

Combination of Agile Development and User Centered Design to Improve the Usability of a Beef-Cattle Farm Simulator

M. Arroqui, P. Mangudo, L. Pellizza, S. Murgolo, A. Ottonello, S. Ferragut, J. R. Alvarez, C.F. Machado and A. Teyseyre

Abstract— At the Faculty of Veterinary Science of the National University of Buenos Aires Center a Beef-Cattle Farm simulator called Simugan has been developed. Simugan allows users to experiment over a virtual farm in a simple and low cost way compared with real farm conditions. Increased number of users allowed deficit identification in Simugan usability, as inconsistency problems with inputs and outputs difficult to understand by novel users. Therefore, a combination of agile development and user centered design (AD&UCD) was applied to fix those identified problems. The results shown a high level of achievement with AD&UCD (94% of methodology required criteria) as reported in user interviews and evaluation tests.

Keywords— Agile Software Development, User Centered Design, Simugan, whole farm, livestock

I. INTRODUCCIÓN

LA ACTIVIDAD ganadera habitualmente es parte de una organización más amplia conocida como la empresa agropecuaria, que constituye la unidad básica para la administración de los distintos recursos y la toma de decisiones. La ganadería, al igual que el sector agropecuario en general, está constituida fundamentalmente por componentes biológicos que son manejados o controlados para lograr un fin económico. La interacción de estos componentes entre si forman un Sistema Ganadero, siendo el

manejo de este tipo de sistemas una prioridad fundamental para lograr el fin económico deseado. Estos sistemas poseen la particularidad de que bajo mismas alternativas de manejo del campo ganadero, se pueden obtener resultados productivos y económicos diferentes por la continua variación de los factores externos que los afectan como el clima y los precios [16]. De esta manera, el sector agropecuario enfrenta en la actualidad la necesidad de gestionar más y mejor la información para entender estas variaciones y tomar mejores decisiones.

El uso de la simulación matemática es recomendado para la formación en la operación y análisis de sistemas complejos como los mencionados [15], así como para la investigación de los mismos [3, 22]. Con estos objetivos, la Facultad de Ciencias Veterinarias de la Universidad Nacional del Centro de la Provincia de Buenos Aires (<http://www.vet.unicen.edu.ar>) desarrolló y esta utilizando un simulador agropecuario Web denominado Simugan [13]. Este simulador ganadero con base pastoril permite obtener la proyección de una empresa agropecuaria mediante simulación, reduciendo así los costos de experimentación (tiempo y recursos) entre otros beneficios. En general, dicho simulador ha presentado una eficiencia aceptable para el cumplimiento de los objetivos planteados [12].

El simulador agropecuario Simugan se desarrolló aplicando el método ágil Cristal Clear (CC) [14]. Este simulador permite ser accedido desde varios dispositivos para diferentes usos [1]. El uso de Simugan puso de manifiesto requerimientos adicionales: (i) vinculados a problemas de usabilidad en las interfaces de carga de información debido a inconsistencias en la visualización de funcionalidades similares, (ii) vinculados a problemas en la organización de los resultados que dificultan su comprensión, y (iii) nuevos requerimientos, como la comparación entre resultados de distintas simulaciones realizadas. De este modo, ante la necesidad de analizar cuál sería la mejor estrategia de desarrollo para abordar la implementación de estos nuevos requerimientos, surge la posibilidad de aplicar la combinación entre metodologías ágiles (MA) y diseño centrado en el usuario (DCU) [19].

Por consiguiente, este trabajo tiene por objetivo describir las experiencias adquiridas durante la aplicación combinada de CC y DCU en la mejora del simulador agropecuario Simugan.

Este trabajo está organizado de la siguiente manera: La Sección II describe las metodologías ágiles, el diseño de interacción centrado en el usuario y la integración de ambos enfoques. Adicionalmente, ofrece una breve descripción del

M. Arroqui, Facultad de Ciencias Exactas, Facultad de Ciencias Veterinaria, Universidad Nacional del Centro de la Provincia de Buenos Aires Argentina, mauriarroqui@gmail.com.

P. Mangudo, Facultad de Ciencias Exactas, Facultad de Ciencias Veterinaria, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, pmangudo@gmail.com.ar.

L. Pellizza, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, pellizza.lucas@gmail.com.

S. Murgolo, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, santi.m00@gmail.com.

A. Ottonello, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, laotto1@gmail.com.

S. Ferragut, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, safia.ferragut@gmail.com.

J. R. Alvarez, Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Facultad de Ciencias Exactas, Facultad de Ciencias Veterinaria, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, jmrodriguez.alvarez@gmail.com.

C. Machado, Facultad de Ciencias Veterinaria, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, cmachado@vet.unicen.edu.ar.

A. Teyseyre, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina, teyseyre@exa.unicen.edu.ar.

Corresponding author: Mauricio Arroqui

simulador agropecuario Simugan. La Sección III describe el rediseño de las interfaces del simulador aplicando este enfoque combinado. Finalmente, la Sección IV presenta los resultados de evaluaciones con usuarios finales y la Sección V presenta las conclusiones de este trabajo.

II. MATERIALES Y MÉTODOS

Mediante la conformación de una red de usuarios de Simugan, se identificó la oportunidad de complementar las prácticas ágiles en uso con un enfoque DCU para las mejoras planteadas, potenciando los beneficios y las mejores características de cada perspectiva [19]. Para esto, se realizó una integración de técnicas y actividades de usabilidad en el proceso de desarrollo que se venía aplicando. En especial las MA y el DCU se combinaron naturalmente dado que las MA asume una cercanía con los clientes y en el caso del DCU, iteraciones tempranas en los diseños de interfaz validadas por los usuarios [19].

A. Metodologías Ágiles

Las MA surgen como una alternativa de desarrollo de software a los problemas que presentan los desarrollos tradicionales en cascada [20]. La filosofía de las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones cortas. Tras una reunión celebrada el año 2001 en Uthah-EEUU, un grupo de 17 expertos en la industria de software redactaron el Manifiesto Ágil [4]. Dicho manifiesto describe 12 principios que un proceso de desarrollo ágil debe seguir. Se trata de un compromiso público en buscar nuevas y mejores formas de desarrollar software poniendo énfasis en las personas y sus interacciones, la colaboración y la respuesta continua al cambio, explorando nuevas formas de realizar tareas, y compartiendo experiencias.

Las MA respetan un ciclo de vida iterativo e incremental. Cada iteración es independiente y posee un objetivo particular, que contribuye con el global. Una iteración es un conjunto de actividades que se deben realizarse en un lapso de tiempo preestablecido por los desarrolladores. Una actividad se considera terminada una vez que ha sido completada, probada e integrada. Los usuarios o clientes reciben periódicamente versiones en funcionamiento del producto a medida que se va construyendo, lo cual les permite evaluar el trabajo realizado, advertir sobre problemas que se detecten, y sugerir mejoras o introducir funcionalidad valiosa no considerada originalmente.

A nivel de proceso, se realizan frecuentes reuniones retrospectivas donde los integrantes de los equipos comentan y discuten en profundidad tanto sus aciertos (para poder repetirlos y convertirlos en hábitos), así como también el trabajo que no se realizó correctamente o no llevó al equipo a obtener los resultados esperados.

B. Diseño centrado en el usuario

El DCU tiene por objetivo mejorar la usabilidad de un sistema buscando optimizar la forma en que los usuarios realizan sus tareas. En particular, la experiencia en numerosos dominios ha mostrado que las aplicaciones interactivas no

diseñadas teniendo en cuenta las capacidades de los usuarios finales requieren de un mayor esfuerzo de aprendizaje [17]. De este modo, suelen existir funcionalidades que no son utilizadas por la falta de conocimiento de las mismas por los usuarios. Por lo tanto, la usabilidad mide el grado en que los usuarios pueden aprender y utilizar un sistema dado para alcanzar sus objetivos deseados y el grado de satisfacción en dicho proceso. Adicionalmente, la usabilidad refiere a cómo los usuarios pueden entender el propósito de un sistema, cómo el sistema va a trabajar para ellos, y deducir cómo hacer para manipular el sistema con el fin de alcanzar una meta determinada [10]. En este sentido, el DCU define algunos criterios básicos que debe cumplir un sistema para aportar a su usabilidad, como por ejemplo la previsibilidad, familiaridad, consistencia, y observabilidad, entre otros. Del mismo modo, el término usabilidad es definido en la norma ISO 9241-11 [9] como el “grado en que un producto puede ser usado por usuarios específicos para lograr objetivos específicos con eficacia, eficiencia y satisfacción, en contextos de uso específicos”.

El proceso de desarrollo centrado en el usuario que sustenta la usabilidad de aplicaciones interactivas es de forma cíclica, iterativa e incremental y consta de 4 fases bien definidas [cita]: en la fase de planificación se identifican las necesidades y se establecen los requerimientos; en la fase de diseño se producen alternativas de solución; la fase de desarrollo se implementan las soluciones interactivas; y por último la fase de evaluación en la cual los distintos diseños se evalúan. Este proceso hace especial énfasis en la participación activa de los usuarios durante todas las fases aportando su punto de vista y necesidades [7, 17].

C. Diseño centrado en el usuario integrado a metodologías ágiles

Las MA definen el proceso mediante el cual se desarrolla una herramienta de software. Por otro lado, el DCU tiene por objetivo generar herramientas de software que sean real mente útiles y usables para los usuarios finales. Por la sinergia evidente entre estas dos metodologías, hay diversos antecedentes en su combinación [5, 11, 18, 21]. La utilización conjunta de estas metodologías presenta desafíos en su aplicación debido a que las MA están centradas en el desarrollador y DCU está centrado en el usuario, que generan conflictos filosóficos que deben ser considerados en su aplicación. En ese sentido, Dickinson et. al. [6] proponen lo siguiente:

- (i) Permitir que el equipo de diseño de las interfaces participe junto al cliente y los desarrolladores de software en la formulación y negociación de estimaciones y objetivos de cada iteración.
- (ii) Integrar el equipo de diseño al equipo de desarrollo de forma que ambos puedan realizar el testeo de forma conjunta.
- (iii) Posibilitar la comunicación “cara a cara” entre el equipo de diseño de interfaces y el líder del desarrollo para mejor comunicación de la transferencia de información entre equipos técnicos con conocimientos heterogéneos.

- (iv) Realizar iteraciones cortas que permitan tener feedback frecuente por parte del cliente y los usuarios para que los cambios propuestos tengan un impacto menor en el tiempo de desarrollo.
- (v) Propiciar un ambiente de trabajo independiente pero colaborativo entre los equipos de diseño de interfaces y desarrollo, para evitar, por ejemplo, diseños imposibles de implementar.

La aplicación conjunta de los puntos mencionados aumenta la probabilidad de obtener un producto de software ajustado a las necesidades del cliente, fácil de aprender y utilizar para los usuarios finales.

D. Simulador Agropecuario

El Simulador bioeconómico Simugan [13] fue desarrollado con el propósito de crear un sistema de soporte a la toma de decisiones en ganadería de carne. Particularmente, el simulador permite modelar las características de un campo y observar las consecuencias de distintos cursos de acción, aprendiendo de los mismos de una forma eficiente y económica, siendo un soporte valioso para la toma de decisiones. Actualmente, su desarrollo ha implicado 12500 horas/hombre de trabajo y debido al uso recurrente y complejidad de cálculo es que se ejecuta sobre una Grid Computacional [2]. En la Fig. 1 se muestra un esquema general de la implementación del simulador. Éste integra un modelo económico, financiero e impositivo a un modelo biológico base, los cuales en conjunto, permiten evaluar cualitativamente varios cursos de acción en diferentes plazos, contribuyendo al aprendizaje y a la toma de decisiones. El proceso de utilización de éste incluye: i) ingresar la información necesaria para definir la estructura general del campo, ii) definir reglas de contingencia (Ej. Realización de reserva forrajeras en caso de excedentes de pasto) y iii) comenzar a ejecutar una simulación. Al finalizar la misma, el usuario obtiene en su casilla de correo electrónico una planilla de calculo con los datos diarios y gráficos pre-configurados que muestran el comportamiento del campo durante el período definido de simulación.

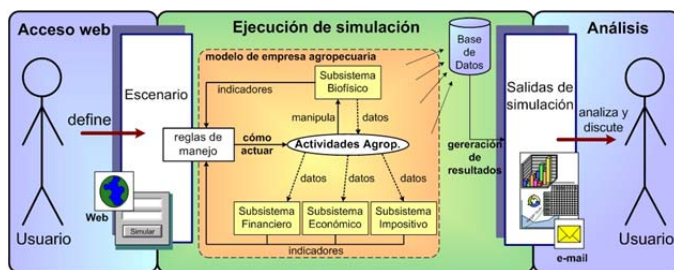


Figura 1. Esquema general de Simugan.

III. APLICACIÓN A SIMUGAN

La experiencia acumulada a lo largo de varios años en la utilización de Simugan muestra una buena aceptación, pero para su uso correcto demanda de un elevado costo de entrenamiento y de un soporte técnico intensivo, mayormente debido a limitaciones de usabilidad. En este sentido, se han detectado en el diseño de la interfaz algunos problemas de organización y de inconsistencia que dificultan la interacción, procedimientos largos y poco intuitivos, así como también

inconvenientes en la recepción e interpretación rápida de los resultados de la simulación. Por otro lado, se observó la necesidad recurrente de realizar análisis comparativos entre diferentes simulaciones, que debería ser de una manera integrada e intuitiva.

A fin de resolver estos problemas y lograr una interacción más simple y eficiente se rediseñó la interfaz web del simulador. Para esto, se integró el método ágil CC, el cual venía siendo aplicado al desarrollo, y técnicas de DCU siguiendo un enfoque generalista como el que se expone en Fox et. al. [7], donde los desarrolladores de software actúan también como especialistas de DCU. En la Fig. 2 se muestra el proceso mixto focalizado en la característica iterativa de los desarrollos ágiles y a los siguientes aspectos primordiales de DCU [8]:

- Centrarse en el usuario desde el principio y a lo largo de todo el proceso de diseño.
- Obtener medidas empíricas del uso de los prototipos y de la aplicación.
- Desarrollar un proceso de diseño iterativo en el cual los prototipos se diseñan, testean y rediseñan hasta conseguir el objetivo óptimo de funcionamiento.

En una etapa inicial (Fig. 2) previa a comenzar el desarrollo, se realizó un análisis exhaustivo del problema a solucionar. Inicialmente, se realizó un evaluación con 15 usuarios finales para determinar el estado actual del sistema. Luego se diseñaron diversos prototipos de baja fidelidad que luego fueron evaluados y analizados por las personas participantes del equipo con el objetivo de identificar y corregir inconvenientes de usabilidad antes de comenzar con el desarrollo. Estas evaluaciones sirvieron también para capturar nuevos requerimientos que debía cumplir el sistema final para lograr satisfacer la usabilidad deseada.

Luego de terminar de definir las especificaciones de usabilidad y el diseño de interfaz de usuario inicial, se dio comienzo a la segunda etapa, la etapa iterativa (Fig. 2). Esta etapa comenzó con reuniones de planificación para determinar que funcionalidades se implementarían en la primera iteración. El resto de las funcionalidades se colocaron en el llamado backlog, para ser realizadas en las sucesivas iteraciones. Luego de determinar que funcionalidades se incluirían en la primera iteración, comenzó el desarrollo. La etapa inicial e iterativa serán explicadas en más detalle a continuación, luego de describir el ambiente de trabajo aplicado junto con los roles que cumplieron las personas participantes del equipo.

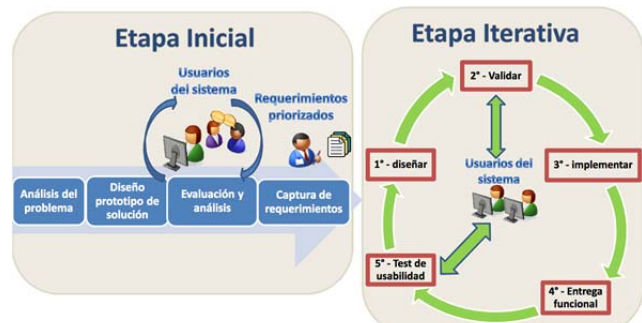


Figura 2. Proceso de desarrollo integrando MA y DCU.

A. Ambiente de Desarrollo y Roles en el Proyecto

El principio fundamental de DCU es la incorporación de los usuarios al equipo de trabajo con el objetivo de una exitosa interacción usuario-sistema. Esta metodología hace que el usuario esté presente en todas las etapas del desarrollo aportando su visión, promoviendo que los desarrollos implementados estén basados en sus necesidades y no en las de los desarrolladores. Para esto, se definieron y cumplieron los siguientes roles de CC y DCU:

- **Sponsor:** Rol que cumple la persona encargada de administrar los recursos que han sido asignados al proyecto y tomar las decisiones adecuadas para satisfacer las necesidades que el proyecto demande en función de la disponibilidad y prioridad impuesta desde los niveles superiores.
- **Usuario Experto:** Rol que cumple la persona que esta muy familiarizada con el modo de operación del sistema.
- **Coordinador:** Rol que cumple la persona que es responsable de comunicar al Sponsor el estado del proyecto, dándole una visión interna del mismo.
- **Diseñador-Programador:** Rol que cumple la persona que codifica el sistema. En CC no existe una separación de estos roles ya que es natural para cualquier programador la necesidad de diseñar la forma en que lleva a cabo la programación de una tarea.
- **Diseñador de interfaces:** Rol que cumple la persona que realiza las tareas de capturar requerimientos del usuario respecto a la interfaces, realizar prototipos de las interfaces, coordinar y liderar reuniones de validación de interfaces de usuario involucrando a los usuarios finales.
- **Testeador:** Rol que cumple la persona que es responsable de testear el sistema y crear los reportes de errores correspondientes.

Los roles de **Sponsor** y **Usuario experto** fueron llevados a cabo por dos personas que conocían en detalle el funcionamiento y desarrollo del simulador, siendo los mismos involucrados activamente en el proyecto sirviendo como soporte de usabilidad. Por su parte, los roles **Diseñador-Programador**, **Diseñador de interfaces** y **Testeador**, en concordancia con el enfoque generalista, fueron llevados a cabo por 4 personas que alternaban entre los 3 roles (en adelante **Grupo de Desarrolladores**). Estas 4 personas fueron coordinadas por una persona con el rol de **Coordinador**.

Para satisfacer las propiedades de *CC Comunicación cercana y Acceso fácil a usuarios expertos*, fue necesario definir un ambiente que asegurara la interacción fluida y continua de los miembros del equipo. Por esto, el **Grupo de Desarrolladores** y las personas con rol de **Usuario experto** y **Sponsor** del sistema se ubicaron dentro de un espacio común de trabajo. La transmisión del conocimiento se hizo efectiva, contribuyendo al consenso para la toma de decisiones frente a problemas, haciendo posible aplicar la práctica de *CC Programación “lado a lado”*. Mediante la utilización de radiadores de información se comunicaba a los todos los integrantes del equipo las planificaciones y el progreso de las actividades realizadas en las iteraciones, utilizando *Diagramas de Gantt* y transformados a medida que se avanzaba en el

desarrollo, en *Diagramas de quemado*.

B. Etapa Inicial

El análisis del problema se realizó aplicando la práctica de DCU *Reunión con los interesados en la herramienta*, en la que el **Grupo de Desarrolladores** llevó a cabo varias reuniones con las personas que cumplían el rol de **Usuario Experto** y **Sponsor**, para obtener la visión general del proyecto y sus objetivos. En base a estas reuniones y con los objetivos planteados, se realizó la planificación global del proyecto identificando las diferentes tareas a realizar con su correspondientes estimaciones en tiempo.

Luego se realizó un análisis más específico del problema a solucionar utilizando prácticas de DCU. En primer término se realizó la práctica *Investigación de contexto* para lograr un mejor entendimiento sobre quiénes eran los usuarios y cuáles eran sus necesidades. Adicionalmente se realizó la práctica de *Evaluación heurística* para determinar potenciales problemas de usabilidad y ponderar su gravedad, comprobando para ello el cumplimiento de principios de diseño usable (principios heurísticos) previamente establecidos. Estos principios de diseño son directrices que establecen requisitos que debe cumplir el diseño con el fin de facilitar su comprensión y uso por el usuario final.

En este sentido, los problemas de usabilidad detectados en las interfaces de carga fueron:

- Baja adaptabilidad de las interfaces de acuerdo al perfil de usuario logeado.
- Baja calidad visual y técnica de la administración de los escenarios de simulación.
- Imposibilidad de administrar simulaciones realizadas con anterioridad.

Por otro lado, los problemas de usabilidad detectados en los resultados de salida fueron:

- Gran tamaño de las planillas de cálculo enviadas como archivos por correo electrónico, con lo cual simulaciones de varios años no puedan ser enviadas al usuario.
- Gran cantidad de información en cada pestaña de las planillas de cálculo, y de este modo se demandan gran cantidad tiempo de aprendizaje.

Como consecuencia, se acordó que la solución para el caso de las interfaces de carga era realizar un rediseño de las mismas. En el caso de los resultados de salida se acordó que la planilla de cálculo no se seguiría enviando al correo electrónico del usuario sino que éste podría acceder a una representación de las mismas en línea a través del sitio Web de Simugan, para lo cual se utilizaría en la implementación la librería de Business Intelligence Eclipse BIRT (<http://eclipse.org/birt/>).

Durante el final de ésta etapa inicial se realizó la práctica de DCU *Prototipado rápido* mediante la cual se diseñó un primer diseño de baja fidelidad de la solución. Los mismos pueden ser desde dibujos hechos a mano alzada (Fig. 3, diseño interfaces de carga) hasta programas con pantallas gráficas interactivos sin funcionalidad (Fig. 4, diseño de los resultados). Los diseños se construyeron de manera iterativa y fueron validados mediante testeos de usabilidad con usuarios

reales. El propósito de estos testeos fue identificar y corregir inconvenientes de usabilidad antes de que el **Grupo de Desarrolladores** iniciase con el diseño de la interfaz de usuario.

Finalmente como resultado de esta etapa inicial se lograron establecer los requerimientos de usabilidad del sistema, como así también, el diseño inicial para comenzar la siguiente etapa. En esta etapa inicial, a diferencia de lo que ocurre con un diseño tradicional de DCU, debido a los tiempos reducidos que caracterizan a las iteraciones de los métodos ágiles, se focalizó en un conjunto acotado de requerimientos y necesidades, dejando el resto para ser abordados en la etapa iterativa.

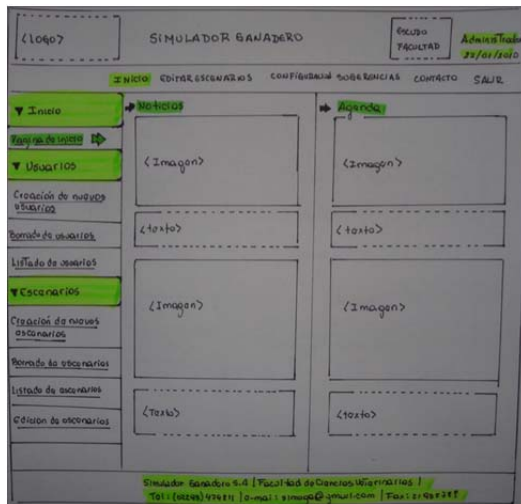


Figura 3. Diseño en papel del re-diseño del sitio Web.

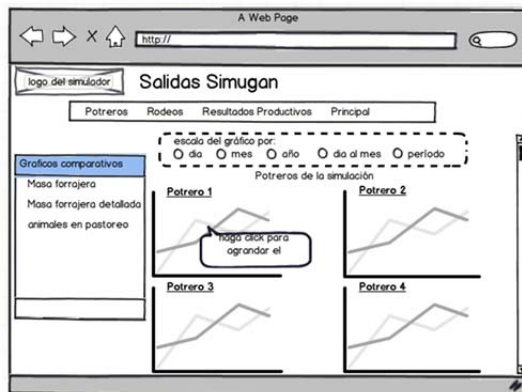


Figura 4. Diseño de baja fidelidad del sitio Web de resultados.

C. Etapa Iterativa

Esta etapa siguió un proceso de desarrollo ágil dividido en una serie de iteraciones cortas de tamaño fijo. De este modo, una iteración es una especie de miniproyecto con un tiempo fijado (propiedad de MA *Respetar plazos de tiempo*), donde se busca agregar al proyecto final una pequeña porción de funcionalidad estable y testeada.

Por la combinación metodológica entre DCU y MA, en la primera iteración se comenzó con la implementación del plan de la etapa inicial y se diseñaron las interfaces a implementar en la próxima iteración. De esta forma, una iteración consta del diseño y validación de las futuras interfaces a implementar

(1° tarea y 2° tarea de la Fig. 2) y de la implementación y testeo de las interfaces validadas en la iteración previa (3° tarea, 4° tarea y 5° tarea de la Fig. 2). Si las funcionalidades implementadas eran correctas y pasaban los testeos de usabilidad, se las marcaba como finalizadas. Al finalizar cada iteración se realizaba una entrega del software al usuario que sea funcional (propiedad de CC *Entrega frecuente*). De esta forma, el usuario testeaba que el sistema cumpliera con la usabilidad deseada, aplicando el principio de DCU *Participación activa de los usuarios*, favoreciendo las revisiones y correcciones continuas del desarrollo. El carácter repetitivo del proceso, organizado en ciclos de iteraciones y entregas de software al usuario con testeo de usabilidad (como se ve en la Fig. 2), permitió la maduración incremental del producto y por ende, de su arquitectura (propiedad de MA *Arquitectura incremental*). De esta forma se disminuyeron tiempos de refactorización al atender los errores de desarrollo de requerimientos en etapas tempranas, siendo ésta una de las características más importantes de las MA. A su vez, al realizar al fin de cada iteración la técnica de MA *Talleres de reflexión* se logró la mejora continua del proceso utilizado en el desarrollo del proyecto (propiedad de CC *Mejora Frecuente*). En las Figuras 5 y 6 se muestra las versiones finales de las interfaces de carga y de resultados de los sitios Web, respectivamente.

IV. RESULTADOS DE LA EVALUACIÓN CON USUARIOS

El proceso aplicado define evaluaciones continuas por parte de los usuarios para validar las decisiones de diseño e implementación tomadas por los desarrolladores. Aparte de los testeos realizados con las personas con el rol de **Usuarios Experto** se realizaron 2 evaluaciones del sistema involucrando a 15 usuarios que usaban o usarían el simulador agropecuario en algún momento. Entre los usuarios se encontraban estudiantes de grado y pos-grado, investigadores y profesionales del sector agropecuario, de los cuales 6 no habían utilizado nunca el simulador y de los restantes la cantidad de horas frente al mismo variaban desde 2 horas semanales a 10 horas semanales. La primer evaluación fue desarrollada antes de comenzar el desarrollo con el objetivo de obtener opiniones y medir experiencias de uso inicial, de la cual se obtuvo diferentes necesidades que tenían los usuarios al realizar distintas tareas.



Figura 5. Sitio Web de carga.

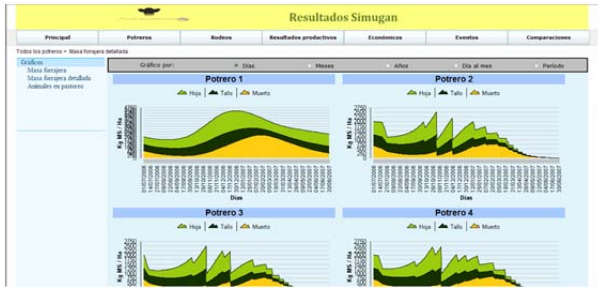


Figura 6. Sitio Web de resultados.

De este modo, esta primera evaluación fue utilizada como complemento del análisis realizado durante la etapa inicial (Fig. 2) con los usuarios expertos y el Sponsor. La segunda evaluación, sirvió para corroborar el cumplimiento de los requerimientos y la usabilidad deseada del sistema, comparando entre la versión inicial y la solución final desarrollada. En ambos casos la evaluación fue empírica y desarrollada aplicando las siguientes tareas definidas por DCU:

- Entrevistas a usuarios: Entrevistas realizadas a usuarios del sistema para capturar y diseñar los distintos perfiles de usuarios que utilizan o utilizarían el simulador.
- Evaluación de diagnóstico: Evaluación con usuarios de la versión inicial del desarrollo, cuyo objetivo fue capturar los requerimientos que los usuarios tenían para con el sistema, y realizar un primer análisis de usabilidad para obtener métricas iniciales. Al final del desarrollo, en la segunda evaluación, esta tarea fue realizada para obtener una métrica comparativa entre la versión inicial y la versión final con los requerimientos incluidos. La misma sirvió para definir luego, el trabajo a futuro y fomentar de esta manera una mejora continua a la solución propuesta.
- Evaluación subjetiva: La misma se realizó para que los usuarios plasmaran las opiniones personales que tenían sobre el sistema que estaban testeando. Sirvió para definir nuevos requerimientos en la primer evaluación realizada y para medir la mejora implementada, en la última evaluación.

En la Fig. 7 a) y b) se presenta el análisis de los resultados de “facilidad de uso” obtenidos en cada una de las evaluaciones con usuarios. En este aspecto se aprecia la mejoría lograda para la segunda evaluación, dado que no se encontraron evaluaciones con puntuación “Regular” y aparecen un 30% de evaluaciones “Excelentes”. La mejora fue lograda mediante la incorporación de funcionalidades nuevas y botones de navegación más intuitivos que los presentados en la versión evaluada en la primera evaluación.

En la Fig. 7 c) y d) se aprecia cómo se logró disminuir la brecha de aprendizaje, dado que no se obtuvieron evaluaciones “Regular” y disminuyeron las evaluaciones “Bueno” en un 5%, dando un incremento de 5% y 10% a las evaluaciones “Muy bueno” y “Excelente”, respectivamente. Esta mejora se debió, entre otros aspectos, a la incorporación de una navegación simple que permite llegar a cualquier dato específico realizando no más 3 clicks. Otro aspecto positivo

fue mantener la misma lógica de funcionalidad en toda las secciones tanto de carga de interfaces como de resultados.

En la Fig. 7 e) y f) se puede ver un gran cambio en la organización de la información, dado que en la segunda evaluación los valores para “Excelente” incrementaron en un 20%, “Muy bueno” incrementó un 10% y los valores de “Regular” desaparecen. Esta mejora es de suma importancia para la comprensión de los resultados de una simulación.

Por último, en el gráfico de la Fig. 8 se advierte un notorio avance de 56% a 94% en el cumplimiento de los diferentes criterios que son manifiestos por el DCU (predictibilidad, familiaridad, consistencia, observabilidad). Los resultados muestran un claro progreso y mejora en términos de usabilidad de las interfaces. Este avance se debe a la incorporación de nuevos aspectos y elementos a las páginas para poder alcanzar un diseño que satisfaga no sólo los criterios de diseño y usabilidad sino las necesidades de los diferentes usuarios.

V. CONCLUSIONES

Los resultados demostraron que la integración de DCU enriqueció el proceso ágil, se lograron mejoras sustanciales en la usabilidad y calidad del producto desarrollado y un incremento en la satisfacción de los usuarios.

En base a la experiencia realizada a continuación se destacan algunas observaciones relativas a tres aspectos centrales:

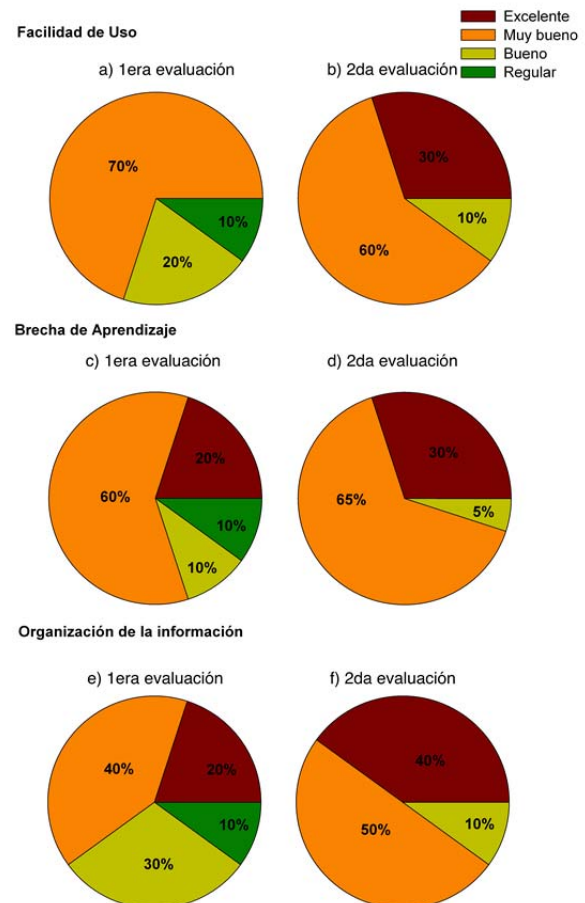


Figura 7. Comparativa entre evaluaciones.

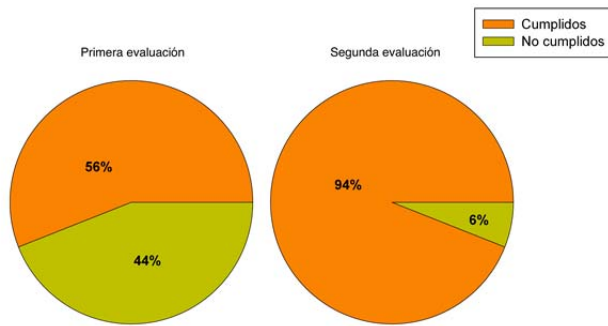


Figura 8. Comparación de resultados de la evaluación de criterios de diseño.

El proceso de desarrollo integrando los métodos ágiles con el diseño centrado en el usuario, técnicas aplicadas y cuestiones relativas a la administración del proyecto y colaboración. Muchas de estas observaciones confirman y son consistentes con lo reportado hasta ahora en la literatura [7, 19]:

- Proceso de Desarrollo:
 - ✓ Existencia de una etapa inicial antes de comenzar el desarrollo para realizar una investigación de contexto y el diseño de prototipos de baja fidelidad.
- Desarrolladores que actúan como desarrolladores de software y como diseñadores de interfaces. Técnicas y prácticas:
 - ✓ Prototipos de alta/baja fidelidad para explorar diseños alternativos que pueden ser comunicados y evaluados.
 - ✓ Evaluaciones de usabilidad con expertos y guías de diseño.
 - ✓ Test de usabilidad con usuarios y reuniones con grupos de usuarios.
- Administración del Proyecto y Colaboración:
 - ✓ La participación de los usuarios que aportan su punto de vista es central en el proyecto para lograr un producto usable.
 - ✓ Propiciar un ambiente de trabajo colaborativo definiendo adecuadamente responsabilidades y formas de comunicación para evitar malos entendidos.

De este modo, la combinación MA y DCU durante el desarrollo de herramientas de software aumenta la probabilidad de que, como lo sugieren las evidencias de nuestras observaciones, obtener beneficios para el negocio y lo más importante para el usuario. Por un lado MA permitió desarrollar software de calidad más rápidamente, y por otro lado, DCU permitió incrementar la satisfacción de los usuarios del sistema. Precisamente es de esperar que se logre:

- Responder a cambios y ofrecer nuevos productos en periodos más cortos de tiempo.
- Reducir los costos de producción: los costos y tiempos de desarrollo totales pueden ser reducidos evitando el rediseño y reduciendo el número de cambios posteriores requeridos en el sistema.
- Reducir los costos de entrenamiento y apoyo: los sistemas que son fáciles de usar requieren menos entrena-

menos soporte para el usuario y menos ayuda para la ejecución de tareas.

- Reducir los costos de uso: los sistemas que mejor se ajustan a las necesidades del usuario mejoran la productividad y la calidad de las acciones y las decisiones. Los sistemas más fáciles de utilizar reducen el esfuerzo y permiten a los usuarios manejar una variedad más amplia de tareas.
- Mejorar la calidad del producto: el DCU da lugar a aplicaciones de mayor calidad de uso, más competitivos en un mercado que demanda productos de fácil uso.

REFERENCIAS

- [1] M. Arroqui, C. Mateos, C. F. Machado, and A. Zunino. Restful web services improve the efficiency of data transfer of a whole-farm simulator accessed by android smartphones. *Computers and Electronics in Agriculture*, 87(0):14–18, 2012.
- [2] M. Arroqui, J. Rodríguez Álvarez, H. Vasquez, C. Mateos, C. Machado, and A. Zunino. A gridification method for agricultural simulation applications. *Concurrency and Computation: Practice and Experience*, 2014.
- [3] J. Bryant and V. Snow. Modelling pastoral farm agro-ecosystems: A review. *New Zealand Journal of Agricultural Research*, 51(3):349–363, 2008.
- [4] A. Cockburn, M. Beck, J. Beedle, R.C. Grenning, A. V. Martin, and Bennekum. *Principles behind the Agile Manifesto [en línea]*. 2001. <http://agilemanifesto.org/principles.html> (consultado en agosyo 2016).
- [5] T. Da Silva, M. S. Silveira, F. Maurer, and T. Hellmann. User experience design and agile development: From theory to practice. *Journal of Software Engineering and Applications*, 5(10):743–751, 2012.
- [6] J. Dickinson and D. Kumana. Getting agile with user-centered design, 2011.
- [7] D. Fox, J. Sillito, and F. Maurer. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In *Agile, 2008. AGILE '08. Conference*, pages 63–72. IEEE, 2008.
- [8] J. D. Gould and C. Lewis. Designing for usability: key principles and what designers think. *Commun. ACM*, 28(3):300–311, March 1985.
- [9] ISO - International Organization for Standardization. ISO 9241-11:1998 ergonomic requirements for office work with visual display terminals (vdt) – part 11: Guidance on usability. Technical report, 1998. <http://www.userfocus.co.uk/resources/iso9241/part11.html>.
- [10] Kolko J. *Thoughts on Interaction Design, Second Edition*. Morgan Kaufmann, 2011.
- [11] Jusoh N. M. User-centered design and their application. *Nor Azman Ismail Nur Zurifah Syazrah Othman*, pages 30–43, 2008.
- [12] C. F. Machado, J. Burges, H. Berger, C. Faverón, and I. S. teffanazzi. *First steps in the use of a web whole-farm model to foster the feedback between beef cattle extension and research*. An Overview of Research on Pastoral-Based Systems in South America, 2010.
- [13] C. F. Machado, S. Morris, J. Hodgson, M. Arroqui, and P. Mangudo. A web-based model for simulating whole-farm beef cattle systems. *Computers and Electronics in Agriculture*, 74(1):129–136, 2010.
- [14] P. Mangudo, M. Arroqui, C. Marcos, and C. F. Machado. Rescue of a beef-cattle system: Crystal clear in action. *International Journal of Agile and Extreme Software Development (IJAESD)*, 2012. In press.
- [15] M. Milrad, J. M. Spector, and P. Davidsen. Building and using simulation based environments for learning about complex domains. 2000.
- [16] C. J. Pearson. *Agronomy of grassland systems*. Cambridge University Press, 1997.
- [17] J. Preece, Y. Rogers, and H. Sharp. *Interaction design: beyond human-computer interaction*. J. Wiley & Sons, 2002.
- [18] T. Da Silva. *A framework for integrating interaction design and agile methods*. PhD thesis, Pontificia Universidade Católica do Rio Grande do Sul, 2012.
- [19] T. Da Silva, A. Martin, F. Maurer, and M. Silveira. User-centered design and agile methods: A systematic review. In *Agile Conference (AGILE), 2011*, pages 77–86, aug. 2011.
- [20] I. Sommerville. *Software engineering*. Pearson/Addison-Wesley, 2004.
- [21] J. Ungar and J. White. Agile user centered design: enter the design studio - a case study. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2167–2178, New York, NY, USA, 2008. ACM.
- [22] S. R. J. Woodward, A. Romera, W. Beskow, and S. J. Lovatt. Better

simulation modelling to support farming system innovation: review and synthesis. *New Zealand Journal of Agricultural Research*, 51:235–252, 2008.



Mauricio Arroqui es docente de la Facultad Ciencias Exactas e investigador de la Facultad Ciencias Veterinarias en la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina) en temas relacionados a AgroTICs. Obtuvo el título de Dr. en Ciencias de la Computación en el año 2014 en la UNCPBA-Argentina.



Pablo Mangudo es docente de la Facultad Ciencias Exactas e investigador de la Facultad Ciencias Veterinarias en la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina) en temas relacionados a AgroTICs. Obtuvo el título de Ingeniero de Sistemas en el año 2014 en la UNCPBA-Argentina.



Lucas Pelliza y Santiago Murgolo son Ingenieros de Sistemas recibidos en el año 2012 en la Facultad Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina). Título de la tesis de grado:

Reformulación de las salidas de un simulador agropecuario aplicando metodologías centradas en el usuario.



Andrea Otonello y Sofia Ferragut son Ingenieras de Sistemas recibidas en el año 2012 en la Facultad Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina). Título de la tesis de grado: Integración de diseño

centrado en el usuario con Métodos ágiles para el rediseño de un Simulador Ganadero Web.



Juan Rodriguez Alvarez es becario doctoral del CONICET e investigador de la Facultad Ciencias Veterinarias en la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina) en temas relacionados a AgroTICs. Obtuvo el título de Ingeniero de Sistemas en el año 2014 en la UNCPBA-Argentina.



Claudio F. Machado es profesor e investigador en la Facultad Ciencias Veterinarias de la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina). Obtuvo el título de Mg. en Producción Animal en el año 1993 en la Universidad de Chile, Chile y en el 2004 el título de Dr. en Ciencia Animal en la Universidad de Massey,

Nueva Zelanda.



Alfredo Tyseyre es profesor e investigador en la Facultad Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA-Argentina). Obtuvo el título de Mg. en Ciencias de la Computación en el año 2001 y el título de Dr. en Ciencias de la Computación en el año 2010 en la UNCPBA-Argentina.