía de estilo, o un estándar): En las inspecciones de conformidad, los participantes inspeccionan el diseño de la interacción del sistema para comprobar que si cumple lo estipulado en determinados estándares o guías de estilo. Todos los participantes deben estar familiarizados con los estándares o guías sobre los que se evalúa.
- Inspecciones de consistencia: En las inspecciones de consistencia, el objetivo consiste en identificar inconsistencias entre contextos de interacción y sus contenidos. El evaluador intenta encontrar inconsistencias en terminología, color, disposición, formatos de entrada y salida de datos, etc. Cuando el producto pertenece a una familia de productos, se juntan representantes de los distintos equipos de diseñadores para inspeccionar la consistencia entre los productos de la familia.”^[1]

5.1.5 Pensar en Voz Alta

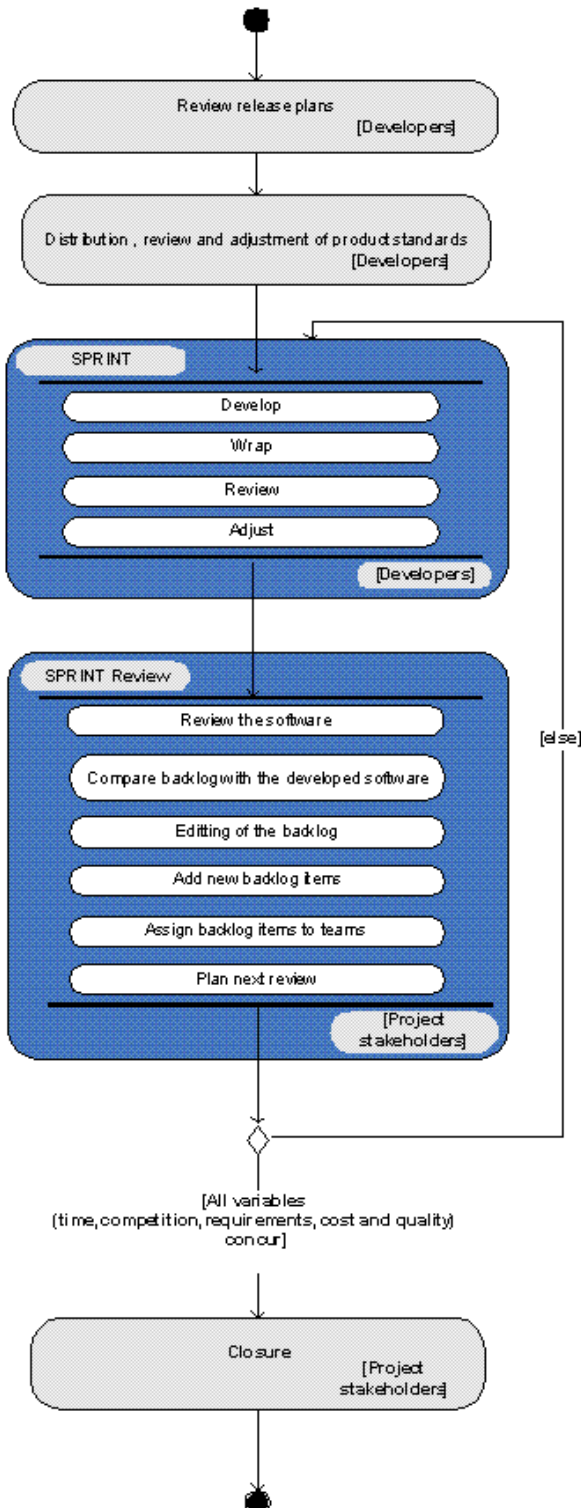
“Tal y como menciona Nielsen en sus técnicas, es un caso especial de protocolo verbal, en el cual el usuario dice en voz alta lo que piensa mientras realiza una tarea o resuelve un problema. Nielsen distingue el pensamiento en voz alta de otras técnicas de usabilidad precisando que puede ser el método de ingeniería de usabilidad más valioso. Una prueba de pensamiento en voz alta implica el tener a un participante usando el sistema mientras que piensa en voz alta. Su efectividad está en datos cualitativos y no en medidas de funcionamiento. La idea es conseguir la opinión del usuario mientras usa el sistema para evitar posteriores racionalizaciones.”^[1]

5.1.6 Retroalimentación del Usuario

“La retroalimentación se puede obtener dando a los usuarios acceso a direcciones especiales de correo electrónico, grupos de noticias o foros. Los usuarios pueden de este modo enviar sus quejas o peticiones de cambio o mejora.” [1]

6. Actividades y ubicación de Técnicas seleccionadas en Scrum en los tres momentos de vida:

6.1 C.Iniciales



Review release plans y Distribution , review and adjustment of product standards:

Actividades IPO: Análisis de Usuario y Especificación de Requisitos

- **Personas;**
- **Especificaciones de Usabilidad;**

SPRINT:

Actividad IPO: Prototipado y Evaluación de Usabilidad

- **Prototipos de papel (sin participación de usuario);**
- **Pensar en voz alta (sin participación de usuario);**

SPRINT REVIEW:

Actividad IPO: Prototipado

- **Prototipos de papel;**

Review release plans y Distribution , review and adjustment of product standards:

En estas primeras fases se introduce la actividad **Personas** para obtener que el equipo tenga claro desde el principio cual es el *target* de usuario para el que se va a realizar el sistema. Como es preciso comprender el propósito para el que el sistema se utilizará, es imprescindible enfocarse en quien lo utilizará. Además tener un contacto con un subconjunto muy limitado pero representativo de los usuarios finales desde el principio puede facilitar muchas sugerencias.

Otra actividad útil en esta fase es la **Especificaciones de Usabilidad** para plantear los objetivos a perseguir relativamente a la usabilidad.

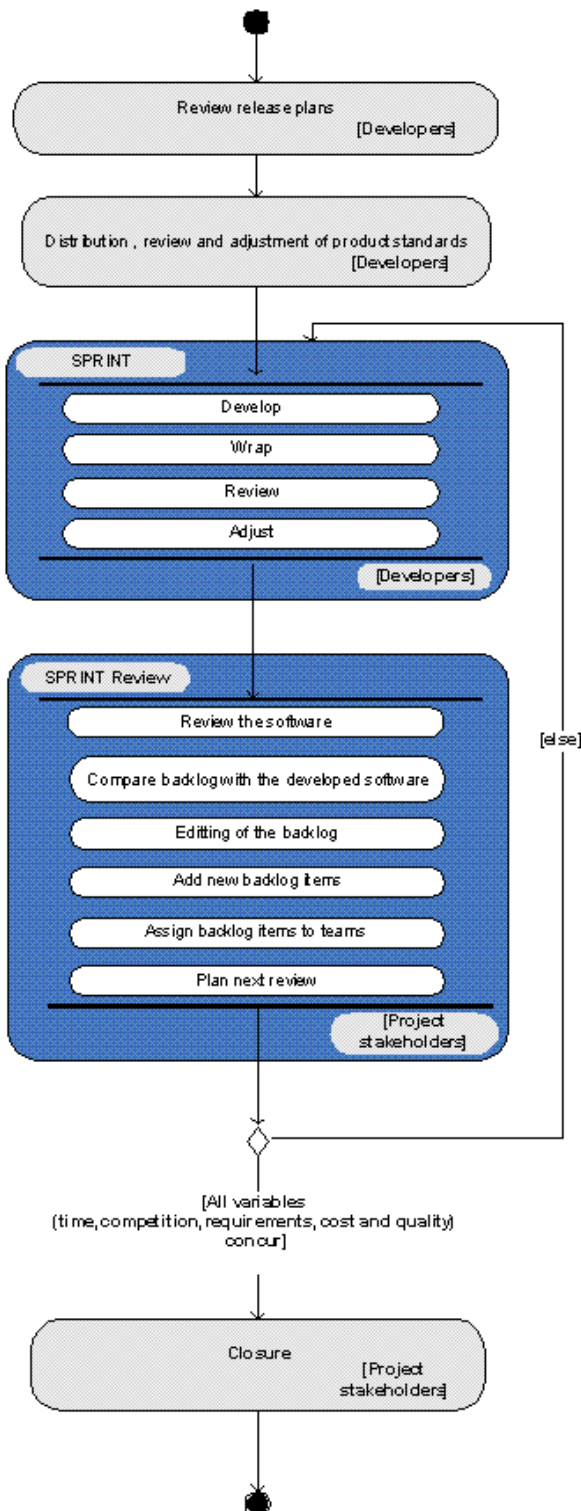
SPRINT:

En esta fase se añaden las actividades de **Prototipos de papel** y **Pensar en voz alta** pero sin participación de usuario. Es decir, durante las reuniones o durante el desarrollo, los proyectistas pueden tomar decisiones y hacer pruebas de usabilidad usando estas dos actividades que, a pesar de no ser muy costosas, pueden ser determinantes para hallar un buen grado de usabilidad. Puede ser incluso muy interesante para los desarrolladores jugar el papel de usuario destacándose de vez en cuando de su entorno puramente tecnológico.

SPRINT REVIEW:

En esta fase se supone que puedan colaborar activamente ejecutivos, clientes y, como no, representantes de los usuarios. Se ha elegido así la actividad de **Prototipos de papel** que resulta muy útiles como

6.2 C.Centrales



Review release plans y Distribution , review and adjustment of product standards:

Actividad IPO: Especificación de Requisitos

- **Especificaciones de Usabilidad;**

SPRINT:

Actividad IPO: Prototipado

- **Prototipos de papel;**

SPRINT REVIEW:

Actividad IPO: Prototipado y Evaluación de Usabilidad

- **Inspecciones;**
- **Pensar en voz alta;**

CLOSURE:

Actividad IPO: Evaluación de Usabilidad y Especificación de Requisitos

- **Retroalimentación del Usuario;**
- **Especificaciones de Usabilidad;**

Review release plans y Distribution , review and adjustment of product standards:

En esta fase se introduce la actividad de **Especificaciones de Usabilidad**, para dar un nuevo impulso en tener en cuenta los objetivos a perseguir relativamente a la usabilidad.

SPRINT:

En esta fase se añade la actividad de **Prototipos de papel**. Durante el desarrollo, los proyectistas pueden tomar decisiones y hacer pruebas de usabilidad usando esta actividad que, a pesar de no ser muy costosas, puede ser determinante.

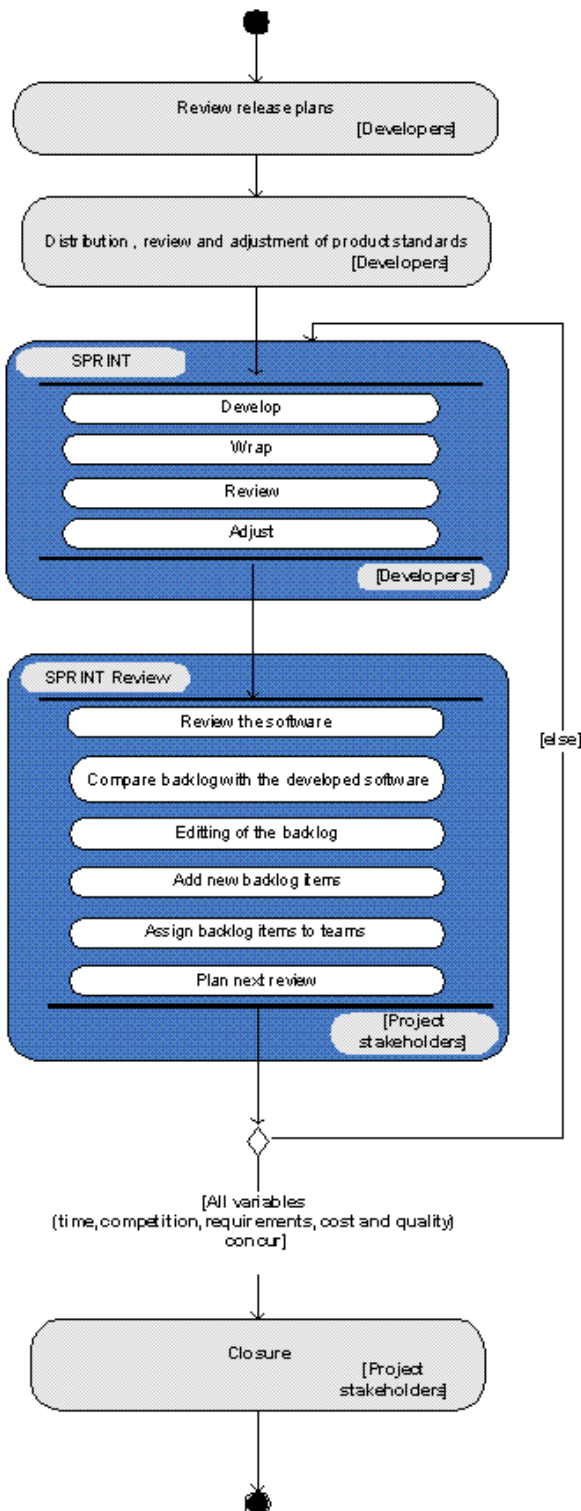
SPRINT REVIEW:

En esta fase se supone que puedan colaborar activamente ejecutivos, clientes y, como no, representantes de los usuarios. Se han elegido así actividades como **Pensar en voz alta e Inspecciones**. Estas resultan muy útiles como benchmark y pueden representar una “zona franca” donde desarrolladores y usuarios hablan el mismo idioma. La actividad de **Inspecciones** además ayuda a encontrar errores o imprecisiones.

CLOSURE:

En la etapa final de estos ciclos de desarrollo, es fundamental tener **Retroalimentación del Usuario**, para encontrar los aspectos y a los errores a corregir. Se añade también la actividad **Especificaciones de Usabilidad** para resumir todas las especificaciones halladas a lo largo del desarrollo y que van a ser el objetivo de los últimos ciclos (C. Evolución)

6.3 C.Evolución



<p>SPRINT: Actividad IPO: Prototipado y Evaluación de Usabilidad</p> <ul style="list-style-type: none"> ▪ Pensar en voz alta (sin participación de usuario);
<p>SPRINT REVIEW: Actividad IPO: Prototipado y Evaluación de Usabilidad</p> <ul style="list-style-type: none"> ▪ Inspecciones;
<p>CLOSURE: Actividad IPO: Evaluación de Usabilidad</p> <ul style="list-style-type: none"> ▪ Retroalimentación del Usuario;

SPRINT:

En esta fase se añade la actividad de **Pensar en voz alta**, pero sin participación de usuario. Es decir, durante las reuniones o durante el desarrollo, los proyectistas pueden tomar decisiones y hacer pruebas de usabilidad usando esta actividad muy rápida para los últimos detalles.

SPRINT REVIEW:

Se han elegido para esta fase la actividad de **Inspecciones**, porque es aquí que es preciso encontrar errores o imprecisiones y pedir que se hagan unas modificaciones al sistema.

CLOSURE:

En la fase final del desarrollo, cuando se testean las últimas funcionalidades y se eliminan los errores, es fundamental tener **Retroalimentación del Usuario**, para encontrar, pero también para dar la justa prioridad, a los aspectos y a los errores a corregir en las próximas versiones.

ventajas e inconvenientes	
ventajas	inconvenientes
el Marco de Integración permite encajar las actividades IPO en el proceso de desarrollo software sin muchos esfuerzos;	los usuarios deben participar bastante en el proceso de desarrollo y esto puede resultar frustrante para los desarrolladores y introducir pérdidas de tiempo
el Marco de Integración permite plantear unas actividades IPO en el proceso de desarrollo y evitar así que no se le dé bastante peso;	el Marco de Integración no indica explícitamente que actividades utilizar, dejando todas las decisiones de responsabilidad del proyectista, que puede que sea poco experto de IPO
el Marco de Integración deja mucha libertad al proyectista que puede evaluar y elegir que técnicas utilizar basándose en los casos concretos;	

7. Conclusiones

El Marco de Integración de que se ha dado, en este trabajo, una posible instanciación en el proceso de desarrollo SCRUM *Development Process* ha resultado muy flexible y cómodo a utilizar. Hasta para un proceso de desarrollo como este, que pretende tener en cuenta la complejidad, la no predictibilidad y aleatoriedad, en el que las actividades no son bien definidas y ordenadas, se ha encontrado una posible aplicación del Marco de Integración para mejorar la usabilidad. El modo en que se presentan las técnicas, relacionadas a las actividades de la Ingeniería del Software, con la misma terminología que se utiliza en este campo y la herramienta web[♦], rinden muy satisfactorio utilizar este Marco de Integración. De alguna manera se podría decir que aún más que ser una herramienta muy útil, es una herramienta muy usable!

[♦] <http://is.ls.fi.upm.es/xavier/usabilityframework/>

8. Referencias:

- [1] Xavier Ferré. *Marco de Integración de la Usabilidad en el Proceso de Desarrollo Software*. Tesis doctoral. 2005
- [2] Larman, C.: *UML y Patrones, Segunda Edición*. Prentice-Hall (2002)
- [3] George M. Donahue. *Usability is good business*. Compuware Corporation. www.compuware.com 1999
- [4] IEEE *standard for developing software life cycle processes*, IEEE Std 1074-1997.
- [5] Takeuchi and Nonaka *New New Product Development Game* (January-February 1986) Harvard Business review
- [6] Rising, Linda *Agile Methods: What's It All About* DDC-I Online News www.ddci.com 2001-11-01
- [7] Schwaber, K. *Advanced Development Methods. SCRUM Development Process* Retrieved August 15, 2006
- [8] Constantine, L. L., and Lockwood, L. A. D. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design* Addison-Wesley, New York, NY, 1999
- [9] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* Addison-Wesley, Reading, MA, 1998.
- [10] D. Norman e S. Draper. *User centered system design: New perspective on human-computer interaction*. Erlbaum Associates, Inc. Mahwah, NJ, USA, 1986

9 Bibliografía Consultada

Mayhew, D.J. *The Usability Engineering Lifecycle* Morgan Kaufmann, San Francisco CA, 1999

www.usabilitynet.org A European Union project that provides usability and user centred design resources to practitioners, managers and EU projects.

Ben Shneiderman . *Leonardo's Laptop: Human Needs and the New Computing Technologies* 2002

McLellan, S.G.; Roesler, A.W.; Tempest, J.T.; Spinuzzi, C.I.; *Building more usable APIs* Software, IEEE, pp78-86, 1998