

# Influencia de la calidad del proceso en la usabilidad del producto: una revisión sistemática

Alberto García-Mireles<sup>1</sup>, Félix García<sup>2</sup>, M<sup>a</sup> Ángeles Moraga<sup>2</sup> y Mario Piattini<sup>2</sup>

<sup>1</sup>Departamento de Matemáticas, Universidad de Sonora, 83000 Hermosillo, Sonora, México

<sup>2</sup>Grupo Alarcos – Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, 13071 Ciudad Real, España  
[mireles@gauss.mat.uson.mx](mailto:mireles@gauss.mat.uson.mx), [Felix.Garcia](mailto:Felix.Garcia), [MariaAngeles.Moraga](mailto:MariaAngeles.Moraga), [Mario.Piattini](mailto:Mario.Piattini@uclm.es)@uclm.es

**Resumen.** Cuando las organizaciones implementan un programa de mejora de procesos de software están buscando mejorar la productividad, reducir costos y aumentar la calidad del software desarrollado. De hecho, la calidad del producto software depende en importante medida de los procesos utilizados para su desarrollo y/o mantenimiento.

Sin embargo, a pesar de la importancia de la relación calidad proceso y producto, es un tema que no ha sido ampliamente abordado en la literatura. Una de las características de calidad de producto que tiene más interés hoy en día es la usabilidad, ya que influye en cómo el usuario percibe el producto. Con este fin, en este artículo se presenta una revisión sistemática de la bibliografía que analiza la relación entre la usabilidad y los procesos software. Como resultado se identifican los procesos que son importantes a la hora de promover una mejora en la usabilidad de los productos.

**Palabras clave:** Proceso de software, usabilidad, revisión sistemática de literatura.

## 1 Introducción

Para mantenerse en condiciones de competir en el mercado, las empresas de desarrollo de software incorporan cambios en sus procesos para producir software a menor costo, de mayor calidad y en el menor tiempo posible. Por tanto, uno de los argumentos más frecuentes para invertir en iniciativas de mejora de procesos (SPI) es incrementar la calidad del software [1].

El enfoque que se aplica con más frecuencia para abordar la calidad de procesos software es el proveniente de la industria de fabricación. En [2] se presentan resultados de algunos estudios empíricos de SPI exitosos en los cuales la calidad está relacionada con la disminución del número de defectos en el software. Con base en los resultados de la encuesta presentada en [3], se concluye, que las iniciativas SPI implantadas en las empresas que participaron en el estudio les permitió disminuir la densidad de defectos del 10 al 94%. Además, encontraron una correlación entre los niveles de la calidad del producto y el nivel de madurez alcanzado (hasta el nivel tres). Por su parte, la percepción general entre los desarrolladores de software que han participado en programas de SPI en países emergentes es positiva respecto de la mejora en la productividad y calidad [4].

Aunque se reconoce que el nivel de madurez del proceso de software influye en la calidad del producto, no se sabe mucho acerca de la influencia específica que puede tener. Tal como se señala en [5, 6] se requiere investigación más profunda que establezca, por una parte, las relaciones que se dan entre la calidad del proceso y la calidad del producto de software, y por

otra, la relación de la calidad con las restricciones del proyecto y las metas del negocio [5, 6]. En esta línea existen algunos trabajos, tales como el propuesto en [8] que sugiere que existe una relación entre los modelos de certificación de procesos y la fiabilidad, facilidad de prueba, usabilidad, eficiencia e integridad del producto. Otros trabajos establecen correlaciones entre una iniciativa de SPI y los atributos de calidad del software a partir de la percepción de directivos y desarrolladores de software [9-11]. Sin embargo, ninguno de ellos aborda a nivel técnico cómo implementar un esfuerzo SPI con base en las características de calidad de producto que se pretenden mejorar.

Con todo ello, en este trabajo se revisa la influencia que pueden tener los procesos de software en la calidad de los productos. El análisis se centra en una de las características de calidad de producto más relevantes desde la perspectiva del usuario, que es la usabilidad. Esta se define como el “grado en el cual un producto o sistema puede ser usado por usuarios específicos para lograr metas específicas con efectividad, eficiencia y satisfacción en un contexto de uso especificado” [7]. Expertos en usabilidad coinciden en que se requieren procesos, métodos y herramientas adecuados para promover una mejora en la usabilidad del producto [8].

El análisis presentado se fundamenta en los resultados de una revisión sistemática de literatura (RSL) realizada con la finalidad de identificar los procesos que son importantes a la hora de promover una mejora en la usabilidad de los productos. El resto del presente artículo está estructurado de la siguiente forma: en la sección dos se describe la metodología aplicada para realizar la RSL y algunas estadísticas relevantes sobre los estudios seleccionados. La sección tres sintetiza los aspectos más destacables de los estudios seleccionados como relevantes. La sección cuatro describe la forma de integrar prácticas de usabilidad en procesos de acuerdo a los resultados de la revisión. Finalmente, se presentan las conclusiones y las líneas de trabajo futuro.

## 2 Revisión sistemática sobre procesos software y usabilidad de producto

La revisión sistemática de literatura está basada en la propuesta metodológica de Kitchenham [9] y Kitchenham y Charters [10]. Para satisfacer el objetivo especificado en la RSL se plantean las siguientes preguntas de investigación: ¿cuál es el estado de la investigación en cuanto a la relación entre los procesos de software y la usabilidad? ¿cuáles son las actividades que se recomiendan para mejorar la usabilidad del software? ¿en qué procesos se debe hacer énfasis para mejorar la usabilidad del producto desarrollado?

**Tabla 1.** Términos para la cadena de búsqueda.

Términos	Cadenas
Usabilidad	usability
Software	software OR product
Proceso de software	“software process” OR “development process” OR “development method”

Como criterios de inclusión para la selección de estudios primarios se ha considerado que sea un artículo de revista o de conferencia publicado hasta el 2010, escrito en inglés, orientado al

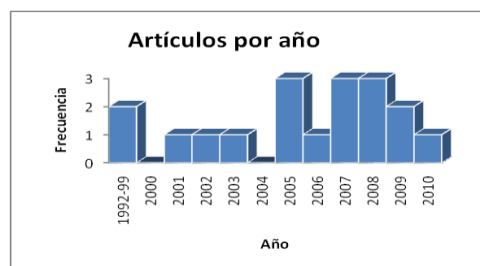
proceso o a actividades relacionadas con la usabilidad y, de preferencia, que incluya validación empírica. Los criterios de exclusión se aplican a los trabajos orientados a arquitecturas dirigidas por modelos y estudios que analizan el grado de integración de prácticas de usabilidad en la industria. La tabla 1 muestra las cadenas de búsqueda empleadas, que se unieron con el operador lógico AND para obtener una sola cadena a la hora de realizar las búsquedas en los motores correspondientes.

Las bases de datos consultadas fueron Scopus, IEEE Computer Society, ACM Computer Library, Science@Direct y Springer. También se aseguró que las bases de datos incluyeran las conferencias EuroSPI, ICSP y PROFES, ya que se trata de conferencias muy relevantes dentro del área de procesos de software.

En la tabla 2 se muestran los artículos obtenidos utilizando la cadena de búsqueda. Se obtuvieron 504 entradas sin eliminar duplicados. Para realizar el primer filtro se aplicaron los criterios de inclusión considerando el título y resumen de los trabajos. Como resultado se obtuvieron 168 trabajos candidatos. Finalmente, en una segunda etapa se seleccionaron los 18 artículos primarios a partir de la lectura completa de los trabajos anteriores verificando que efectivamente satisficieran los criterios especificados. La tabla 3 muestra la relación de artículos.

**Tabla 2.** Selección de artículos primarios.

Base de datos	Relevantes	Candidatos	Primarios
Scopus	238	20	1
Science @Direct	40	30	3
IEEE	133	48	6
ACM	44	25	1
Springer	49	45	7
<i>Sumas</i>	<i>504</i>	<i>168</i>	<i>18</i>



**Fig. 1.** Frecuencia de artículos publicados por año.

De los trabajos primarios seleccionados, destaca que más de la mitad (56%) han sido publicados en conferencias. En la Figura 1 se muestra el número de publicaciones por año. Como se puede observar, de los noventa sólo se tiene dos referencias en total, mientras que en la década pasada se observa una mayor frecuencia de trabajos.

La Figura 2 muestra el tipo de validación aplicada en los estudios primarios. La mayoría de los artículos utilizan métodos empíricos, sobre todo casos de estudio (56%), un 6% utiliza el método cualitativo de Investigación-Acción y un 17% son propuestas a nivel teórico. En la siguiente sección se describe el análisis de los estudios primarios.



Fig. 2. Tipo de validación de los estudios primarios.

### 3 Procesos de software para mejorar la usabilidad

En esta sección se describe, a partir del análisis de los estudios primarios, qué aspectos se deben considerar en los procesos software a la hora de promover el desarrollo de productos con mejor usabilidad. Los resultados se clasifican teniendo en cuenta el modelo de referencia de procesos en los que se basa cada propuesta. En este sentido, algunos trabajos consideran al *Rational Unified Process* (RUP) ([11], [12], [13]), mientras que otros utilizan modelos de referencia generales ([14], [15], [16]). También se encontraron propuestas que utilizan procesos orientados al dominio de aplicaciones web o de dispositivos móviles ([17], [18], [19]), así como enfoques basados en procesos ligeros o ágiles ([20], [21], [22], [23]). La última categoría agrupa las propuestas que inciden en la especificación de requisitos de usabilidad ([24], [25], [26], [27], [28]). A continuación se sintetizan las propuestas de acuerdo a esta clasificación.

#### 3.1 Propuestas basadas en RUP

Las propuestas que están basadas en el RUP enfocan sus esfuerzos en las etapas de inicio y elaboración del ciclo de vida. Se aprovecha la naturaleza iterativa e incremental para evaluar prototipos que se elaboran con distinto nivel de detalle. En [11] proponen una adaptación para integrar las actividades de usabilidad en el RUP. Las actividades que abordan, desde la perspectiva de usabilidad, para definir requisitos son: creación de conceptos, obtención de requisitos, análisis de requisitos y diseño de la visión del producto. Identifican tres nuevos roles que denominan ingeniero de usabilidad, diseñador de interfaz de usuario y evaluador de usabilidad. Señalan que se requiere soporte de la dirección, e identificar y apoyar las tareas del desarrollo del software que necesitan más atención para mejorar la usabilidad del producto.

**Tabla 3.** Relación de artículos primarios.

Clave	Año	Tipo	Base de datos	Validación
anderson2001[11]	2001	Revista	IEEE	Caso de estudio
biel2010 [25]	2010	Revista	Scimedirect	Caso de estudio
bonacin2009 [21]	2009	Conferencia	Springer	Caso de estudio
castro2008 [27]	2008	Conferencia	IEEE	Ninguna
constantine2002 [29]	2002	Revista	IEEE	Caso de estudio
ferre2005 [14]	2005	Conferencia	Springer	Ninguna
glissmann2005 [17]	2005	Conferencia	IEEE	Caso de estudio
goransson2003 [12]	2003	Revista	Scopus	Ninguna
helms2006 [15]	2006	Revista	Scimedirect	Caso de estudio
hornbaek2007 [24]	2007	Conferencia	Springer	experimento
hussain2008 [22]	2008	Conferencia	Springer	Caso de estudio
lif1999 [26]	1999	Revista	Scimedirect	investigación acción
memmel2007 [23]	2007	Conferencia	Springer	Caso de estudio
moreno2009 [19]	2009	Conferencia	Springer	Caso de estudio
nielsen1992 [16]	1992	Revista	IEEE	Encuesta
rafla2007 [28]	2007	Revista	Springer	Experimento
sousa2005 [13]	2005	Conferencia	ACM	Caso de estudio
wang2008 [20]	2008	Conferencia	IEEE	Caso de estudio

Por su parte, en [12] se resalta que es importante involucrar al usuario, conocer el contexto de uso del sistema, realizar desarrollo evolutivo y evaluar de forma continua los prototipos. Las actividades que incluyen son: crear un plan de diseño de usabilidad, realizar análisis de la competencia, conducir estudios de usuario, elaborar el diseño conceptual, elaborar el diseño de interacción, elaborar el diseño detallado, desarrollar la asistencia al usuario, evaluar la usabilidad, monitorizar el trabajo de usabilidad y refinar el plan de diseño de usabilidad.

Finalmente, en [13] presentan UPi (Proceso Unificado para el desarrollo de sistemas Interactivos) como proceso unificado para desarrollar sistemas interactivos. Su propuesta se basa en casos de uso, modelos de tareas, requisitos de usabilidad y patrones de diseño de usabilidad.

En la etapa de diseño elaboran un plan de definición de usabilidad en donde se definen valores y patrones de usabilidad para cada elemento de la interfaz y construyen prototipos para facilitar la evaluación de los patrones por parte de los usuarios. Destaca la adaptación de la fase de elaboración para orientarla más hacia la usabilidad.

### 3.2 Propuestas basadas en un modelo de procesos genérico

Estas propuestas toman como base las etapas generales del ciclo de vida para indicar las actividades y prácticas que contribuyen a mejorar el nivel de usabilidad. En [14] elaboran un marco de trabajo para ofrecer a los desarrolladores de software una selección de las técnicas de HCI (*Human Computer Interaction*) apropiadas para organizaciones con interés en mejorar la usabilidad de sus productos. Las prácticas están organizadas por tipo de actividad y por la temporalidad de las iteraciones del ciclo de desarrollo. Encontraron gran similitud entre las actividades de HCI y aquellas relacionadas con los requisitos del software, ya que para ambas es

muy relevante el contexto de uso, conocer al usuario y sus tareas, especificar requisitos de usabilidad, desarrollar el concepto del producto, producir prototipos de bajo costo y validarlos con representantes de los usuarios. Señalan que el diseño de interacción se puede realizar en paralelo e independiente del diseño del software y la evaluación de la usabilidad requiere un tipo de pruebas diferente a las típicas de Ingeniería del Software, como la evaluación de expertos, pruebas de usabilidad y estudios de seguimiento de sistemas instalados.

En [15] construyen un modelo de proceso denominado *Wheel* y lo prueban en la industria. Es un modelo iterativo que busca la flexibilidad y menor coste. Incluyen una actividad de evaluación que controla las iteraciones. Las actividades que se especifican son la concepción del producto, modelo de usuario/tareas, escenarios de uso, diseño de pantallas, prototipo de bajo costo, prototipo detallado, integración, despliegue, sistema operacional y cierre. Dentro de cada actividad se pueden realizar el análisis, diseño, implementación y evaluación. El modelo de proceso se aplicó en una empresa estadounidense con el fin de diseñar dos interfaces de usuario para comunicación con dispositivos conectados a Internet. Los prototipos obtenidos fueron útiles para obtener financiación para el proyecto.

La propuesta de ingeniería de usabilidad de [16] promueve un esfuerzo sistemático usando métodos establecidos para lograr interfaces usables. Los elementos esenciales son las pruebas empíricas en las que participan usuario, prototipos y diseño iterativo. Su propuesta incluye tres fases: pre-diseño, diseño y post-diseño. En el pre-diseño el objetivo es entender a los usuarios y sus tareas y se puede partir de estudios de mercado, realizar un análisis de tarea y construir prototipos. En la etapa de diseño se utiliza un enfoque iterativo para refinar la interfaz de usuario que deberá evaluarse empíricamente. Por último, la etapa de post-diseño tiene por objetivo obtener datos para la siguiente versión del producto y recomendar estudios de seguimiento.

### **3.3 Propuestas basadas en procesos para un dominio particular**

Este grupo de trabajos está orientado al desarrollo de software, considerando la usabilidad del producto, pero a partir de las características particulares de la tecnología empleada para desarrollar la solución. En [17] presentan una propuesta de guías para el desarrollo de aplicaciones para dispositivos móviles considerando los principios de HCI. El modelo tiene cinco etapas: identificar necesidades, establecer requisitos, diseño conceptual, diseño físico e implementación. Consideran el diseño conceptual como aspecto clave porque es ahí en donde se consideran las capacidades del dispositivo y su alineación con los objetivos del proyecto. En el diseño físico se considera el aspecto de las pantallas y la estructura de navegación.

En [18] describen un enfoque flexible, dirigido por modelos para la ingeniería de aplicaciones web denominado diseño centrado en el uso. El proceso separa a los actores humanos de los no humanos. Los pasos preliminares identifican el propósito esencial y se realiza un modelado exploratorio. En la primera iteración se realiza un modelo de roles, modelo de tareas, agrupamiento de tareas, diseño preliminar de la interfaz y esquema de interacción, prototipo abstracto del modelo de contenido, diseño detallado de la interfaz y construcción. Las iteraciones sucesivas refinan los productos de iteraciones previas. Sugieren que se requiere de entrenamiento especial y coordinación estrecha entre diseñadores de interfaz de usuario y programadores.

Por último, en [19] proponen integrar aspectos de accesibilidad en el proceso de ingeniería web. Consideran como base para la accesibilidad los mecanismos identificados en las guías

WCAG 1.0 y 2.0. Proponen el uso de técnicas de usabilidad en las etapas de obtención y análisis de requisitos para generar modelos conceptuales, de navegación y de presentación.

### **3.4 Propuestas basadas en procesos ágiles**

Este grupo de propuestas tienen como marco de referencia los métodos ágiles, principalmente la programación extrema (XP). Realizan propuestas en donde se siga manteniendo la agilidad, pero que se logren mejores niveles de usabilidad del software. En [20] proponen el desarrollo de software dirigido por el modelo de usuario (UMDD) el cual es un método centrado en el usuario y consiste de cuatro fases: modelado de usuario, modelado del diseñador, modelado de la implementación y evaluación de usabilidad. El modelado del usuario tiene como meta la construcción de la interfaz de usuario de acuerdo a la experiencia previa del usuario, su conocimiento y preconcepciones acerca de las tareas. Se realizan las actividades de definición de requisitos, modelado de la interfaz de usuario, prototipado de la interfaz de usuario y evaluación temprana de la usabilidad. Utiliza el modelo de roles, tareas y de dominio. La otra fase relevante para la usabilidad es la evaluación, en donde se establecen metas, métodos y criterios de evaluación.

En [21] presentan un modelo de proceso basado en métodos ágiles que fomenta la participación del usuario a través del diseño participativo y métodos de la semiótica organizacional. El ciclo de vida de la propuesta sigue la idea de XP. Incluye actividades como el análisis de expertos, prototipos de bajo costo, análisis de aceptación de usuario y talleres con usuarios. Indican algunas lecciones aprendidas de aplicar el modelo en un proyecto, como los beneficios que ofrecen las reuniones cara a cara para evaluar los diseños de interfaces y enfatizan el balance entre el nivel de calidad requerido para usabilidad y accesibilidad y la rapidez del proceso de desarrollo.

Hussain et al. [22] describen el proceso de desarrollo en el cual integran XP con el diseño centrado en el usuario. Señalan que los aspectos similares entre XP y UCD (*User Centered Design*) son la orientación al usuario final, prueba continua y desarrollo iterativo. El proceso comienza con la creación de historias y a continuación se elabora el prototipo en papel que evalúa el cliente. A partir de lo anterior se crea un prototipo final de acuerdo a los deseos de cliente y se implementa. Las pruebas de usabilidad se realizan por expertos en usabilidad y por usuarios finales. Por su parte, en [23] presentan una propuesta para integrar prototipos de bajo costo y alto costo en la metodología ágil XP. Señalan que los prototipos de bajo costo ofrecen realimentación inmediata, mientras que con los de alto costo identificaron problemas en el contenido de pantallas.

### **3.5 Propuestas que inciden en los requisitos**

Este grupo de propuestas están orientadas a obtener una especificación más completa y precisa de los requisitos de usabilidad. Para lograr este objetivo, los autores plantean la modificación de técnicas que inciden sobre los procesos de análisis del sistema y/o software.

En [24] proponen un método para evaluar casos de uso denominado Evaluación de Casos de Uso (UCE). Consta de tres actividades: inspección de casos de uso, valoración de casos de uso y documentación de la evaluación. Utilizan una evaluación heurística basada en guías de inspección de usabilidad. Sugieren que este método de evaluación puede ser complementario al método de evaluación denominado pensar en voz alta.

Por su parte, en [25] diseñaron un método que analiza la arquitectura de software de una aplicación móvil usando el método SATURN (*software architecture analysis of usability requirements realization*). Para realizar la evaluación de usabilidad se requiere de un prototipo funcional o producto final y los pasos son: describir el contexto, determinar el análisis de escenarios, evaluar escenarios, interpretar resultados y revisar métodos y herramientas. Los resultados del caso de estudio señalan que el 74% de los problemas de usabilidad pueden estar relacionados con problemas en el diseño arquitectónico. Encontraron el 30% de los problemas en el análisis de la arquitectura y el resto, durante la evaluación de usabilidad.

En [26] se presenta un método para obtener requisitos de interfaz de usuario basado en el modelado de la interfaz de usuario (UIM). Para ello se utilizan modelos de actor, de metas y de trabajo y un proceso iterativo con enfoque top-down. Los modelos se elaboran en papel y se distribuyen a todos los participantes de las sesiones para evaluarlos. El método se validó a través de investigación acción en una empresa sueca. Como inconvenientes se detectó que el método era dependiente del líder y hay dificultades para identificar la magnitud del conjunto de tareas adicionales asociadas a un escenario de trabajo particular.

En [27] adaptan la técnica “Personas”, utilizada para describir perfiles de usuario, para integrarla en la actividad de análisis del proceso de desarrollo de software. Identifican 11 actividades en “Personas” y las relacionan con las actividades de ingeniería de requisitos: obtención, análisis, especificación y validación. Finalmente, en [28] proponen una adaptación al Taller de Atributos de Calidad (*Quality Attribute Workshop*) para que sea dirigida por la usabilidad. El objetivo es descubrir y documentar requisitos de usabilidad utilizando como guías las propiedades de usabilidad de Folmer y los escenarios de Bass. Se espera que los usuarios identifiquen aquellos requisitos que crean relevantes para el software y que escriban un ejemplo concreto asociado. Con esas tareas se pretende identificar las tareas de alto nivel que los usuarios esperan del sistema. Se seleccionan los escenarios más representativos y se detallan los casos de uso. La validación de la técnica se realizó con estudiantes de grado y encontraron que las guías son útiles para articular requisitos de usabilidad.

#### **4 Incidencia de las prácticas de usabilidad en los procesos del ISO 12207**

Una vez presentadas las contribuciones de los artículos primarios, se realiza un análisis de los procesos a tener en cuenta para mejorar el nivel de usabilidad del software. Como referencia para este análisis se utiliza el estándar ISO/IEC 12207:2007.

Una premisa fundamental del estándar ISO/IEC 12207:2007 es que el software siempre exista en el contexto de un sistema, por tanto, en este trabajo se consideran los procesos técnicos del contexto del sistema y los procesos de implementación de software para integrar las recomendaciones encontradas en la literatura. Los procesos técnicos se usan para definir los requisitos para un sistema, la generación de un producto, su uso, mantenimiento hasta su retirada. Entre los beneficios que se buscan está la optimización de la usabilidad del sistema. Por su parte, los procesos de implementación de software se usan para producir un elemento de software que satisface los requisitos derivados a partir de los requisitos del sistema.

En la tabla 4 se presentan los procesos del estándar que pueden ser considerados para introducir prácticas que incrementen la usabilidad del software. Se observa una incidencia mayor de prácticas en los procesos de análisis de requisitos del sistema, identificación de requisitos de participantes, análisis de los requisitos de software y diseño arquitectónico. Esto coincide con [14], quienes señalan la convergencia en intereses entre las propuestas de diseño



centrado en el usuario y las prácticas de ingeniería de software en las etapas tempranas del desarrollo. Sin embargo, existen pocas propuestas en las etapas de diseño, construcción e integración, pruebas de aceptación y operación.

**Tabla 4.** Relación de procesos software de ISO 12207 con usabilidad de acuerdo a los resultados de la RSL.

Artículo ↓	Proceso →												
	6.4.1 Req. participantes	6.4.2 An. Req. Sistema	6.4.3 Dis. Arq. Sistema	6.4.4 Implementación	7.1.1 Implementación	7.1.2 Análisis	7.1.3 Diseño	7.1.4 D. Detallado	7.1.5 Construcción	7.1.6 Integración	7.1.7 Pruebas	6.4.8 Pruebas aceptación	6.4.9 Operación
anderson2001	x	x		x		x	x	x	x			x	x
biel2010				x			x				x		
bonacin2009	x	x		x	x				x				
castro2008	x	x		x		x							
constantine2002	x	x		x		x	x	x	x		x		
ferre2005	x	x	x	x		x	x					x	x
glissmann2005	x	x	x	x				x					
goransson2003	x	x		x	x		x	x				x	x
helms2006	x	x	x	x		x	x	x	x	x	x	x	x
hornbaek2007				x		x							
hussain2008		x		x		x			x	x	x		
lif1999	x	x		x		x	x						
memmel2007	x	x											
moreno2009	x	x		x		x	x		x		x		
nielsen1992	x	x	x	x	x	x	x	x	x		x	x	x
rafla2007	x	x		x		x							
sousa2005	x	x		x		x	x	x	x		x	x	
wang2008	x	x	x	x		x	x		x		x	x	
<i>Sumas</i>	<i>15</i>	<i>16</i>	<i>5</i>	<i>17</i>	<i>3</i>	<i>13</i>	<i>11</i>	<i>7</i>	<i>9</i>	<i>2</i>	<i>8</i>	<i>7</i>	<i>5</i>

Las prácticas relevantes asociadas a los procesos técnicos de sistema se presentan en la tabla 5. El énfasis de ellas está en conocer a los usuarios, sus metas y necesidades y la forma más eficiente en que la tecnología se puede adaptar en su ambiente de trabajo para lograr una experiencia más eficaz y eficiente. En los procesos de soporte a la aceptación y operación las prácticas se centran en el entrenamiento y las pruebas de usabilidad con usuarios, para conocer el nivel en que el sistema satisface los requisitos establecidos en el análisis del sistema.

**Tabla 5.** Técnicas y actividades referidas en los artículos primarios para los procesos técnicos de sistema del ISO 12207.

Proceso	Descripción	Técnicas y actividades referidas
6.4.1 Definir requisitos de participantes	Identificar a grupos y participantes Identificar contexto de uso, Identificar requisitos de usabilidad con base a capacidades y limitaciones de usuarios Verificar requisitos	Estudios etnográficos, entrevistas, grupos focales, personas, UQAW, modelos de actor y de metas, clasificación de usuarios, identificar características del ambiente, identificar características del hardware y software de soporte, modelo de roles, modelo de tareas, investigación contextual, perfiles de usuarios, propuesta de concepto de producto, comprender el contexto social Modelo de casos de uso, construcción de prototipos, modelos de situaciones de trabajo (agrupamiento de casos de uso para lograr meta de alto nivel), especificar requisitos de usabilidad y metas de usabilidad, historias de usuario, diagramas de afinidad, modelos/análisis de tarea, escenarios de uso, validación por usuarios y expertos
6. 4.2 Análisis de requisitos del sistema	Describir funciones y capacidades del sistema incluyendo requisitos de usuario. Verificación de requisitos	
6.4.4 Implementación	Elaborar el elemento de sistema especificado. Reemplazado por el 7.1.1 Implementación del software	Ver tabla 6.
6.4.3 Diseño arquitectónico	Se asignan requisitos del sistema a los componentes. Se identifican y realizan actividades de diseño centrado en el humano. Evaluación de arquitectura	Selección de plataformas para el desarrollo, elaboración de modelos abstractos de presentación, interacción y navegación, diseño de ambientes de interacción y el comportamiento asociado, diseño participativo con prototipos
6.4.7 Instalación del software	Instalar el producto en el ambiente destino	Desplegar el sistema en el ambiente destino
6.4.8 Soporte a la aceptación del software	Asistir al cliente a lograr la confianza de que el producto satisface los requisitos. Se provee entrenamiento	Evaluar el sistema con los usuarios para verificar que es usable, eficiente y preciso. Evaluación acumulativa, estudios de usabilidad en el sitio del usuario, pruebas de usabilidad empíricas
6.4.9 Operación del software	Operar el producto de software en el ambiente destino y proporcionar soporte a los clientes	Establecer criterios de evaluación cuando el producto está en operación, gestión del cambio, entrenamiento, evaluación de usabilidad a largo plazo, obtener datos para la siguiente versión

**Tabla 6.** Técnicas y actividades referidas en los artículos primarios para los procesos de implementación de software del ISO 12207.

Proceso	Descripción	Técnicas y actividades referidas
7.1.1 Implementación	Planeación del desarrollo de software y selección de métodos, herramientas, estándares adecuados	Elaboración de calendario para ciclo de prototipos, planeación del diseño y asignación de recursos, planear cómo usar los métodos
7.1.2 Análisis de requisitos de software	Establecer requisitos del software incluyendo características de calidad: Interacción hombre-equipo Áreas sensibles a errores y que requieren capacitación, verificación de requisitos	Casos de uso con escenarios alternativos, prototipos de baja fidelidad, ordenación de tarjetas, modelo de dominio, identificación de objetos, atributos y operaciones que deben estar presentes en la interfaz de usuario, viñetas, evaluación de productos de la competencia para establecer metas de usabilidad, modelado de tareas de usuario, modelado conceptual Validación con usuarios representativos
7.1.3 Diseño arquitectónico del software	Construir estructura de alto nivel e identificar componentes e interfaces Verificación de arquitectura	Elaborar modelo de navegación, presentación, prototipo de interfaz gráfica, usar patrón MVC para definir la arquitectura, modelo de datos, de componentes y de procesamiento/modelo de interacción Evaluaciones heurísticas y pruebas de usuario sobre prototipos
7.1.4 Diseño detallado de software	Especificar las características de los componentes en un nivel que permita la codificación y prueba Verificación del diseño detallado	Determinar objetos visuales y patrones de la interfaz de usuario, prototipos de alta fidelidad, consideración de principios de diseño en dominios particulares, definir las características de elementos visuales y de interacción de las pantallas, elaborar diseño detallado para contextos de interacción seleccionados Validación con usuarios y expertos
7.1.5 Construcción del software	Desarrollar y documentar cada unidad de software. Probar cada unidad	Implementación del prototipo, presentación de contenidos en interfaz web, refactorización de código, uso de herramientas automáticas, uso de patrones de diseño y bibliotecas de componentes, reutilizar código Evaluación por expertos y usuarios
7.1.6 Integración del software	Desarrollar plan de integración de componentes y requisitos de prueba Realizar integración y prueba	Integrar los componentes
7.1.7 Prueba de cualificación del software	Se realizan pruebas de cualificación de acuerdo a requisitos especificados.	Evaluaciones heurísticas y caminatas, pruebas de usabilidad guiadas por expertos, establecer meta, método y criterio de evaluación, inspecciones de usabilidad colaborativas

Las técnicas y/o prácticas asociadas a los procesos de implementación del software se muestran en la tabla 6. En primer lugar se recomienda la planificación de actividades y la selección de métodos adecuados de soporte a la usabilidad considerando un enfoque iterativo, así como las propuestas basadas en casos de uso para documentar las tareas del usuario y responder a condiciones de error a través de escenarios alternativos. Para abordar tareas de diseño de interfaz resaltan los modelos de interacción, de navegación, y la construcción de prototipos en diferentes niveles de fidelidad para organizar los objetos de interacción y de información y refinar el comportamiento esperado. En los procesos de diseño y construcción se recomienda el uso de patrones, estándares y generación automática de código.

Finalmente, en la evaluación del producto, previo a su instalación, se pueden usar diversas técnicas de evaluación de usabilidad, como evaluación heurística y pruebas empíricas de usabilidad.

El análisis de estos estudios primarios contribuye a la identificación de tareas que pueden ser integradas en el proceso de desarrollo de software. Sin embargo, todavía hay poca evidencia empírica del nivel de efectividad de las prácticas en los resultados del proyecto. Por otro lado, aunque algunas propuestas consideran los estándares de la ISO en lo que respecta a integrar tareas de soporte a la usabilidad, no se encontraron referencias a estándares de la calidad del software, como el ISO 9126.

Desde una perspectiva de ingeniería es importante medir los atributos de calidad, y en este sentido fueron muy pocas las propuestas que incluyen medición del producto o proceso. Por tanto, la solicitud de un proceso de desarrollo de software completo, que integre la usabilidad, como indica [8], sigue vigente.

## 5 Conclusiones

En este artículo se han presentado los resultados de una revisión sistemática para conocer en qué procesos de desarrollo de software se debe incidir para mejorar la usabilidad de los productos finales. Como resultado de la revisión se puede concluir que el número de trabajos relevantes no es muy elevado, lo que motiva la necesidad de seguir profundizando en la relación calidad proceso producto desde la perspectiva de la usabilidad.

La clasificación de las propuestas ayuda a comprender en qué áreas se está trabajando con usabilidad y procesos de software. Así, las áreas que han recibido más interés se basan en adaptaciones al RUP, elección de marcos generales de procesos, métodos ágiles y mejoras en la etapa de análisis y especificación de requisitos de usabilidad.

Las propuestas que introducen prácticas de diseño centrado en el usuario coinciden en la necesidad de obtener los requisitos de usabilidad a partir del usuario, comprender el contexto de uso, desarrollar interfaces y mecanismos de interacción a través de prototipos, y de evaluación por los usuarios. Por su parte, los trabajos primarios orientados a la fase de análisis estudian la adaptación de técnicas de usabilidad al proceso de análisis y adecuación de técnicas de obtención de requisitos y evaluación de arquitectura para incluir aspectos relativos a la usabilidad.

Las actividades y prácticas recomendadas en los artículos primarios se proyectaron en los procesos técnicos de sistema y de implementación de software del ISO/IEC 12207 con la finalidad de identificar los procesos que una organización debe modificar si pretende introducir prácticas de usabilidad. Se encontró que la mayoría de las prácticas incide en los procesos de identificación de requisitos de participantes, análisis de requisitos del sistema, análisis de

requisitos de software y diseño de alto nivel del software. También se encontraron algunas prácticas que inciden en los procesos de prueba de cualificación, pruebas de aceptación y operación.

Como trabajo futuro se espera continuar este análisis para conocer el impacto de las prácticas de usabilidad en otras categorías de procesos del ISO/IEC 12207, como aquellas relacionadas con el aseguramiento de la calidad. Además, este trabajo se enfocó en las actividades del proceso, pero es necesario considerar otros elementos, como los artefactos asociados y roles. Por otra parte, se requieren estudios empíricos para evaluar el impacto de las prácticas en la usabilidad del software y en éstos se deberían establecer indicadores para medir el grado de usabilidad logrado. Finalmente, el análisis de la relación entre el proceso de desarrollo y las características de calidad del software se puede extender a otros factores, como aquellos descritos por el ISO/IEC 25010.

**Agradecimientos.** Este trabajo ha sido financiado por los proyectos: PEGASO/MAGO (MICINN y FEDER de España, TIN2009-13718-C02-01), VILMA (JCCM, Fondo Social Europeo, PEII 11-0316-2878), ALTAMIRA (JCCM, Fondo Social Europeo, PII2I09-0106-2463).

## Referencias

1. Staples, M. and M. Niazi, *Systematic review of organizational motivations for adopting CMM-based SPI*. Information and Software Technology, 2008. **50**: p. 605-620.
2. Agrawal, M. and K. Chari, *Software effort, quality, and cycle time: a study of CMM level 5 projects*. IEEE Transactions on Software Engineering, 2007. **33**(3): p. 145-156.
3. Herbsleb, J., et al., *Software quality and the Capability Maturity Model*. Communications of the ACM 1997. **40**(6): p. 31-40.
4. De Oliveira, S., R. Valle, and C. Mahler, *A comparative analysis of CMMI software project management by Brazilian, Indian, and Chinese companies*. Software Quality Journal, 2010. **18**: p. 177-194.
5. Allen, J., B. Kitchenham, and M. Konrad, *Theme Q. The relationships between processes and product qualities.*, in *A Process Research Framework*, E. Forrester, Editor. 2006, Software Engineering Institute, Carnegie Mellon University. pp. 172. En <http://www.sei.cmu.edu/publications/books/process/iprc.html>.
6. Kitchenham, B. and S. Pfleeger, *Software quality: the elusive target*. IEEE Software, 1996. **13**(1): p. 112-21.
7. JTC1/SC7/WG6, *ISO/IEC FCD 25010: Systems and software engineering – System and software product Quality Requirements and Evaluation (SQuARE) – System and software quality models* 2010.
8. Seffah, A. and E. Metzker, *The obstacles and myths of usability and software engineering*. Comm. Of the ACM, 2004. **47**(12): p. 71-76.
9. Kitchenham, B., *Procedures for Undertaking Systematic Reviews*. . 2004. Joint Technical Report, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd (0400011T.1).
10. Kitchenham, B. and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering* 2007. Technical Report EBSE/EPIC-2007-01.
11. Anderson, J., et al., *Integrating usability techniques into software development*. Software, IEEE, 2001. **18**(1): p. 46-53.
12. Göransson, B., J. Gulliksen, and I. Boivie, *The usability design process - Integrating user-centered systems design in the software development process*. Software Process Improvement and Practice, 2003. **8**(2): p. 111-131.

13. Sousa, K., E. Furtado, and H. Mendonça. *UPi - A software development process aiming at usability, productivity and integration*. in *ACM International Conference Proceeding Series 124*. 2005. p. 76-87.
14. Ferre, X., N. Juriste, and A.M. Moreno. *Framework for integrating usability practices into the software process*. in *Lecture Notes in Computer Science 3547*. 2005. p. 202-215.
15. Helms, J.W., et al., *A field study of the Wheel-a usability engineering process model*. *Journal of Systems and Software*, 2006. **79**(6): p. 841-858.
16. Nielsen, J., *The usability engineering life cycle*. *Computer*, 1992. **25**(3): p. 12-22.
17. Glissmann, S., et al. *Proposition of an m-business procedure model for the development of mobile user interfaces*. in *Mobile Business, 2005. ICMB 2005. International Conference on*. 2005. p. 308-314.
18. Constantine, L.L. and L.A.D. Lockwood, *Usage-centered engineering for web applications*. *IEEE Software*, 2002. **19**(2): p. 42-50.
19. Moreno, L., P. Martínez, and B. Ruiz-Mezcua, *Integrating HCI in a web accessibility engineering approach*. 2009. p. 745-754.
20. Wang, X. and Y. Shi. *UMDD: User model driven software development*. in *5th International Conference on Embedded and Ubiquitous Computing*. 2008. p. 477-483.
21. Bonacin, R., B. Calani, and M. Rodriguez, *An Agile Process Model for Inclusive Software Development*. 2009. **24 LNBIP**: p. 807 - 818.
22. Hussain, Z., et al., *Agile user-centered design applied to a mobile multimedia streaming application*. 2008. p. 313-330.
23. Memmel, T., H. Reiterer, and A. Holzinger, *Agile methods and visual specification in software development: A chance to ensure universal access*. 2007. p. 453-462.
24. Hornbæk, K., et al., *Use case evaluation (UCE): A method for early usability evaluation in software development*. 2007. p. 578-591.
25. Biel, B., T. Grill, and V. Gruhn, *Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application*. *Journal of Systems and Software*, 2010. **83**(11): p. 2031-2044.
26. Lif, M., *User-interface modelling--adding usability to use cases*. *International Journal of Human-Computer Studies*, 1999. **50**(3): p. 243-262.
27. Castro, J.W., S.T. Acua, and N. Juristo. *Integrating the Personas Technique into the Requirements Analysis Activity*. in *Computer Science, 2008. ENC '08. Mexican International Conference on*. 2008. p. 104-112.
28. Rafla, T., P.N. Robillard, and M. Desmarais, *A method to elicit architecturally sensitive usability requirements: Its integration into a software development process*. *Software Quality Journal*, 2007. **15**(2): p. 117-133.
29. Constantine, L.L. and L.A.D. Lockwood, *Usage-centered engineering for Web applications*. *IEEE Software*, 2002. **19**(2): p. 42-50.