Un Marco de Trabajo para Analizar y Mejorar la Calidad de Datos dentro de su Ciclo de Vida[1]

Gonzalo Domingo
Proyectos de Telesupervisión y Geociencias
D.S.I. Cuenta E&P - Argentina Sur
Repsol YPF
gedomingoe@repsolypf.com
y
Agustina Buccella, Alejandra Cechich
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue, Neuquén, Argentina
{abuccell, acechich}@uncoma.edu.ar

Resumen En la actualidad pocas empresas en la Argentina tienen en cuenta a la calidad de datos como requisito fundamental en todo desarrollo, implementación y uso del sistema. Es muy común que al momento de diseñar la aplicación, la calidad de datos sea muchas veces obviada y no exista una metodología o técnicas para su análisis. En este trabajo se propone una metodología orientada a pensar los sistemas desde la óptica de la calidad de los datos durante todo el ciclo de vida de un desarrollo de software, desde el momento del relevamiento y hasta la puesta en producción. La metodología cuenta con una serie de prácticas a realizarse de manera de garantizar dentro de una empresa la calidad de los datos cuando el sistema este en funcionamiento. Ilustramos la propuesta con un caso de estudio.

Palabras Clave: Calidad de Datos, Ciclo de Vida del Dato, Ciclo de Vida del Desarrollo de Software

1. Introducción

El término *Calidad de Datos* posee varias definiciones en la literatura [7,2,6], pero todas convergen en que el concepto calidad del dato es relativo al uso del dato [7]. Esto implica que este concepto es relativo, datos considerados con calidad para cierto uso pueden considerarse con insuficiente calidad para otros usos. Siguiendo con la definición, frases como: "Basura adentro, basura afuera"; "si se ingresa información inexacta, se obtendría información inexacta"; "pagar ahora o pagar mas tarde más" son muy comunes dentro del ámbito de calidad del dato. En nuestro trabajo, tomamos la definición de calidad del modelo FUNDIBQ (Fundación Iberoamericana de la Calidad) que establece, *la calidad es el conjunto de características propias de un producto, servicio, sistema o proceso imprescindibles para cumplir las necesidades o expectativas de partes interesadas*, es decir que la calidad es un punto de acuerdo entre las partes interesadas.

Una baja calidad de datos dentro de una empresa o institución lleva por ejemplo a clientes insatisfechos cuando sus datos personales, sus pedidos o sus facturas no son correctas; a empleados insatisfechos ya que cometen errores o no conocen cierta información, lo que los hace cometer a su vez más errores; a toma de decisiones erróneas porque los datos usados por los gerentes también pueden tener errores y es sabido que las decisiones no van a ser mejores que los datos en los que están basadas; etc. Por lo tanto se busca aprovechar los beneficios de una buena calidad de datos que se ven reflejados en la empresa o empresas que hacen uso de los mismos datos. Por ejemplo, mejora en el soporte a la

[1] Este trabajo esta parcialmente soportado por el proyecto UNCOMA 04/E059 (Mejora del Proceso de Desarrollo de Software Basado en Componentes)

toma de decisiones, reducción del tiempo necesario para obtener un informe, sustitución de actividades de bajo valor por otras de mayor valor, mejora de la imagen de la empresa, etc.

Para la elaboración de nuestro trabajo, hemos analizado varias propuestas que actualmente existen en la literatura. Entre ellas podemos citar a [3, 4, 8] ya que poseen en algún punto similitudes a nuestra propuesta. La propuesta de Wang [8] define una metodología denominada *Administración de la Calidad del Dato Total* (TDQM - Total Data Quality Management) cuyo objetivo es generar productos de información de alta calidad para los consumidores de información. La metodología propone, luego de analizar y conceptualizar el producto de información, la construcción de sistemas que fabrican o manufacturan la información (SMI). Estos SMI detallan las funcionalidades del sistema con los controles de calidad que debería poseer. Es justamente aquí donde se identifican posibles problemas de calidad analizando cómo se producen los datos.

En la propuesta de Ken Orr [3] el concepto de calidad de datos se basa en el uso del mismo y se establecen seis reglas para la calidad del dato como, los datos que no son usados no se mantienen correctos por mucho tiempo, la calidad de los datos está en función de su uso, no de su obtención, la calidad de los datos no será mejor que su uso más riguroso, etc. Teniendo en cuenta las reglas se definen una serie de actividades que apuntan a la evaluación y el análisis de la calidad de los datos. La actividad de *auditoria* consiste en determinar que tan buenos son los datos hoy. El *rediseño* se refiere a volver sobre las aplicaciones que están funcionando, enfocándose sobre todo en aquellos datos que puedan resultar más críticos para los procesos de negocio soportados por la aplicación y analizando cuidadosamente el uso que se le está dando a estos datos. La actividad de *entrenamiento* se centra en hacer comprender a los usuarios la importancia de la calidad de los datos. Así, se dedica tiempo a educación y entrenamiento. Por último la actividad de *medición* se refiere a medir constantemente la calidad de los datos, es decir, todas las actividades anteriores deben repetirse en el tiempo, haciendo a este un proceso iterativo. En comparación con nuestro trabajo, la metodología que proponemos intenta definir una guía práctica, aplicable a todos los sistemas que se van a construir en la empresa. Las recomendaciones o prácticas que creamos serán luego calificadas y no se basarán en su uso, sino en cómo afectan a las dimensiones de calidad del dato. Sin embargo, muchas de las prácticas que recogimos e incluimos en el marco de trabajo fueron recopiladas pensando en las reglas definidas en esta propuesta.

Por último, la propuesta de Pierce [4] plantea medir la calidad de los productos de información emanados de los sistemas a través de múltiples dimensiones tales como la certeza, la accesibilidad, la consistencia, etc. Para esto se utilizan matrices de control para combinar problemas con controles de calidad y así evaluar los productos de información. Las columnas de la matriz enumeran los problemas de la calidad de los datos que pueden afectar el producto de información; y las filas de la matriz son los controles de la calidad ejercitados durante el proceso de fabricación de la información para prevenir, detectar, o corregir estos problemas de la calidad del dato. Así, estos controles ayudan a evitar que cierto error aparezca en el producto de información. Nuestro trabajo tomó de esta propuesta la idea de basar el concepto de calidad en el ciclo de vida del dato. También implementamos la evaluación de las aplicaciones utilizando una matriz. Luego, clasificamos las distintas recomendaciones de calidad del dato encuadrándolas en el ciclo de vida del mismo y realizamos el análisis para evaluar si se respetaban los parámetros que se establecieron.

Este trabajo está organizado como sigue: La Sección 2 muestra el marco en el que se desarrolló la metodología. La Sección 3 explica la metodología en sí. La Sección 4 ilustra la propuesta con un caso de estudio. Conclusiones y trabajo futuro se discuten en la última sección.

2. La Calidad del Dato

En Redman [6] se describe el ciclo de vida del dato, el cual se compone de cuatro etapas fundamentales: *modelado del dato*, *captura del valor*, *almacenamiento* y *visualización*. La Figura 1 muestra gráficamente estas cuatro etapas.

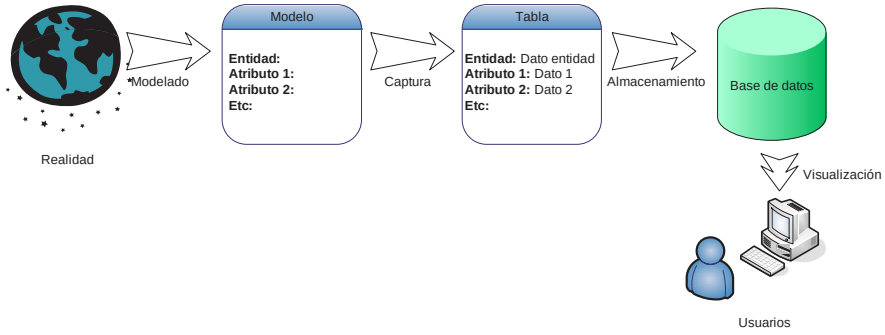

Figura 1. Ciclo de vida del dato.

La etapa de *modelado del dato*, se refiere a la elaboración de una abstracción representando la realidad una vez que han sido relevados los requerimientos o necesidades del cliente. Esta abstracción constituye un modelo lógico donde se establecen qué datos se tomaran y cómo fluirá la información por la aplicación y los roles de los actores que interactúan con ella. En la etapa de *captura del valor*, el dato es tomado de la realidad a través de interfaces del sistema o con otros sistemas. Aquí deben tenerse en cuenta aspectos como máscaras, fechas, validaciones y/o reglas en las interfaces de usuario. En la etapa de *almacenamiento*, el dato pasa de la interface de captura al repositorio de datos donde quedará almacenado. Por último la etapa de *visualización* se refiere a la presentación de los datos al usuario. Se deben tener en cuenta aspectos como la comprensión de la información, la identificación de errores o inconsistencias, la robustez, etc.

Para analizar y evaluar la calidad de los datos dentro de un proceso de desarrollo de software debemos tener en cuenta cuatro dimensiones principales [1]:

- *Exactitud*: ¿Representan los datos exactamente la realidad o fuentes verificables? La exactitud del dato esta relacionada con su fuente; es decir, es el nivel de correspondencia entre el dato y el mundo real.
- *Completitud*: ¿Todos los datos necesarios están presentes? ¿Qué cantidad de datos no están presentes? Esta dimensión se refiere a los datos necesarios que debe contener un sistema de información.
- *Consistencia*: ¿Los datos fueron consistentemente definidos y entendidos? Se refiere a la definición de estándares y protocolos para los datos. Todos los datos se representan en un formato compatible, que además es el más adecuado para la tarea que se está desarrollando.
- *Temporalidad*: ¿Los datos están disponibles cuando se necesitan? Por ejemplo, los datos ¿están disponibles cuando se deben tomar decisiones? Dentro de esta dimensión se enmarca el concepto de *Volatilidad*, el cual se refiere a la cantidad de tiempo que el dato se mantiene válido.

Estas cuatro dimensiones serán utilizadas para evaluar la calidad de datos en una empresa, ya que en ellas se enmarca la clasificación de las prácticas de acuerdo a la etapa del ciclo de vida del dato que afecten.

3. Un Enfoque para Mejorar la Calidad del Dato durante el Ciclo de Vida

Para la elaboración de nuestro nuevo enfoque, se relevaron las prácticas más utilizadas dentro de todo proceso de desarrollo de software y se analizaron de acuerdo a la calidad de los datos. Basados en las cuatro dimensiones de la calidad del dato, vistas en la sección anterior, clasificamos estas prácticas de acuerdo a la etapa del ciclo de vida del dato en que se encuentran. Definimos 64 prácticas en total – 27 prácticas para la etapa de modelado del dato, 22 para captura del valor, 4 en almacenamiento, y 11 en visualización. Por razones de brevedad no explicamos cada una de ellas, pero si daremos una breve descripción de algunas.

Ciclo de Vida del Dato: Modelado del Dato

- **Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas**: De esta forma se mantiene actualizado el flujo de negocio ya que a veces serán necesarias acciones derivadas de este cambio. De esta forma, se afecta la *Temporalidad,* ya que el dato se mantendrá valido en el tiempo. Si cambia, se notifica a los responsables del dato (los consumidores de información definidos como referentes) detectando tempranamente diferencias con la realidad. La *exactitud* se ve afectada ya que permite detectar de forma temprana diferencias entre el dato almacenado y la realidad, minimizando el impacto negativo de la duplicidad de datos.
- **Si el dato existe en un sistema fuente, tomarlo de la misma**: Hay aplicaciones que por su dominio son consideradas fuentes de datos ya que son las que capturan el dato lo más cerca posible de su generación. Una vez que un sistema está definido como tal, si el dato es necesario para otra aplicación, no debe duplicarse, sino que debe existir una interface entre este y el sistema fuente. De esta forma no solo se garantiza unicidad sino que se mejora el uso e implementación del sistema fuente. Esto mejora la dimensión de *temporalidad* puesto que con esta práctica se favorece a que el dato se mantenga válido por más tiempo y que se detecten cambios tempranamente mejora la *consistencia* ya que al relacionar los sistemas se mantiene la definición del modelado del dato y se mejora la misma. La *completitud* mejora ya que se favorece que todos los elementos necesarios del dato estén presentes al interrelacionar los sistemas que hacen uso del mismo. La *exactitud* también mejora, al mejorar el uso e implementación del dato, la relación del mismo con la realidad se ve favorecida. Por otro lado al evitar la duplicación de datos, se disminuye la probabilidad de errores.
- **Evitar que un dato esté duplicado en más de un sistema**: Con esta práctica no solo se garantiza unicidad sino que se mejora el uso e implementación de los sistemas participantes. Con esto, se gana en *temporalidad* porque el dato se mantendrá válido por más tiempo y se podrán detectar cambios tempranamente y en *exactitud* al mejorar el uso e implementación del dato se favorece la relación del mismo con la realidad. Por otro lado al evitar la duplicación de datos, se disminuye la probabilidad de error.

Ciclo de Vida del Dato: Captura del Valor

- **Para la Codificación de las tablas tipificadoras, realizar consultas *like* antes de realizar una nueva inserción**: Este punto se refiere a permitir ingresar un dato luego de haber hecho una comprobación de si el mismo existe en la base de datos. Por ejemplo, si se ingresa una calle y se coloca como dato "Rivadavia" el sistema debiera consultar en la base de calles y comprobar que existen dos datos coincidentes. Por lo tanto se debe preguntar al usuario si se refiere a "Comodoro Rivadavia" o a "Bernardino Rivadavia". Así se mejora la *consistencia* ya que se asegura que el dato se mantiene consistente ya que no se guarda el ingreso sino la coincidencia con la tabla tipificada, la *completitud* guardando el dato completo y no solo lo que se ingresa y la *exactitud* disminuyendo la posibilidad de error de carga y eliminando la posibilidad de error de tipeo.
- **Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez**: Esto debe ser evaluado con el usuario referente para evitar que la carga sea tediosa. Por otro lado minimiza el error al combinar las probabilidades. Se mejora la *exactitud* al minimizar la probabilidad de error de tipeo.

Ciclo de vida del Dato: Almacenamiento

- **Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar**: Esta regla de inferencia debe estar modelada en la aplicación para evitar así el error de ingreso de datos. Así se puede mejorar la *temporalidad* al ayudar a mantener el dato válido porque al cambiar los datos que le dieron origen, estos se actualizarán; la *consistencia* ya que el dato tendrá el formato esperado porque se define dentro de la aplicación; la *completitud* debido a que no hay ingreso humano y la regla deberá validar que los datos que le dan origen lo hacen en toda su completitud y la *exactitud* ya que la correspondencia con la realidad se mantendrá mientras la regla de inferencia esté bien modelada.
- **Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado y filtrado por estas reglas. El mismo debe ser dinámico**: Si se filtra la información por las reglas de negocio se puede detectar tempranamente falta de correspondencia entre el dato ingresado y lo esperado, evitando así el ingreso de datos basura. De esta forma mejora la *temporalidad* ya que el dato que ingresa es más estable y tiende a no quedar desactualizado; la *consistencia* ya que se ayuda a que el dato ingrese en el formato esperado y la *exactitud* porque se minimizan errores de ingreso de datos.

Ciclo de Vida del Dato: Visualización

- **El sistema debe alertar sobre vencimientos**: De esta forma el responsable de los datos, que es el usuario referente de la aplicación o quién se haya designado, es avisado cuando de acuerdo a la lógica de la aplicación algún dato está por perder validez. Mejora así la *temporalidad* ayudando al responsable a tomar acciones preventivas y la *exactitud* ya que al perder validez el dato se vuelve inexacto, por lo cual si esto se sabe con anticipación y se re valida, el riesgo de pérdida de correspondencia con la realidad disminuye.
- **El sistema debe verificar y advertir cambios en la tendencia de los datos**: De esta forma, se puede advertir en modo preventivo un cambio de tendencia. Para determinar si se trata de un error o de un cambio efectivo en la tendencia de la realidad, se requerirá un análisis funcional que

deberá realizar el dueño de los datos. Puede servir para detectar tempranamente errores en el registro del dato. Mejora así la *exactitud* al alertar de desvíos para tomar acciones preventivas destinadas a localizar falta de calidad en el registro del dato.

- **El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)**: Esta práctica es más de negocio que de sistemas. Pero es una recomendación que debemos realizar al negocio cuando estamos observando un proceso que debiera alimentar o alimentarse de otros procesos de la compañía pero no lo hace. Se espera que de esta forma mejore la calidad, intensificando el uso de los datos.

La clasificación de las prácticas está afectada por dos criterios según la independencia del sujeto que realiza el análisis, *Objetivo* y *Subjetivo*. El primero de ellos se realiza de acuerdo a un comité de calidad que evalúa las prácticas y el segundo se aplica el criterio del evaluador que podrá variar de acuerdo a su conocimiento del dominio y a su experiencia previa. Ambos determinan mediante una tabla de puntaje un valor de error si la práctica no se cumple.

La Clasificación Objetiva se enumera a continuación:

- *Práctica Estándar (E)*: La aplicación de este punto es considerada un estándar en la industria y su aplicación debería masificarse.
- *Buena práctica (B)*: Los puntos calificados de esta forma deberían aplicarse siempre que sea posible. Se considera que son muy importantes para garantizar que los datos sean de calidad en las dimensiones que afectan, pero se contempla que a veces resultan difíciles de aplicar por su costo, siendo antieconómicas. En estos casos, se debe documentar la decisión de no aplicarlas como parte del diseño.
- *Recomendación (R)*: La aplicación de estos puntos se considera favorable, quedando a criterio del líder del proyecto la evaluación costo/beneficio para su aplicación.

La Clasificación Subjetiva posee los siguientes valores:

- *Sin Error*: El soporte funcional observa que la práctica en cuestión está aplicada de forma correcta en la aplicación.
- *No Aplica*: La práctica recomendada no se observa, pero no se considera error ya que esta decisión de no aplicarla fue tomada en tiempo de diseño y la misma está documentada.
- *Leves*: Situaciones que pueden disminuir la calidad del dato. Cuando se permite el ingreso de datos de baja calidad pero no afecta a los datos que son críticos ni al éxito de la tarea que esta realizando el usuario.
- *Graves*: Situaciones propensas a disminuir la calidad del dato que pueden afectar el éxito de la tarea. Cuando la aplicación permite que se ingrese un dato que al ser erróneo pueda comprometer la tarea que se está realizando.
- *Fatales*: Errores conceptuales, aplicación de un modelo erróneo o errores que impiden terminar la tarea exitosamente. Son los más peligrosos ya que permiten que se ingresen datos que impiden terminar la tarea para la cual se los está capturando.

3.1. Mejora al Proceso de Desarrollo

Nuestro trabajo fue desarrollado en una empresa del medio que por confidencialidad, llamaremos “El Petróleo SA”. En dicha empresa, el desarrollo de sistemas, desde la obtención de

requerimientos hasta la puesta en producción del mismo, estaba guiado por un proceso que no consideraba la tarea de verificar y controlar la calidad de los datos como requisito temprano. Luego del análisis exhaustivo de calidad de datos realizado por dicha empresa, en donde fueron definidas las prácticas y recomendaciones antes mencionadas, el proceso de desarrollo fue modificado. La Figura 2 muestra la parte de este proceso con los controles de calidad agregados.

En la figura se observa cómo interactúan los actores que intervienen en el desarrollo de una aplicación. Estos actores son: *Usuario Referente*, *Analista Funcional*, *Proveedor* y *Soporte Funcional*.

En las etapas de relevamiento y elicitación de requerimientos, el usuario referente aporta las historias denotando su forma de representar los requerimientos funcionales del sistema. Por otro lado, el analista funcional determina las pautas de calidad. Este trabajo se realiza junto con el usuario para llegar a un acuerdo de cuáles de las prácticas definidas (evaluando el peso y costo/beneficio de cada una) serán puestas en práctica en la aplicación. Este trabajo genera un documento con las pautas de calidad acordadas que se entrega al proveedor de aplicación, sea este el equipo de desarrollo o el proveedor de aplicación en caso de productos que ya están desarrollados.

El equipo de proyectos se pregunta (no mostrado en la figura por cuestiones de espacio) , en esta instancia, si conoce una solución corporativa que aplique a la necesidad. Si es así, se implementa con consenso de la gerencia de proyectos. Si no se conoce una solución preestablecida, se elabora el documento de visión que enumera las necesidades del negocio, el objetivo de una solución informatizada y su alcance.

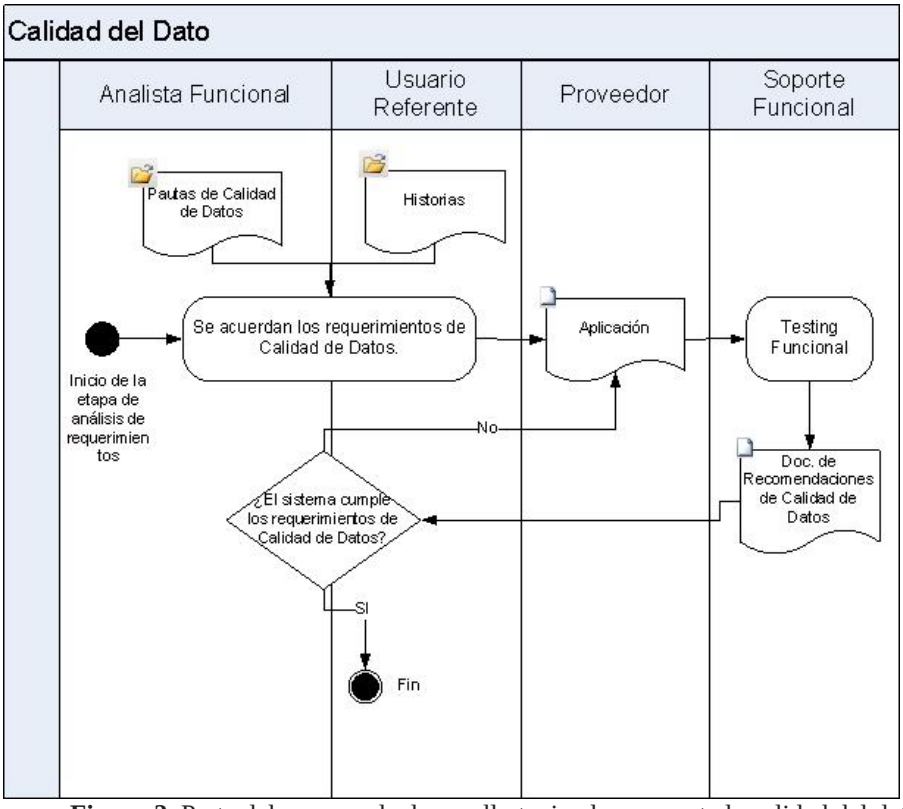

Figura 2. Parte del proceso de desarrollo teniendo en cuenta la calidad del dato.

Si el desarrollo será local, se empieza con la etapa de relevamiento detallado y diseño. El documento de historias se pasa al grupo de programación para que estime esfuerzo de desarrollo en

tiempo por historia. Con esto, el grupo de proyectos arma el cronograma y el plan de entregables. El mismo se valida con el usuario para ajustar prioridades y negociar tiempos de entrega teniendo como variable de ajuste el alcance de cada entrega. De estas entrevistas surgen como documentos el cronograma y el plan de entrega que detalla cada entregable con su alcance.

Se agregan a la documentación que se entregará al grupo de programación las pautas de calidad de datos. Estas son las 64 prácticas excepto que alguna se desestime por una decisión de diseño.

Luego, avanzando con el cronograma, se devuelve el código fuente de la aplicación, la aplicación funcionando sin errores unitarios ni de integración[2] y el manual de instalación continuando con la programación de la siguiente entrega. El grupo de proyectos recibe dicha entrega y la instala en un servidor de prueba donde realiza las pruebas funcionales generando documentos de errores y mejoras detectadas. A su vez, el Soporte Funcional utiliza estas pruebas funcionales como guía para realizar las pruebas de calidad de datos. Aplica el marco de trabajo para obtener así los valores de error y la recomendación de aprobación o no de la prueba.

En base a los resultados obtenidos, el Soporte Funcional, elaborará también el documento de recomendaciones de calidad del dato, donde dejará explicitada la descripción de las dimensiones que presentan errores, comentando todos los puntos en los que considere que no se respetaron las pautas o no se cumplen las recomendaciones y el motivo por el cual no encuadra con lo esperado. También se detallan las acciones para acceder o visualizar el dato en cuestión.

Con el documento de recomendaciones de calidad del dato, entre el analista y el usuario referente, decidirán si la aplicación cumple con un mínimo de pautas para entrar a producción o si se deben re-enviar los resultados al proveedor para que mejore el producto antes de iniciar su ciclo productivo. Si aparecen correcciones o ajustes tanto de calidad como funcionales, se re planifica y se devuelve a desarrollo. Si el usuario referente da su aprobación, pasa a un ambiente de producción donde lo pueden acceder todos los usuarios.

3.2. Aplicación de nuestro Marco de Trabajo: Verificando Aplicaciones

En nuestro marco de trabajo, cada práctica está ponderada de acuerdo a cada dimensión de la calidad del dato. La escala se puede ver en la Figura 3.

Ponderación	
Estándar	3
Buena práctica	2
Recomendación	1
No aplica	0

Figura 3. Tabla de ponderación de las prácticas (Clasificación Objetiva).

Esta ponderación objetiva se combina con la apreciación del soporte funcional. Éste, en el momento de la verificación, evalúa la aplicación de acuerdo al cumplimiento de cada práctica. Esta evaluación se efectúa de acuerdo a la escala de la Figura 4.

[2] En programación, una prueba unitaria es una forma de probar la corrección de un módulo de código, esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego con las Pruebas de Integración se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión

Luego, el valor de error se multiplica para cada práctica y cada dimensión por el de ponderación dando el valor de incumplimiento de cada práctica. Finalmente se suman los valores de todas las prácticas para determinar el valor final. De esta manera, si el valor de error es menor a 46, se recomienda aprobar la prueba. Si está entre 46 y 119 se recomienda aprobarla con observaciones y si el valor de error es mayor a 120, la recomendación de la prueba será rechazar la aplicación.

Estos valores fueron escogidos para que si una práctica estándar tiene un error fatal, (equivale a 40 x 3) se genere un valor de error de rechazo, y así cualquier valor de error mayor dará una recomendación de rechazo también. La aprobación se dará con cualquier valor de error menor o igual a 45. Esto se estableció considerando que un error grave en una práctica estándar era lo máximo aceptable para aprobar. La aprobación con observaciones será cualquier combinación de valores intermedios.

Error	Puntos	Descripción
Sin Error	0	
No Aplica	0	
Leves	5	Situaciones que pueden disminuir la calidad del dato
Graves	15	Situaciones propensas a disminuir la calidad del dato que pueden afectar el éxito de la tarea
Fatales	40	Errores conceptuales, aplicación de un modelo erróneo o errores que impiden terminar la tarea exitosamente.

Figura 4. Tabla de errores de las prácticas (Clasificación Subjetiva).

Los resultados serán validados con el usuario referente ya que el rechazo implica volver a desarrollar y planificar las entregas del producto. Será una decisión consensuada ya que existe la posibilidad de que el usuario referente igualmente apruebe el pasaje a producción de la entrega y que los errores encontrados se solucionen en la siguiente etapa.

Todos los errores encontrados y las decisiones tomadas en base a éstos quedarán documentados en el Documento de Recomendaciones de Calidad de Datos (visto en la Figura 2). En el mismo se detallan las recomendaciones que no se cumplen, cómo afectan a cada dimensión describiendo el error y el motivo por el cual no encuadra con la dimensión en cuestión.

4. Un Caso de Estudio

Como caso de estudio se utilizó una entrega de una aplicación que administra la electricidad de la compañía “El Petróleo SA”. Esta aplicación centraliza la información acerca de la electricidad generada por la compañía, la electricidad comprada y registra las ventas de energía eléctrica, como así también los datos de los equipos generadores de electricidad. La disponibilidad de esta información tiene, entre otros objetos, la generación de informes solicitados por la Secretaría de Energía de la Nación.

Antes de una solución informática, la información era mantenida en planillas, lo que dificultaba la generación de los informes a la Secretaría de Energía y otros entes reguladores. La generación manual de estos informes era compleja ya que había que corroborar grandes cantidades de datos para evitar inconsistencias. Los datos de la energía generada eran mantenidos localmente en cada planta y centralizados en forma manual por personal del área de energía eléctrica.

Para el desarrollo de este módulo y como era la primera vez que se aplicaba, se reunió a los desarrolladores, el analista, el soporte funcional de la aplicación y se los instruyo en los roles que cada

uno asumiría para esta prueba del proceso modificado y orientado a la calidad del dato en las aplicaciones.

Antes de comenzar la programación de la entrega que se tomó para la prueba, se les entregó a los desarrolladores las 64 prácticas definidas, junto con la categorización de las mismas de acuerdo a las cuatro dimensiones del dato y agrupadas de acuerdo a su ciclo de vida.

Cuando el equipo de desarrollo superó las pruebas unitarias y las pruebas de integración para asegurar el correcto funcionamiento del sistema o subsistema en cuestión, se le aplicó al entregable una prueba funcional. En ese momento el soporte funcional tomó la aplicación y ejecutó la prueba funcional, la prueba de usabilidad y el de calidad del dato, de acuerdo a la modificación implementada en el proceso de desarrollo.

Para esto, el soporte funcional, revisó cada una de las recomendaciones y verificó su cumplimiento, registrando si detectaba alguna falta o falla.

A continuación se explicarán algunas de las observaciones encontradas tal y como se documentaron en el "Documento de Recomendaciones de Calidad de Datos".

Ciclo de Vida del Dato: Modelado del Dato

- **Cuando se desactiva una cuenta de usuario se debe notificar, dependiendo del rol, al responsable del flujo sobre acciones preventivas**: En la dimensión temporalidad se consideró un error "Leve", ya que no se controlaba. Igualmente, la cantidad de usuarios en esta etapa del ciclo de vida de la aplicación no justificaba que se considere como un error de mayor envergadura. En exactitud se colocó "No aplica" por la razón antes explicada. De esta forma quedó documentado para ser tenido en cuenta en las siguientes entregas del desarrollo.
- **Si el dato existe en un sistema fuente, tomarlo de la misma**: En la aplicación existen datos que están en sistemas fuentes como los contratos de comercialización que se llevan con SAP. El sistema tiene una interface con éste por lo que se pudo verificar que no hay error.
- **Evitar que un dato esté duplicado en más de un sistema**: Se puso un valor de error "Leve" en temporalidad y "No aplica" en exactitud. Ya que por disposiciones de seguridad informática los datos tomados del sistema fuente SAP, con el cual se conecta esta aplicación, no pueden ser accedidos en línea sino que se exportan una vez al día a los sistemas satélites. Por lo cual la temporalidad puede hacer que el dato no sea valido como máximo por 24 horas. Pese a ello para este desarrollo se acordó que no afectaría a la exactitud ya que los datos importados son poco dinámicos.

Ciclo de Vida del Dato: Captura del Valor

- **Para la Codificación de las tablas tipificadoras, realizar consultas like antes de realizar una nueva inserción**: Se consideró un error "Leve" para la exactitud y "No aplica" para consistencia y completitud ya que las mismas no se ven afectadas porque la redundancia está controlada por la base de datos.
- **Si los datos a ingresar son críticos, evaluar el ingreso de los datos más de una vez**: En esta práctica se puso un valor de "No aplica" en la dimensión exactitud, ya que hay circuitos y cadenas de aprobación definidos para los datos más críticos. Como por ejemplo, cuando se carga el valor de una factura de distribuidor de un determinado proveedor de energía eléctrica, se dispara un mail a la gente de "Cuentas a Pagar" quienes con una copia de la factura verifican que el valor sea correcto y aprueban el pago en la aplicación.

Ciclo de vida del Dato: Almacenamiento

- **Si existe una regla matemática para inferir un campo a través de otro, este no se debe cargar**: Se verificó y se consideró sin error.
- **Las reglas de negocio relevadas deben ser parte de la aplicación para que el dato sea almacenado filtrado por estas reglas. El mismo debe ser dinámico**: Se verificó y se consideró sin error.

Ciclo de Vida del Dato: Visualización

- **El sistema debe alertar sobre vencimientos**: Se verificó y se consideró sin error. Por ejemplo, una vez cargada la factura de distribución, el personal de "Cuentas a Pagar" tiene 5 días hábiles para pagar, la aplicación verifica este vencimiento y va alertando del mismo.
- **El sistema debe verificar y advertir cambios en la tendencia de los datos**: Se colocó error "Leve" en exactitud ya que está en desarrollo y se espera implementarlo en una etapa más avanzada de la implementación.
- **El proceso de negocio soportado tiene que estar abierto a otros procesos (cultura de compartir los datos)**: Se verificó y se consideró sin error. De hecho el usuario referente es partidario de la sinergia entre áreas. Valoró que Sistemas de Información tenga entre sus recomendaciones el verificar que esto se cumpla. Esta aplicación que es del área "Ingeniería de Petróleo Gas y Electricidad" será también usada por el área "Cuentas a Pagar" gracias a esta cultura de procesos abiertos.

Estos son algunos de los puntos que se analizaron y documentaron. La valoración final de puntos de error dio un valor de 90 puntos compuestos por: 15 puntos por una práctica estándar con error leve, 10 puntos por una buena práctica con error leve, 10 puntos por una buena práctica con error leve, 5 puntos por una recomendación con error leve, 15 puntos por una práctica estándar con error leve, 10 puntos por una buena práctica con error leve, 15 puntos por una práctica estándar con error leve, 10 puntos por una buena práctica con error leve. Luego de comparar el peso de 90 con la tabla antes explicada, la recomendación final fue: "Aprobar con observaciones".

Con esta recomendación, se decidió pasar a producción la aplicación, documentar estas observaciones, analizarlas y mejorar en la siguiente entrega modular de la aplicación. Con lo que las observaciones encontradas se solucionaron en la siguiente iteración del proceso, sin demorar la planificación de la entrega del módulo analizado.

5. Conclusiones y Trabajo Futuro

En este trabajo hemos partido de la definición de calidad de los datos como un punto de acuerdo entre las partes interesadas, es decir, las características que un producto debe cumplir para satisfacer las expectativas de los interesados. Es sabido, que los datos de las organizaciones son propensos a dejar de satisfacer las necesidades de dichas partes rápidamente.

Para dar soluciones a este problema, hemos creado un marco de trabajo el cual ha generado la modificación de nuestro ciclo de vida en el proceso de desarrollo de nuestros sistemas. Este marco

funciona como una guía de recomendaciones a aplicar en los sistemas que se desarrollan, permitiendo evaluar los mismos de acuerdo a la forma en que están construidos y haciendo énfasis en la calidad de los datos que manipularán.

La metodología descripta debe ser parte integral de una organización, del grupo de desarrollo y de la mentalidad de sus componentes. Para lo cual es necesario mejorar los procesos y la manera de trabajar; dar a la planificación el lugar que se merece y producir un cambio de cultura. Es necesario persuadir a los empresarios de que los beneficios de medidas preventivas son tangibles. La calidad de los datos no debe ser un agregado a las aplicaciones, sino algo que surja desde el propio diseño.

De cara al futuro nos queda, implementar este proceso en toda la cultura de la organización para que los sistemas sean desarrollados atendiendo a factores como la calidad del dato y decidiendo de este modo invertir ahora y no pagar más adelante el costo de la falta de calidad. De esta manera el marco será probado en varios sistemas y podremos medir su eficacia y proponer en casos de que se necesario nuevos cambios.

Referencias

[1] G. Brackstone**.** Managing data quality in a statistical agency. Survey Methodology, (25):139-179, 1999.

[2] E. M. Burns, O. MacDonald, and A. Champaneri**.** Data quality assessment methodology: A framework. In Joint Statistical Meetings Section on Government Statistics, pages 334-337, 2000.

[3] K. Orr. Data quality and systems theory. - Communications of the ACM, 41(2):66-71, February 1998.

[4] E. Pierce. Assessing data quality with control matrices. Communications of the ACM, 47(2):82-86, February 2004.

[5] T. Redman. The impact of poor data quality on the typical enterprise. - Communications of the ACM, 41(2):79-83, February 1998.

[6] T. Redman. Data Quality: The Field Guide. Digital Press, January 15 2001.

[7] G. Tayi and D. Ballou. Examining data quality. - Communications of the ACM, 41(2):54-57, February 1998.

[8] R.Wang. A product perspective on total data quality management. - Communications of the ACM, 41(2):58{65, February 1998.