Análisis de Procesos de Software Orientados a la Usabilidad

José Ignacio Rocca
Universidad ORT Uruguay

Martin Solari
Universidad ORT Uruguay

Abril 2018

Resumen. Las principales tendencias tecnológicas de software (Internet of Things, computación ubicua, sistemas contextuales, etc.) promueven la adopción masiva de aplicaciones por parte de usuarios en contextos heterogéneos. El software está cada vez más presente en la vida diaria de las personas y la usabilidad es de crítica importancia para su adopción y utilidad. Es en este contexto, que surge, la necesidad de definir un enfoque sistemático para la adopción de técnicas y prácticas que posibiliten el desarrollo de productos con alta usabilidad. El objetivo de este trabajo es presentar un análisis detallado de las propuestas de procesos de software orientadas a la usabilidad que fueron detectadas previamente mediante un mapeo sistemático de la literatura. Es posible concluir que existen factores comunes a todas las propuestas de procesos de software orientados a la usabilidad y aspectos o prácticas complementarias que son opcionales y adaptables al contexto específico del proceso propuesto.
Palabras clave: proceso de software, usabilidad.

Abstract. The main technological software trends (Internet of Things, ubiquitous computing, context-aware systems, etc.) promote the massive adoption of applications by users in heterogeneous contexts. Software is increasingly present in people's daily lives and usability is critical for its adoption and usefulness. It is in this context, that arises, the need to define a systematic approach for the adoption of techniques and practices that enable the development of products with high usability. The objective of this paper is to present a detailed analysis of the usability-oriented software process proposals that were previously detected through a systematic mapping of the literature. It is possible to conclude that there are common factors to all the proposals of usability-oriented software processes and aspects or complementary practices that are optional and adaptable to the specific context of the proposed process.
Keywords: software process, usability.

Documento de Investigación, Nro.14, abril, 2018. Universidad ORT Uruguay.
Facultad de Ingeniería. ISSN 1688-6372.

1 Introducción

1.1 Procesos de software y usabilidad

Como se estableció en el mapeo sistemático de la literatura [1] realizado previamente por los autores, en el contexto tecnológico actual existe alta demanda de usabilidad por parte de los usuarios finales. De igual manera se detecta un interés creciente por parte de la comunidad de la ingeniería de software por enfocar de manera sistemática en sus procesos de desarrollo la adopción de métodos y prácticas que generen productos con mayores niveles de usabilidad.

Para la ingeniería de software los procesos brindan un enfoque sistemático tanto para el desarrollo como para la evolución del software. Feiler y Humphrey [2] coinciden en que más que representar una prescripción rígida de actividades los procesos de software pueden definirse como una serie de criterios adaptables al contexto. Los procesos de software son útiles para facilitar la comunicación y coordinación de personas, brindar soporte a la gestión de proyectos, soportar la mejora continua del equipo, pero principalmente para asegurar el desarrollo de un producto de calidad. A lo largo del tiempo, en la ingeniería del software han surgido dos grandes filosofías (predictiva y ágil) asociadas a los procesos de software. De igual forma, se han definido propuestas de procesos de software que se orientan o consideran diferentes atributos de calidad de software como ser mantenibilidad, seguridad o performance.

Por otro lado, la usabilidad es definida como la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a los usuarios en un contexto de uso determinado [3]. Dentro de la usabilidad se puede analizar un conjunto de sub-atributos como ser: prevención de errores, facilidad de aprendizaje, eficiencia en la tarea, facilidad para recordar, estética y accesibilidad [4].

Históricamente la ingeniería de software y las disciplinas asociadas a la usabilidad como ser la interacción hombre computadora (en inglés Human Computer Interaction, HCI), la Ingeniería de la Usabilidad (Usability Engineering, UE en inglés) o el diseño centrado en el usuario (User Centered Design, UCD en inglés) han evolucionado en forma paralela y sin conexión [5], generando una brecha entre distintas disciplinas.

1.2 Motivación

El proceso que integra nuevos roles y considera prácticas de usabilidad específicas a lo largo del ciclo de vida es considerado un proceso de software orientado a la usabilidad. El objetivo del presente trabajo, es extender y profundizar el análisis de los procesos orientados a la usabilidad que fueron identificados en el mapeo sistemático de la literatura [1] de referencia.

Si bien la mayoría de las propuestas de proceso detectadas en el mapeo tienen su origen conceptual dentro de la ingeniería de software, también se detectaron propuestas provenientes de disciplinas alternativas como ser HCI, UE y UCD. Esto tiende a disminuir la brecha existente entre la ingeniería de software y dichas disciplinas. Otro aspecto a destacar es que dentro de los procesos con origen en la ingeniería de softwa-

re existe un número similar de procesos ágiles y predictivos. Esto indica que la importancia y criticidad de la usabilidad es independiente de la filosofía de desarrollo utilizada en el proceso de base y la integración de la usabilidad se busca de forma transversal tanto en procesos ágiles como en procesos predictivos.

No obstante, las propuestas de proceso detectadas poseen bajos niveles de replicabilidad, documentación o difusión hacia la comunidad profesional de la ingeniería de software. Esto complejiza y limita la aplicabilidad de los procesos en la industria, así como también dificulta la profundización e investigación futura en el campo.

Es en este contexto que resulta de interés, analizar teóricamente la información disponible de los 31 procesos de software orientados a la usabilidad detectados en el estudio de mapeo [1]. De este análisis será posible extender o generalizar algunos aspectos comunes a todos los procesos, comprender las técnicas más utilizadas, analizar la integración de equipos multidisciplinarios, la participación de usuarios y otros aspectos relevantes. De igual manera resulta de interés contrastar y comparar los procesos dentro de su misma familia u origen conceptual como para comprender aspectos y discusiones comunes. El objetivo de este trabajo es facilitar y promover la integración de la usabilidad en los procesos de software a partir del análisis teórico de las propuestas detectadas y contribuir a la aplicación de dichos procesos por parte de las comunidades profesionales de la ingeniería de software.

A continuación se detalla la metodología de análisis utilizada para los diferentes procesos. En la sección 3 se analizan los distintos procesos de software orientados a la usabilidad agrupando los mismos según origen conceptual. Por último en sección 4 se presenta la discusión y conclusión del análisis realizado.

2 Metodología de análisis

El objetivo de esta sección es detallar la metodología de análisis aplicada a los distintos procesos. Para facilitar la lectura se presentan sub secciones explicando diferentes aspectos de la metodología aplicada.

2.1 Agrupación y origen conceptual

Un aspecto relevante de la extracción de datos realizada en el mapeo fue establecer el origen conceptual de los distintos procesos. En ese sentido se codificaron las siguientes opciones:

- Ágil: para aquellos procesos con origen en metodologías ágiles.
- Predictivo: para aquellos procesos que no se presentan como ágiles y son adaptaciones de procesos predictivos como ser (UP, Waterfall, RAD, etc)
- UCD: para aquellas propuestas de procesos cuyo origen conceptual es la disciplina de diseño centrado en el usuario (UCD)
- UE: procesos que explícitamente se basan en la propuesta Usability Engineering de Nielsen [6].

- Cognitiva/psicología: procesos basados en aspectos de psicología provenientes del área de HCI.
- No aplica (N/A): para aquellas propuestas de procesos que no tiene un origen conceptual claramente identificable.

A efectos del análisis teórico, los procesos se agruparon bajo procesos predictivos, procesos ágiles y procesos con origen en UCD, UE, HCI y otros.

2.2 Identificación

Otro aspecto analizado es la existencia de un nombre propio para la propuesta de proceso. Si bien muchos de los 31 procesos de software orientados a la usabilidad detectados previamente son identificados mediante un nombre, a efectos de facilitar la trazabilidad con los artículos detectados originalmente en el mapeo se utiliza el apellido del autor principal y su año publicación al inicio de cada análisis. Ejemplo: Ben Ayed2010 – UP/U (nombre autor y año – nombre del proceso). En los casos donde no se utiliza un nombre propio para el proceso se mantiene únicamente la codificación Autor – Año.

2.3 Aspectos generales

En la sección de aspectos generales se detallan características generales del proceso que surgen de la presentación introductoria del mismo. Otros aspectos analizados son la evolución en la línea de investigación seguida por los autores del proceso, aspectos relacionados con la especialización del proceso a contextos específicos y su validación mediante casos de estudio u otros estudios empíricos.

2.4 Declaración de principios

En esta sección se analizan los principios y valores que sustentan los procesos propuestos. Tanto los principios como los valores contemplados por un proceso describen aspectos fundamentales que hacen a definiciones políticas y/o estratégicas del deber ser de un proceso. Establecen límites y conductas deseadas que luego son implementadas por las actividades y prácticas en los procesos.

2.5 Representación gráfica

En el mapeo, se analizó la notación utilizada para describir y comunicar el proceso. Como resultado se obtuvo que el 87% de los procesos fueron descriptos utilizando imágenes y prosa, mientras que solo un 13% utilizaba notación específica de procesos como ser BMPN o SPEM. La representación gráfica de un proceso es de vital importancia para el entendimiento y difusión del mismo. En esta sección se extrae de las distintas propuestas de proceso las imágenes que ilustran y mejor representa el proceso analizado en cuestión. En todos los casos se ha citado la representación gráfica usada por los autores en forma literal (imagen de la publicación original).

2.6 Detalles del proceso

En esta sección del análisis para cada uno de los procesos identificados se presentan detalles específicos. Roles que se incorporan, fases recomendadas para el proceso, técnicas utilizadas a lo largo del ciclo de vida, herramientas que soporten determinada actividad.

3 Análisis y resumen de procesos orientados a la usabilidad

3.1 Procesos Predictivos

Ben Ayed2010 – UP/U

Aspectos generales: El proceso propuesto integra un modelo de procesos existente (U model) con aspectos de HCI y UP. La propuesta de proceso se centra en el desarrollo de aplicaciones de soporte a la decisión basada en procesos de Data Mining. El proceso se valida empíricamente presentando un caso de estudio en el dominio de Salud.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

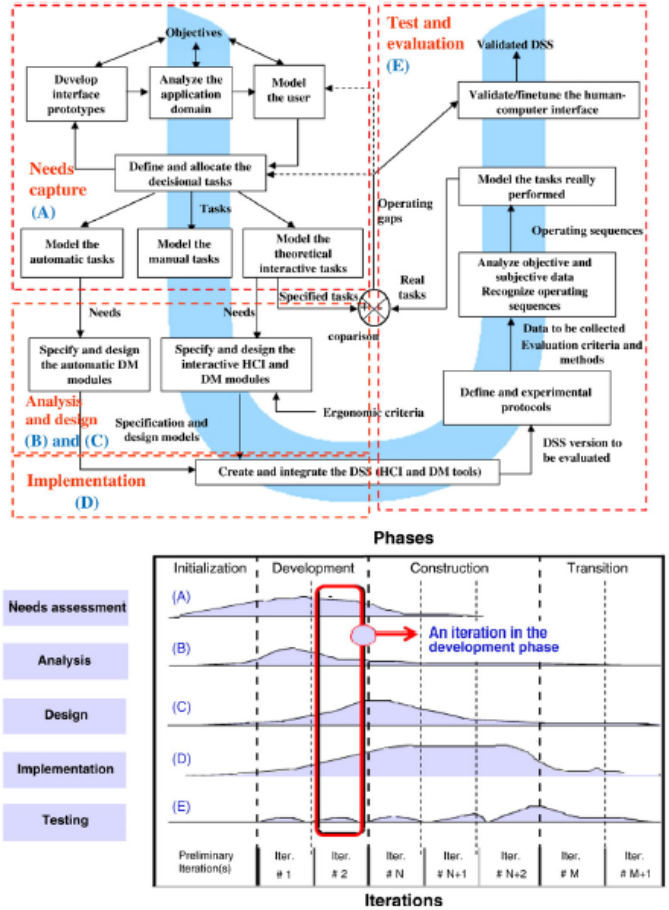


Fig. 1. Tomado de [7]

Detalles del proceso: El proceso iterativo propuesto mapea las disciplinas de UP, con las actividades que se deben ejecutar en el marco del modelo U enriquecido. El modelo U enriquecido posee una fase descendente y una ascendente que se ejecutan en cada iteración del proceso.

Al comienzo de la fase descendente, el proceso propone el modelado de los objetivos, los usuarios y las distintas tareas de toma de decisiones. Se propone tipificar las tareas de toma de decisión por ejemplo cuales decisiones son automáticas, manuales o interactivas. Sobre el final de la fase descendente se hace el análisis y diseño detallado e implementan las soluciones propuestas.

En la fase ascendente del modelo se evalúan experimentalmente los aspectos relevantes del sistema desarrollado (por ejemplo, efectividad, eficiencia, satisfacción o tasa de errores).

Benbelkacem2014 - 2TUP modificado para HCI

Aspectos generales: La propuesta toma como base el proceso (proveniente de la ingeniería de software) 2TUP (Two Tracks Unified Process) e integra prácticas, técnicas y modelos específicos de HCI para diseñar interfaces de usuario de realidad aumentada (mixed reality) y tradicionales. El proceso es validado empíricamente con un caso de estudio de una aplicación de mantenimiento de maquinaria industrial.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

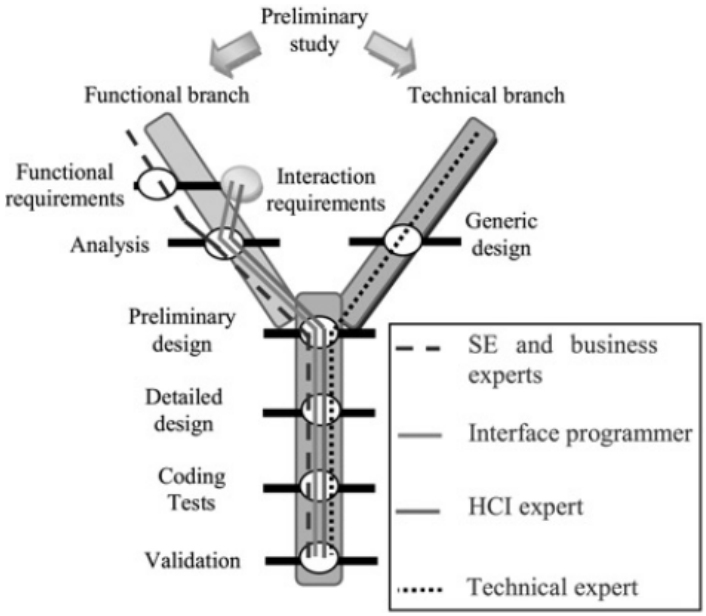


Fig. 2. Tomado de [8]

Detalles del proceso: El proceso es iterativo dado que toma como base el 2TUP. Se centra en describir las intenciones e interacciones de los usuarios en lugar de centrarse en las tareas a diferencia de otros procesos basados en UCD. Para esto se utilizan modelos específicos de usuarios e interacciones.

Mediante una extensión de UML, se agrega un concepto de "interaction object" y una relación de "describe" para materializar la correspondencia entre los "bussines objects" e "interacion objects". En el proceso, las especificaciones de interacción se establecen al mismo nivel que las especificaciones funcionales. De igual manera, la colaboración entre los distintos actores (expertos en ingeniería de software, HCI, en el negocio, programadores de UI, entre otros) es formalizada mediante el uso de artefactos comunes.

Fischer2011 - Fischer2011 Holistic Model

Aspectos generales: La propuesta de proceso integra Usability Engineering, software engineering y potenciales nexos con marketing. Se identificaron actividades y artefactos comunes que se amplían desde la perspectiva de marketing. Se proponen 3 niveles de integración: estándares, procesos y operaciones o procedimientos. Todo es representado con un modelo holístico que muestra actividades y artefactos comunes. Dicho modelo se encuentra en detalle en la tesis de Fischer que está redactada en alemán, por lo que no es posible analizarla en detalle.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

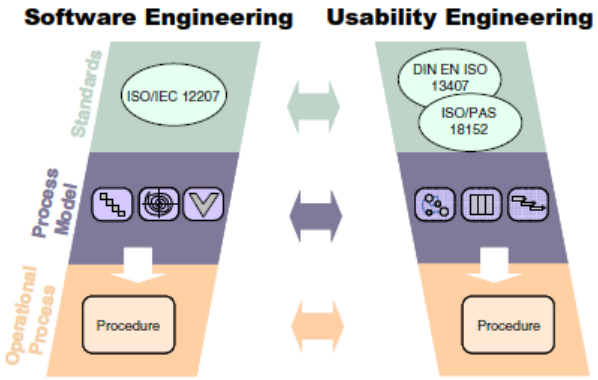


Fig. 3. Tomado de [9]

Detalles del proceso: No se tiene acceso al detalle del proceso propuesto.

Goransson2003 - Goransson2003

Aspectos generales: La propuesta de proceso busca adaptar RUP para que sea más centrado en el usuario. Para lograr esto, se propone agregar una nueva disciplina al

proceso e introducir 5 roles nuevos. La propuesta de proceso integra la experiencia y conocimientos previos de los investigadores en temas como: la definición de métodos para diseño y modelado de interfaz de usuario que complementen a UML y casos de uso, la inclusión de técnicas específicas y la incorporación de un nuevo rol en el proceso el usability designer.

Declaración de principios:

- Foco en el usuario
- Involucramiento activo de usuarios
- Desarrollo de sistemas en evoluciones
- Representaciones de diseño sencillas
- Uso de prototipos
- Evaluar en el contexto de uso
- Actividades de diseño explicitas y consientes
- Actitud profesional
- Incluir expertos en usabilidad durante todo el ciclo
- Diseño holístico
- Adaptación del proceso UCD a cada realidad
- Establecimiento de una actitud UCD

Representación gráfica:

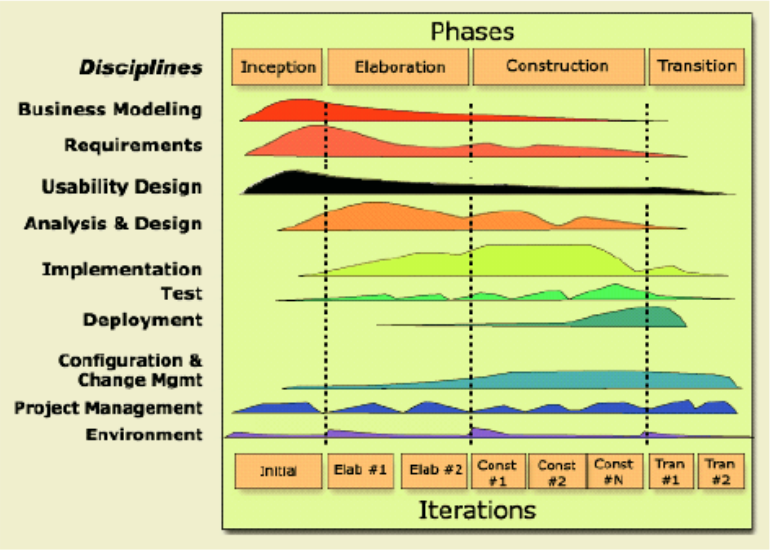


Fig. 4. Tomado de [10]

Detalles del proceso: El proceso es una adaptación RUP por lo que puede considerarse un proceso iterativo. Se establecen 5 roles nuevos, lo cual no implica que tengan que ser ejecutados por personas diferentes.

Todo el proceso está descripto en términos de UML como todo el modelo RUP para que sea consistente. Cada una de estas actividades define la aplicación de ciertos métodos y técnicas para lograr determinados entregables.

Algunas de las prácticas y técnicas específicas propuestas por el proceso incluyen: el uso de guías de estilo corporativas, el análisis de tareas y de utilización de información por parte de los usuarios, el uso de metáforas de workspace, entre otras.

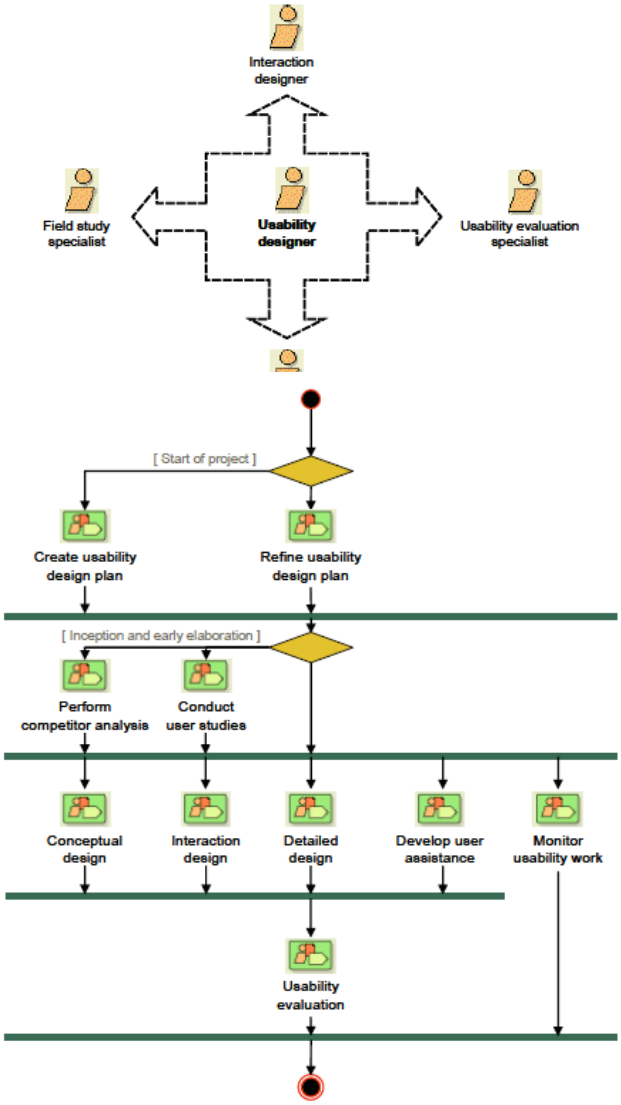


Fig. 5. Tomado de [10]

Konstantinidis2012 - Konstantinidis2012

Aspectos generales: La propuesta de proceso busca integrar aspectos de usabilidad en Unified Process (UP) para el desarrollo de un sistema de información clínica.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

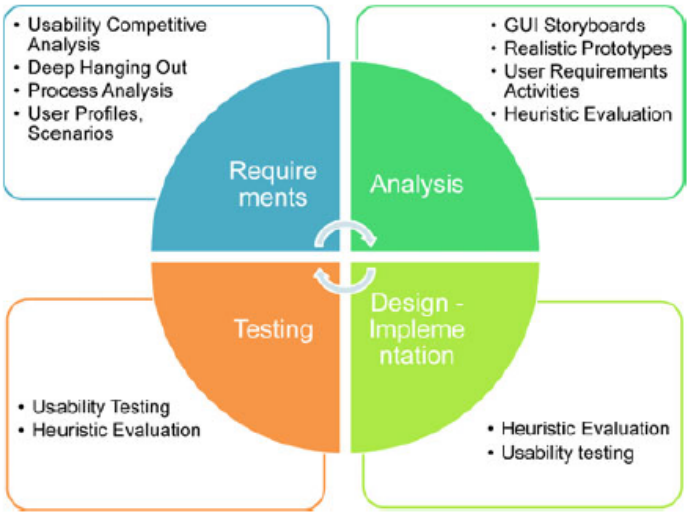


Fig. 6. Tomado de [11]

Detalles del proceso: El proceso es iterativo ya que parte de UP y se realiza una adaptación del mismo, utilizando sólo 4 de los 13 diagramas UML. Igualmente se propone simplificar la forma de documentar los requerimientos mediante descripción de escenarios y user profiles.

Para cada fase del UP, Inception, Elaboration, Construction y Transition, se ejecutan iteraciones del workflow básico de UP con las actividades y técnicas específicas establecidas en la representación gráfica. A modo de ejemplo en la disciplina de Análisis se propone la realización de Storyboards de UI, prototipos, evaluación heurística entre otras.

Maguire2001 - Maguire2001

Aspectos generales: El proceso propone instanciar el Framework de ISO 13407 (Human Centred Design Framework) acompañando cada etapa del proceso de Human centred design con métodos y técnicas específicas.

El proceso resulta adaptable a diferentes contextos debido a que las técnicas propuestas son evaluadas y comparadas para posibilitar el ajuste del proceso al proyecto particular.

Declaración de principios:

- Involucramiento activo de usuarios y clara comprensión de usuarios y tareas.

- Equilibrio apropiado entre funcionalidad provista y acciones/actividades a cargo de los usuarios.
- Iteración en el diseño de soluciones
- Equipos de diseño multi-disciplinarios.

Representación gráfica:

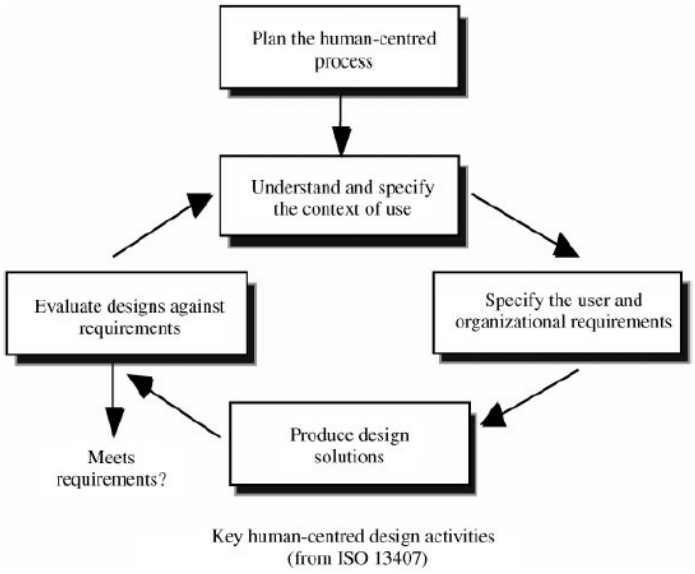


Fig. 7. Tomado de [12]

Detalles del proceso: El proceso propuesto es iterativo. Se definen 5 subprocesos claves de acuerdo a la ISO 13407 que tienen que seguir el flujo descripto la representación gráfica. Para cada subproceso, se presentan los métodos y técnicas propuestos.

- Planificar el proceso de diseño human-centred
- Entender y especificar el contexto de uso
- Especificar los usuarios y los requerimientos organizacionales
- Producir diseños y prototipos
- Llevar a cabo evaluaciones basadas en usuarios

Methods for human-centred design

Planning (Section 3)	Context of use (Section 4)	Requirements (Section 5)	Design (Section 6)	Evaluation (Section 7)
3.1. Usability planning and scoping 3.2. Usability cost–benefit analysis	4.1. Identify stakeholders 4.2. Context of use analysis 4.3. Survey of existing users 4.4. Field study/user observation 4.5. Diary keeping 4.6. Task analysis	5.1. Stakeholder analysis 5.2. User cost–benefit analysis 5.3. User requirements interview 5.4. Focus groups 5.5. Scenarios of use 5.6. Personas 5.7. Existing system/ competitor analysis 5.8. Task/function mapping 5.9. Allocation of function 5.10. User, usability and organizational requirements	6.1. Brainstorming 6.2. Parallel design 6.3. Design guidelines and standards 6.4. Storyboarding 6.5. Affinity diagram 6.6. Card sorting 6.7. Paper prototyping 6.8. Software prototyping 6.9. Wizard-of-Oz prototyping 6.10. Organizational prototyping	7.1. Participatory evaluation 7.2. Assisted evaluation 7.3. Heuristic or expert evaluation 7.4. Controlled user testing 7.5. Satisfaction questionnaires 7.6. Assessing cognitive workload 7.7. Critical incidents 7.8. Post-experience interviews

Fig. 8. Tomado de [12]

Nosseir2012 - Mobile Development Process Spiral

Aspectos generales: Se propone un proceso de desarrollo de software usable especializado para el contexto de aplicaciones móviles. El proceso es predictivo ya que utiliza como base el modelo de espiral de Bohem que tiene orientación a la evaluación periódica y reducción de riesgo. El modelo espiral de Bohem además es reportado por estudios previos como uno de los modelos más utilizados para incorporar actividades, métodos y técnicas que aseguren la usabilidad.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

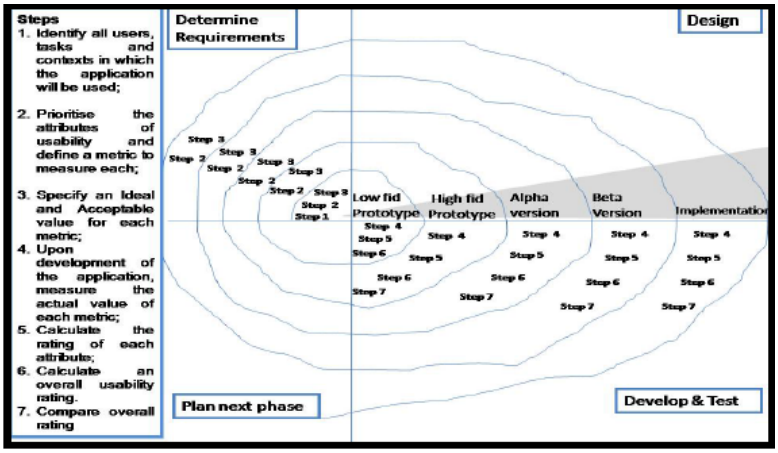


Fig. 9. Tomado de [13]

Detalles del proceso: El proceso es iterativo y es descripto a partir de las primeras iteraciones. Al comienzo del proyecto deben definirse los atributos de usabilidad que se quieren medir y el valor objetivo para cada atributo.

A medida que pasan estás iteraciones se van incorporando técnicas y actividades específicas. Por ejemplo en las primeras iteraciones se hacen prototipos o sketches de baja fidelidad y esto se va evolucionando a medida que pasan las iteraciones.

Se describen en total 5 iteraciones. En cada iteración se hace Determinación de requerimientos, diseño, desarrollo y prueba y planificación de la siguiente iteración.

- Técnicas recomendadas iteración 1: holistic design, product visión, análisis de necesidades, análisis de usuarios, análisis de mercado, personas, modelado organizacional, observación, entrevistas, definición de métricas de usabilidad.
- Técnicas recomendadas iteración 2: entrevistas, escenarios, card sorting, brainstorming.
- Técnicas recomendadas otras iteraciones: flujos de estado, diagramas de navegación, diagramas de transición de interfaces, análisis cognitivo de tareas. Prototipos de baja y alta fidelidad, mago de oz, test de usabilidad, JAD, cognitive walkthorugh, evaluación heurística. Planificar y medir las métricas de usabilidad.

Rodriguez2014 - OpenUP/UMM-ISO

Aspectos generales: La propuesta de proceso busca extender Open UP para dar cumplimiento al modelo Usability Maturity Modelo UMM – ISO. Para esto, se hace un diagnóstico previo para evaluar la conformidad de Open UP con el modelo y se detectaron gaps y oportunidades de mejora. La propuesta proceso se valida mediante un proyecto de desarrollo para una aplicación de gestión financiera de Gobierno.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

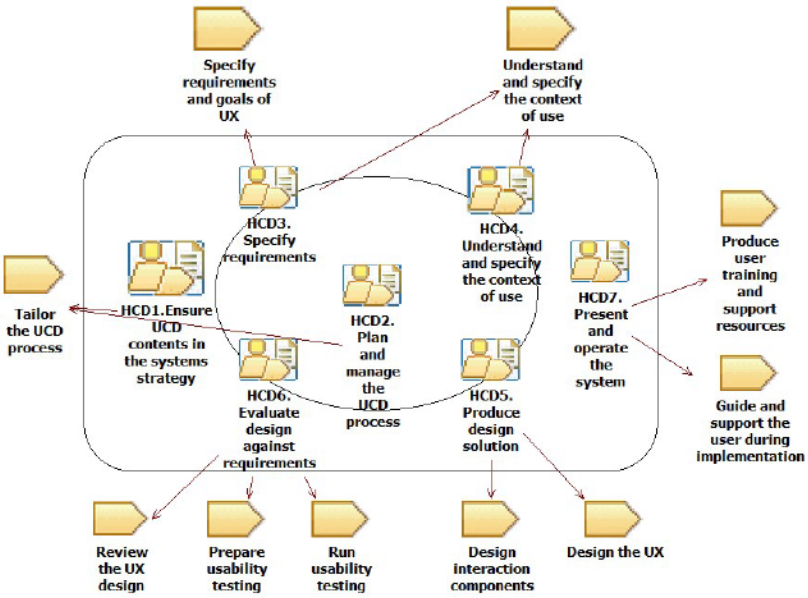


Fig. 10. Tomado de [14]

Detalles del proceso: El proceso parte de Open UP que es la versión libre de Unified Process, por lo tanto es considerado un proceso iterativo y predictivo. El proceso toma como base la definición de Open UP disponible dentro del Eclipse Process Framework (EPF) siguiendo la notación SPEM.

Para dar conformidad al modelo UMM, se realizó una extensión a la Eclipse Process Framework Practice Library (EPL) incorporando los gaps detectados durante el assesment o diagnóstico de procesos en prácticas de Human Centered Development,

El proceso final adiciona ítems a las prácticas base, modifica las prácticas cross process y propone roles alternativos. En EPF se crea una práctica (de SPEM) User Centered Development (UCDev) que incluye los roles, tareas, entregables, métodos y guías necesarios para involucrar activamente a los stakeholders durante todo el ciclo de vida.

Roles: 4 Bussiness actors (sponsor, domain technical leader, user representative, end user). 1 Analyst (UX specialist), 1 developer (UX designer), 1 Tester (UX tester)

Work Products: User model, Usability Goal, Task model, UX Concept, UX prototype, UX Storyboard, Navigation map and User and training document.

Tareas: Las tareas especifican, un propósito, relaciones con otros elementos (ej roles o work products) y una serie de pasos a seguir. Se agregaron las siguientes tareas:

- Tailor the UCD process
- Specify requirements and objectives of user experience
- Understand and specify the context of use
- Designing the user experience
- Components of interaction design

- Review the design of the user experience
- Preparing for usability testing
- Run usability tests
- Design and produce material training and user support
- Providing training and user support

El proceso resultado fue adaptado para que sea conformante con los distintos niveles de UMM-ISO. Por lo tanto la extensión de EPF desarrollada puede utilizarse para OpenUP/UMM-ISO N1, OpenUP/UMM-ISO N2 y OpenUP/UMM-ISO N3.

Soares2003 - RUPi

Aspectos generales: El proceso se llama RUPi por Rational Unified Process for interactive systems. La propuesta busca integrar HCI en el Software development process. La base del proceso RUPi es UMLi, una extensión a UML para modelar aspectos de sistemas interactivos como ser la UI.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

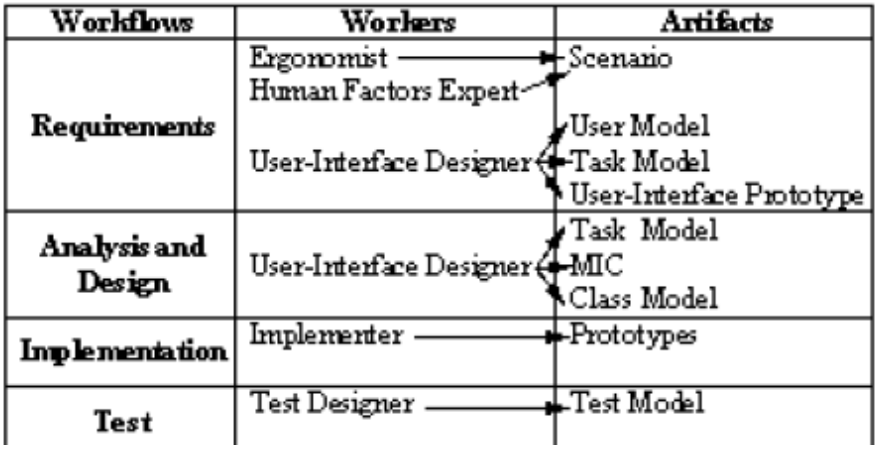


Fig. 11. Tomado de [15]

Detalles del proceso: El proceso es considerado iterativo y predictivo porque utiliza como proceso base RUP. Se incluyen mecanismos de diseño y modelado de UI basado en UML para las etapas iniciales del ciclo de vida. Las fases son las mismas que el RUP. Se cambia el contenido de cada disciplina del workflow de RUP, esto es, las actividades que son ejecutadas y los entregables resultantes.

En etapas tempranas se propone el modelado de tareas, escenarios y la utilización de un formalismo LEMD. Dicho formalismo en conjunto con técnicas y modelado UML tradicional se utiliza para generar prototipos validables por usuarios.

Prácticas clave del RUPi:

- Desarrollo iterativo

- Gestión de requerimientos
- Arquitectura basada en componentes
- Modelado visual de usuarios, tareas, contextos y dominio.
- Verificación continúa del software con foco en usabilidad, accesibilidad y aceptación.
- Control de cambios al software.

Cambios en las disciplinas:

1. Requerimientos: Modelado de usuarios, modelado de tareas, modelado del contexto (lugar de uso, escenario de uso del sistema), modelado de dominio, definición de guidelines basadas en los usuarios, definición de los requerimientos de usabilidad.
2. Análisis y diseño: Refinamiento del modelo de tareas, modelado conceptual de UI, modelado de clases.
3. Implementación: Utilización de guías de estilo.
4. Testing: Evaluación de la usabilidad, especificando el caso de pruebas basado en los valores establecidos para los requerimientos de usabilidad. Se plantean las pruebas con usuarios como método base.

Velmourougan2014 - U-SDLC

Aspectos generales: Se propone un enfoque sistemático para incorporar actividades y técnicas de usabilidad a lo largo de Software development lifecycle (SDLC). Usability Software SDLC es el nombre del proceso propuesto el cual es validado mediante una comparación con otros SDLC más tradicionales: Modelo en V, RUP, Scrum, Waterfall, Spiral, Prince2.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

Fig. 12. Tomado de [16]

Detalles del proceso: El proceso es iterativo y surge a partir de la comparación y experimentación con SDLC más tradicionales. De dicho análisis experimental se define el U-SDLC que ataca proactivamente las vulnerabilidades detectadas en los SDLC previos.

A las actividades básicas de Requerimientos, Diseño, Codificación, Pruebas y Mantenimiento, se le adicionan técnicas de usabilidad como ser: Entrevistas contextuales con usuarios, definición de requerimientos y métricas de usabilidad, definición de guías de usabilidad, prototipación, test de usabilidad con usuarios, entre otras.

3.2 Procesos Ágiles

Beyer2004 – Rapid contextual design

Aspectos generales: El proceso propuesto está basado en Contextual design que es un método propuesto y publicado en un libro [17] por los mismos autores. El método original (Contextual Design) no es considerado ágil, por lo que el proceso propuesto es una adaptación del método original a un equipo utilizando XP. El nuevo método lo publican en [18].

Declaración de principios: El proceso discute los principios y valores de la agilidad. Se incorporan los mismos como supuestos y axiomas para definir el proceso Rapid contextual design.

Representación gráfica: No hay acceso a la representación gráfica.

Detalles del proceso: El proceso es iterativo y ágil por adaptarse a la realidad de un equipo de desarrollo de XP. Como aspecto base, se propone un rol de representante de cliente y diseñadores de UI.

Grandes fases del método:

1. Set Project focus (se planifica y define a grandes rasgos la visión. Se definen los representantes de cliente)
2. Contextual inquiry with potential customers (se aplica observación, entrevistas con los representantes de cliente/usuarios claves)
3. Build an affinity (se presenta como las tareas son hechas en la actualidad. El proceso "as is" en la actualidad)
4. Introduce to larger team (se presenta el proceso "as is" al equipo)
5. Identify issues (se identifican los aspectos críticos a resolver y generan ideas para responder a estos)
6. Build user stories (se generan las US, guiadas por modelos de secuencia del proceso)
7. Run the planning game (para ejecutar el juego de planificación, se deben tener high level UI mockups, con esto se priorizan y estiman las US. Se define que se va a ejecutar en las próximas iteraciones)
8. Design detailed user interfaces (se diseña la UI detallada separando el diseño de la implementación del software, hay roles diferenciados para esto incluso)
9. Test UIs with User (en mockups en papel con los representante de clientes/usuarios clave)
10. Deliver to Development (se entregan las UIs detalladas a desarrollo y a Test para la aceptación de las US contra los diseños definidos)
11. Continue iterations in parallel (durante el desarrollo de la iteración 1 el equipo de UI desarrollo las UIs para la segunda y hacen otras actividades de apoyo como el paso 9)

Constantine2002 – Usage centered engineering

Aspectos generales: El proceso propuesto puede ser considerado un MDD (ya que el desarrollo se encuentra fuertemente influenciado por modelos) para web applications. Los autores presentan el proceso como una evolución de User-centered design (UCD).

Declaración de principios: El proceso discute e incorpora principios y valores de la agilidad. Se combinan prácticas de Extreme programming para priorizar y trabajar mediante iteraciones de tamaño fijo. El proceso discute e incluye Little Design Upfront para asegurar y validar la UI (el refactoreo y evolución se aplican en el código pero no a la UI, ya que se busca que la misma se valide al comienzo).

Representación gráfica:

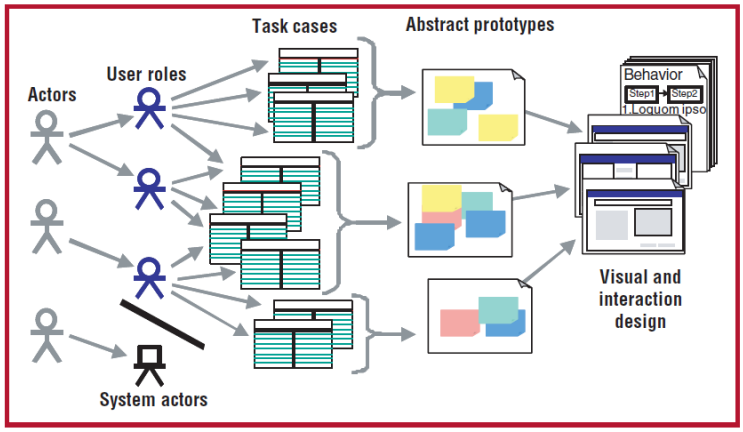


Fig. 13. Tomado de [19]

Detalles del proceso: El proceso propuesto es iterativo. Se definen 3 modelos básicos: roles, tareas y contenidos.

- Modelo de roles: describe usuarios e interacción con el sistema (tipo, frecuencia, propósito, volumen de información intercambiada)
- Modelo de tareas: describe los "casos de tareas" (Task cases) que son un tipo de caso de uso. Secuencia de acciones que un sistema hace que determina un resultado observable por un actor. Los Task cases también son llamados casos de uso esenciales (sin asumir aspectos tecnológicos o de interfaz). Se ordenan por frecuencia de uso e importancia.
- Modelo de contenido: prototipo abstracto que representa los contenidos de la UI en forma independiente de la apariencia y el comportamiento. Este modelo muestra como estos prototipos se agrupan en "contextos de interacción" reconocibles para el usuario. Funcionan como puente entre el modelo de tareas y prototipos de mayor fidelidad tipo sketches o mockups (más realistas)

Estos modelos se complementan con otros modelos como ser: modelo de dominio/glosario, modelo de reglas de negocio y modelo operacional (que muestra el ambiente o contexto de operación del sistema).

Se define un esquema visual y de interacción que representa una guía abstracta de estilos y elementos de interacción. Estos elementos se utilizan como base para el desarrollo y la evaluación de la aplicación. En la siguiente imagen se presenta el detalle del proceso.

Preliminary steps

1. *Essential purpose and preconception*—Clarify business and user purposes, then fantasize and set aside preconceptions of features, facilities, content, and capabilities.
2. *Exploratory modeling*—Identify questions, ambiguities, and areas of risk and uncertainty.

First iteration

3. *Role modeling*—Inventory and prioritize all user roles and select initially targeted subset.
4. *Task modeling*—Inventory and prioritize all tasks and select initially targeted subset.
5. *Task clustering*—Group all tasks by affinity and draft overall navigation architecture.
6. *Design*—Draft visual and interaction scheme; review and revise aesthetic design.
7. *Abstract prototyping*—Develop content model for interaction contexts supporting selected subset of tasks.
8. *Design*—Develop detailed user interface design for selected interaction contexts.
9. *Construction*—Program designed portions of user interface.

Successive iterations (similarly)

3a. Review role model and revise as needed, select next subset.
4a. Review task model and revise as needed, select next subset.
5a. Review navigation architecture and revise as needed.
6a. Review visual and interaction scheme and revise as needed.
7a. Develop content model for interaction contexts supporting next selected task subset.
8a. Design user interface for selected interaction contexts.
9a. Construct newly designed portions.

Fig. 14. Tomado de [19]

Holzinger2005 – Extreme Usability (XU)

Aspectos generales: En el proceso se analizan las similitudes entre los principios y prácticas de ambos métodos (alto involucramiento de clientes/usuarios en el desarrollo, trabajo en iteraciones, equipos pequeños multidisciplinarios, entre otros aspectos clave). El proceso fue evaluado empíricamente en un experimento académico con alumnos que poseen conocimientos de ambos métodos.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

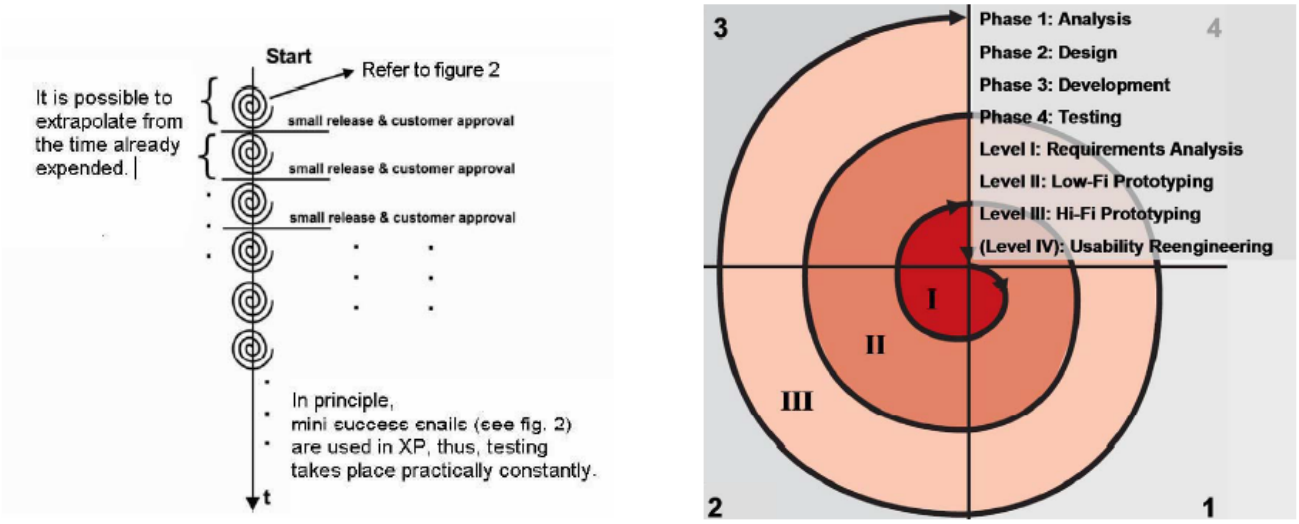


Fig. 15. Tomado de [20]

Detalles del proceso: El proceso es iterativo y ágil porque toma como base XP. Se propone en cada iteración de XP ejecutar las actividades de UE: análisis, diseño, desarrollo y evaluación, con los 3 pasos de modelado (paper mockup, prototipo funcional y producto final)

Humayoun2011 – Humayoun2011 Three Fold Integration

Aspectos generales: La propuesta de proceso se presenta como un framework que integra UCD en una metodología ágil. Se propone una integración en 3 capas (proceso, iteración y ambiente de desarrollo). Además se presentan herramientas automáticas (UEMan y TaMUlator) para la integración en la capa de ambiente de desarrollo.

Declaración de principios: El proceso discute e adopta los principios de UCD y Agile. Ambas filosofías si bien tienen orígenes distintos, no poseen contradicciones.

Representación gráfica:

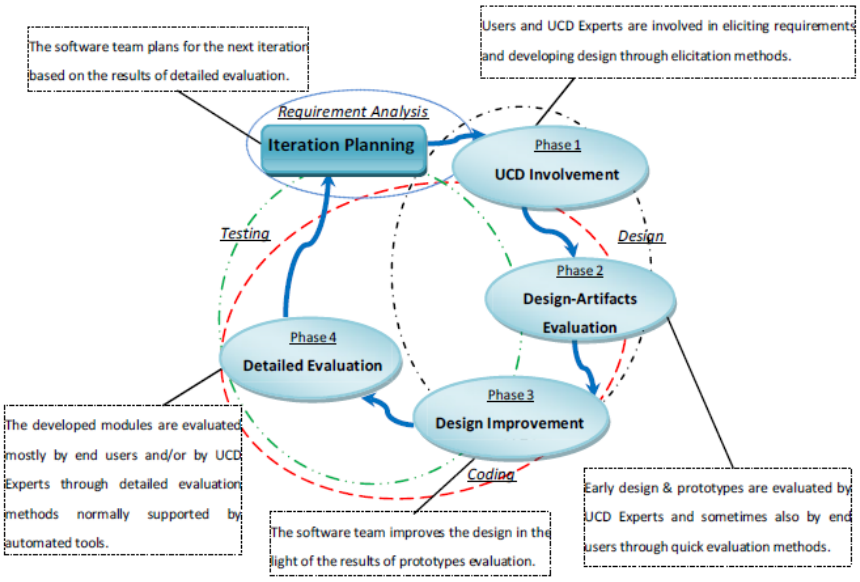


Fig. 16. Tomado de [21]

Detalles del proceso: El proceso es iterativo y ágil. Se propone la integración de prácticas de UCD y expertos en 3 capas diferentes.

Integración a nivel de proceso/ciclo de vida: Este nivel de integración implica seleccionar los métodos y técnicas correctas de UCD para aplicar en el proyecto en los momentos correctos. Se identifican técnicas y métodos para Elicitación y Evaluación. Dentro de los métodos de elicitación se propone integrar aquellos que requieran menos tiempo y esfuerzo y brinden buen nivel de feedback (focus groups, card sorting) durante los ciclos iniciales del método ágil. Para los métodos de evaluación, se recomienda utilizar distintos tipos de técnicas para abarcar mayor cantidad de issues de usabilidad. Métodos basados en expertos en UCD (evaluación heurística) es preferible hacerlos en etapas tempranas, sobre prototipos en papel incluso. Métodos que involucren usuarios conviene hacerlos sobre etapas más tardías.

Integración a nivel de iteración: Este nivel de integración permite que se integren las técnicas de UCD dentro de cada iteración del método ágil. Se añaden actividades de diseño y refinamiento del diseño en cada iteración, se toman medidas específicas de UCD y se definen e integran roles específicos también. Se incorpora el rol de Experto en UCD para apoyar al equipo.

Integración a nivel de ambiente de desarrollo: En este nivel de integración se pretende definir experimentos de evaluación de usabilidad y permitir ejecutar los mismos directamente en el ambiente de desarrollo IDE para tener un feedback continuo en el desarrollo. Se pueden definir experimentos (que en definitiva son instancias de evaluación) basado en 3 tipos. Expert based (heurísticas y cognitive walkthrough), user based y system based (basado en logs o en task analisis). Estos experimentos se evalúan contra un resultado esperado y pueden generar tareas que corrijan el sistema para

permitir alcanzar el nivel deseado. Debe existir trazabilidad entre el experimento y el cambio que se debe realizar. Este nivel de integración se logra con la utilización de las herramientas UEMan y TaMUlator.

Krohn2009 – Krohn2009

Aspectos generales: Se propone un proceso que integra UCD en un método ágil Feature Driven Development (FDD). El proceso es validado empíricamente en un proyecto de desarrollo de un sistema de gestión de contenidos web que soporta la actividad de docentes y alumnos para una universidad.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica: No hay acceso a la representación gráfica.

Detalles del proceso: El proceso es ágil e iterativo porque se basa en FDD. En las primeras iteraciones y fases tempranas se hace un estudio de los distintos tipos de usuarios y público objetivo del sistema web. En función de este relevamiento y análisis surgen casos de uso prototipables que describen la interacción entre un usuario y el sistema para alcanzar un cierto objetivo.

Técnicas propuestas en Requerimientos:

- Entrevistas semi-estructuradas.
- Benchmarking
- Organización de contenidos y arquitectura de información
- Workshops

A partir de estas técnicas se define un backlog priorizado de features con sus dependencias que permite desarrollar mediante un esquema iterativo e incremental usando FDD. Las features más prioritarias son implementadas en forma ágil en un prototipo funcional que es evaluado por los usuarios objetivo/clave que brinda feedback y cambios al respecto. Las features menos prioritarias se van incorporando paulatinamente en el proceso.

Técnicas propuestas en Diseño:

- Sketch en papel
- Screen mockups

Krusche2014 – Rugby + Tornado

Aspectos generales: El proceso propuesto presenta la integración de 2 enfoques (Rugby y Tornado) para permitir la mejora continua de la usabilidad y experiencia de usuario en aplicaciones móviles. Tornado es un enfoque liviano que se basa en la definición de escenarios y generación de prototipos ejecutables validables por los usuarios. Por otro lado Rugby es una adaptación de SCRUM para permitir al desarrollador liberar prototipos en cualquier momento del sprint con el objetivo de obtener feedback de usuarios con mayor frecuencia. La integración de estos 2 enfoques se da por la importancia que cobra el feedback de los usuarios en el contexto de aplicaciones móviles.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

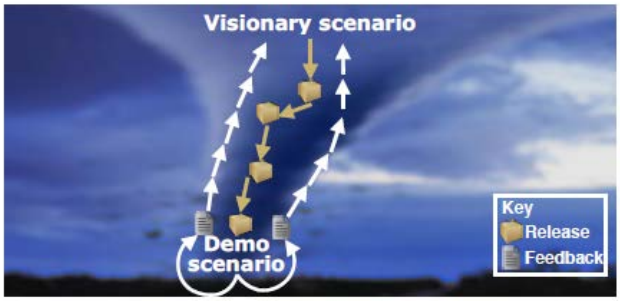


Fig. 17. Tornado. Tomado de [22]

Detalles del proceso: Los 2 enfoques que forman parte de la base del proceso propuesto son iterativos.Tornado es un enfoque que tiene su origen en procesos de innovación. El proceso se inicia con escenarios o visiones de alto nivel que luego se detallan en escenarios de demo más concretos utilizando feedback y entregas de prototipos tempranas. Durante este proceso, se comienza por prototipos de baja fidelidad incrementando hacia mayores niveles de detalle en el prototipado.

Rugby es una adaptación de SCRUM que incorpora algunas prácticas de UP. En este proceso se definen 2 flujos de trabajo para el equipo. Se presenta a continuación el flujo de trabajo de release y de feedback.

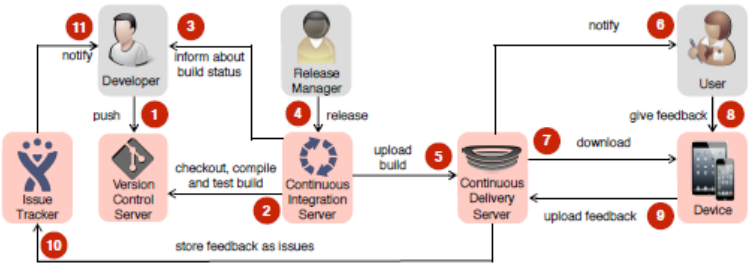


Fig. 18. Rugby feedback workflow. Tomado de [22]

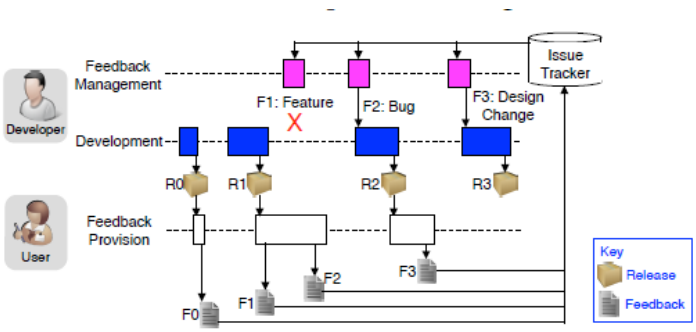


Fig. 19. Rugby release workflow. Tomado de [22]

Losada2012 – InterMod + UE

Aspectos generales: La propuesta de proceso busca integrar la ingeniería de la usabilidad (UE) en el método Intermod. Intermod es un proceso ágil basado en una herramienta Model Driven. El enfoque presentado fue evaluado empíricamente en el contexto del desarrollo de aplicaciones móviles

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

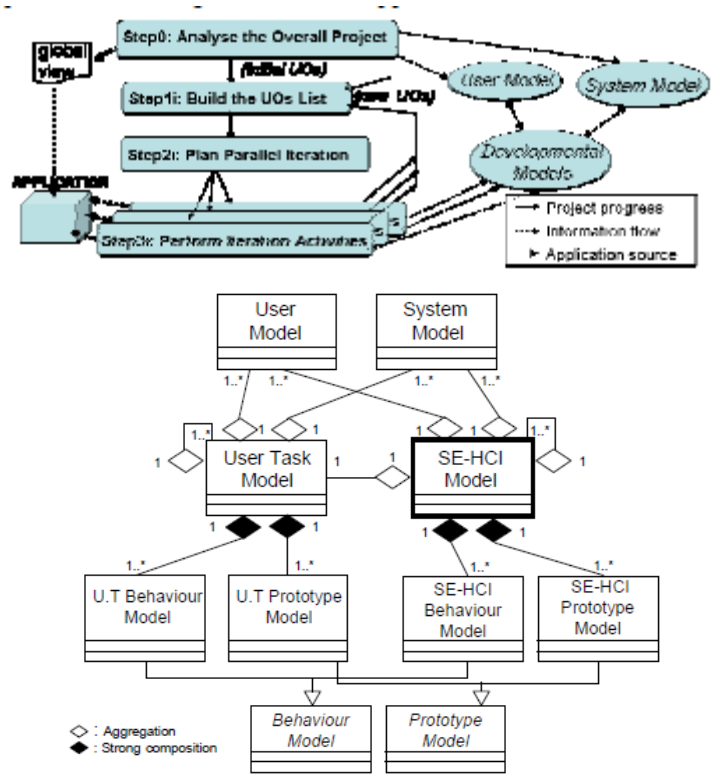


Fig. 20. Tomado de [23]

Detalles del proceso: Intermod es un proceso iterativo Model Driven (MDD) que implementa los principios y valores de la agilidad. En el proceso, se definen iteraciones y dentro de cada iteración se trabaja en distintos objetivos de usuario (user objectives). Estos objetivos de usuario pueden ser cumplidos con uno o más requerimientos funcionales y no funcionales.

Existen 3 grandes actividades de desarrollo: Análisis y diseño de navegación, construcción de interfaz, codificación de capa de negocio. Por otro lado existen 3 tipos de

actividades de integración: Integración de modelo de requerimientos, integración de la interfaz, integración del código y refactoring. Todas las actividades son model driven.

Se define un modelo de requerimientos el cual incluye un modelo de HCI (Semantically enriched HCI). Mediante transformaciones de este modelo abstracto se genera el modelo de presentación que luego se integra con otros modelos para construir la UI y lógica funcional final.

El modelo Semantically enriched HCI incluye:

- User model task: que describe a los distintos tipos de usuarios y la performance deseada de los usuarios para cada user objective.
- SE-HCI model: que es una abstracción del modelo anterior y también describe la interacción, los posibles errores y las características de diseño básicas (colores, secciones, tipos de botones, entre otras)

Para integrar conceptos de UE al método Intermod, se propone agregar las siguientes actividades.

- Métodos de elicitación: entrevistas a usuarios, observación, focus groups, etc
- Inspecciones: evaluación heurística y cognitive walkthrough.
- Métodos basados en test: experimentos con usuarios usando think aloud, medición de performance.

Perez2014 – AGILEUX

Aspectos generales: La propuesta presenta un modelo de madurez para incorporar agilidad y técnicas y métodos de usabilidad en pequeñas empresas alineado con los modelos de buenas prácticas CMM y MPS.BR.

Declaración de principios: La propuesta adopta los valores y principios de la agilidad integrando los mismos con principios de UCD.

Representación gráfica: No hay acceso a la representación gráfica.

Detalles del proceso: El proceso propuesto es iterativo. Para detallar las actividades y productos resultantes del proceso, se define para las áreas de proceso de nivel 2 de CMM, los artefactos y técnicas de usabilidad que deben aplicarse.

Área de proceso	Artefactos y técnicas
RM	Suggested Artifacts: personas, sketches, prototypes of low, medium and high fidelity; mockups; story cards, conceptual models, wireframes, wikis and storyboards. Suggested techniques: interviewing users; observations and contextual inquiries; focus groups; brainstorming; informal cognitive walkthroughs; card sorting, peer review design
PP&PM	Suggested Techniques: Workshops, Focus Groups, Brainstorming.
PPQA	Suggested artifacts: personas, sketches, prototypes of low, medium and high fidelity, mockups, user story cards, wireframes, storyboards. Suggested techniques: interviewing users; focus groups; brainstorming; informal cognitive walkthroughs; simplified

	think-aloud; heuristic evaluation; peer review design; card sorting; user testing (recommended:RITE - Rapid Iterative Testing and Evaluation).
M&A	Suggested artifacts: specifications, database and tools to collect and analyze base and derived measures

Table 1. Tomado de [24]

3.3 Procesos con origen en UCD, UE, HCI y otros

Balland2013 – Balland2013

Aspectos generales: La propuesta proceso extiende un lenguaje de diseño de UI en el marco un proceso de desarrollo basado guiado por modelos que se implementa en la herramienta DiaSuite propuesto en [25]. El proceso original es un MDD que sirve para desarrollar aplicaciones que interactúan con sensores en dominios como telecomunicaciones, aviación y domótica. El compilador DiaSuite genera un framework Java basado en artefactos de diseño que el programador luego extiende y complementa para brindar las funcionalidades deseadas. El proceso se valida con un caso de estudio de una aplicación para personas de tercera edad.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

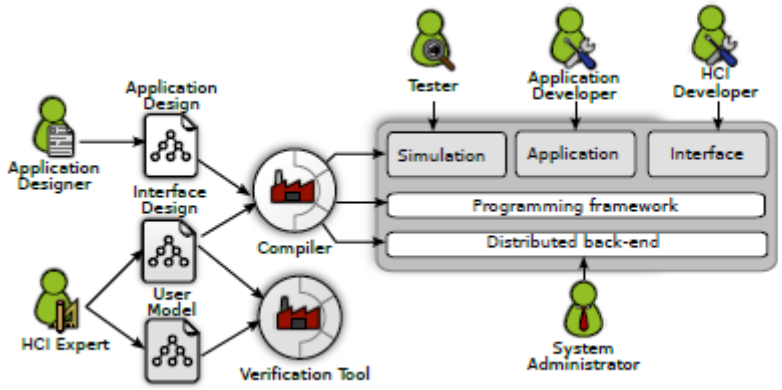


Fig. 21. Tomado de [26]

Detalles del proceso: El proceso propuesto es iterativo. Al proceso guiado por modelos original, se le definen los siguientes modelos.

- Un modelo de los usuarios: especifica preferencias de los usuarios, capacidades y habilidades.
- Un modelo de la UI: define aspectos de presentación de UI y de interacción.

- Un modelo de aplicación: entrelaza aspectos de interacción y los abstrae de la presentación.

Costa2014 – HUCDM (Hybrid User Centered Development Methodology)

Aspectos generales: El proceso propone un desarrollo iterativo-incremental, basado en los principios de UCD especificados en la ISO 9241-210 (Ergonomics of Human-System Interaction). La propuesta adopta prácticas ágiles y se especializa el dominio de sistemas educacionales.

Declaración de principios:

1. Conocimiento explícito de usuarios, tareas y ambientes/contextos
2. Establecimiento de un equipo multidisciplinario
3. Interacción entre los usuarios y el sistema
4. Involucramiento activo de los usuarios
5. Experiencia de usuario
6. Iteración sobre el diseño de soluciones

Representación gráfica:

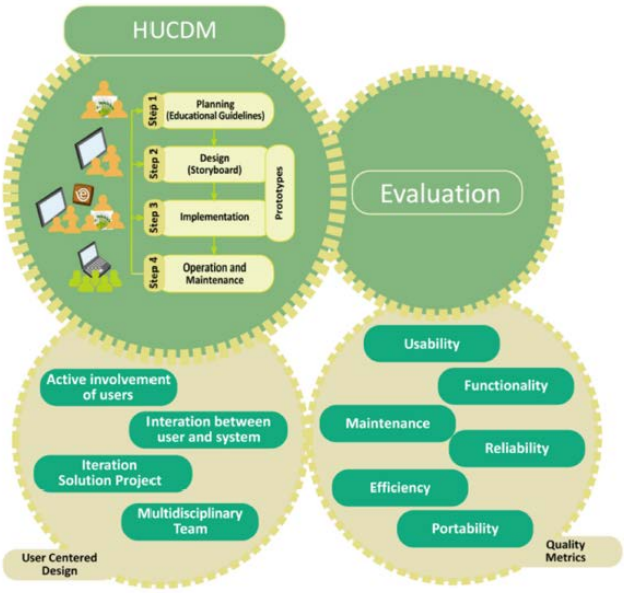


Fig. 22. Tomado de [27]

Detalles del proceso: El proceso propuesto es iterativo y se definen 4 fases: Planificación, Diseño, Implementación y Operación/Mantenimiento. La evaluación y el prototipado son prácticas que se aplican durante todas las fases.

- Planificación: Definición del público objetivo (nivel de educación, audiencia), contenido didáctico, temática educativa, aspectos de organización y navegación.
- Diseño (storyboarding): Armoniza, analiza e integra los conceptos definidos anteriormente con aspectos de interacción de software, particularmente navegación, arquitectura de la información y UI. Considera el uso de escenarios o modelos de tarea para representar interacciones claves.
- Implementación: Se dividen dos frentes de trabajo, uno educacional y uno técnico. Durante esta fase se evalúa y testea el software generado por parte de equipos multidisciplinarios. Se usan diferentes niveles (Papel, Pantallas clave (mockups), prototipos funcionales (código programado)) de prototipos colaborativos. En la evaluación participan Profesores y alumnos (usuarios clave del sistema)
- Operación/Mantenimiento: Corrección de defectos tanto por el frente educativo como por el técnico. Mantenimiento evolutivo del software.

Ferre2001 - The usability process

Aspectos generales: La propuesta de proceso presenta una generalización y abstracción de procesos previamente propuestos como ser Usability Engineering de Nielsen, Contextual Desing de Beyer y Usage Centered Design de Constantine y Loockwood. Se analizan los aspectos comunes a todos estos procesos.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

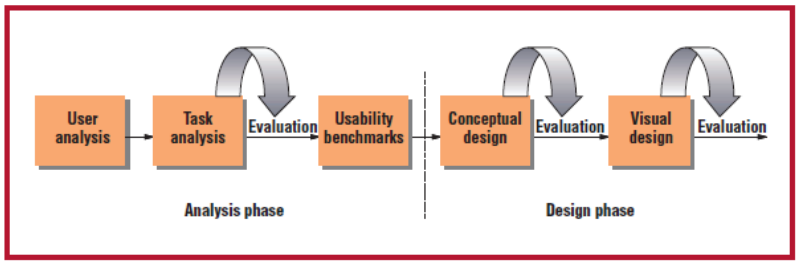


Fig. 23. Tomado de [28]

Detalles del proceso: Se propone un proceso iterativo con 3 grandes fases: Análisis, diseño y evaluación.

- Análisis:
  - Se hace un análisis de los usuarios (site visits y contextual inquiry, focus groups, encuestas y otras técnicas)
  - Task analysis que se priorizan por importancia y frecuencia de uso. El análisis de tareas se puede hacer en paralelo con lo anterior. Se propone un modelado de tareas como en Usage centered design.

- Usability benchmark: Se definen los valores actuales y objetivos además de métodos de medición para los sub atributos de usabilidad ejemplo(eficiencia, tasa de errores o satisfacción)

- Diseño:
  - Se hace el diseño conceptual de la interacción basado en las tareas relevadas. Se utilizan sketches en papel o mockups de baja fidelidad, luego se testean o evalúan contra el set de tareas definidos (primero internamente y luego en una recorrida con usuarios). Para hacer el diseño conceptual se recomienda apoyarse en los principios de diseño básicos.
  - Se hace el diseño visual o de alta fidelidad de la UI a partir del diseño conceptual. El resultado son prototipos o mockups con mayor fidelidad que son evaluados.
  - Durante esta fase se utilizan diversos tipos de prototipos para comunicar la especificación del sistema e involucrar al usuario en etapas de evaluación de usabilidad. Paper mockups, mago de oz y storyboards.
- Evaluación:
  - Test de usabilidad con usuarios. Realizado de ser posible a lo largo de todo el proceso, definiendo de antemano tareas a evaluar, grupos de usuarios objetivo y demás aspectos relevantes. Se propone el uso de think aloud durante la ejecución de los test de usuarios
  - Evaluación heurística: Un experto puede realizar evaluación heurística para acortar los tiempos de evaluación en las iteraciones. Sin embargo se definen como técnicas complementarias y es necesario realizar test con usuarios igualmente.
  - Otra técnica propuesta son evaluaciones colaborativas de usabilidad. Esta técnica, bajo un set de normas definidas, examina desde distintos puntos de vistas (usuarios, expertos de dominio, expertos en usabilidad, diseñadores y el equipo de desarrollo) la usabilidad de una aplicación.

Ferre2005 - Ferre2005

Aspectos generales: Se propone un framework para integrar prácticas y técnicas de HCI dentro de un proceso de desarrollo de software.

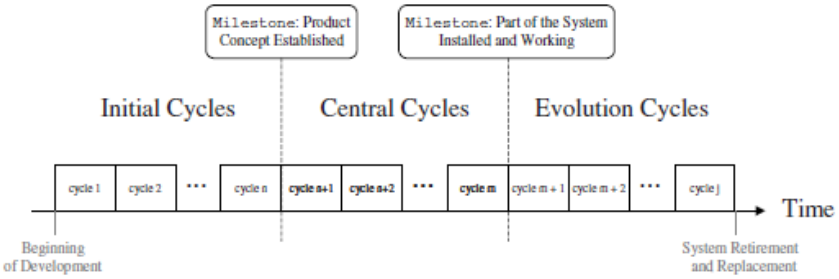
Declaración de principios:

Los principios o aspectos base que guían el framework propuesto son:

1. Se trate de un ciclo de vida iterativo
2. Involucramiento alto de usuarios
3. Conocimiento de los usuarios y tareas
4. Conocimiento multidisciplinario

Representación gráfica:

.



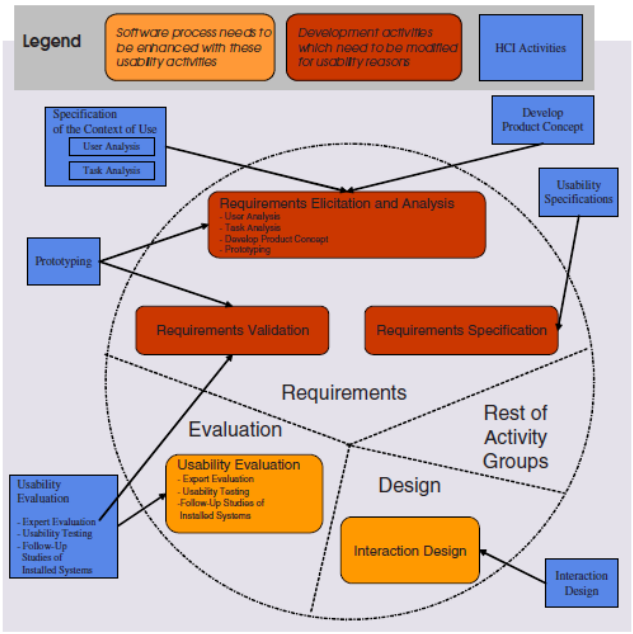


Fig. 24. Tomado de [29]

Detalles del proceso: No se tiene acceso al detalle del proceso.

Godet-Bar2010 – Symphony adaptado

Aspectos generales: La propuesta de proceso adapta el proceso surgido dentro del marco de software engineering, Symphony, integrando prácticas y modelos de HCI para diseñar interfaces tradicionales e interfaces de realidad aumentada.

Se agregan al Symphony:

- Actividades que involucran expertos en HCI y usabilidad.

- Mecanismos de colaboración sobre artefactos comunes.
- La especificación de un sub-proceso de HCI que se ejecuta en paralelo.
- Un modelo de análisis común con objetos y relaciones específicos.

Declaración de principios:
Principios esenciales del método propuesto:

1. Integración indolora de HCI en SE: la integración de expertos en HCI y prácticas específicas no debería revolucionar el proceso de desarrollo.
2. Trazabilidad y consistencia de modelos y comunicaciones: deben establecerse modelos comunes entre los expertos y estos estar trazados con los requerimientos.
3. Soporte de actividades con herramientas.

Representación gráfica:

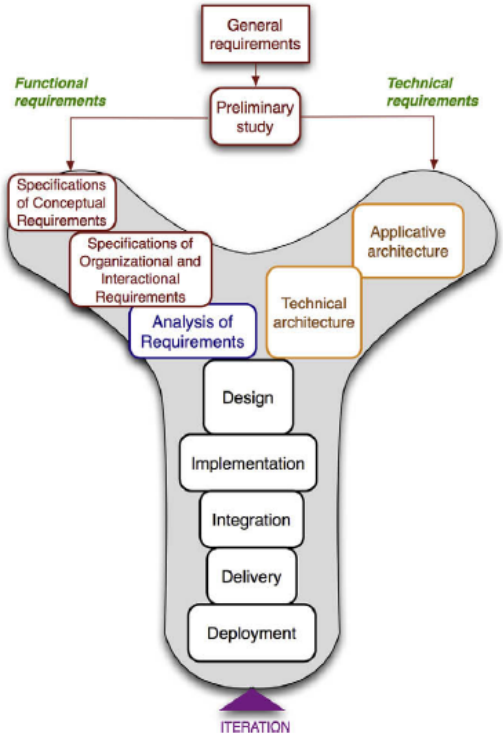


Fig. 25. Tomado de [30]

Detalles del proceso: El proceso es iterativo. En la rama izquierda, funcional de método, se mezclan e integran actividades de HCI.

Helms2006 – Wheel

Aspectos generales: Wheel es un framework general al cual se pueden incorporar actividades, técnicas y métodos (nuevos o existentes) para aplicar las mejores prácticas de usabilidad. Surge del análisis, adaptación y extensión de diversos métodos provenientes de la Usability Engineering y la Software Engineering (LUCID Framework, Star Life cycle, Watefall and Spiral).

Declaración de principios:

Wheel fue creado con el fin de satisfacer los siguientes principios/objetivos:

1. Flexibilidad del modelo permitiendo una progresión iterativa a través del ciclo de vida de la UE.
2. Posibilidad de incorporar métodos y técnicas existentes para varias actividades de desarrollo.
3. Posibilidad de adaptar el modelo de proceso para cumplir con ciertas necesidades, restricciones o contextos de proyecto.
4. Aplicabilidad del modelo en distintos tipos de aplicaciones y estilos de interacción.
5. Permitir la generación de documentación de los productos de trabajo como parte integral del proceso.
6. Integración con el proceso de desarrollo de software utilizado para desarrollar el core funcional del producto.

Representación gráfica:

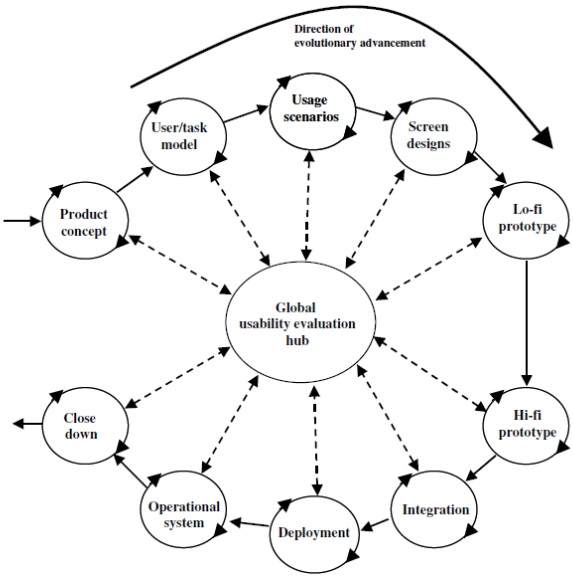


Fig. 26. Tomado de [31]

Detalles del proceso: El proceso es iterativo. En cada ciclo se ejecutan las actividades de análisis, diseño, implementación y evaluación**.** En la representación gráfica se aprecian las técnicas propuestas para distintas actividades como ser el modelado de

usuarios y tareas en los ciclos iniciales, el modelado de escenarios de uso, diseño de UI y distintos niveles de fidelidad en el prototipado.

Nielsen1992 – Usability Engineering (UE)

Aspectos generales: En 1992, Nielsen detecta la necesidad de tratar la usabilidad de forma sistemática dentro del proceso de desarrollo de software y por lo tanto propone la Usability engineering como un proceso práctico para lograr esto.

Si bien el proceso propone todo el modelo de la UE al detalle, Nielsen concluye luego de hacer un análisis de técnicas y factores clave, que los elementos básicos de un proceso orientado a la usabilidad: son testing con usuarios empírico, prototipación y diseño iterativo.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

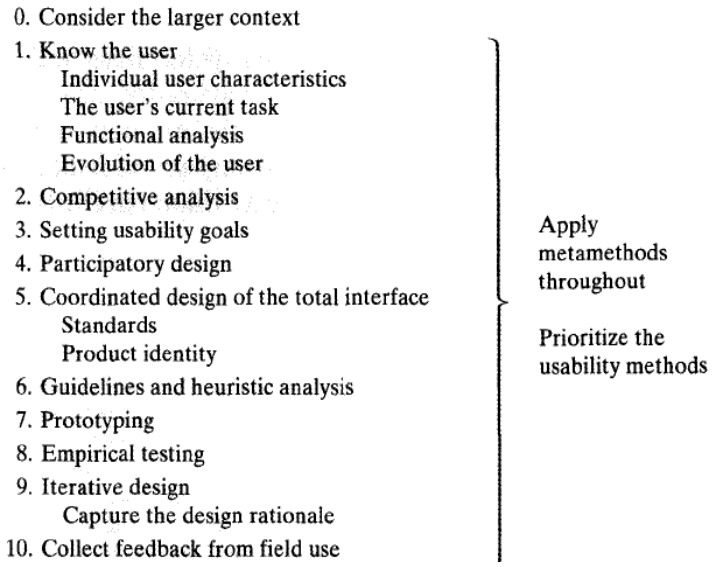


Fig. 27. Tomado de [6]

Detalles del proceso: El proceso de Usability engineering es iterativo y define 3 grandes fases que se describen a continuación.

- La etapa pre diseño:
  - Conocer al usuario: características individuales, tareas, análisis funcional, evolución del usuario.
  - Análisis competitivo.
  - Definición de objetivos de usabilidad: facilidad de aprendizaje, eficiencia, eficacia, satisfacción, prevención de errores.
- La etapa de diseño:

- Diseño participativo
- Diseño coordinado y consistente: seguimiento de estándares, identidad de producto
- Guías de diseño y análisis heurístico: guías generales, por categoría/control o por producto.
- Prototipación
- Testeo empírico: thinking aloud, constructive interaction (2 usuarios trabajan juntos para ejecutar una tarea)
- Diseño iterativo

• La etapa post diseño:
  - Obtener feedback del contexto de uso y el entorno

Rodriguez2015 – Generic GDP (GUI Desing Process)

Aspectos generales: La propuesta abstrae actividades y elementos comunes de varios GDPs para lograr una aproximación genérica al tema. Una vez definido el GDP genérico se integra el mismo en un ciclo de desarrollo conocido usando como base el modelo en espiral de Boehm.

Se valida empíricamente el proceso con un caso de estudio de una aplicación web de e-commerce.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

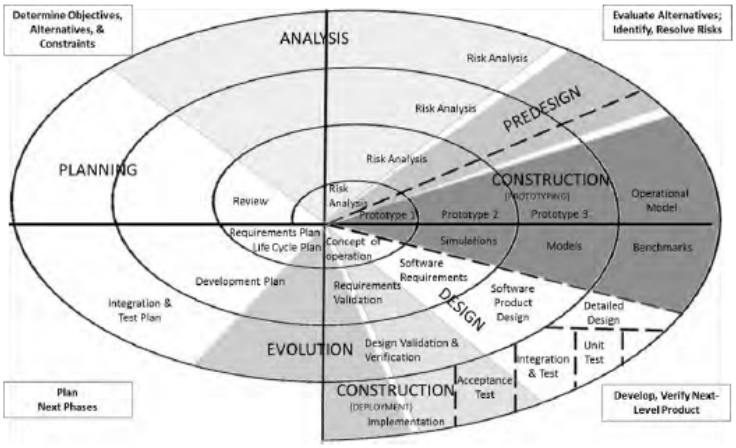


Fig. 28. Tomado de [32]

Detalles del proceso: El proceso propuesto es iterativo. El GDP genérico se presenta como una evolución de otros GDPs evaluados previamente. La propuesta de proceso GDP se define lo suficientemente genérico para adaptarse bien en los Software Development Lifecycles (SDLC) que se elijan. El detalle del GDP se describe en la siguiente imagen:

	Activity	*Activity's Description*
Planning Phase	Project Selection.	This activity refers to the project formulation. In this activity it is defined the objective of the project and if the project will be subcontracted or not.
	Marketing Activities.	This activity refers to the execution of market activities, (i.e. carrying out an analysis of the software systems that are competing against the system under construction).
	Organize the Work.	This activity regards project planning involving the definition of activities, roles (including interdisciplinary roles like definers, designers, implementers, application writers, and manual writers that constitute large groups by themselves). Other activities involved include usage scenarios, preliminary meetings with prospective users, relevant prior experience, intense concentration on the design problem, and periods of non involvement with the design problem.
Analysis Phase	Needs Analysis.	In this activity, it is determined the type and objectives of the system, and the users.
	User Analysis.	In this stage, it is determined in a specific way the kind of user that will use the system, defining details such as the user's computer skills.
	Task Analysis.	In the task analysis, they are established the tasks that the user will perform with the system to reach their business objectives.
	Functional Analysis.	In this activity, they are defined the computing functions that are required to perform the tasks.
	Requirements Specification.	In this activity, formal specifications are described for the system implementation. Specifications like: Data dictionary, Entity-Relationship Diagram, etc.
Pre – design Phase	Preliminary Specification of UI.	In this activity, it is sketched the GUI design.
Design Phase	Conceptual Design.	The aim here is to design the system concept. This activity is repeated in an evolutive and incremental way.
	Formal Design.	This activity aims to construct the design models of the system with formal tools. This activity is repeated in an evolutive and incremental way.
	Detailed Design.	This activity aims to detail the formal design. This activity is repeated in an evolutive and incremental way.
Construction Phase	Prototyping	The goal of this activity is building any kind of prototype type that is necessary for the software development. For example, a prototype can be a high- or low fidelity prototype, an evolutive or disposable prototype, a global or local prototype, etc. There should, at least, be a global evolutive prototype that in the first versions could be of low fidelity and even disposable. Following this stage, a working prototype will be produced based on an iterative and incremental process. The prototype increments can be local prototypes that are integrated to the working prototype at a certain point of the development process. This activity includes testing each working prototype version. An evolutive prototype is a prototype that evolves to the final system by adding increments. A disposable prototype is a prototype that is no longer used to build the final system. A high-fidelity prototype is similar to the final system whereas a low-fidelity prototype involves initial prototype sketches.
	Deployment.	This activity refers to either release the evolving prototype or release the final system.
Evolution Phase	Operation, Evaluation & Evolution	At the end of each prototype increment, the working prototype is evaluated and the prototype evolution is planned.

Fig. 29. Tomado de [32]

Seffah2009 – Adoption-centric usability engineering

Aspectos generales: Seffah tiene varios trabajos sobre los problemas de integración entre UE, HCI, UCD y Software engineering (SE) en general. El define que el problema de integración de métodos y técnicas específicas de la usabilidad en SE es

en general un problema de adopción. La propuesta es un meta-modelo de procesos que incluye un framework de medición (Adoption Centric Usability engineering – ACUE) que brinda apoyo a los equipos de desarrollo de software para incorporar métodos de usabilidad en sus procesos y medir el grado de adopción de los mismos.

Declaración de principios: No tiene declaración de principios o valores explicita. Se define que ACUE debe retroalimentarse del uso de los propios practioners. Con la suficiente cantidad de replicaciones y perfeccionamiento en los datos, la comunidad de practitiones se verá motivada a integrar las prácticas que apliquen a sus contextos de trabajo.

Representación gráfica:

Category	Popular methods
Inquiry	– Contextual inquiry – Ethnographic study/Field study – Observation – Interzviews – Focus groups – Survey – Questionnaires – Self-reporting logs – Screen snapshots
Inspection	– Heuristic evaluation – Cognitive walkthroughs – Formal usability inspections – Feature inspection – Consistency inspection – Standards inspection – Guidelines and checklists
Testing	– Thinking aloud protocol – Co-discovery method – Question-asking protocol – Performance measurement – Eye tracking
Prototyping	– Paper, Pictive and video prototyping – Storyboarding – Scenario sketching
Cognitive modeling	– Affinity diagrams – Archetypes and personae – Blind voting – Card sorting – Education evaluation
User requirements analysis	– Task analysis – Contextual inquiry – Focus groups and surveys – Personae and scenarios – Use case maps – Affinity diagramming – Brainstorming – Card sorting
Analytical and predictive methods	– Goals, Operators, Methods, and Selection (GOMS) performance analysis – Cognitive task analysis – Task-environment analysis – Knowledge analysis – Design analysis – Programmable user models – Simulations

Fig. 30. Taxonomía y métodos de UE. Tomado de [33]

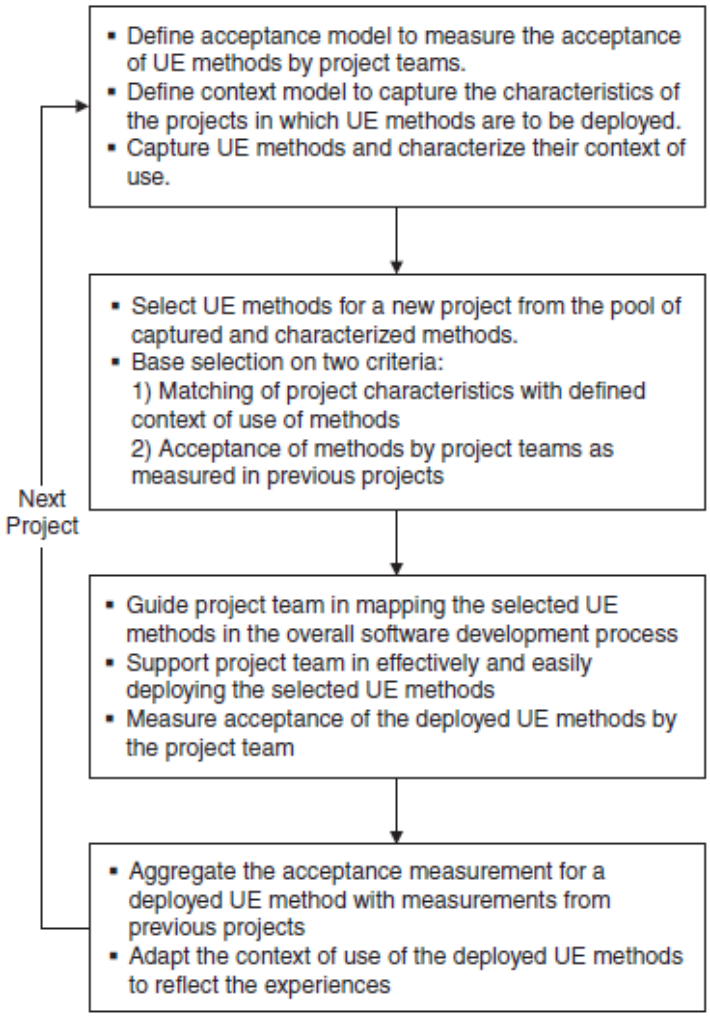


Fig. 31. Pasos del meta-modelo ACUE. Tomado de [33]

Detalles del proceso: El proceso propuesto es iterativo y busca mediante la introducción de algunos conceptos base, la adopción de los métodos de Usability Engineering dentro de un equipo de desarrollo de software.

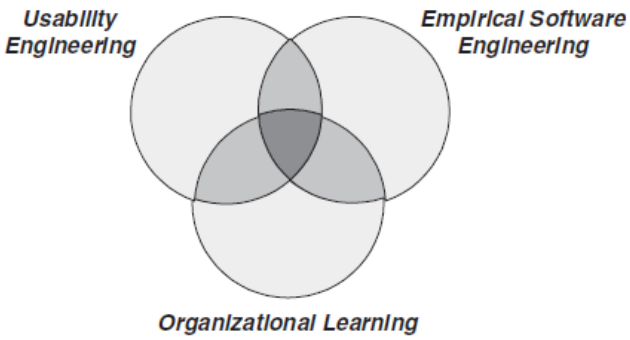
El modelo presenta un empaquetamiento de los métodos de usabilidad llamado USEPack que además define en que contextos conviene utilizar dicho método de acuerdo a evaluaciones de proyectos pasados.

Otro concepto que se presenta es el de Usability Engineering Kit que abstrae las actividades de un método de usabilidad de las técnicas o cómo deberían realizarse dichas actividades. Las prácticas específicas para realizar cada actividad en los distintos contextos se asocian mediante los USEPacks. Se elegirá el USEPack que haya tenido mayor aceptación proyectos anteriores de contextos similares. La aceptación de

los USEPacks en los distintos contextos de proyectos se mide usando teorías de aceptación de tecnología. Luego de algunas iteraciones de trabajo con el método se obtienen perfiles o descripciones de los distintos contextos y qué factores afectan la aceptación o el rechazo de un USEPack en un equipo. Los conceptos presentados por ACUE se implementaron en una herramienta web (ProUSE) que permite a los equipos de proyecto definir su contexto de acuerdo al proyecto y en función de esto seleccionar los USEPack que le apliquen. La herramienta también hace la evaluación de la aceptación o rechazo de los USEPack previamente seleccionados y explota la información para retroalimentar el modelo. 44 miembros de proyecto de 5 empresas desarrolladoras distintas participaron de la evaluación del método y la herramienta propuesta.

El método considera que la adopción de técnicas y métodos de usabilidad puede hacerse por 2 caminos: institucionalización del método basada en expertos o adopción basada en evidencia.

La solución potencial se ve en la intersección de UE, ESE y Aprendizaje organizacional.



Requirements Development
Identify user tasks
Identify user attributes
Identify organizational, technical and physical environment
Define system
Generate requirements
Set usability objectives

System Evaluation
Evaluate against requirements
Evaluate to improve design
Evaluate against required practice
Evaluate in use

Fig. 32. Tomado de [33]

Sivaji2013 – HUM (LBUT + EHE)

Aspectos generales: El proceso propuesto integra Laboratory based usability testing (LBUT) y Exploratory Heuristic evaluation (EHE) en una metodología hibrida (Hybrid usability methodology - HUM). Se evaluó la metodología híbrida propuesta en 4 casos de estudio. La propuesta se centra en las etapas de evaluación de usabilidad.

Declaración de principios: No tiene declaración de principios o valores explicita.

Representación gráfica:

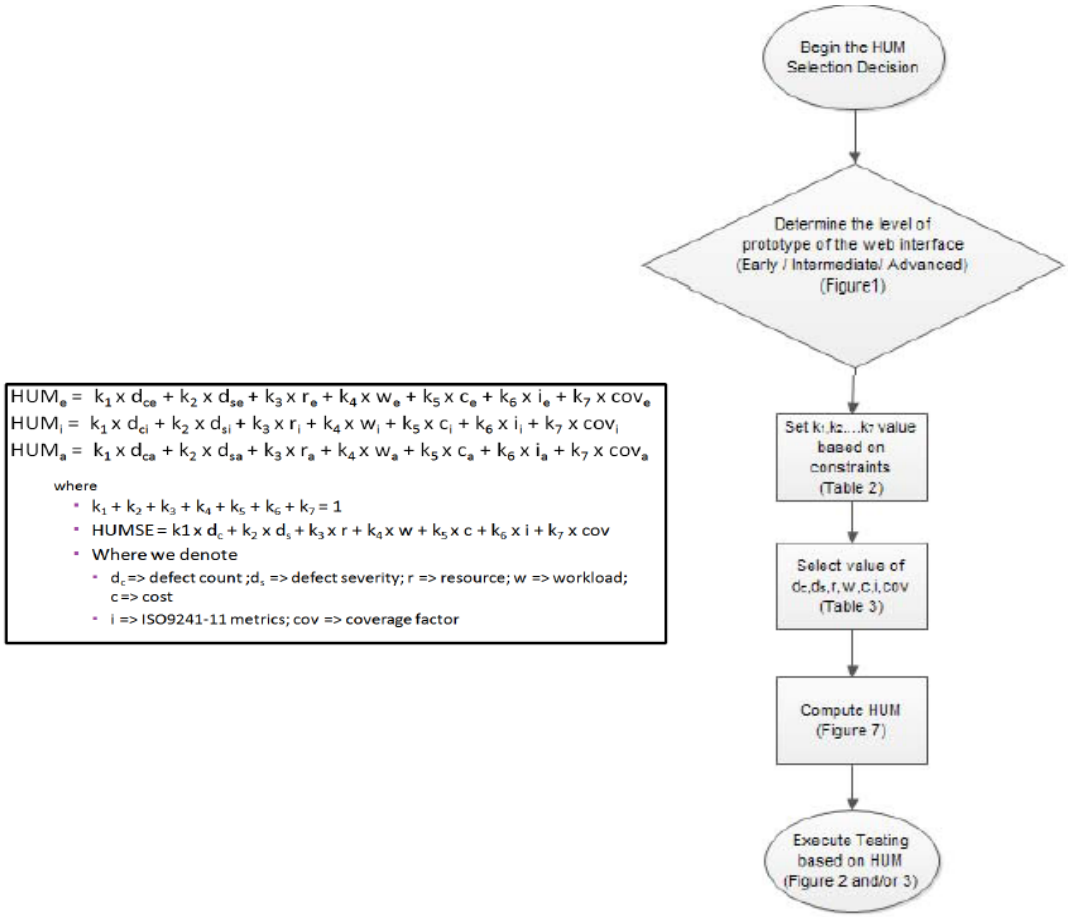


Fig. 33. Tomado de [34]

Detalles del proceso: El proceso propuesto es iterativo. El modelo hibrido fue desarrollada mediante la evaluación experimental de la performance relativa de cada método (LBUT y EHE) en distintas etapas del software development lifecycle de websites.

A raíz de la evaluación experimental, el proceso propuesto HUM, se puede ver como una función dependiente del k1 (cantidad de defectos), k2 (severidad de defectos), k3 (recursos), k4 (workload), k5 (costo), k6 (compliance con ISO 9241) y k7 (cobertura). Estas fueron las variables analizadas en el experimento. El HUM se ve como una función de estas variables que representan en cierta forma el contexto de un proyecto y como resultado de setear cada una de estas variables podemos obtener una definición de HUM distinta. Básicamente el HUM resultado (para ciclos tempranos, intermedios y tardíos) nos dirá el porcentaje de EHE y LBUT que debemos aplicar en nuestro proyecto. De esta forma los autores consideran al HUM un framework situacional o contextual.

Smith2004 – Smith2004

Aspectos generales: La propuesta es un modelo de procesos para el desarrollo de cross-cultural websites con alta usabilidad, compatibles con ISO 13407 (Human Centred Design Framework). El proceso incluye una variedad de técnicas centradas en el usuario y basadas en expertos que se incluyen en el análisis, diseño, construcción y evaluación de websites multiculturales.

Declaración de principios: Sigue los principios de UCD pero tiene en consideración que algunos de estos principios deben ser adaptados o flexibilizados para el contexto del desarrollo cross cultura o cross country. Por ejemplo la participación activa de usuarios tiene una interpretación diferente en culturas occidentales que orientales.

Representación gráfica:

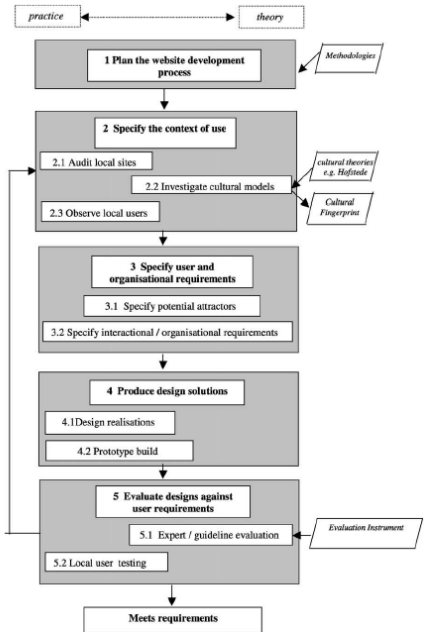


Fig. 34. Tomado de [35]

Detalles del proceso: El proceso es iterativo y considera varios aspectos estudiados por parte de los autores a la hora de desarrollar sitios multiculturales. El modelo es compatible con el StarLife cicle (con origen en UCD) propuesto por Hix y Hartson y también con el modelo en V.De igual manera se considera compatible con la ISO 13407. A continuación se explicita el detalle del proceso.

Table 6
Details of activities within the process model

Activity 1	Plan the website development process		Select development framework (e.g. RAD, DSDM, XP). Identify technical architecture options. Consider implications for user centred design/participation
Activity 2	Specify the context of use		
	Activity 2.1	Audit local sites	Remote/local audit, expert based. Identify semiotic attractors operating in domain.
	Activity 2.2	Investigate cultural models	Identify Hofstede scores and cultural fingerprints
	Activity 2.3	Observe local users	Virtual ethnography. Local ethnography
Activity 3	Specify user and organisational requirements		
	Activity 3.1	Specify potential attractors	Select appropriate attractors suitable to the particular site.
	Activity 3.2	Specify interactional requirements	Select suitable metaphors, navigational styles (strong menus vs hypertext) etc.
Activity 4	Produce design solutions		
	Activity 4.1	Specify potential design realizations for attractors	Build low-fidelity prototypes
	Activity 4.2	Prototype build	Build high-fidelity prototypes
Activity 5	Evaluate designs against requirements		
	Activity 5.1	Expert/guideline evaluation	From a developing base of guidelines
	Activity 5.2	Local user testing	Remote or local testing. Use of culturally specific task scenarios. Verbal protocols.

Fig. 35. Tomado de [35]

Taylor2011 – The website developmental for the healthcare consumer con heuristicas.

Aspectos generales: Se propone un proceso centrado en el usuario para el diseño de healthcare consumer web-sites que incorpora buenas prácticas de arquitectura de información, conocimientos prácticos, calidad de información y Human Computer Interaction.

Declaración de principios: No tiene declaración de principios o valores explicita. Se menciona adherencia a principios de UCD.

Representación gráfica:

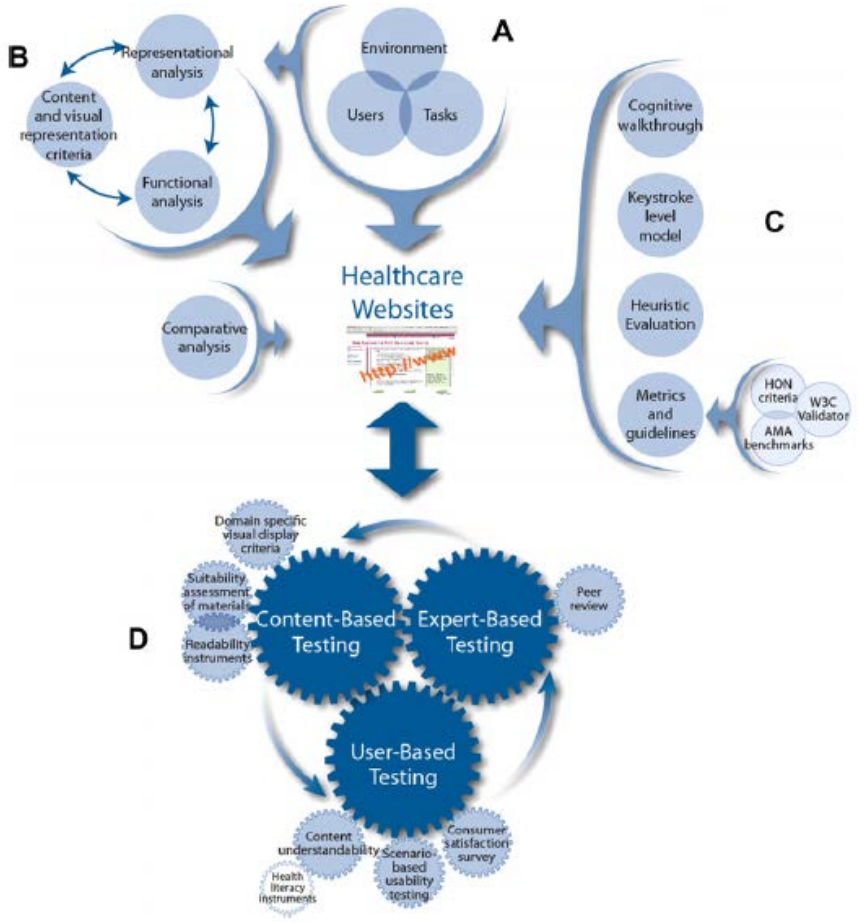


Fig. 36. Tomado de [36]

Detalles del proceso: El proceso es iterativo e incluye varios métodos y técnicas para diseño y evaluación de usabilidad:

- Diseño y rediseño de sitios centrados en el usuario para Salud.
- Modelado de: usuario, ambiente/contexto, tareas, análisis funcional.
- Representación gráfica y visual.
- Análisis comparativo
- Cognitive walkthrough
- Evaluación heurística
- Uso de estándares del dominio (ej Heath on Net Foundation provee estándares para salud)
- Estudios de usabilidad a escala reducida
- Testing del contenido
- Testing basado en expertos

Xiong2010 – Inter-Combined Model

Aspectos generales: El proceso propuesto (Inter-Combined Model) pretende cerrar la brecha (mejorar la comunicación y la transferencia de conocimiento) entre diseñadores y desarrolladores. El modelo presentado se valida con un estudio de caso de una aplicación mobile. Utilizan como base prácticas de prototipación y modelos livianos como agile.

Declaración de principios: No tiene declaración de principios o valores explicita. Se menciona adherencia a principios de UCD.

Representación gráfica:

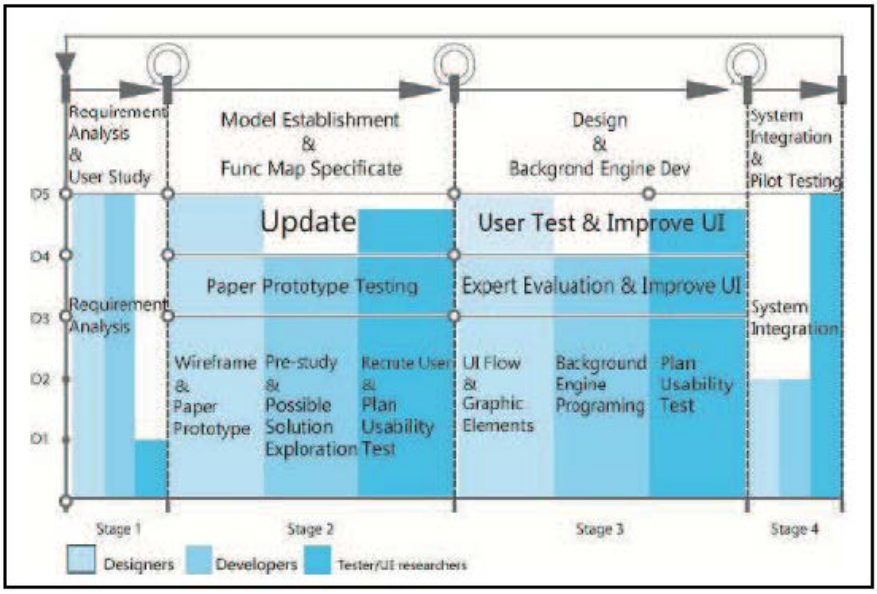


Fig. 37. Tomado de [37]

Detalles del proceso: El proceso propuesto es iterativo y se divide en 4 fases clave:

- Análisis de requerimientos y estudio de los usuarios: Utilizan persona, escenarios, entrevistas, workshops y cuestionarios. En esta fase el especialista en UCD tiene mayor participación.
- Establecimiento del modelo y mapeo de la funcionalidad: Diseñadores, developers y especialistas trabajan en la arquitectura de la aplicación. Los diseñadores prototipan en función del modelo establecido. Se usan diagramas de actividad para modelar y maperar las funcionalidades. Al final de esta fase se valida el prototipo con usuarios reclutados y especialistas.
- Diseño y desarrollo en backgroud: En estas fases las funcionalidades se ven en un prototipo de baja fidelidad (ej wireframe) inicialmente, que también es validado. Por parte de los desarrolladores se hace el diseño de la aplicación y documentación de las decisiones.

- Integración del sistema y codificación: En esta fase los desarrolladores codifican el software desde el “backend hacia adelante”. Se va haciendo test/evaluación de usabilidad a lo largo del ciclo en varias etapas.

4 Discusión

4.1 Procesos iterativos y ciclo de feedback

Del análisis de los 31 procesos de software orientados a la usabilidad se concluye que todos los procesos proponen ser instanciados mediante ciclos de vida iterativos. La interactividad en el desarrollo del software permite que los potenciales usuarios e interesados del software puedan evaluar y conocer el producto mientras el mismo se desarrolla. Otro aspecto analizado que resulta común a la mayoría de los procesos es la importancia del ciclo de feedback y evaluación del software. Tanto los procesos con origen conceptual en ingeniería de software como los procesos con origen en otras disciplinas, son centrados en la evaluación. El feedback de los usuarios finales cobra un rol fundamental a lo largo de todo el ciclo de vida. Los procesos de software orientados a la usabilidad son procesos altamente empíricos y se centran en los usuarios más que en los productos resultados.

4.2 Roles

Uno de los principios del diseño centrado en el usuario establece que es necesario formar equipos interdisciplinarios con expertos de diferentes disciplinas o roles que incorporen competencias y habilidades específicas. Este es un aspecto que resulta interesante analizar en los procesos propuestos. Más de un 35% de los procesos ([8], [10], [14], [38], [21], [26], [27], [30], [35], [36], [37]) proponen la incorporación de roles a los equipos. Estos roles proveen nuevas competencias y habilidades que permiten por ejemplo el diseño de la UI, la participación de los usuarios finales, el modelado de la interacción o la ejecución de evaluación de usabilidad basada en heurísticas y métodos que requieren experiencia.

4.3 Orígenes conceptuales de las propuestas de procesos

Del análisis del origen conceptual de las propuestas de procesos analizados es posible decir que existen más propuestas de procesos con origen en la ingeniería de software (tanto ágiles como predictivas) en comparación a las otras disciplinas (UE, UCD y HCI). Dentro de los procesos con origen en la ingeniería de software, se identifican procesos predictivos, principalmente basados en Unified Process (UP, RUP o OUP) y esprial de Boehm y procesos ágiles fuertemente basados en eXtreme programming y Scrum. Como se concluyó en el mapeo sistemático [1], existe un número similar de procesos predictivos y ágiles. Un aspecto interesante que surge del análisis detallado de los procesos es el uso de modelos de buenas prácticas basados en están-

dares como ser ISO 13407, Usability Maturity Model del ISO TR 18529, CMMi y MPS.BR.

4.4 Prácticas adoptadas en procesos de software orientados a la usabilidad

A continuación, se presentan algunas prácticas y técnicas fueron analizadas en todos los procesos. Es importante destacar que las prácticas analizadas pueden denominarse con diferentes nombres en distintos procesos o detallar particularidades, pero conceptualmente a efectos del análisis es tratada como la misma práctica.

Definición de objetivos y métricas de usabilidad

Este aspecto analizado implica la definición sistemática de requerimientos y objetivos de usabilidad como una actividad explícita en el proceso en la etapa de requerimientos. Es importante destacar que la definición de objetivos de usabilidad implica además la definición de métricas asociadas a los sub-atributos para permitir la verificación del objetivo. Un 39% de los procesos analizados ([7], [10], [12], [13], [15], [16], [28], [29], [6], [32], [33], [35]) proponen la incorporación de esta práctica de forma sistemática.

Caracterización de usuarios

Esta práctica analizada en los procesos es incluida en los procesos de requerimientos y puede ser implementada mediante la utilización de diversas técnicas como ser Persona Design, definición de perfiles o análisis de usuarios. La caracterización de los usuarios resulta ser una práctica común incorporada en el 77% de los procesos analizados ([7], [10], [11], [12], [13], [14], [15], [16], [17], [19], [21], [39], [23], [24], [27], [28], [29], [31], [6], [32], [33], [35], [36], [37]) .

Modelado de contextos, tareas o escenarios

Esta práctica analizada en los procesos es incluida en los procesos de requerimientos y puede ser implementada mediante casos de uso esenciales, modelado de tareas o task cases, análisis de contexto, entre otras técnicas. En el 55% de los procesos se detectó algún tipo de modelado de contextos, tareas o escenarios ([7], [10], [11], [12], [13], [14], [15], [19], [23], [24], [27], [28], [29], [31], [33], [36]).

Prototipado

Dentro de este aspecto se analizan los distintos niveles y técnicas de prototipación posibles. Desde prototipos de UI en papel de baja fidelidad a prototipos funcionales e interactivos que modelan la transición, navegabilidad e interacción de la interfaz pasando por prototipos desechables de fidelidad media. El prototipado resulta ser un aspecto altamente adoptado (87%) por los procesos de software orientados a la usabilidad que se analizaron.

Evaluación heurística

Este aspecto analizado incluye la evaluación de usabilidad basada en las heurísticas propuestas por Nielsen pero también otro tipo de evaluación basada en expertos y experiencia. Es importante destacar que sólo el 42% de los procesos analizados prescriben el uso de este tipo de evaluación ([11], [12], [13], [21], [23], [24], [28], [29], [6], [33], [34], [35], [36]).

Pruebas con usuarios

Se detectó la utilización de algún tipo de pruebas basadas en usuarios como un aspecto común a la mayoría (97%) de los procesos analizados. Este aspecto puede ser instanciado mediante prácticas y técnicas específicas como el think aloud protocol, experimentos en laboratorio con usuarios, walkthrough y otro tipo pruebas.

5 Conclusiones

Del análisis de los procesos es posible concluir que existen características comunes a los procesos analizados que son independientes a la familia u origen conceptual del proceso. Estas características comunes son, la instanciación del proceso en ciclos de vida iterativos, la caracterización y análisis de los usuarios objetivo, el foco en el feedback y en el involucramiento de los usuarios mediante distintos niveles de prototipación y la presencia de algún nivel o técnica de pruebas con usuarios. Estos aspectos del proceso son comunes tanto para procesos con origen en la ingeniería del software (predictivos o ágiles) como para procesos con origen en el diseño centrado en el usuario, la ingeniería de la usabilidad o la interacción hombre computadora.

En este contexto, es posible concluir que los demás aspectos analizados de los procesos pueden ser opcionales o adaptables al contexto del proceso o empresa. Estos otros aspectos incluyen, la inclusión de roles específicos a los equipos, la definición de métricas y objetivos de usabilidad, el modelado de tareas o escenarios o contextos y la evaluación basada en heurística o en expertos. Es importante destacar que estas prácticas o aspectos de proceso, resultan complementarios y no contradictorios con los aspectos comunes detectados en todos los procesos.

Por último, del origen conceptual de los procesos, la mayor presencia de procesos que provienen de la ingeniería de software está ligada al foco histórico hacia la sistematización y desarrollo de procesos que la ingeniería de software como disciplina tiene. Es posible decir que las disciplinas complementarias (UE, UCD y HCI), poseen más foco en un nivel de principios o políticas y en el nivel de técnicas y prácticas específicas, pero en muchos casos, los autores no llegan a definir procesos con la rigurosidad y detalle que se hace en la ingeniería de software.

6 Referencias

[1] J. I. Rocca and M. Solari, “Procesos de Software Orientados a la Usabilidad Un Mapeo Sistemático de la Literatura,” in *ESELAW*, 2016.

[2] P. H. Feiler and W. S. Humphrey, “Software process development and enactment: concepts and definitions,” *Softw. Process 1993 Contin. Softw. Process Improv. Second Int. Conf.*, vol. Berlin, Ge, pp. 28–40, 1993.

[3] “‘Software Engineering - Product quality. Part 1: Quality model,’” *Software Engineering - Product Quality - Part 1: Quality Model*. .

[4] J. Nielsen, “The usability engineering life cycle,” *Computer (Long. Beach. Calif).*, vol. 25, no. 3, 1992.

[5] A. Seffah and E. Metzker, “The obstacles and myths of usability and software engineering,” *Commun. ACM*, vol. 47, no. 12, pp. 71–76, Dec. 2004.

[6] J. Nielsen, “The Usability Engineering Life Cycle,” *Ieee*, vol. 25, no. 3, pp. 12–22, 1992.

[7] M. Ben Ayed, H. Ltifi, C. Kolski, and A. M. Alimi, “A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain,” *Decis. Support Syst.*, vol. 50, no. 1, pp. 64–78, 2010.

[8] M. Belhocine, N. Zenati-Henda, S. Benbelkacem, A. Bellarbi, and M. Tadjine, “Integrating human–computer interaction and business practices for mixed reality systems design: a case study,” *IET Softw.*, vol. 8, no. 2, pp. 86–101, 2014.

[9] H. Fischer, K. Nebe, and F. Klompmaker, “A Holistic Model for Integrating Usability Engineering and Software Engineering Enriched with Marketing Activities,” pp. 28–37, 2011.

[10] B. Göransson, M. Lif, and J. Gulliksen, “Usability Design – Extending Rational Unified Process with a New Discipline,” pp. 316–330, 2003.

[11] G. . Konstantinidis, G. C. . Anastassopoulos, A. S. . Karakos, E. . Anagnostou, and V. . Danielides, “A user-centered, object-oriented methodology for developing health information systems: A clinical information system (CIS) example,” *J. Med. Syst.*, vol. 36, no. 2, pp. 437–450, 2012.

[12] M. MAGUIRE, “Methods to support human-centred design,” *Int. J. Hum. Comput. Stud.*, vol. 55, no. 4, pp. 587–634, 2001.

[13] A. Nosseir, D. Flood, R. Harrison, and O. Ibrahim, “Mobile development process spiral,” *Proc. - ICCES 2012 2012 Int. Conf. Comput. Eng. Syst.*, pp. 281–286, 2012.

[14] A. Rodríguez, “Extending OpenUP to Conform with the ISO Usability Maturity Model - 21 gs excel,” *5th IFIP WG 13.2 Int. Conf. HCSE 2014 Paderborn, Ger. Sept. 16–18, 2014 Proc.*, vol. 8742, pp. 90–107, 2014.

[15] R. Kazman, L. Bass, and J. Bosch, “Bridging the Gaps Between Software Engineering and Human-Computer Interaction,” *25th IEEE Int. Conf. Softw. Eng. ICSE 2003*, 2003.

[16] D. P. Velmourougan, “Software development Life cycle model to build

software applications with Usability," pp. 1–6, 2014.
[17] H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*. Elsevier Science, 1997.
[18] K. Holtzblatt, J. B. Wendell, and S. Wood, *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. Elsevier Science, 2004.
[19] L. L. Constantine and L. A. D. Lockwood, "Usage-centered engineering for Web applications," *IEEE Softw.*, vol. 19, no. 2, pp. 42–50, 2002.
[20] A. Holzinger, M. Errath, G. Searle, B. Thurnher, and W. Slany, "From Extreme Programming and Usability Engineering to Extreme Usability in Software Engineering Education (XP + UE o XU)," pp. 3–6, 2005.
[21] S. R. . Humayoun, Y. . Dubinsky, and T. . Catarci, "A three-fold integration framework to incorporate user-centered design into agile software development," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6776 LNCS, pp. 55–64, 2011.
[22] K. S. and B. B., "User feedback in mobile development," *2nd Int. Work. Mob. Dev. Lifecycle, MobileDeLi 2014*, pp. 25–26, 2014.
[23] B. Losada, M. Urretavizcaya, J.-M. López-Gil, and I. Fernández-Castro, "Combining InterMod agile methodology with usability engineering in a mobile application development," in *Proceedings of the 13th International Conference on Interacción Persona-Ordenador - INTERACCION '12*, 2012, pp. 1–8.
[24] A. L. Peres, T. S. Da Silva, F. S. Silva, F. F. Soares, C. R. M. De Carvalho, and S. R. D. L. Meira, "AGILEUX Model: Towards a Reference Model on Integrating UX in Developing Software Using Agile Methodologies," in *2014 Agile Conference*, 2014, pp. 61–63.
[25] D. Cassou, J. Bruneau, C. Consel, and E. Balland, "Toward a tool-based development methodology for pervasive computing applications," *IEEE Trans. Softw. Eng.*, vol. 38, no. 6, pp. 1445–1463, 2012.
[26] E. Balland, C. Consel, B. N'Kaoua, and H. Sauzeon, "A case for human-driven software development," in *2013 35th International Conference on Software Engineering (ICSE)*, 2013, pp. 1229–1232.
[27] L. Costa P, Reis P, "Hybrid User Centered Development Methodology: An Application to Educational Software Development," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8699, pp. 243–253, 2014.
[28] X. Ferré, N. Juristo, H. Windl, and L. Constantine, "usability engineering Usability Basics for," *IEEE Softw.*, vol. 18, no. 1, pp. 22–29, 2001.
[29] X. Ferre, N. Juristo, and A. M. Moreno, "Framework for Integrating Usability Practices into the Software Process," *Proc. 6th Int. Conf. Prod. Focus. Softw. Process Improv.*, vol. 3574, pp. 202–215, 2005.
[30] G. Godet-bar, D. Rieu, and S. Dupuy-chessa, "HCI and business practices in a collaborative method for augmented reality systems," *Inf. Softw. Technol.*, vol. 52, no. 5, pp. 492–505, 2010.
[31] J. W. Helms, J. D. Arthur, D. Hix, and H. R. Hartson, "A field study of the

Wheel—a usability engineering process model," *J. Syst. Softw.*, vol. 79, no. 6, pp. 841–858, Jun. 2006.

[32] L. Rodriguez-Martinez, H. Duran-Limon, R. Mendoza-González, and J. Muñoz, "Identifying common activities in the graphical user interface development process and their integration into the software-system development life cycle," *Comput. Sci. Inf. Syst.*, vol. 12, no. 1, pp. 323–348, 2015.

[33] A. Seffah and E. Metzker, *Adoption-Centric Usability Engineering*. 2009.

[34] A. Sivaji, M. R. Abdullah, A. G. Downe, and W. F. W. Ahmad, "Hybrid Usability Methodology: Integrating Heuristic Evaluation with Laboratory Testing across the Software Development Lifecycle," in *2013 10th International Conference on Information Technology: New Generations*, 2013, pp. 375–383.

[35] A. Smith, L. Dunckley, T. French, S. Minocha, and Y. Chang, "A process model for developing usable cross-cultural websites," *Interact. Comput.*, vol. 16, no. 1, pp. 63–91, Feb. 2004.

[36] H. A. Taylor, D. Sullivan, C. Mullen, and C. M. Johnson, "Implementation of a user-centered framework in the development of a web-based health information database and call center," *J. Biomed. Inform.*, vol. 44, no. 5, pp. 897–908, 2011.

[37] Y. Xiong and A. Wang, "A new combined method for UCD and software development and case study," in *The 2nd International Conference on Information Science and Engineering*, 2010, pp. 1–4.

[38] H. Beyer, K. Holtzblatt, and L. Baker, "An Agile Customer-Centered Method :," no. Cd, pp. 50–59, 2004.

[39] T. . Krohn, M. C. . Kindsmüller, and M. . Herczeg, "User-centered design meets feature-driven development: An integrating approach for developing the web application myPIM," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5619 LNCS, pp. 739–748, 2009.

José Ignacio Rocca - jrocca@uni.ort.edu.uy

,Martin Solari - martin.solari@ort.edu.uy