

Hacia un Enfoque Ligero para el desarrollo de Aplicaciones Web de Calidad

Una aproximación guiada por calidad para un proceso de desarrollo ligero de aplicaciones web

Guillermo Covella³, Ivana Miaton², Cristina Coulleri¹, Luis Olsina³, Luisa Mich¹

¹e-Tourism Group-DISA, Università degli Studi di Trento. {mich, coulleri}@cs.unitn.it

²III-LIDI-Facultad de Informática, Universidad Nacional de La Plata. imiaton@lidi.info.unlp.edu.ar

³GIDIS-Facultad de Ingeniería, Universidad Nacional de La Pampa. {covellag, olsinal}@ing.unlpam.edu.ar

Abstract

A medida que el mercado de aplicaciones Web va madurando y con él metodologías y herramientas de desarrollo, las exigencias de calidad sobre productos y procesos son más importantes y frecuentes. También crecen las restricciones de tiempo y presupuesto. Este fenómeno tiende a dejar fuera del mercado a las pequeñas empresas que desarrollan software Web con procesos caóticos y no emplean un modelo de calidad mínimamente definido. Seguramente la brecha que separa un proceso caótico de uno consistente –iterativo, adaptable al contexto- es vista como insalvable, dado que esas organizaciones no disponen del tiempo y los recursos necesarios para afrontarla.

Nosotros proponemos una aproximación liviana pero con un enfoque de ingeniería Web a este problema: flexible, progresivo e intuitivo, basado particularmente en modelos empleados para la evaluación de calidad de aplicaciones Web existentes.

Para ilustrar nuestra propuesta hacemos referencia a una experiencia que está desarrollándose en un proyecto Italo-Argentino¹ y que involucra investigadores y colaboradores con competencias diversas, en un ámbito multidisciplinario. Las primeras conclusiones que extrajimos están relacionadas con la posibilidad de evaluar, rápida y eficazmente, los beneficios de emplear un modelo de calidad como marco de referencia para el desarrollo Web, específicamente para una pequeña organización.

Palabras clave: ingeniería Web, modelo de calidad de producto, proceso ligeros de desarrollo.

¹ “Comunità virtuali nello sviluppo del turismo sostenibile in Argentina”. Soportado y financiado por el Gruppo e-Tourism, DISA, Università degli Studi di Trento, Italia. Con aportes de la Associazione Trentini nel Mondo y colaboración del grupo GIDIS, Facultad de Ingeniería, Departamento Informática, UNLPam, Argentina.

1. Introducción

El diseño de una aplicación Web tiene características particulares ya que requiere un balance entre contenidos, aspectos estéticos y rendimiento [18], y es por ello imprescindible una visión amplia del concepto de calidad de producto y un enfoque interdisciplinario y flexible del proceso de desarrollo [20]. Se habla de calidad de producto por simplicidad, dado que en una aplicación Web producto y servicios están embebidos en la tecnología y en algunos casos el producto *es* la empresa, por ejemplo aquellos comercios o bancos que tiene sólo existencia en la Web.

El crecimiento en la demanda de aplicaciones Web ha creado una nueva oportunidad de negocios, en particular para organizaciones pequeñas que podrían usufructuar ventajas competitivas (alta capacitación, acceso a recursos tecnológicos) y no competitivas (tipo de cambio) [17]. Una limitación importante para ello, en este contexto, es una tendencia a emplear métodos de aseguramiento de calidad y procesos de desarrollo ad-hoc o “caóticos”, sin considerar como mínimo:

- una especificación de calidad de producto de la aplicación Web
- un proceso más o menos definido para el desarrollo y mantenimiento de la aplicación Web
- un proyecto que permita estimaciones de tiempo y presupuesto

Por otro lado los nuevos modelos de calidad de producto como ISO 9126 [12], modelos de madurez y capacidad como CMMI [13], y los modelos de procesos de desarrollo que se pueden adoptar y adaptar para desarrollo de aplicaciones Web, como RUP [14] o Agile Development [1], no son percibidos como factibles de adoptar rápidamente y a un costo razonable por las organizaciones pequeñas. En el caso de los modelos más formales, porque las tareas de gestión del proyecto quitan tiempo y recursos a la producción de código, y en el caso de Agile Development porque una organización pequeña no dispone siempre de recursos humanos altamente calificados ni de usuarios con disponibilidad ilimitada de tiempo para involucrarse en el desarrollo, tal como se requiere y se compara en [24].

En ese sentido, en el trabajo “How Internet Companies Negotiate Quality” [7] se da por hecho que muchas compañías pequeñas o dedicadas a nichos, que producen exitosamente software para la Web, reducen sus prácticas de aseguramiento de la calidad por un lado, y por otro lado la calidad del producto es motivo de negociación permanente entre clientes y proveedores, en pos de acortar los ciclos de vida. En cambio no está totalmente claro qué aspectos de un proceso formal subsisten. Sin embargo muestra tres características de esta visión particular de calidad y proceso:

Times drives: indica que el tiempo es la guía principal. Para ser competitivo, útil o innovador un producto debe ingresar al mercado antes que su competencia. Esto puede causar que otros requisitos tales como performance, costo y calidad pierdan prioridad. Tres atributos de esta característica son: “First to market”, orientado a releases y especificaciones mínimas.

Quality depends: la noción de calidad es menos prioritaria y tiende a encontrar su definición en cada producto.

Process adjusts: al menos en las compañías encuestadas en por los autores de [7], los procesos se ajustan de acuerdo a necesidades y circunstancias, pero más bien argumentan que si cuentan con recursos de calidad (principalmente humanos) el proceso bien definido no es importante.

Por otro lado, y en relación a la evaluación de calidad de aplicaciones Web, son incipientes los trabajos sobre “construcción de la calidad” como parte de las tareas del proceso de desarrollo [4] pero hay trabajos específicos, con resultados concretos, que facilitan la evaluación de sitios y aplicaciones web operativos considerando distintos enfoques: cuantitativos [5] o bien de propósitos

generales [3]. No obstante algunas de las metodologías, modelos y técnicas propuestos en esos trabajos se pueden utilizar, empírica e intuitivamente, desde el momento inicial del desarrollo de una aplicación Web, como forma de planificar la calidad del producto.

Sin desconocer la importancia que tienen los recursos humanos altamente calificados en el éxito de un desarrollo de tipo ligero, como Extreme Programming [26], pensamos que contar con un marco de referencia y algunos aspectos definidos del proceso (adaptados de procesos modernos y específicos para la Web) facilita la transición de un proceso ad-hoc a uno más formal y consistente [2] -que puede adaptarse a distintos contextos, orientado hacia una práctica reflectiva más que a seguir un idéntico proceso-, evaluando rápidamente los beneficios potenciales y sin descuidar la calidad del producto.

En este trabajo informamos de una experiencia realizada en el marco de un proyecto Italo-Argentino siguiendo esas líneas guía. Particularmente las condiciones de los proyectos realizados son equiparables a las que caracterizan a las organizaciones pequeñas en proyectos Web de pequeña o mediana escala. Específicamente en casos donde el personal tiene alta rotación, una formación y capacidad diversas; el ciclo de vida debe ser corto y los requisitos de calidad pueden jugar un rol importante para el éxito del proyecto.

En el plano metodológico el objetivo principal de este proyecto era investigar la factibilidad de una aproximación ligera para el desarrollo de aplicaciones Web guiada por requisitos de calidad, desde etapas tempranas del desarrollo

El trabajo está organizado de la siguiente manera: en el capítulo 2 presentamos la motivación de esta propuesta donde se explicita además un marco de referencia o framework. En el capítulo 3 comentamos una experiencia de implementación concreta sobre tres proyectos, ahora en su etapa final. En el capítulo 4 se encuentran las Conclusiones. Finalmente, en el capítulo 5 se mencionan los posibles Trabajos Futuros.

2. Motivación y propuesta metodológica

La motivación del trabajo está en proponer una vía intermedia, entre un desarrollo basado en modelos –de calidad y de proceso- ad hoc y otro formalizado, que además permita a una pequeña organización un desarrollo de aplicaciones Web con calidad, pero sin detener su proceso de producción y permitiéndole a corto plazo una evaluación cualitativa y cuantitativa de los beneficios que potencialmente implica esta aproximación.

Desde el punto de vista metodológico la propuesta se concentra en diseñar la calidad de las aplicaciones desde el comienzo, empleando modelos, metodologías y técnicas que son empleadas preferentemente en la evaluación de calidad de sitios ya operativos .

Para articular efectivamente la propuesta es necesario un marco de trabajo flexible y efectivo que tenga en cuenta el carácter interdisciplinario del proceso de desarrollo para la Web [6] y algún mecanismo de supervisión de los riesgos que implica una aproximación ligera .

En ese sentido valoramos la importancia de un marco de referencia como guía en el proceso de construcción de la calidad que permite identificar las áreas claves donde concentrar el esfuerzo; adoptar y adaptar modelos, métodos, técnicas e instrumentos y también asignar los recursos disponibles.

En este trabajo consideramos el modelo expresado en la Tabla 1 (la V en latín está por U), llamado *7Loci* [3], como marco de referencia de calidad. Este modelo fue desarrollado teniendo en cuenta la

naturaleza comunicacional de un sitio Web y, usando como referencia los siete *loci* descritos en la obra *De inventione* de Tulio Marco Cicerón(106 aC), permite identificar las dimensiones fundamentales de una aplicación Web.

En otras palabras los 7*Loci* o dimensiones constituyen un framework o meta-modelo donde se pueden articular distintos modelos y heurísticas de calidad y es independiente del dominio de la aplicación.

También permite un diseño de la calidad del producto de acuerdo a las prioridades que se fijen para cada proyecto, dejando de lado aspectos que no se consideran relevantes en ese dominio o para ese entregable (release) del producto.

Tabla 1. Marco de Referencia de Calidad 7loci.

<i>Loci</i>	Dimensión	Aspectos considerados
¿QVIS? (¿Quién?)	Identidad	Imágenes y otros elementos que definen la identidad de la persona u organización propietaria del sitio.
¿QVID? (¿Qué?)	Contenido	Información que tiene el sitio.
¿CVR? (¿Por qué?)	Servicios	Servicios disponibles para los usuarios.
¿VBI? (¿Dónde?)	Ubicación	Visibilidad del sitio y espacios de comunicación entre usuarios y de éstos con la organización.
¿QVANDO? (¿Cuándo?)	Mantenimiento	Considera la definición de actividades (y el momento de su realización) que garantizan el funcionamiento y la operatividad de la aplicación Web.
¿QVOMODO? (¿Cómo?)	Usabilidad	Concierno a cómo son provistos contenidos y servicios a los usuarios.
¿QVIBUS AVXILIIS? (¿Con qué medios?)	Factibilidad	Administración del proyecto.

Algunos aspectos que el modelo especifica y típicamente no se tienen en cuenta cuando lo que importa es “time to market”, pero que pueden tener un impacto significativo sino se consideran en el momento oportuno, son:

- la imagen (marca, logo, diseño, personalización, etc.) de la organización
- valor de la información y links ofrecidos
- calidad del contenido
- las distintas necesidades de los distintos perfiles de usuarios
- Recursos necesarios para mantener la aplicación (contenidos, servicios y código).

Para especificar estos atributos de calidad se proponen algunas técnicas sencillas, como entrevistas o brainstormings focalizados con usuarios, plasmando rápidamente las propuestas sobre un prototipo no funcional de la interfaz de la aplicación, un medio económico y eficaz [16] para un desarrollo Web interdisciplinario.

El proceso para la especificación de calidad propuesto implica el siguiente conjunto mínimo de actividades:

- Adopción –si no existe- y/o adaptación de un proceso de desarrollo.
- Definición del perfil o perfiles de usuario del sitio o aplicación Web. En este caso un paso clave en la definición de un proceso de evaluación o especificación de atributos de calidad teniendo en cuenta los propuestos en los trabajos [5] y [9].
- Especificación de los requisitos más importantes de Usabilidad y Funcionalidad (*Servicios del Marco de Referencia 7loci*). Empleando para ello un Árbol de Requisitos [8] que agrupa jerárquicamente atributos, sub características y características deseables para la aplicación en acuerdo a las prescripciones de la norma ISO/IEC 9126-1 [12].

Para completar el Árbol de Requisitos se pueden utilizar inicialmente un conjunto de heurísticas bien conocidas [23] para Usabilidad y algunos estándares formalizados o de facto, propios de cada dominio, para Funcionalidad [8].

2.1 Características de un proceso de desarrollo adaptable

Un proceso de desarrollo definido permite conocer tareas y actividades necesarias para llevar adelante un proyecto, planificando las iteraciones, y consecuentemente los recursos necesarios al estimar un presupuesto. Desde ese punto de vista tiene que ver con la dimensión *Factibilidad* del metamodelo de calidad. Para adaptar o personalizar un proceso de desarrollo sin detener un proceso de producción ya iniciado es necesaria una *aproximación* flexible y progresiva, en este caso los modelos de proceso bien conocidos pueden constituir líneas guía para la personalización de un modelo adaptado a las necesidades y posibilidades de cada organización y/o proyecto.

Consideramos las siguientes características esenciales a tener en cuenta:

Consistente: que pueda ser utilizado en distintos contextos de desarrollo y para distintos dominios.

Guiado por riesgo: en base a una lista básica de riesgos técnicos, jerarquizada, que funciona como guía para el proceso ya que permite direccionar el esfuerzo, los recursos y prever mecanismos de mitigación que garanticen la continuidad del proyecto.

Iterativo: donde el equipo se pueda fijar metas para cada una de las iteraciones, que pueden ser evaluadas y dan una idea clara del progreso del desarrollo. El cumplimiento de esas metas, a su vez, da una idea de que se tiene control sobre los pasos del proceso de desarrollo.

Incremental: al final de cada iteración, dentro de la fase de construcción, se debe contar con un nuevo entregable (release) de la aplicación.

Mínimamente documentado: los documentos son considerados como artefactos producidos durante el proceso de desarrollo por lo tanto incluimos también algunos criterios de calidad mínimos para construirlos:

- Formato estándar –adaptado para nuestra experiencia de la implementación del RUP [14]-.
- Revisados periódicamente
- Control básico sobre los cambios -Software Configuration Management-.

2.1.1 Tareas del proceso

Considerando esta propuesta como una aproximación ligera y flexible, las tareas del proceso de desarrollo, respetando las características enunciadas, deben estar orientados principalmente a producir un nuevo release de la aplicación. Dependiendo del alcance, dominio y complejidad de la aplicación y del tipo de proyecto –propio, para entregar llave en mano, subcontrato, etc.- se pueden considerar un conjunto diferente de artefactos a producir, y para cada iteración podemos considerar como mínimo las siguientes actividades:

- Elicitar requisitos a implementar en el release actual –funcionales, no funcionales-
- Establecer los modelos básicos a especificar y artefactos necesarios a construir
- Desarrollo y prueba del código
- Distribución del release construido

De acuerdo a las necesidades de cubrir un mercado o de mejorar las condiciones frente a la competencia, las especificaciones de calidad pueden ser flexibles en los primeros releases en beneficio de poner rápidamente la aplicación a disposición de los usuarios, y los requisitos funcionales reducidos a un conjunto mínimo, para acortar el ciclo de vida. Pero una vez cumplido ese objetivo inicial será necesario ajustar la calidad de la aplicación a un mercado ya maduro y competitivo.

2.1.2 Artefactos del proceso

Los artefactos que se decidan construir son un conjunto mínimo de documentos y modelos generados y mantenidos junto al código fuente a lo largo del proceso de desarrollo de la aplicación. Tienen el objetivo principal de explicitar los conceptos encontrados o elaborados en la fase de elicitación de requisitos. También constituyen acuerdos y líneas guía sobre los que se construyen nuevos releases y versiones. Pueden servir para introducir el uso de un lenguaje de modelado estándar como UML, pero usando inicialmente un número reducido de conceptos, modelos y diagramas.

En la sección Experiencias de Implementación se detalla, como referencia, el conjunto de documentos desarrollados a propósito de la experiencia realizada en el proyecto Italo-Argentino, utilizando este marco de referencia.

3. Experiencias de implementación

El proyecto “Comunità virtuali nello sviluppo del turismo sostenibile in Argentina” se lleva a cabo en el marco de las actividades del grupo e-tourism, perteneciente al Dipartimento di Informatica e Studi Aziendale en la Universidad de Trento, Italia. Para participar del mismo fueron convocados en Argentina un grupo de tres profesionales con competencia en distintas áreas del desarrollo de aplicaciones Web, con la particularidad de que dos de ellos pertenecen a la colectividad Trentina en Argentina.

3.1 Aplicaciones Web para el grupo *e-tourism*, la comunidad *e-migrati* y la *Associazione Trentini nel Mondo*.

Uno de los objetivos del proyecto es el desarrollo de una aplicación Web con las características de una “comunidad virtual” o Comunidad on-line según la clasificación dada en [6], destinada a los miembros de la “comunidad real” de la región del Trentino (Italia) que emigraron, llamada *e-migrati* [versión beta en <http://www.economia.unitn.it/etourism/emigrati>]. Este proceso migratorio se dio en la primera mitad del siglo pasado, principalmente hacia Sudamérica y en particular a la República Argentina. Ahora, por distintos motivos, un número crecientes de descendientes de aquellos emigrantes tienen interés en retornar a Italia y constituyen un grupo potencial de visitantes de este sitio comunitario.

Otro objetivo es realizar una propuesta para el rediseño del sitio de la “Associazione Trentini nel Mondo” [<http://www.trentininelmondo.it>], vinculada directamente al proyecto de comunidad virtual y con filiales de segundo y tercer nivel, llamadas federaciones y círculos, en los cinco continentes,

con la intención de mejorar aspectos de presentación, funcionalidad, accesibilidad y mantenibilidad. El portal de la Asociación constituye un punto de ingreso al sitio de la comunidad virtual e-migrati.

Por otro lado los miembros del grupo *e-tourism* consideraban necesario rediseñar su propio sitio Web [<http://www.economia.unitn.it/etourism>], desarrollado en el año 2000 y que tres años después presentaba algunos problemas de mantenibilidad –tanto de código fuente como de contenidos-, Funcionalidad y Usabilidad.

El rediseño del sitio *e-tourism* presentó características prácticamente ideales para un desarrollo piloto que permitiera experimentar nuestra propuesta. Además permitió integrar el equipo de desarrollo, formado inicialmente por personal con competencias diversas y una formación profesional heterogénea.

La adaptación de un modelo de proceso se facilitó porque existía, al momento de iniciar el trabajo, un conjunto mínimo de requisitos ya establecidos, los usuarios principales son los propios miembros del grupo, disponibles con ciertas restricciones de tiempo para un trabajo interdisciplinario.

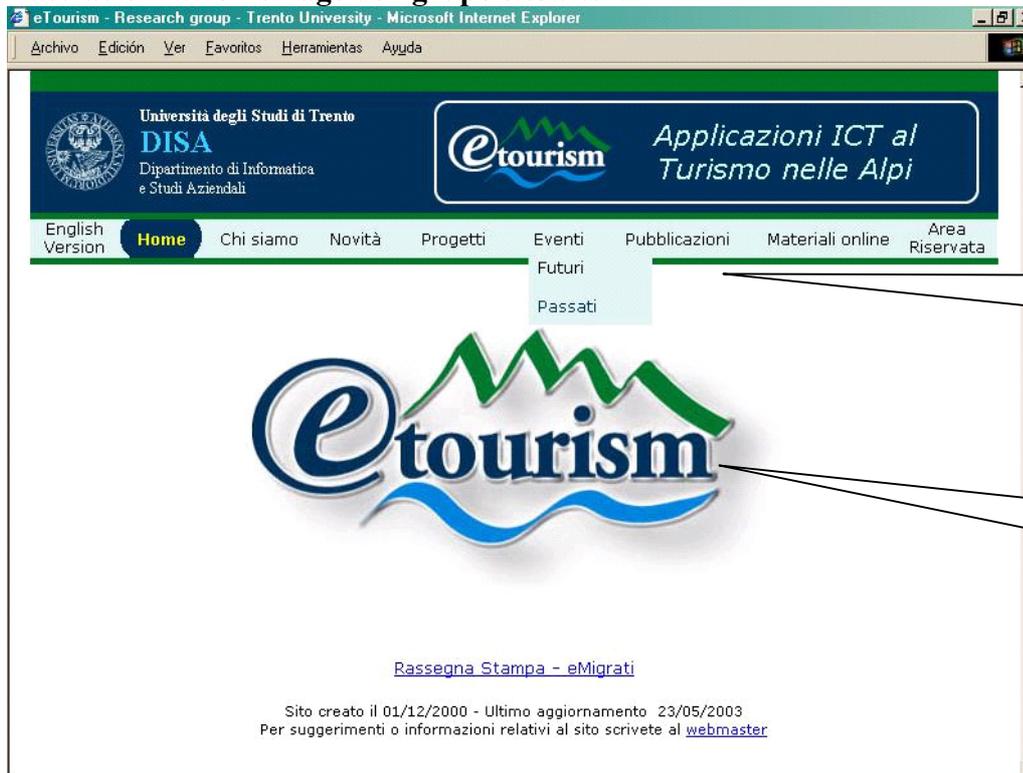
La necesidad de algunos documentos y modelos se justificó en el hecho de que la aplicación será mantenida por un grupo diferente al que la desarrolló. Así, documentos y modelos constituyen líneas guía para subsiguientes procesos de rediseño o modificación de la aplicación a partir de la versión entregada.

3.2 Usando el marco de referencia

Durante la tarea de elicitación de requisitos, como parte del proceso de rediseño del sitio *e-tourism*, se identificaron problemas a nivel de:

Mantenibilidad: el código fuente, en Perl, de todo el sitio estaba en un único archivo fuente, con mínima documentación “en pequeño”, los datos correspondientes a la parte dinámica estaban en parte en un archivo de texto plano y en parte embebidos en el código fuente. La sección *Area Riservata*, destinada a la actualización del sitio no permitía más de un usuario a la vez.

Ilustración 1.Home Page del grupo e-tourism



Menú horizontal.
En la versión anterior los ítems eran links

Acceso rápido a Chi Siamo.
Sin funcionalidad en la versión anterior

Funcionalidad: ausencia de un mecanismo de búsqueda de publicaciones por autor o por título.
Comprensibilidad y Accesibilidad: aspecto de los links no estandarizados, ausencia de texto alternativo, imágenes e iconos empleados con criterios ambiguos, etc.

Tomando como referencia el modelo 7 Loci estos atributos corresponden a las dimensiones: *Mantenimiento, Servicios y Usabilidad*. Por otro lado los aspectos del diseño relacionados a las dimensiones *Identidad y Contenido*, considerados relevantes por los usuarios y que fueron exhaustivamente evaluados para la versión original, permanecen sin cambios.

3.2.1 Artefactos

Los artefactos –documentos y modelos- que acordamos desarrollar, además del código necesario para implementar la aplicación, brevemente justificados, son los siguientes:

Visión de la aplicación: explica aspectos claves para la toma de decisiones. Constituye un punto de partida para entender la misión de la aplicación porque explica los “que” y “porque” del proyecto. Es frecuente encontrar que esta información suele ser de “transmisión oral” entre los usuarios clave y desarrolladores.

Glosario: especifica conceptos claves, un vocabulario común para usuarios y desarrolladores y el modo de interpretarlos en el contexto del dominio de la aplicación, contribuyendo a la comunicación entre los miembros del equipo.

Lista jerarquizada de riesgos: un instrumento simple para una aproximación a un proceso guiado por riesgo. Constituida por un detalle de los riesgos detectados, ordenada jerárquicamente de acuerdo al impacto que tendrán sobre el proyecto en caso de ocurrir y la probabilidad (baja, media, alta) de que efectivamente acontezcan. Los riesgos de mayor impacto y alta probabilidad de ocurrencia estarán al tope de la lista y concentrarán el esfuerzo para mitigar su impacto sobre el proyecto.

Prototipos no funcionales (“paper mock-up”): son sketches a mano alzada o impresiones de diapositivas. Así rescatamos el valor de los prototipos no funcionales como un medio eficiente para elicitación de requisitos de la interfaz de usuario, algunos requisitos funcionales y no funcionales [11] y también como una forma de introducir ingeniería de usabilidad desde una etapa temprana del proceso de desarrollo [10] evitando trasladar implícitamente esta tarea a los usuarios finales.

Especificación de requisitos funcionales: funciona como “acuerdo” entre desarrolladores, y usuarios principales. Es una lista de los requisitos elicitados con referencias a documentos de origen de los requisitos (apuntes de reuniones, síntesis de brainstormings, prototipos, etc.), estándares y heurísticas.

Gestión de configuración: para este proyecto consiste simplemente de una guía acerca de cómo resguardar y archivar las distintas versiones de los artefactos (documentos, código y ejecutables) y cómo proceder con el código fuente cuando se deben incorporar o promover componentes nuevos o modificados como parte estable de la aplicación.

Prueba: para este proyecto tuvimos en cuenta alguno de los principios de testing de Agile processes [1], donde el proceso de testing se realiza dentro de la misma iteración, totalmente ligado a la actividad de escribir código. Solo se explicitaron casos de testing sobre requisitos considerados críticos para la liberación de un nuevo release.

3.2.2 Modelos básicos

Están propuestos como parte de los artefactos que producen las actividades de análisis de requisitos y como una introducción al uso de un lenguaje de modelado. Para los diagramas empleamos una herramienta de uso libre y gratuito llamada Visual Paradigm for UML, Community Edition, compatible con herramientas comerciales (por ej.: IBM-Rational Rose).

Modelo de arquitectura: el valor de este modelo crece si se desarrolla en una etapa temprana del proceso. Una visión gráfica de las partes más importantes del sistema y la interacción necesaria entre ellas. Constituyen una herramienta de comunicación entre los miembros del equipo de desarrollo porque da una idea aproximada de los componentes, las interfaces y la tecnología que se necesitan desarrollar o con los que se debe interactuar.

Casos de uso: se emplean solo para describir y representar aquellos requisitos funcionales identificados, en el Documento de Especificación de Requisitos, como más complejos o sobre los que es necesario explicitar roles de los actores y comportamiento del sistema. Para la descripción personalizamos una plantilla presentada en [20].

Datos: un diagrama en UML representando las tablas y sus relaciones con un detalle básico de propiedades [15].

Componentes para páginas Web: un diagrama con el despliegue de los componentes y sus relaciones, poniendo énfasis en la Arquitectura de la información y en aspectos navegacionales, empleando para ello un subconjunto básico de estereotipos adaptados de [22].

4. Conclusiones

Consideramos que la experiencia tiene validez para una organización pequeña, que desarrolla aplicaciones Web para un mercado competitivo porque es una aproximación liviana, progresiva y flexible que no introduce ningún factor que detenga la producción de código, que generalmente es

la tarea que insume más tiempo en este tipo de organizaciones. También permite una evaluación temprana de la calidad del producto y el proceso.

Las experiencias realizadas en los proyectos mencionados en la sección 3 nos permiten afirmar que con este enfoque es factible articular rápidamente un nuevo equipo de desarrollo, basado en personal con distintas competencias y perfiles, multidisciplinario y con una perspectiva puesta en la calidad del producto desde el inicio del proceso de desarrollo. Además se pueden especificar nuevos proyectos, ajustados a partir de datos históricos relevantes. Por ejemplo para el desarrollo del sitio *e-migrati* y el rediseño del sitio de la asociación *Trentini nel Mondo* empleamos las líneas guía surgidas del caso piloto con pequeños ajustes. En ambos casos pudimos definir un proyecto con estimaciones razonables de tiempo y recursos, y realizar una asignación de tareas teniendo una visión completa del proceso y calidad del producto a desarrollar.

5. Trabajos futuros

En este trabajo iniciamos un proceso de articulación entre el meta-modelo 7Loci y alguna de las metodologías de evaluación de calidad como WebQEM, pero para diseñar la calidad de aplicaciones como tarea del proceso de construcción. En ese sentido pensamos avanzar para formalizar esa articulación y hacerla extensible a todo el ciclo de vida de una aplicación, integrando aspectos que pueden evaluarse cuantitativamente, en base a métricas bien conocidas por Ej. sobre atributos vinculados a características como Usabilidad y Funcionalidad, y otros eminentemente cualitativos como aspectos Estéticos y de Contenidos.

Otro aspecto que nos interesa evaluar es la calidad en uso de las aplicaciones desarrolladas. Es decir la calidad que perciben los usuarios al visitar los sitios e interactuar con las aplicaciones. Para ello consideraremos la guía que ofrecen la norma ISO/IEC 9126-1 y el draft ISO/IEC 9126-4, y diseñaremos un experimento que nos permita conseguir un indicador global de calidad en uso, articulando evaluaciones de efectividad, productividad y satisfacción.

Finalmente pensamos conducir un survey para evaluar la percepción inicial que tienen los desarrolladores a quienes se les transferirán las aplicaciones para mantener, respecto a la calidad interna –código fuente, modelos y documentos- del producto (en este caso cada una de las aplicaciones Web).

6. Referencias

- [1]. Kohn M., Ford D., *Introducing an Agile Process to an organization*. IEEE Computer, June 2003.
- [2]. Lycett M, Macride R., Ch. Patel, Paul R. *Migrating Agile Methods to Standardized Development Practices*. IEEE Computer, June 2003.
- [3]. Mich L., Franch M., Gaio L., *Evaluating and Designing the Quality of Web Sites*. IEEE Multimedia, Jan-Mar, 2003, pags. 34-43.
- [4]. Mich L., Cilione G., Franch M., *The 2QCV3Q Model to Analyse Requirements of a Web Site: The Case of the Association "No Pain for Children*. Anales del congreso "ICSE2002-Web Engineering Workshop", Orlando, Florida, 2002, p. 1-10.
- [5]. Olsina L., Rossi G., *Measuring Web Application Quality with WebQEM*. IEEE Multimedia, oct-dic 2002.
- [6]. Ginige A., Murugesan S., *WebEngineering: An Introduction*. IEEE Multimedia jan-mar 2001.
- [7]. Baskerville R., Levine L., Pries-Heje J., Slaughter S. *How Internet Software Companies Negotiate Quality*. IEEE Computer, May, 2001.pag. 51-57.

- [8]. Olsina L., Lafuente G., Rossi G., E-Commerce Site Evaluation: A Case Study, Lecture Notes in Computer Science 1875, Proc. 1st Int'l Conf. Electronic Commerce and Web Technologies (EC-Web 2000), Springer-Verlag, Heidelberg, 2000, pag. 239-252.
- [9]. Barnes S., Vidgen, R. An integrative approach to the assessment of e-commerce quality. *Journal of Electronic Commerce Research*. 3(3). 2002.
- [10]. Juristo N., Windl H., Constantine L., Introducing Usability, *IEEE Software* jan-feb, 2001.
- [11]. Hammar Cloyd M., *Designing User-Centered Web Applications in Web Time*. IEEE Software jan-feb, 2001.
- [12]. ISO/IEC 9126-1:2001, *Software Engineering— Product Quality—Part 1: Quality Model*, Int'l Org. For Standardization, Geneva, 2001.
- [13]. Carnegie Mellon University /Software Engineering Institute, *Capability Maturity Model® Integration Technical Report*, CMU/SEI-2002-TR-011, <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr011.pdf>, March 2002.
- [14]. Rational software , *Rational Unified Process*. <http://www.rational.com/products/rup/index.jsp> [última visita 09/06/03].
- [15]. Rational Software , *The UML and Data Modeling*, 2000 , <http://www.rational.com/products/whitepapers/101516.jsp> [última visita 09/06/03].
- [16]. Nielsen J., *Paper Prototyping: Getting User Data Before You Code*. Alertbox, April 14, 2003: <http://www.useit.com/> [última visita 3/06/2003].
- [17]. Rabinovich E., *Argentina Makes Its Software Play*. *Wired News*, <http://www.wired.com/news/infostructure/0,1377,57929,00.html> [última visita 3/06/2003].
- [18]. Pastor O., *Tutorial: OOWS: Una Aproximación para el Modelado Conceptual de Aplicaciones Web*, 2nd Ibero American Conference on Web Engineering. Santa Fe, Argentina, September 2002. <http://www.ing.unlpam.edu.ar/icwe2002/tutorial.html#Web-app> .
- [19]. Dustin E., Rashka J., McDiarmid D., *Quality Web Systems, Performance, Security and Usability*. ISBN: 0-201-71936-3. Addison Wesley, 2002.
- [20]. Vidgen R., Avinson D., Wood B., Wood-Harper T., *Developing Web Information Systems*. ISBN: 0-7506-5763-4. Butterworth Heinemann, 2002.
- [21]. Schneider G., Winters J.P., *Applying Use Cases, Second Edition, A Practical Guide*. ISBN: 0-201-70853-1. Addison Wesley, 2000. <http://books.txt.com>.
- [22]. Conallen, J. *Building Web Applications with UML*, ISBN: 0-201-61577-0, Addison Wesley 2000.
- [23]. Nielsen J., *Alertbox*. <http://www.useit.com>, [última visita 3/06/2003].
- [24]. Orr K., *CMM Versus Agile Development: Religious Wars and Software Development*. Executive Report Vol 3 N° 7, Cutter Consortium, 2003. <http://www.com/consortium/>, [última visita 3/06/2003].
- [25]. Mich L., Franch M., Novi Inverardi P., Marzani P., *Choosing the “Rightweight” Model for Web Site Quality Evaluation*. Third International Conference on Web Engineering, ICWE 2003, Springer, LNCS 2722, Page(s): 334-337, July 2003. ISBN: 3-540-40522-4.
- [26]. Maurer F., Martel S., *Extreme Programming, Rapid Development for Web-Based Applications*. *IEEE Internet Computing*, Jan-Feb, 2002, pag. 86-90.