

Diagnóstico del Uso de Técnicas de Revisión de Software en Uruguay

Darío Macchi, Martín Solari

Universidad ORT Uruguay

macchi@uni.ort.edu.uy, martin.solari@ort.edu.uy

Abstract. Existen distintas técnicas estáticas de detección de defectos entre las que se diferencian las revisiones por no requerir de la ejecución del artefacto a evaluar. El objetivo de esta investigación es conocer de qué manera están siendo revisados los distintos artefactos generados durante el proceso de desarrollo de software en Uruguay. Se realiza una encuesta *online* dirigida a profesionales en desarrollo de software, vinculados a actividades de aseguramiento de calidad o definición de procesos. Los resultados de la encuesta indican que no se utilizan técnicas formales de revisión aunque distintos artefactos de software son revisados mediante técnicas menos formales y sistemáticas (técnicas *ad-hoc* y revisiones de pares). Respecto a los artefactos, los más revisados son claves para el proceso de desarrollo, de alto riesgo para el éxito del proyecto y de alta visibilidad de cara al usuario. Estos son las especificaciones de requerimientos y documentos de diseño, el código y la interfaz de usuario respectivamente. Dado que el uso de revisiones no formales es amplio, se pueden buscar mejoras en la calidad de dichos procesos adoptando prácticas de técnicas más formales como el uso de técnicas de lectura y herramientas de apoyo a través de la capacitación permanente.

1 Introducción

La detección de defectos en el desarrollo de software se realiza mediante un conjunto de acciones que pueden clasificarse como técnicas dinámicas y estáticas. Las técnicas dinámicas (como el *testing*) basan la evaluación de un sistema o componente en el comportamiento exhibido durante su ejecución. Para ello se introducen valores de entrada comparando la salida con los resultados esperados. En contraposición, las técnicas estáticas son aquellas que evalúan un sistema o componente basado en su forma, estructura, contenido o documentación. Esto significa que no requieren de la ejecución del artefacto para realizar su análisis [1]. Un ejemplo de técnica estática son las revisiones, de las cuales existen varios tipos dependiendo de su objetivo y formalidad.

La elección de qué familia de técnicas utilizar (estática o dinámica) así como la elección de las técnicas dentro de cada familia depende de diversos factores. Por ejemplo, dependen del artefacto sobre el que se quiere trabajar, quién realiza la detección y cómo se realiza la misma. También dependen de la etapa del proceso de desarrollo en que se genere el artefacto. Por ejemplo, las técnicas estáticas son más adecuadas para su uso en documentos de requerimientos y diseño (etapas tempranas del ciclo de desarrollo) mientras que para el código resulta más adecuado el *testing* [2].

Las técnicas de revisión estática, en particular, son claves para mejorar la calidad del software debido a los siguientes motivos:

- Se afirma que las distintas técnicas de revisión de software permiten la detección y corrección de defectos con una eficacia de entre un 50 y 90 por ciento [3].
- Son técnicas aplicables a cualquier artefacto generado durante el proceso de desarrollo de software [4], en particular sobre documentos de especificación de requerimientos y diseño [2].
- Permiten la temprana detección y remoción de defectos [5] [6] [7].
- El costo asociado a la detección y corrección de defectos en etapas tardías disminuye, al igual que el retrabajo [8] [9], ya que los defectos típicamente son encontrados ni bien son introducidos al artefacto revisado [10]. Esto sucede gracias a que el proceso que genera el defecto y el que lo detecta están en un mismo nivel de abstracción [2].

A pesar de los beneficios mencionados, parece existir un problema de adopción que se da cuanto más formal es la técnica. Este hecho pudo comprobarse en una investigación anterior sobre inspecciones de software (la más formal de las revisiones) identificándose además factores causantes del problema [11]. Estos resultados motivaron este nuevo estudio ya que no resulta claro cómo están revisando sus artefactos de software aquellas empresas que no realizan inspecciones.

El objetivo de este estudio es reportar los resultados de un diagnóstico acerca del uso de técnicas de revisión por parte de profesionales de la industria del software en Uruguay. Dicho diagnóstico se realizó a través de una encuesta *online* y permitió obtener resultados que describen la situación de la industria del software uruguayo sobre revisiones de software.

El resto del documento se estructura de la siguiente forma: la sección 2 plantea definiciones de técnicas de revisión y trabajos previos; en la sección 3 se explica el método de investigación y los detalles de preparación y ejecución de encuesta realizada; la sección 4 muestra los hallazgos de la investigación, (tanto resultados cuantitativos como cualitativos); en la sección 5 se discuten los hallazgos y se analizan las implicancias de los mismos en relación con los objetivos generales del trabajo; por último, la sección 6 resume las conclusiones a las que se llegó durante la realización de este trabajo.

2 Estado de la cuestión

La IEEE define como revisión al proceso o reunión mediante la cual un producto de software o un proceso es mostrado a todos los interesados para inspeccionarlo, realizar comentarios o aprobaciones [12]. Ciolkowski *et al.* definen la revisión como un enfoque para examinar productos de software en busca de defectos, desvíos respecto a estándares establecidos y otros problemas [3]. Para realizar revisiones existen distintas técnicas que se pueden clasificar según el grado de formalidad con las que se lleven a cabo. Se entiende como formal un proceso cuyas etapas se encuentran definidas, tiene roles y responsabilidades establecidas [13], con una agenda específica [14] y del cual puede medirse su capacidad para así mejorarlo.

Wieggers define una lista de técnicas [15] en orden creciente de formalidad y sugiere que una empresa impulsada por la generación de productos de calidad debe practicar una

variedad de técnicas de revisión de acuerdo a un amplio espectro de formalidad, rigor, eficacia y costo [16]. La lista consta de las siguientes técnicas: *Ad-hoc*, Revisión de pares, Revisión de pares múltiple, Programación de a pares, Presentación, Revisión en equipo e inspección de software.

2.1 Técnicas de revisión

Ad-Hoc: es una forma rápida de obtener otra perspectiva para encontrar problemas que el autor del artefacto no puede ver por sí mismo. La técnica no sigue un proceso establecido y sus resultados no se registran de una manera predefinida.

Revisión de pares (*Peer Deskcheck*): involucra a una sola persona (distinta al autor del artefacto) quien recibe una copia del artefacto para su revisión. El revisor puede utilizar *checklists* u otras técnicas de análisis que incrementen la efectividad del proceso.

Revisión de pares múltiple (*Passaround*): es una revisión de pares realizada concurrentemente. El artefacto a revisar es distribuido entre 3 y 15 personas quienes revisan y dan su devolución por separado. Una variante es compartir un documento donde los revisores realicen sus comentarios para minimizar la redundancia y mostrar diferentes interpretaciones de lo revisado.

Programación de a pares (*Pair Programming*): esta técnica consiste en dos desarrolladores trabajando en un mismo producto simultáneamente, en una sola computadora, compartiendo un teclado y un monitor [15]. Uno hace de controlador y es el que efectivamente programa; el otro es el navegador que observa el trabajo del controlador e identifica errores [17].

Presentación (*Walkthrough*): es una técnica donde el diseñador o desarrollador guía a los miembros de un equipo a través del artefacto a revisar. Los participantes hacen preguntas o comentarios sobre posibles anomalías, violación de estándares u otros problemas [12]. El autor toma un rol dominante donde el objetivo es satisfacer sus necesidades en lugar de los objetivos de calidad del equipo [15]. Típicamente no se sigue ningún procedimiento definido, no requiere reportes de gerencia y no genera métricas.

Revisión en equipo (*Team Review*): Comienza con una planificación seguida de una de preparación donde los participantes reciben el material necesario varios días antes. Luego el equipo (con roles definidos) se reúne para discutir los resultados donde participan el líder de revisión (puede ser el propio autor del artefacto revisado), registrador y moderador (no puede ser del propio autor). La última etapa es la corrección de los defectos encontrados. Existe un cierto paralelismo entre la Revisión en equipo de Wieggers y la Revisión técnica planteada en el estándar IEEE 1028-2008 [12]. Ambos métodos tienen en común la idea de un rol de líder de revisión o moderador y de un registrador. También ambas técnicas incluyen una etapa de reunión y la necesidad de una etapa de preparación previa a la misma.

Inspecciones es software: Es la principal actividad de evaluación estática [18] ya que es la más formal, sistemática y rigurosa [15] en cuanto a sus procedimientos. Una inspección de software tradicional (método de Fagan) consta de siete etapas y roles bien definidos. Estos equipos son formados por 4 o 5 compañeros de trabajo, no permitiendo la

participación de puestos gerenciales por condicionar los resultados [19]. Cada inspector dedica entre una y cuatro horas a la revisión del producto de trabajo. Luego, en reuniones de 2 horas máximo, quién tiene el rol de lector presenta el material poco a poco al resto del equipo quiénes señalan posibles defectos. El lector ayuda al equipo a llegar a la misma interpretación de cada parte del producto analizado y la reunión termina decidiendo cómo verificar los cambios que el autor deberá realizar en la etapa de retrabajo [19].

2.2 Uso de técnicas y antecedentes

Es comúnmente aceptado que la inspección de software es una técnica de gran utilidad a la hora mejorar la calidad del software ya que produce resultados más repetibles [6]. Laitenberger dice que permite la detección de defectos ni bien el artefacto es creado [20] y otros afirman que es uno de los métodos más efectivos de asegurar la calidad [21]. De todas formas y a pesar del gran número de reportes positivos sobre esta técnica, se afirma que su adopción por parte de la industria es muy baja [22] [3]. La existencia de evidencia respecto al problema planteado y factores causantes del mismo [11] despiertan la duda acerca de qué técnicas de revisión menos formales están siendo usadas en lugar de la inspección de software.

En la literatura se encuentran pocos trabajos que incluyan estudios de técnicas de revisión menos formales como la *ad-hoc*, la de revisión de pares, la de revisión de pares múltiple y la programación de a pares (en adelante todas serán denominadas técnicas *peer review*). Sobre estas técnicas se desconoce el grado de penetración y la extensión que tiene su uso en la industria [3]. Por este motivo en 2002 la red ISERN (*Software Engineering Research Network*) y el instituto Fraunhofer deciden llevar a cabo un trabajo en conjunto que consiste en una encuesta *online* para evaluar el estado de la práctica de revisiones de software. Esta encuesta se lleva a cabo en dos etapas; la primera en el contexto del VISEK que es un proyecto alemán orientado a la creación de un centro virtual de métodos de Ingeniería de Software [23] y la segunda ya a nivel mundial con la colaboración del ISERN [24].

Los resultados indican que, quiénes respondieron la encuesta, realizan revisiones de sus artefactos de software regularmente pero que dichas revisiones no son sistemáticas. Esta afirmación se realiza explícitamente a través de la respuesta a una de las preguntas e implícitamente ya que los procesos de revisión descritos no mencionan la etapa de preparación la cual es considerada como crucial en cualquier técnica formal [3]. Los datos finales de este estudio indican que la técnica utilizada con mayor frecuencia (50% de quienes respondieron) es la denominada *peer review*. La siguiente técnica más utilizada es la inspección de software, practicada por el 21% de los encuestados. La técnica menos utilizada es la de *walkthrough* con un 16,5% [23].

3 Método de investigación – Encuesta

Esta sección describe el proceso seguido para elaborar y poner en funcionamiento la encuesta llevada cabo en este trabajo. Los pasos que componen dicho proceso se pueden ver en la **Fig. 1** y son: (1) Definición y objetivos – se establece cual es la finalidad de la encuesta; (2) Diseño – definición de la muestra, modelo, formulario y validación; (3) Implementación y ejecución – se pone en marcha el proceso de recabar datos; (4)

Procesamiento de respuestas – se realiza el tratamiento de los datos y la obtención de conclusiones.

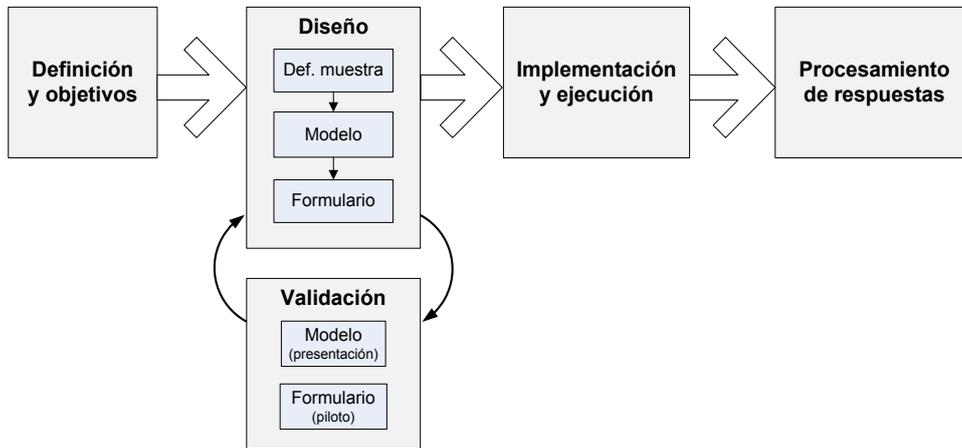


Fig. 1. Esquema del proceso de elaboración y ejecución de la encuesta.

El proceso de investigación fue elaborado en base a los trabajos de Kitchenham y Pfleeger (*Principles of Survey Research*) [25] y lo reportado por Ciolkowski *et al.* como un enfoque sistemático y disciplinado para realizar encuestas en el área de la Ingeniería de Software [24]. Por otra parte la encuesta realizada en este trabajo es del tipo descriptiva y exploratoria según la clasificación por objetivos de Wholin [26]. Las encuestas descriptivas son las que buscan realizar afirmaciones acerca de la distribución de cierta característica o atributo en una población sin llegar a explicar el porqué de la misma. Las exploratorias son aquellas que se realizan de forma previa a un estudio más profundo para asegurarse que ningún aspecto importante haya sido pasado por alto.

3.1 Definición y objetivos

El objetivo de la encuesta es conocer si los profesionales de la industria del software en Uruguay realizan revisiones, qué artefactos de software son revisados y que técnicas utilizan.

3.2 Diseño

Definición de la muestra

Tomando como base el objetivo planteado se decide dirigir la encuesta a profesionales en desarrollo de software, vinculados a actividades de aseguramiento de calidad y/o definición de procesos en Uruguay. Este grupo de profesionales es tomado como la población objetivo del estudio. Debido a la imposibilidad de hacerles llegar la encuesta a todos los integrantes de este grupo se decide definir una muestra que involucre a un subconjunto representativo de dichos profesionales [27] [24] [28].

El primer grupo de profesionales representativos (grupo 1) se determina a partir de una selección manual de profesionales, a los que tienen acceso los autores, con el perfil

definido anteriormente. A este grupo se le suma luego un comunicado enviado a la Cámara Uruguaya de Tecnologías de la Información (CUTI) solicitando la participación de aquellos profesionales que posean el perfil descrito en las empresas asociadas (grupo 2). Por último se publica un llamado a participar de la encuesta en el boletín de graduados de Universidad ORT Uruguay (grupo 3). Debido a la alta probabilidad de que los tres grupos contactados tengan profesionales en común, los comunicados instan a aquellos que ya hayan completado la encuesta a ignorar la solicitud de participación. Finalmente, y como forma de lograr un mayor alcance, se solicita a quienes reciben la invitación que reenvíen el comunicado a aquellos profesionales uruguayos que coincidan con el perfil, emulando la técnica de *snowballing* [24].

Se considera que la muestra definida tiene un buen alcance (sin tener en cuenta los posibles referidos) ya que involucra directamente a 63 profesionales (del primer grupo) quienes pertenecen a 43 empresas del sector. Además, el comunicado enviado a CUTI llega a 256 empresas cuyos profesionales seguramente pertenezcan, en parte, al conjunto alcanzado por el tercer comunicado del boletín de egresados (2780 graduados).

Dado que la participación en la encuesta es voluntaria, se cree que la muestra tenga cierto sesgo. Dicho sesgo está causado por la predominancia de respuestas de profesionales que contestan la encuesta por utilizar prácticas de revisión o por encontrarse interesados en temas de calidad. De todas formas, como el objetivo es conocer cómo se revisan ciertos artefactos y con qué técnicas, la posible parcialidad de las respuestas no afecta a las conclusiones que se obtengan.

Elaboración del modelo conceptual

A través de la construcción de un modelo conceptual se busca describir los objetos a investigar, las variables involucradas en el estudio y la relación entre ellas [24]. En este caso también se realiza como ejercicio para refinar los objetivos generales planteados y generar preguntas concretas y sin ambigüedades. El modelo se elabora utilizando la técnica *Goal, Question, Metric* (GQM) [29] y el resultado puede verse en la **Tabla 1**. Dado que la encuesta es en parte descriptiva, algunas preguntas son precodificadas lo que facilita su posterior análisis. Las preguntas correspondientes a la parte exploratoria son abiertas para capturar respuestas no previstas o con mayor detalle [24].

Construcción del formulario

El formulario se construyó directamente con las preguntas obtenidas de la creación del modelo. A los efectos de que los encuestados no tengan dudas acerca de la definición del proceso de “Revisión estática” se comienza la encuesta dando una definición del mismo: *“Se define como Revisión Estática al proceso mediante el cual se realiza un análisis que no implica la ejecución del artefacto inspeccionado (distinto al testing). Dicho proceso es aplicable a cualquier artefacto del ciclo de desarrollo en busca de identificar anomalías o desvíos respecto a estándares.”*

La primera pregunta sin embargo busca determinar si se utiliza el proceso de inspección de software específicamente, por lo que se realiza un agregado a la definición anterior dentro de la propia pregunta. Se define inspección de software como: *“un tipo de revisión estática según lo definido anteriormente pero siguiendo procedimientos formales con roles y fases bien definidas”*. Las siguientes preguntas sí se enfocan en el uso de otras técnicas (menos formales) para revisar artefactos de software.

Tabla 1. Modelo del cual se obtienen las preguntas para la construcción del formulario.

G	Determinar el número de encuestados que realiza inspecciones según la definición anterior.
Q	1) En su actividad profesional ¿utiliza el método Inspección de Software? <input type="checkbox"/> Sí <input type="checkbox"/> No
M	- % de empresas que respondieron sí.
G	Determinar si se realiza algún tipo de análisis que no implique la ejecución del artefacto inspeccionado (distinto al testing).
Q	2) ¿Realizan algún tipo de análisis que no implique la ejecución del artefacto inspeccionado (distinto al testing) sobre alguno de los artefactos generados durante el ciclo de desarrollo de software? <input type="checkbox"/> Sí <input type="checkbox"/> No
M	- % de empresas que respondieron sí.
G	Conocer que artefactos son revisados. Descubrir otros artefactos (quizás menos convencionales) donde se apliquen revisiones estáticas.
Q	3) ¿Cuál/cuáles de los siguientes artefactos del proceso de desarrollo son revisados? <input type="checkbox"/> Especificación de requerimientos <input type="checkbox"/> Documentos de diseño <input type="checkbox"/> Código <input type="checkbox"/> Casos de prueba <input type="checkbox"/> Interfaz de usuario <input type="checkbox"/> Documentos de arquitectura <input type="checkbox"/> Otro _____
M	- # de respuestas por artefacto. - Codificación de respuesta Otro. - Lista de artefactos revisados.
G	Determinar qué número de empresas tiene definido un proceso de revisión. Conocer de qué forma se revisa cada artefacto.
Q	4) ¿Tienen un proceso definido/establecido de inspección/revisión de algún artefacto? <input type="checkbox"/> Sí <input type="checkbox"/> No Describa brevemente el proceso de revisión (aunque no esté definido) para cada artefacto.
M	- % empresas que respondan que sí. - Codificación de la descripción del procedimiento de revisión.
G	Conocer si se asigna tiempo a actividades de inspección/revisión al momento de planificar un proyecto o por el contrario surgen de la necesidad de asegurar su calidad una vez terminado el artefacto.
Q	5) ¿La revisión de un artefacto surge de la necesidad de asegurar su calidad al momento de terminar el mismo (de forma <i>ad hoc</i>, espontánea y sin planificación) o en la gestión del proyecto se reserva tiempo para inspeccionar cada artefacto según la instancia del proyecto en que se encuentre? <input type="checkbox"/> Surge de forma espontánea y sin planificación, simplemente por necesidad de asegurar la calidad del artefacto terminado. <input type="checkbox"/> Se planifica en qué momento del proyecto se inspeccionará qué artefacto. <input type="checkbox"/> Otro _____
M	- % de empresas que sí asignan tiempo. - % de empresas que revisan ni bien se finalizó el artefacto.
G	Determinar si existen roles específicos de inspección o si la revisión la hace un par. Determinar quiénes se encuentran involucrados en las inspecciones/revisiones.
Q	6) Indique si existen roles o equipos especializados en inspeccionar/revisar. <input type="checkbox"/> Sí <input type="checkbox"/> No Explique quiénes participan en la revisión/inspección de cada artefacto.
M	- % de empresas con roles o equipos revisores. - Lista de personas involucradas en la revisión de cada artefacto.
G	Conocer qué se hace con la información generada producto de estas inspecciones.
Q	7) ¿De qué manera comunica el resultado de la inspección/revisión? <input type="checkbox"/> Reuniones (todos los integrantes del equipo o involucrados) <input type="checkbox"/> Feedback directo <input type="checkbox"/> Repositorio (wiki, estándares, checklists, etc.) <input type="checkbox"/> Otro _____
M	- % de respuestas por medio de comunicación. - Codificación de respuestas "Otro".

Validación de la encuesta

Antes de realizar los comunicados a los distintos grupos que componen la muestra y comenzar con la ejecución de la encuesta se decide realizar un trabajo de validación. La primera parte de esta validación se da como una presentación, tanto del modelo como del formulario final, al grupo de estudiantes de la Maestría en Ingeniería de Universidad ORT Uruguay en un seminario interno. Cabe destacar que muchos de estos estudiantes realizan actividades relacionadas con el aseguramiento de la calidad a nivel industrial. De esta presentación se obtiene un primer *feedback* que se incorpora tanto al modelo como al formulario. En particular se decide agregar a la encuesta preguntas para validar resultados mediante la técnica de triangulación [30], la cual consiste en cruzar información de distintas preguntas para chequear la consistencia entre las respuestas.

La segunda validación se realiza a través de una encuesta piloto. Su objetivo es chequear que las preguntas se encuentran redactadas correctamente, que son claras y que permiten obtener respuestas correctas y completas acerca de lo que motiva la pregunta (modelo). El piloto se realiza encuestando a equipos del área de *Software Factory* de la Universidad ORT Uruguay que se encuentran desarrollando el proyecto de fin de carrera. El resultado de esta encuesta se consolida en un documento donde se realiza el análisis de los datos correspondiente y llegando incluso a esbozarse algunas conclusiones. Dicho documento es enviado a los participantes del piloto y a algunos docentes de la universidad.

3.3 Implementación y ejecución

Para llevar a cabo la encuesta *online* se decide utilizar la herramienta de formularios de Google Docs. Dicha herramienta es seleccionada por facilitar la difusión de la encuesta y por la capacidad de generar reportes preliminares a demanda. La herramienta dio buenos resultados en el piloto por lo que se decide proseguir con su uso. Además de crear el formulario de carga, se elabora el comunicado donde se deja claro a quién va dirigida la encuesta, para que se lleva a cabo la misma y como pueden hacer los profesionales para difundir la invitación. Se utiliza un servicio de urls cortas para generar distintos *links* y llevar registros de la cantidad de *clicks* que recibe cada *link*. De esta manera se puede medir la efectividad de los comunicados en cada etapa de del proceso de difusión de la encuesta (Tabla 2).

Tabla 2. Envío de comunicados y recordatorios por grupo

Grupo	Fecha	Recordatorios	Comentarios
Grupo 1	17-21/06/2012	01/07/2012 10/07/2012	Enviado en forma masiva a la mayoría y en forma individual a ciertos profesionales para lograr mayor atención y difusión.
Grupo 2	04/08/2012	16/08/2012	Comunicado replanteado para hacer algunos términos más accesibles a profesionales. En este ya no se insta a difundir la encuesta a otros profesionales.
Grupo 3	08/08/2012	N/A	La Coordinación de Graduados envía el comunicado, el cual es modificado siguiendo ciertas pautas del boletín.

3.4 Procesamiento de las respuestas

Por las características de la encuesta (descriptiva y exploratoria) el análisis de los datos se realiza básicamente a través de estadísticos descriptivos simples y de la agregación de datos [24]. Las preguntas sobre qué artefactos se revisan, si se realiza o no revisiones, etc., son cuantitativas (representadas de forma discreta) y son de más fácil análisis ya que existe una gran variedad de herramientas (muchas de ellas informáticas) que automatizan la tarea.

De todas formas, durante el análisis de las respuestas del piloto surgió la necesidad de procesar de otra manera preguntas abiertas como, por ejemplo, la que pide la descripción el proceso de revisión de cada artefacto. El problema con las preguntas abiertas es que requieren otro tipo de análisis dada su naturaleza cualitativa (información expresada en palabras o imágenes) y generalmente requiere un mayor esfuerzo. Si bien para el análisis cualitativo también existen algunas herramientas informáticas, para en este estudio se utiliza un método manual denominado de Comparaciones Constantes [30].

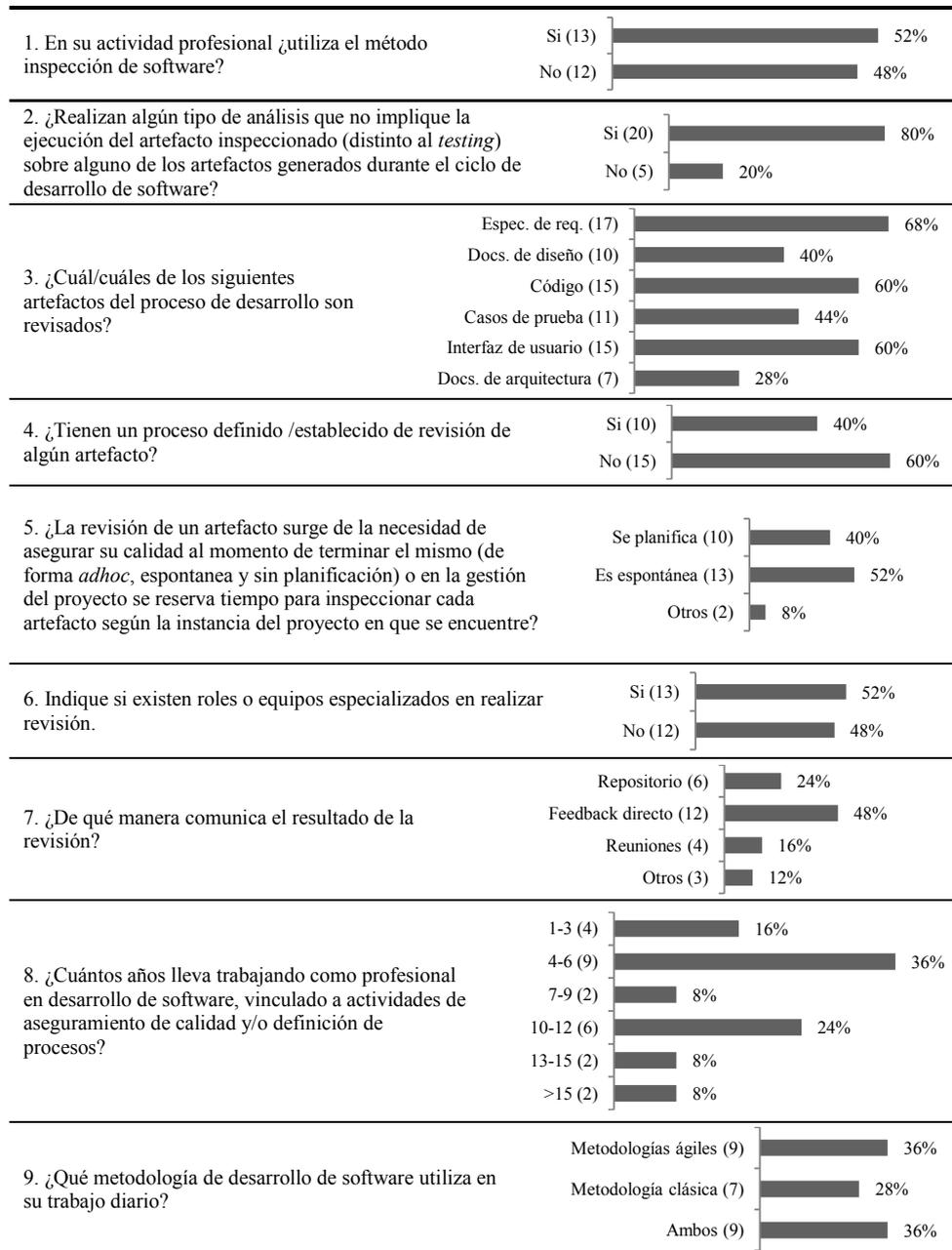
El método de Comparaciones Constantes comienza con el etiquetado de las cadenas de texto de cada respuesta según el tema de interés del estudio. Luego dichos textos se agrupan según las etiquetas correspondientes de forma de ver si aparecen patrones que generen conocimiento. El último paso es escribir una hipótesis preliminar o proposición en base a los patrones encontrados. Como puede apreciarse más allá del proceso de codificar y agrupar, esta es una actividad principalmente creativa, por no decir puramente subjetiva y que requiere un fuerte apoyo en datos cuantitativos que respalden las proposiciones generadas [30].

En este trabajo, las respuestas que describen el proceso de revisión usado para cada artefacto y los roles que participan de dichas revisiones fueron procesadas utilizando el método de Comparaciones Constantes. Ambas preguntas están muy relacionadas (se espera que aparezcan roles mencionados en las descripciones de los procesos) por lo que las respuestas de ambas se tomaron en conjunto y se procesaron al mismo tiempo para no contabilizar más de una vez la aparición de cada rol. Como ejemplo del proceso de codificación se puede tomar la respuesta “*Código: revisión entre pares cotejando contra los Coding Standards definidos en la Wiki*” que describe el proceso que sigue un profesional para revisar el código. El proceso de etiquetado asigna la etiqueta *código* a toda la frase para identificar el artefacto, la etiqueta *un par* a la referencia *entre pares* para indicar quiénes son los revisores y la etiqueta *herramientas* a *Coding Standards* para indicar cómo se realiza. Luego, extrayendo todas las frases completas etiquetadas con *código* se puede inferir cómo se realizan las revisiones sobre este artefacto. Si por el contrario se extraen las etiquetadas como *un par* sabremos que artefactos son revisados por pares.

4 Resultados

En total se obtuvieron respuestas de 25 profesionales de la industria del software uruguayo. La **Tabla 3** muestra las respuestas de cada pregunta usando estadísticos descriptivos simples (frecuencia), donde los valores absolutos y relativos se pueden ver en forma de gráfica.

Tabla 3. Resultados obtenidos de procesar las respuestas



La pregunta número 4 se complementa con una pregunta adicional que busca que cada encuestado describa el proceso de revisión que utiliza para cada artefacto en su actividad diaria. Del análisis de dichas respuestas se elaboró una lista de proposiciones obtenidas con el método de Comparaciones Constantes [30] (Tabla 4). Esta tabla ordena las

proposiciones descendientemente por frecuencia de aparición y el valor numérico es la cantidad de veces que la etiqueta correspondiente a cada proposición aparece en las respuestas de los encuestados.

Tabla 4. Proposiciones que surgen de la descripción de los procesos de revisión usados.

Id	Proposiciones	#
H1	Las revisiones en general involucran a un par	11
H2	Generalmente se asignan tareas de revisión a roles ya existentes	10
H3	La UI se valida contra la especificación de requerimientos y se observa al usuario usarla	7
H4	En general se hacen revisiones sin un proceso definido	6
H5	El código se revisa contra estándares de desarrollo	5
H6	Se revisa especificación de requerimientos para descartar ambigüedades y se valida con el cliente	5
H7	Se usan herramientas de apoyo al proceso de revisión	5
H8	Cliente y desarrolladores participan en revisiones de etapas tempranas del proyecto	3
H9	Existen procesos novedosos de revisión	3
H10	Se mencionan pocas intervenciones externas en procesos de revisión	3
H11	Los documentos de arquitectura se validan con pares y contra objetivos de arquitectura definidos	2
H12	Los documentos de diseño se revisan de alguna forma	2
H13	Los casos de prueba se revisan de alguna forma	1
H14	La técnica de revisión depende de la complejidad y el costo-beneficio ante fallas	1
H15	La definición de revisión estática no fue bien comprendida	1

5 Discusión

El primer resultado obtenido es que ninguno de los profesionales que respondió la encuesta utiliza la técnica formal de inspección de software. Esta afirmación parece contradecir lo afirmado en los resultados, ya que el 52% de los encuestados dicen realizar inspecciones. La explicación surge al cruzar las respuestas de quiénes dicen realizar inspecciones (pregunta 1) y quiénes indican tener un proceso definido de revisión (pregunta 4). El valor obtenido de este cruce debería ser igual, sin embargo solo el 40% de encuestados dicen tener un proceso establecido de revisión. Por otra parte, la descripción del proceso utilizado y los roles involucrados en la revisión de cada artefacto (pregunta 6) no indican un proceso definido en fases y con roles de inspección específicos. Esto lleva a concluir que ninguna de las descripciones dadas se corresponde a un proceso de inspección como el definido en la encuesta y por lo tanto ninguno de los encuestados realiza inspecciones de software.

Un motivo por el que se da esta contradicción es que los profesionales tienen la auto-percepción de que sus procesos son más formales de lo que realmente son. Un ejemplo de esto puede apreciarse en las descripciones de procesos dadas por aquellos que dicen realizar inspecciones de software. Dichos profesionales consideran formales sus procesos ya que incluso cuentan con roles y fases definidas pero si se comparan con las

definiciones de la sección 2, coinciden con técnicas de revisión menos formales. Por ejemplo, la descripción del proceso de revisión de código de un profesional que dijo realizar inspecciones de software es “*Código: revisión entre pares cotejando contra los Coding Standards definidos en la Wiki.*”. Este proceso tiene definido un estándar contra el cual comparar los hallazgos, un repositorio (no se aclara si éste se actualiza luego de cada revisión) y es llevado a cabo por pares indicando al menos un rol de revisor. Si bien es un proceso definido, está muy lejos de ser una inspección formal de software. Este problema puede estar potenciándose por las diferencias en la terminología utilizada al referirse a inspecciones de software y el mal uso de sinónimos reportado por Aurum *et al.* [4].

El segundo resultado obtenido es la confirmación de que los artefactos generados durante el desarrollo de software están siendo revisados de alguna manera, sobre todo los correspondientes a etapas tempranas del ciclo de vida. El 80% de los encuestados respondió que revisa alguno de los artefactos mencionados (pregunta 2). Esto sugiere que hay un 20% que confía únicamente en el *testing* como medio de detección de defectos o ni siquiera realiza *testing*. Los artefactos revisados (pregunta 3) son principalmente la especificación de requerimientos, documentos de diseño y código. Esto resulta lógico desde el punto de vista de reducción de costos por el beneficio de la temprana detección de defectos [31] siguiendo la tendencia de lo reportado a nivel mundial [24]. Además resulta interesante destacar que otro artefacto reportado como muy revisado (60%) es la interfaz de usuario. El conocimiento de qué artefactos se revisan permite visualizar el espacio de oportunidad que existe para trabajar en la adopción de técnicas de revisión, técnicas de lectura y herramientas.

Por otra parte, de las proposiciones de la **Tabla 4** se infiere que se realizan revisiones frecuentemente sobre diversos artefactos de software pero que falta profundizar en el uso de técnicas y sistematizar el proceso. Un ejemplo de esto es que no se menciona el uso de técnicas de lectura (sobre todo en artefactos como la especificación de requerimientos) lo que mejoraría la calidad del proceso de revisión. Otro ejemplo es la asignación de tareas de revisión a roles ya existentes, sobrecargando la actividad de esos roles lo que podría tener consecuencias inesperadas. Estas inferencias basadas en preguntas abiertas son apoyadas por ciertas respuestas cuantitativas. Por ejemplo, un indicador de falta de sistematización es que las revisiones surgen espontáneamente y sus resultados son devueltos directamente al autor sin dejar registros de la actividad realizada. En estos casos el uso de herramientas y técnicas de apoyo a la revisión resultarían beneficiosos para mejorar la calidad del proceso y en este sentido algunos profesionales mencionan su uso.

Un último resultado es que los profesionales encuestados realizan revisiones *ad-hoc*, revisiones de pares o de pares múltiple y en muy pocos casos revisiones en equipo (estas últimas con muy pocas de las características descritas por Wiegers [15]). Esto se infiere de las descripciones de procesos de revisión realizadas para cada artefacto. Debido a que el uso de revisiones no formales resulta ser amplio, se puede pensar en mejorar la calidad de dichos procesos adoptando prácticas de técnicas de revisión más formales. Se podría comenzar incentivando el uso de *checklists* y técnicas de lectura para hacer de la revisión un proceso más sistemático, más predecible y no tan dependiente del profesional que lo realiza. De esta manera, quiénes revisan espontáneamente podrían aprovechar mejor el tiempo del revisor y no alejarlo tanto de su actividad planificada. Por otra parte, quiénes planifican las revisiones podrían conocer *a priori* el tiempo y costo de dichas actividades de forma de ajustarlas al calendario del proyecto.

6 Conclusiones

La revisión de artefactos de software es una técnica estática de aseguramiento de la calidad aplicable a cualquier artefacto y en cualquier momento, con una alta tasa de eficacia en la remoción de defectos. Existen técnicas con distintos niveles de formalidad, aunque un estudio anterior a éste indica que la más formal de las técnicas (la inspección de software) tiene una baja adopción en la industria. Respecto a técnicas menos formales de revisión existen pocos antecedentes, destacándose la categorización de técnicas realizada por Wiegers y el diagnóstico de Ciolkowski *et al.* acerca del uso de dichas técnicas. El presente estudio realizó un diagnóstico a través de una encuesta para conocer qué técnicas de revisión están siendo utilizadas en la industria uruguaya del software.

Los resultados de la encuesta indican que un 80% de quienes la contestaron utilizan técnicas de revisión menos formales en los artefactos generados durante el desarrollo de software. Por lo tanto existe un 20% que confían únicamente en el *testing* para la detección de defectos o no realizan ninguna actividad de evaluación. Este dato muestra que todavía existe espacio para trabajar en la adopción de procesos de revisión y evaluación en general. Respecto a los artefactos revisados, predominan los correspondientes a etapas tempranas del desarrollo como la especificación de requerimientos y documentos de diseño, aunque hay un número importante de referencias a revisiones de código e interfaz de usuario.

De las descripciones de los procesos de revisión utilizados para cada artefacto se concluye que los profesionales que respondieron la encuesta tienen una auto-percepción de que sus procesos son más formales de lo que realmente son. También se obtuvo una serie de proposiciones que deberán ser probadas en futuros trabajos empíricos pero que indican ciertas tendencias en el uso de revisiones por parte de estos profesionales. Por ejemplo se destaca el uso de revisiones de pares, la asignación de tareas de revisión a roles ya existentes y el uso de documentos de referencia (especificación de requerimientos y estándares) contra los que validar los artefactos.

Las respuestas dadas en general indican que si bien se realizan revisiones frecuentemente, falta profundizar en el uso de técnicas y sistematizar los procesos. Como trabajo futuro se podrían plantear actividades empíricas donde se incentive el uso de técnicas de lectura sobre ciertos artefactos y herramientas de apoyo. Dichas actividades permitirían probar que el uso de ciertas prácticas pertenecientes a técnicas de revisión más formales pueden incrementar la calidad las revisiones menos formales haciéndolas más predecibles y no tan dependientes del profesional revisor. La adopción de nuevas técnicas y herramientas por parte de la industria puede acortar las diferencias con la academia de modo que futuros trabajos en conjunto se lleven a cabo con la uniformidad de criterios necesaria para el beneficio de ambas partes.

7 Referencias

- [1] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," vol. 121990, 1990.
- [2] P. Runeson, C. Andersson, T. Thelin, A. Andrews, and T. Berling, "What Do We Know about Defect Detection Methods?," *IEEE Software*, vol. 23, no. 3, pp. 82–90, 2006.
- [3] M. Ciolkowski, O. Laitenberger, and S. Biffel, "Software reviews: The state of the practice," *IEEE software*, vol. 20, no. 6, pp. 46–51, 2003.

- [4] A. Aurum, H. Petersson, and C. Wohlin, "State-of-the-art: software inspections after 25 years," *Software Testing, Verification and Reliability*, vol. 12, no. 3, pp. 133–154, Sep. 2002.
- [5] O. Laitenberger, "Cost-effective Detection of Software Defects through Perspective-based Inspections," Fraunhofer IESE / University of Kaiserslautern, 2000.
- [6] M. Fagan, "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, vol. 15, no. 3, pp. 182–211, 1976.
- [7] D. Mishra and A. Mishra, "Simplified software inspection process in compliance with international standards," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 763–771, Jun. 2009.
- [8] D. Winkler and S. Biffl, "An Empirical Study on Design Quality Improvement from Best-Practice Inspection and Pair Programming," in *Product-Focused Software Process Improvement*, 2006, vol. 4034, pp. 319 – 333.
- [9] D. O. Neill, "Determining Return on Investment Using Software Inspections," *CrossTalk - The Journal of Defense Software Engineering*, 2003.
- [10] O. Laitenberger, T. Beil, and T. Schwinn, "An industrial case study to examine a non-traditional inspection implementation for requirements specifications," *Empirical Software Engineering*, vol. 7, no. 4, pp. 319–343, 2002.
- [11] D. Macchi and M. Solari, "Mapeo Sistemático de la Literatura sobre la Adopción de Inspecciones de Software," in *Conferencia Latinoamericana de Informática (CLEI 2012)*, 2012.
- [12] IEEE, "IEEE Standard 1028-2008 for Software Reviews and Audits," 2008.
- [13] S. Rakitin, *Software verification and validation for practitioners and managers*, Second edi. Artech House, 2001.
- [14] J. Juran and B. Godfrey, *Juran's quality handbook*, 5th ed. McGraw-Hill, 1999, p. 1872.
- [15] K. Wiegers, "When Two Eyes Aren't Enough," *Software Development*, vol. 9, no. 10, 2001.
- [16] K. Wiegers, "Seven Truths About Peer Reviews," *Cutter IT Journal*, vol. 15, no. 7, 2002.
- [17] L. Williams and R. Kessler, *Pair programming illuminated*. Boston: Addison-Wesley Longman Publishing Co., 2003.
- [18] A. Endres and H. Rombach, *A handbook of software and systems engineering: Empirical observations, laws, and theories*. Pearson Addison-Wesley, 2003.
- [19] B. Brykczynski, R. Meeson, and D. Wheeler, "Software Inspection: Eliminating Software Defects," in *In Proceedings of the Sixth Annual Software Technology Conference*, 1994, p. 12.
- [20] O. Laitenberger, "A survey of software inspection technologies," in *Handbook on Software Engineering and Knowledge Engineering*, vol. 2, 2002, pp. 517–556.
- [21] O. Laitenberger and J.-M. DeBaud, "An Encompassing Life-Cycle Centric Survey of Software Inspection," *Journal of Systems and Software*, vol. 50, no. 1, pp. 5–31, 2000.
- [22] R. Radice, *High Quality Low Cost Software Inspections*. Paradoxicon Publishing, 2001, p. 479.
- [23] O. Laitenberger, S. Vegas, and M. Ciolkowski, "The state of the practice of review and inspection technologies in germany," 2002.
- [24] M. Ciolkowski, O. Laitenberger, S. Vegas, and S. Biffl, "Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering," *ESERNET*, vol. 2765, pp. 104–128, 2003.
- [25] B. Kitchenham and S. L. Pfleeger, "Principles of Survey Research - Part 1-6," *Software Engineering Notes*.
- [26] C. Wohlin, M. Höst, and K. Henningsson, "Empirical Research Methods in Software Engineering," *ESERNET*, vol. 2765, pp. 7–23, 2003.
- [27] Statistics Canada, *Survey Methods and Practices*, no. 12. Social Survey Methods Division, 2003, p. 396.
- [28] B. Kitchenham and S. L. Pfleeger, "Principles of Survey Research - Part 2: Designing a Survey," vol. 27, no. 1, pp. 18–20, 2002.
- [29] V. R. Basili and D. Weiss, "A methodology for collecting valid software engineering data," *IEEE Computer Society Trans. Software Engineering*, vol. 10, no. 6, pp. 728–738, 1984.
- [30] C. Seaman, "Qualitative Methods in Empirical Studies of Software Engineering," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 557–572, 1999.
- [31] B. Boehm, *Software engineering economics*. Prentice Hall, 1981, p. 767.