Un Estudio de la Calidad de los Datos Recolectados durante el Uso del Personal Software Process

Carolina Valverde, Fernanda Grazioli, Diego Vallespir
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
{mvalverde, grazioli, dvallesp}@fing.edu.uy

Resumen

Al usar un proceso de desarrollo de software, los individuos y los equipos generan datos acerca de su uso. Estos datos son fundamentales para el seguimiento de los proyectos de desarrollo de software. Es necesario contar con datos de calidad para tomar las decisiones acertadas durante un proyecto. Sin embargo, muchas veces estos datos no cuentan con la calidad necesaria. Este artículo presenta un estudio de la calidad de los datos del uso del Personal Software Process (PSP). Identificamos 10 posibles problemas de calidad y definimos 91 métricas para medirlos. Encontramos que un 1,34 % del total de los objetos (datos) medidos tiene algún error. Conocer y limpiar los datos de mala calidad ayudan a prevenir la toma de decisiones inadecuadas durante un proyecto de desarrollo de software.

1. Introducción

Los proyectos de desarrollo de productos de software utilizan (generalmente) procesos de desarrollo de software. Se busca mediante el uso de estos procesos construir productos de calidad, dentro del plazo y los costos establecidos. Durante el uso de un proceso los individuos y equipos generan datos acerca de su uso. Por ejemplo, registros de esfuerzo (tiempo empleado en cada fase), registros de fallas (y/o defectos) y registro de versiones. Normalmente, existen herramientas que dan soporte a los procesos para, entre otras cosas, simplificar el registro de los datos mencionados [13].

Los datos que se registran del uso del proceso son fundamentales para el seguimiento del proyecto. El análisis de los mismos influye directamente en la toma de decisiones. Sobre estos datos, en particular en los procesos que se basan en el control estadístico, se realizan análisis estadísticos para controlar y seguir el proyecto así como para realizar estimaciones y predicciones [7].

Los datos recolectados pueden ser de mala calidad. Esto provoca que los análisis que se realicen (estimación de tamaño, costo, plazo) brinden muchas veces resultados incorrectos y entonces que las decisiones que se tomen (basadas en esos resultados) sean las equivocadas. Decisiones equivocadas pueden llevar a grandes pérdidas y al fracaso del proyecto [10].

Lamentablemente, en una extensa revisión de la literatura, Bachmann encuentra que casi no existen trabajos que estudien la calidad de los datos que se recolectan durante el uso de un proceso de desarrollo de software [3]. La mayoría de esos pocos trabajos encontrados estudia los datos de los registros de defectos y los datos del software de control de versiones, dejando de lado otros datos generados durante el uso de un proceso de desarrollo de software.

En otro estudio los autores encuentran que la mala calidad de los datos del proceso de desarrollo afecta la calidad del producto de software desarrollado [2]. Debido al severo impacto negativo que la mala calidad de los datos puede tener es que Shepperd manifiesta "...por esto sugiero que este tema [calidad de datos] debe convertirse en una alta prioridad entre los investigadores de ingeniería de software empírica" [12].

También se debe tener en cuenta que la Calidad de Datos es un área de investigación en sí misma, en la cual se ha generado un gran volumen de trabajo (sobre todo en los últimos años) enfocado principalmente a: definir los distintos aspectos de la calidad de los datos [4, 9, 11, 14], y proponer técnicas, métodos y metodologías para la medición y para el tratamiento de la calidad de los datos [4, 9, 15].

Indudablemente, importa menos la cantidad de datos de la que se disponga que la calidad de los mismos. Dicho de otra forma, la calidad de las decisiones es fundamental en cualquier proyecto; y decisiones de calidad solamente se pueden tomar contando con datos de calidad [6].

Nuestra investigación busca realizar un aporte en el estudio de la calidad de los datos recolectados durante el uso de procesos de desarrollo de software.[1] Esta investigación

[1] Debe quedar claro que la calidad de los datos del uso de un proceso, la calidad del proceso y la calidad del producto son diferentes. Tanto la calidad del proceso como la del producto son extensamente abordadas por

se diferencia de los artículos encontrados en la revisión bibliográfica de Bachmann en dos grandes aspectos:

- Buscamos analizar todos los tipos de datos que se generan durante el uso de un proceso de desarrollo de software y no solamente algunos tipos de datos.

- Utilizamos la disciplina Calidad de Datos como el fundamento y la base de nuestro estudio.

En este artículo presentamos una primera evaluación de la calidad de los datos del uso de un proceso de desarrollo de software utilizando datos del Proceso Personal de Software (*Personal Software Process*, de ahora en mas PSP). El PSP es un proceso para un individuo para desarrollar módulos y pequeños programas de software [7]. El PSP es altamente instrumentado e incluye un marco de medición riguroso; esto lo hace un proceso adecuado para nuestra investigación.

Nuestros datos provienen de cursos dictados desde junio de 2006 hasta setiembre de 2010. Estos cursos fueron dictados por el *Software Engineering Institute* (SEI) de la Universidad Carnegie Mellon o por asociados (*partner*) del SEI; incluyendo un número diferente de instructores en múltiples países. Contamos con 408 sujetos (ingenieros que realizaron el curso) en nuestros datos.

Encontramos un único artículo que evalúa la calidad de los datos del PSP [8]. En dicho artículo se presenta la evaluación de una versión "vieja" del curso del PSP. En esa versión del curso los estudiantes deben realizar muchos cálculos de forma manual para obtener ciertos datos del uso del proceso. En las versiones de cursos más nuevas, que son las usadas en nuestro trabajo, se utiliza una herramienta que hace los cálculos automáticamente. Los estudiantes, por ejemplo, tenían que realizar estimaciones utilizando el método de regresión lineal de forma manual. Los tipos de problemas en los datos que presentan los autores son diferentes a los que nosotros proponemos. Esto se debe a la diferencia en las herramientas utilizadas, cómo se recolectan los datos y cómo se realizan los cálculos derivados de esos datos (manual o automáticamente).

Para realizar nuestro estudio adoptamos un enfoque sistemático, disciplinado y estructurado para la evaluación de la calidad de los datos. Utilizamos la propuesta de Batini y Scannapieco, que proviene de la disciplina Calidad de Datos [4]. Utilizamos e instanciamos dicha propuesta para luego aplicarla en los datos del PSP. Esta instanciación puede resultar repetible en estudios similares.

La Figura 1 presenta el trabajo completo que realizamos. Luego del dictado de los cursos del PSP se identificaron los posibles problemas de calidad de los datos. Para cada problema, se identificaron y ejecutaron métricas específicas cuyos resultados fueron registrados. Luego se realizó la limpieza y migración de los datos. Este artículo presenta los problemas de calidad, las métricas definidas y la medición realizada en la base de datos del caso de estudio, dejando fuera las actividades de limpieza de datos por un tema de espacio.

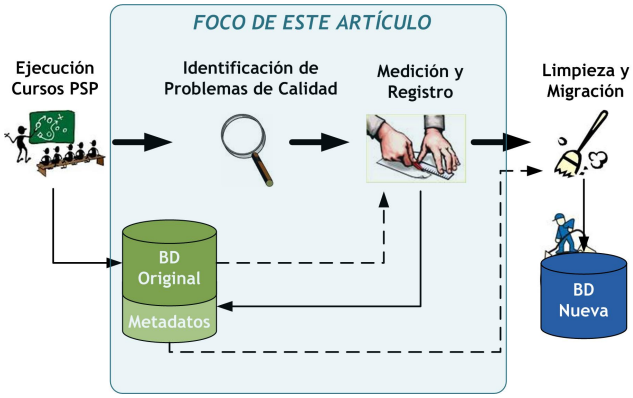

Figura 1. Etapas del estudio realizado

El artículo está organizado de la siguiente manera. El PSP se presenta en la sección 2. La sección 3 presenta las dimensiones y factores de la calidad de datos que son utilizados en este trabajo. La sección 4 presenta la metodología de trabajo y los problemas de calidad de datos en el uso del PSP. La sección 5 presenta un ejemplo de una métrica en particular. En la sección 6 se presentan los resultados y en la 7 las conclusiones.

2. El Personal Software Process

"El PSP es un proceso de mejora personal que ayuda a controlar, gestionar y mejorar la forma de trabajo" [7]. El proceso está dividido en fases que se van completando mientras se desarrolla el producto de software.

Para cada una de las fases del proceso existen guías que ayudan a seguir de forma correcta las actividades a desarrollar en cada fase. En cada una de estas el ingeniero de software recolecta datos sobre el tiempo utilizado en la fase y datos sobre los defectos que se han removido en la fase. Para cada defecto encontrado se registra: el tipo (siguiendo una taxonomía de defectos), el tiempo que llevó detectarlo y removerlo, la fase en la cual fue inyectado y la fase en la cual fue removido. La Figura 2 presenta las fases del PSP, las guías y la recolección de datos.

El PSP se enseña mediante un curso armado especialmente para ese fin. Durante el curso los estudiantes desarrollan programas mientras, de forma progresiva, aprenden el PSP. Existen diferentes versiones del curso, unas con 10 programas, otras con 8 y una con 7. En este artículo presentamos el análisis de la calidad de los datos recolectados

la literatura en ingeniería de software y en forma genérica por la literatura en gestión de la calidad. Sin embargo, como ya presentamos, la calidad de los datos recolectados durante el uso de un proceso de desarrollo de software casi no es abordada en la literatura.

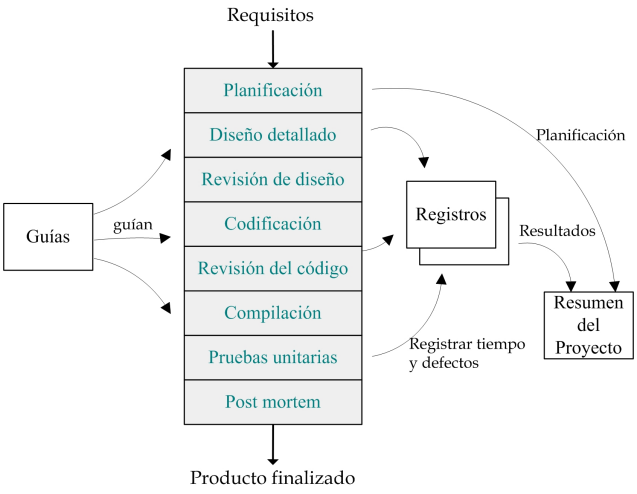

Figura 2. El Proceso Personal de Software

durante el dictado de cursos de 7 y 8 ejercicios. Los cursos fueron dictados por el *Software Engineering Institute* (SEI) de la Universidad Carnegie Mellon o por SEI *Partners*.

Durante el desarrollo de los programas del curso los estudiantes aprenden a planificar, desarrollar y evaluar el propio proceso utilizando las prácticas propuestas por el PSP. Para realizar el primer ejercicio del curso el estudiante utiliza un proceso definido y simple llamado PSP 0. A medida que el curso avanza se agregan nuevas actividades, fases y elementos: formas de estimar tamaño y esfuerzo, cómo realizar una planificación en el PSP, revisiones de código, elementos del diseño, revisiones de diseño, etc. A medida que estos elementos se agregan el proceso cambia. El nombre del proceso y los elementos fundamentales que se agregan en cada uno se presentan en la Figura 3. El PSP 2.1 es el PSP completo.

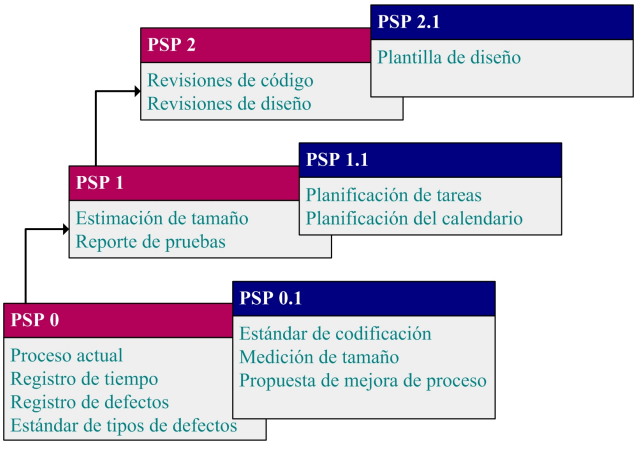

Figura 3. Los niveles del PSP en el curso

3. Calidad de Datos

Los datos constituyen un recurso muy valioso para las organizaciones al ser utilizados principalmente para la toma de decisiones, siendo de suma importancia para garantizar la sobrevivencia y éxito de las organizaciones.

La mala calidad de los datos influye de manera significativa y profunda en la efectividad y eficiencia de las organizaciones así como en todo el negocio, llevando en algunos casos a pérdidas multimillonarias [4]. Cada día se hace más notoria la importancia y necesidad en distintos contextos de un nivel de calidad adecuado para los datos.

Calidad de Datos es un área de investigación en sí misma, que ha avanzado mucho en los últimos años, generándose un gran volumen de trabajo científico en conferencias y *workshops* específicos [9, 11, 14, 15].

Existen distintos aspectos que hacen a la calidad de datos. Estos se conocen como dimensiones de calidad. En los trabajos del área de Calidad de Datos se propone gran variedad de conjuntos de dimensiones y de definiciones para las mismas [9, 11, 14, 15]. Sin embargo, existe un núcleo de dimensiones que es compartido por la mayoría de las propuestas. Este trabajo se basa en la propuesta de Batini y Scannapieco [4], que reúne estas dimensiones consensuadas.

En este trabajo utilizamos una abstracción de la calidad de datos [5], donde además de las dimensiones se definen otros conceptos para la clasificación y el manejo de la misma. Estos conceptos son el de factor, métrica y método de medición. Una dimensión de calidad captura una faceta (a alto nivel) de la calidad de los datos. Por otra parte, un factor de calidad representa un aspecto particular de una dimensión de calidad. Una dimensión puede ser entendida como un agrupamiento de factores que tienen el mismo propósito de calidad. Una métrica es un instrumento que define la forma de medir un factor de calidad. Un mismo factor de calidad puede medirse con diferentes métricas. A su vez, un método de medición es un proceso que implementa una métrica. Se pueden utilizar distintos métodos de medición para una misma métrica.

Las mediciones en una base de datos relacional se pueden realizar a varios niveles de granularidad: celda, tupla, tabla, e incluso a nivel de la base de datos entera. Por esto se definen funciones de agregación, que permiten pasar de un nivel de granularidad de datos a otro, obteniendo la calidad resumida para ese nuevo nivel. Por ejemplo, es posible obtener una medida de calidad de una tupla a partir de las medidas de calidad de cada una de sus celdas.

A continuación se presentan las dimensiones y factores de calidad utilizadas en este trabajo. De la propuesta de Batini y Scannapieco [4], no se considera la dimensión frescura relacionada con el tiempo, ya que los datos se consideran "frescos" y vigentes (respecto a la ejecución del proceso los

datos son eternamente vigentes).

Dimensión: Exactitud

La exactitud indica que tan precisos, válidos y libres de problemas están los datos. Establece si existe una correcta y precisa asociación entre los estados del sistema de información y los objetos del mundo real.

Existen tres factores de exactitud: exactitud semántica, exactitud sintáctica y precisión. La exactitud semántica se refiere a la cercanía que existe entre un valor v y un valor real v'. Interesa medir que tan bien se encuentran representados los estados del mundo real en el sistema de información.

La exactitud sintáctica se refiere a la cercanía entre un valor v y los elementos de un dominio D. Interesa saber si v corresponde a algún valor válido de D, sin importar si ese valor corresponde a uno del mundo real.

La precisión, por otra parte, se refiere al nivel de detalle de los datos.

Dimensión: Completitud

La completitud indica si el sistema de información contiene todos los datos de interés, y si los mismos cuentan con el alcance y profundidad que sea requerido. Establece la capacidad del sistema de información de representar todos los estados significativos de una realidad dada.

Existen dos factores de la completitud: cobertura y densidad. La cobertura se refiere a la porción de datos de la realidad que se encuentran contenidos en el sistema de información. La densidad se refiere a la cantidad de información contenida y faltante acerca de las entidades del sistema de información.

En un modelo relacional la completitud (en particular, la densidad) se caracteriza principalmente por los valores nulos. Un nulo puede indicar que dicho valor no existe, que existe pero no se conoce, o que no se sabe si existe en el mundo real.

Dimensión: Consistencia

Esta dimensión hace referencia al cumplimiento de las reglas semánticas que son definidas sobre los datos. La inconsistencia de los datos se hace presente cuando existe más de un estado del sistema de información asociado al mismo objeto de la realidad, y hay contradicciones entre dichos estados.

Las restricciones de integridad definen propiedades que deben cumplirse por todas las instancias de un esquema relacional. Se distinguen tres tipos de restricciones de integridad, las cuales se corresponden con los factores de esta dimensión.

Las restricciones de dominio, se refieren a la satisfacción de reglas sobre el contenido de los atributos de una relación.

Las restricciones intra-relación, se refieren a la satisfacción de reglas sobre uno o varios atributos de una relación. Las restricciones inter-relación, se refieren a la satisfacción de reglas sobre atributos de distintas relaciones.

Dimensión: Unicidad

La unicidad indica el nivel de duplicación de los datos. La duplicación ocurre cuando un objeto del mundo real se encuentra representado más de una vez en los datos, esto es, varias tuplas representan exactamente el mismo objeto. Distinguimos dos factores de la dimensión Unicidad. La duplicación, cuando la misma entidad aparece repetida de manera exacta, y la contradicción, cuando la misma entidad aparece repetida con contradicciones.

4. Problemas de Calidad en los Datos

Esta sección se subdivide en dos subsecciones. En la primera se presenta la metodología de trabajo y en la segunda se presentan los problemas de calidad identificados.

4.1. Metodología de Trabajo

El primer paso hacia la identificación de los problemas de calidad que podrían contener los datos bajo estudio, fue conocer la realidad y el contexto a analizar. Esto incluye el PSP, la herramienta para el registro de los datos y el propio modelo de la base de datos que utiliza la herramienta.

Una vez conocida la realidad y el contexto procedimos a analizar las dimensiones y factores de calidad propuestos por Batini y Scannapieco [4], que resulten interesantes de medir para dicho conjunto de datos.

Teniendo en cuenta cuáles son los datos relevantes y las dimensiones y factores de calidad a medir, se realizó una exploración de la herramienta utilizada en los cursos del PSP para registrar los datos del proceso, y un análisis de la estructura de la base de datos que los contiene. El foco estuvo en los valores que se ingresan de forma manual en la herramienta, ya que es allí donde ocurrirán la mayor cantidad de errores en los datos.

También fueron analizadas las *Grading Checklists*, utilizadas por los instructores para la corrección de los ejercicios realizados durante el curso, las cuales resultaron ser un gran aporte para identificar métricas relevantes para la calidad de datos en el PSP.

Clasificamos las métricas obtenidas en dos tipos: las que seguro miden un error en los datos y aquellas que permiten identificar datos sospechosos pero no nos permiten asegurar sea un error de datos. Por ejemplo, un valor negativo en un registro de tiempo es indudablemente un error en los datos. Por otro lado, resultaría sospechoso si un ingeniero no hubiese registrado ningún defecto considerando todos los programas desarrollados. En ese caso, es probable que el ingeniero haya olvidado registrar algún defecto durante el desarrollo, pero no podemos asegurarlo.

4.2. Definición de los Problemas de Calidad

La Figura 4 muestra la relación que existe entre dimensiones, factores, problemas y métricas, y cómo se aplican estos conceptos en un caso particular. Mientras que la definición de dimensión y factor de calidad es general y resulta aplicable en cualquier contexto, los problemas y métricas son específicos y definidos para el PSP, pudiendo ser utilizados para otros procesos de desarrollo de software.

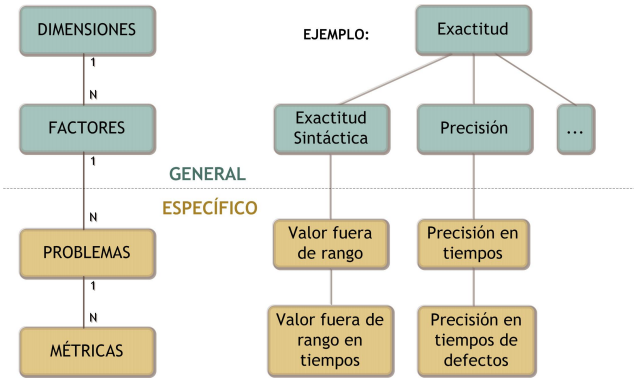

Figura 4. Relación entre Dimensiones, Factores, Problemas y Métricas

Las dimensiones de calidad de datos que se miden son: Exactitud, Completitud, Consistencia y Unicidad. En el Cuadro 1 se muestran todos los problemas identificados para cada factor y cada dimensión de calidad.

Dimensión	Factor	Problema de Calidad
Exactitud	Exactitud sintáctica	Valor fuera de rango
	Exactitud semántica	Identificador de proyecto incorrecto
	Precisión	Precisión en tiempos
Completitud	Densidad	Valor nulo
	Cobertura	Registro inexistente
Consistencia	Integridad de dominio	Reglas de integridad de dominio
	Integridad intra-relación	Reglas de integridad intra-relación
	Integridad referencial	Reglas de integridad referencial
		Referencia inválida
Unicidad	Duplicación	Registro Duplicado

Cuadro 1. Problemas de Calidad en los datos

A continuación se describe brevemente cada problema de calidad identificado. Para todos los casos, la unidad de medida del resultado es booleana: se mide si el objeto medido contiene o no un problema. Los problemas se miden, en la mayoría de los casos y salvo que se indique lo contrario, mediante la definición y ejecución de consultas *SQL*.

Valor fuera de rango

Los valores fueras de rango son aquellos que se sitúan fuera de un rango previamente definido como válido. Dichos valores podrían corresponder a valores anómalos y hacer que los resultados y conclusiones obtenidas al analizar los datos no reflejen fielmente la realidad.

Para cada uno de los casos identificados (por ejemplo, los tiempos) se establecen los criterios para determinar apropiadamente el rango que se va considerar para evaluar los valores, y se identifican los *outliers* mediante consultas en SQL. Un *outlier* es un valor que es inusualmente mayor o menor que otros valores en un conjunto de datos, pero que no corresponde necesariamente a un valor erróneo.

El rango se determina considerando el valor medio y la desviación estándar de los valores registrados. Los valores que se sitúen fuera de dicho rango se considerarán candidatos a contener errores, y por lo tanto serán analizados de manera aislada.

Identificador de proyecto incorrecto

Todos los usuarios deberían utilizar los mismos identificadores para hacer referencia a los mismos proyectos de la realidad.[2] De no ser así, resulta inviable poder realizar análisis de datos por proyecto.

Con este problema se identifican todos los proyectos que tienen asociado el proceso PSP correcto, pero su identificador de proyecto no se corresponde con la realidad.

Precisión en tiempos

Con este problema se desea medir el nivel de precisión de los tiempos registrados, ya que interesa conocer el momento de tiempo exacto en el cuál fue registrado un defecto. No debería suceder que las horas, minutos y segundos de los tiempos registrados sean todos iguales a 0.

Valor nulo

Interesa conocer qué información fue registrada y cuál fue omitida. Para aquella información que fue omitida, interesa conocer la causa de la omisión, y en caso que sea posible, determinar el valor que debería tomar en lugar de nulo.

Se identifican los campos que admiten nulos, pero deberían en la realidad contener algún valor distinto de vacío (el hecho de que admitan nulos es debido a un diseño incorrecto de la base de datos que utiliza la herramienta del curso del PSP).

Registro inexistente

Se identifican aquellos registros que no existen en la base pero sí existen en la realidad, y por lo tanto se omitió su ingreso. Esto significa que existe una porción de datos de la realidad que no se encuentra reflejada en la base de datos. Si

[2] En el PSP cada ejercicio (programa) tiene asignado un proceso (desde PSP0 a PSP2.1). Cada uno de estos programas tiene un identificador de proyecto.

no contamos con el universo total de datos, los análisis estadísticos que se lleven a cabo reflejará solamente una parte de la realidad estudiada.

Reglas de integridad de dominio

Para algunos atributos, es posible definir el dominio al cual sus valores deben pertenecer. En este caso, se define que el dominio válido para ciertos valores debe ser siempre mayor a cero.

Reglas de integridad intra-relación

Se definen un conjunto de reglas sobre ciertos atributos de una misma tabla, que deben ser satisfechas en la base bajo estudio. El hecho de que alguna de estas reglas sea violada, afecta la consistencia de los datos y por lo tanto cualquier análisis que se lleve a cabo a partir de estos.

Reglas de integridad referencial y Referencia inválida

Se definen un conjunto de reglas sobre ciertos atributos de diferentes tablas, que deben ser satisfechas en la base bajo estudio.

En particular, se identifican referencias hacia determinadas tuplas que no existen en la base y por lo tanto resultan ser referencias inválidas. Esto se debe a un error en el diseño del esquema de la base de datos, ya que se omite la definición de *foreign keys* sobre ciertos atributos.

Registro duplicado

Se identifica este problema de calidad cuando existen dos o más registros que aparecen repetidos de manera exacta. Existen dos situaciones:

- Cuando contienen el mismo valor en la clave y demás atributos (o en su defecto valores nulos). Este caso se contempla con controles del SGBD.
- A pesar de contener distinta clave primaria, hacen referencia al mismo objeto de la realidad y contienen los mismos datos en los campos que se definan. Para este caso se verifica que no existan registros repetidos (según el criterio definido) en la base bajo estudio.

Se consideran dos casos. La duplicación de defectos, que corresponde a defectos registrados a la misma hora exacta, y la duplicación de estudiantes, que corresponde a estudiantes que contienen el mismo nombre, mismo instructor y misma fecha de creación de perfil en la herramienta.

5. Ejemplo de Métrica

A modo de ejemplo en esta sección se presenta cómo se aplicó una métrica de un problema de calidad concreto, a un objeto de la base en particular. De esta forma se puede entender el trabajo realizado para cada uno de los problemas y métricas identificados.

Recordemos que una métrica es un instrumento que define la forma de medir un factor de calidad. En nuestro trabajo, definimos métricas para los problemas de calidad como el instrumento que asigna un número a un problema sobre uno o varios objetos de la base (atributos, tuplas y/o tablas específicas), dependiendo de su granularidad. Una misma métrica puede entonces ser utilizada sobre diferentes objetos.

Presentamos como ejemplo la métrica “Precisión en tiempos de defectos”, que se encuentra definida dentro del problema de calidad “Precisión en tiempos”. Este problema lo ubicamos dentro del factor Precisión y dimensión Exactitud.

Los ingenieros que realizan los cursos de PSP deben ingresar el momento exacto (fecha incluyendo hora, minuto y segundo) en el cual un defecto es removido. Esto se registra en la herramienta de forma manual, lo que puede ocasionar que se introduzcan errores en los datos. Dado que durante la exploración de la base de datos realizada para identificar los problemas de calidad detectamos que existían registros sin la precisión adecuada, resulta interesante medir cuáles son los registros que no cuentan con el detalle de hora, minuto y segundo.

Como vimos en la sección anterior, se estudió la realidad y el contexto, y se identificó un problema correspondiente a un factor de calidad y a una dimensión. Luego se define la métrica como instrumento de medición de dicho factor. Para esta métrica particular, la definición es: “Las horas, minutos y segundos, que corresponden a un registro de tiempo no son todos iguales a cero”. También se debe especificar la forma de medición, la unidad del resultado y el método de medición. En este caso, la forma de medición consiste en encontrar los registros de defectos que cumplen que “hh:mm:ss” es igual a “00:00:00”, donde el formato de fecha es “yyyy-mm-dd hh:mm:ss”. Se define la unidad del resultado como un booleano, por lo tanto lo que obtenemos como resultado es si el registro medido es erróneo o no. El método de medición definido para esta métrica es la ejecución de una consulta SQL.

Al ejecutar la medición detectamos 1512 registros que contienen valores en el tiempo de defectos sin la precisión adecuada de un total de 20916 registros de defectos (7,23 %). Si lo vemos a nivel de ingeniero, detectamos 149 ingenieros distintos que cometieron este error al menos una vez de un total de 408 ingenieros (36,52 %).

En la Figura 5 se muestra un histograma que nos permite ver qué porcentaje de error cometen los ingenieros y cómo se distribuyen los mismos. En el mismo podemos observar que si bien el 63,48 % de los ingenieros tuvieron la totalidad de sus registros de defectos sin este error, también existen ingenieros que cometieron este error en la totalidad de sus registros (2,45 %).

Para poder entender por qué los ingenieros cometieron este error, decidimos explorar más a fondo la herramienta de registro utilizada en los cursos de PSP y encontramos que hay un defecto en la misma. Ya que al momento de re-

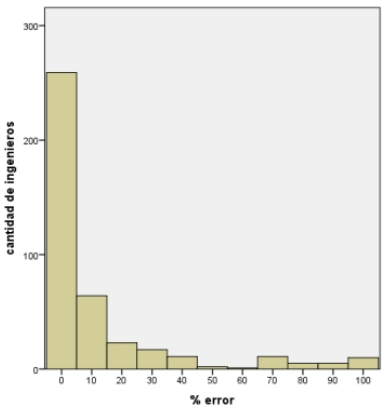

Figura 5. Distribución de ingenieros según porcentajes de error

gistrar un defecto, si se hace "doble *click*" en el campo de la fecha, el valor se pone correctamente con el formato "yyyy-mm-dd hh:mm:ss", pero sin embargo si se hace *click* en el calendario y se elige la fecha del día, la parte de "hh:mm:ss" queda con el valor "00:00:00", siendo esto último difícil de detectar por el ingeniero.

En esta sección no sólo vimos como definir una métrica y ejecutar la medición, sino una forma de visualizar los resultados y la interpretación de los mismos. Existen otras maneras de observar los resultados de la medición, como puede ser a través de un diagrama de dispersión para ver cómo evoluciona el porcentaje de error a medida que van avanzando los ejercicios del curso. De esa forma se puede detectar, por ejemplo, si los porcentajes de errores van bajando, si se mantienen constantes o si aumentan a medida que avanza el curso. Esta información puede ayudar (de forma temprana) a un instructor a ajustar problemas que tengan los estudiantes con la recolección de los datos. Recordemos que es importante lograr tener la mejor calidad de los datos posible, ya que muchas veces los ingenieros basan sus decisiones y estimaciones en datos históricos, y si los mismos tienen problemas de calidad, es probable que las decisiones tomadas no sean las más acertadas.

6. Resultados Generales

Se identificaron 10 problemas de calidad, y se definieron un total de 91 métricas para medir estos problemas aplicados sobre objetos (celdas y/o tuplas) de la base. Realizamos la medición de la totalidad de las métricas que fueron definidas. La ejecución de la medición para estas métricas se realizó en el 100 % de los casos de manera automática mediante sentencias SQL, donde para un 20 % de los casos se utilizaron algoritmos programados en PHP. Esta sección presenta los resultados más importantes y de forma general. Los resultados completos de nuestro análisis se encuentran en: www.fing.edu.uy/inco/grupos/gris/psp-data-quality.htm.

En el Cuadro 2 se muestra la cantidad de métricas definidas para cada problema de calidad, la cantidad de métricas que corresponden a errores y la cantidad que corresponden a casos sospechosos. Además se indica, para cada problema de calidad, el porcentaje de objetos con la presencia de ese problema de calidad según se consideren las métricas que miden errores o casos sospechosos.

Luego de ejecutar todas las mediciones, observamos que un 1,34 % del total de los objetos medidos tiene algún error. Si consideramos únicamente las métricas que miden errores (y no contamos los casos sospechosos) ese valor baja a un 0,99 %, y si consideramos solamente las métricas que miden casos sospechosos el porcentaje de objetos con error asciende a un 2,10 %. El estudio de Johnson y Disney presentaba un 4,8 % de errores en los datos [8]. En ese caso muchos de los errores en los datos se debían a cálculos manuales de datos derivados, situación que no ocurre con las herramientas actuales de soporte al PSP. Esta diferencia en la calidad de los datos debido a las herramientas utilizadas es también mencionada por Bachmann y Bernstein: "También discutimos el impacto de las características de los proyectos [estudiados] y concluimos que, por ejemplo, la naturaleza de los procesos de ingeniería de software, el uso de distintas herramientas de soporte a los procesos, [...] resultan en características diferentes en los datos" [1].

Estos resultados de calidad, vistos tanto a nivel global como a nivel particular para cada problema de calidad, pueden estar indicándonos que es necesaria una limpieza de los datos para garantizar que los futuros análisis estadísticos de los datos de estos cursos sean sobre datos válidos. Caso contrario los resultados y las conclusiones de dichos análisis pueden no corresponderse con la realidad.

Por otro lado, desde el punto de vista de la práctica de la ingeniería de software, queda claro que es necesario un análisis de la calidad de datos del uso de un proceso antes de utilizarlos. Evitar usar datos de mala calidad puede prevenir la toma de decisiones inadecuadas en el transcurso de un proyecto.

7. Conclusiones

En este artículo presentamos el uso de un enfoque sistemático, disciplinado y estructurado de la disciplina Calidad de Datos para identificar y medir los problemas de calidad en los datos recolectados durante el uso del PSP. No encontramos en la literatura otros estudios o publicaciones que aborden esta problemática con un enfoque como el que aquí se presenta. Esta propuesta puede ser utilizada, con adaptaciones, en otros procesos de desarrollo de software.

Los resultados muestran que el porcentaje de datos de

Problema de Calidad	# total de métricas	# métricas error	# métricas sospechosos	% objetos con error	% objetos sospechosos
Valor fuera de rango	8	0	8	0,00 %	3,20 %
Identificador de proyecto incorrecto	1	1	0	8,66 %	0,00 %
Precisión en tiempos	1	1	0	7,23 %	0,00 %
Valor nulo	26	26	0	1,32 %	0,00 %
Registro inexistente	3	1	2	0,12 %	16,90 %
Reglas de integridad de dominio	15	15	0	1,29 %	0,00 %
Reglas de integridad intra-relación	5	2	3	1,05 %	0,40 %
Reglas de integridad referencial	10	9	1	11,94 %	3,44 %
Referencia inválida	20	20	0	0,04 %	0,00 %
Registro duplicado	2	2	0	1,62 %	0,00 %

Cuadro 2. Cantidad de métricas y porcentajes de objetos con error para cada problema de calidad

mala calidad es menor al presentado en otro análisis realizado anteriormente [8]. Existe una diferencia importante entre nuestro estudio y el anterior: el uso de una herramienta de soporte al proceso. En el estudio anterior los datos eran recolectados en papel y los cálculos que había que realizar con dichos datos eran manuales. En nuestro estudio utilizamos cursos del PSP que ya contaban con el soporte de una herramienta que permite una mejor recolección de los datos y el cálculo automático de datos derivados.

La detección de errores en los datos así como su limpieza es importante en el contexto del uso de los procesos de desarrollo de software. Los datos que se generan durante el uso de un proceso son utilizados luego para predicciones, cálculos de avance del proyecto, etc. Si los datos que se utilizan son de mala calidad puede suceder que las decisiones que se tomen durante el proyecto, o acerca de un nuevo proyecto, sean las decisiones equivocadas.

Por otro lado, la ingeniería de software empírica utiliza datos de diversos experimentos o casos de estudio que luego analizar para confirmar o descartar ciertas hipótesis. Si los datos que se utilizan son de mala calidad las confirmaciones o rechazos realizados durante el estudio empírico pueden ser incorrectos.

Desde la perspectiva de la ingeniería de software empírica este artículo busca concientizar acerca de la importancia que tiene la disciplina de calidad de datos para los procesos de desarrollo de software. Desde la perspectiva de la calidad de datos, este trabajo muestra una aplicación de las técnicas de medición de calidad y de limpieza de datos a un dominio particular.

Referencias

[1] A. Bachmann and A. Bernstein. Software process data quality and characteristics: a historical view on open and closed source projects. In *Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops*, IWPSE-Evol '09, pages 119–128, New York, NY, USA, 2009. ACM.

[2] A. Bachmann and A. Bernstein. When process data quality affects the number of bugs: Correlations in software engineering datasets. In *Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on*, pages 62 –71, may 2010.

[3] A. J. E. Bachmann. *Why Should We Care about Data Quality in Software Engineering?* PhD thesis, University of Zurich, 2010.

[4] C. Batini and M. Scannapieco. *Data Quality: Concepts, Methodologies and Techniques*. Springer-Verlag Berlin Heidelberg, 2006.

[5] L. Etcheverry, V. Peralta, and M. Bouzeghoub. Qbox-foundation: a metadata platform for quality measurement. In *4th Data and Knowledge Quality Workshop*, 2008.

[6] E. Harper and D. Zubrow. Should you trust your data? *Software Quality Professional*, 13(4):4–8, 2011.

[7] W. Humphrey. *PSP A Self-Improvement Process for Software Engineers*. Addison-Wesley, 2005.

[8] P. M. Johnson and A. M. Disney. A critical analysis of PSP data quality: Results from a case study. *Empirical Softw. Engg.*, 4(4):317–349, Dec. 1999.

[9] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang. Aimq: a methodology for information quality assessment. *Inf. Manage.*, 40:133–146, 2002.

[10] L. More. No data quality control? Expect to count the cost. *Computing*, pages 28–29, 2006.

[11] M. P. Neely. The product approach to data quality and fitness for use: A framework for analysis. In *Proceedings of the 10th International Conference on Information Quality MIT*, 2005.

[12] M. Shepperd. Data quality: Cinderella at the software metrics ball? In *Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics*, WETSoM '11, pages 1–4, New York, NY, USA, 2011. ACM.

[13] I. Sommerville. *Software Engineering*. Addison-Wesley, 9th edition, 2010.

[14] D. M. Strong, Y. W. Lee, and R. Y. Wang. Data quality in context. *Commun. ACM*, 40:103–110, 1997.

[15] R. Y. Wang, M. P. Reddy, and H. B. Kon. Toward quality data: an attribute-based approach. *Decis. Support Syst.*, 13:349–372, 1995.