

Estudio sobre las prácticas de la ingeniería de *software* en Costa Rica: Resultados preliminares

Christian Quesada-López^{1,2}, Marcelo Jenkins^{1,2}

CITIC, Universidad de Costa Rica, San Pedro, Costa Rica¹
Ministerio de Ciencia, Tecnología y Telecomunicaciones, San José, Costa Rica²
{cristian.quesadalopez, marcelo.jenkins}@ucr.ac.cr

Resumen. El objetivo del presente estudio es caracterizar las prácticas de la ingeniería de *software* (SE) en la industria costarricense. Para alcanzar el objetivo se diseña y ejecuta una encuesta en línea con 46 preguntas basadas en el *Software Engineering Body of Knowledge (SWEBOK)* y estudios previos en el área. Los resultados confirman que (1) La calidad y la productividad siguen siendo los principales desafíos. (2) Los profesionales requieren evidencia sobre las prácticas esenciales de la SE que generan mayor valor agregado a sus organizaciones. (3) Los principales desafíos los enfrentan en las actividades de aseguramiento de la calidad y pruebas, y en la comunicación con la alta gerencia y los usuarios finales. (4) Las metodologías ágiles son las más utilizadas (46.7%), principalmente *Scrum* (24.5%), seguido por las metodologías tradicionales como cascada (19.9%), prototipos (19.0%), incremental (10.2%) y espiral (3.7%). (5) En el 85% de los casos el uso de las metodologías ágiles y cascada es excluyente. (6) Los lenguajes más utilizados son los de la familia .Net (48%), seguido por Java (26%). (7) Las pruebas más comunes son las de unidad, de aceptación, funcionales, de sistema y de integración. (8) Las pruebas manuales prevalecen sobre las automatizadas, solo el 33% reportan el uso frecuente de pruebas automatizadas. La industria costarricense se encuentra en el momento justo para adoptar prácticas de aseguramiento de la calidad y mejoramiento y medición de los procesos. Los resultados de la encuesta pueden ser de interés para la industria y la academia y confirman algunos de los reportados por estudios similares en otros países.

Palabras clave. Prácticas en la industria, Costa Rica, encuesta, Ingeniería de *software* experimental.

1 Introducción

El área de la ingeniería de *software* (SE) ha sido un campo de investigación activo desde 1968 y una disciplina que ha ido incrementando su nivel de madurez con el paso de los años. La IEEE estableció su comité para el desarrollo de estándares en la SE en 1976 y proporciona el *Software Engineering Body of Knowledge (SWEBOK)* como la guía base de conocimiento y prácticas relacionadas con la SE [1] [2]. El SWEBOK establece la línea base de las prácticas de la SE para impulsar el avance del conocimiento y la práctica en el área. El propósito del SWEBOK es proveer una caracterización de los

límites de la disciplina que se encuentra en constante evolución [1] [2]. El SWEBOK se compone de 15 áreas de conocimiento que describen el cómo desarrollar, administrar y dar mantenimiento al *software*, entre otras, tal como se detalla en [2].

En Costa Rica la industria del *software* se ha consolidado en los últimos 15 años y actualmente cuenta con más de 200 organizaciones de desarrollo de *software* [3]. Los servicios de *software* representan el 1.31% del producto interno bruto (PIB) y las exportaciones del sector representan el segundo lugar en producción de divisas. En la actualidad, el país constituye el primer exportador *per cápita* de alta tecnología en la región y el cuarto en el mundo [4]. Durante los últimos años, en el país se han realizado estudios para caracterizar y entender la evolución y el avance de la industria del *software* [3] [5] [6]; sin embargo en nuestro conocimiento, ninguno está enfocado específicamente en las prácticas de la SE.

La caracterización de las prácticas de la SE utilizadas por la industria permite entender la naturaleza y la madurez de las organizaciones de desarrollo de *software* y así dirigir esfuerzos para el mejoramiento de la competitividad del sector. El objetivo de este estudio es caracterizar las prácticas de la SE en la industria de *software* costarricense, identificando las prácticas, métodos y herramientas más utilizadas en la práctica profesional y los desafíos actuales. Este estudio compara algunas de las tendencias reportadas en estudios previos [7] con los resultados obtenidos.

2 Estudios sobre las prácticas de la ingeniería del *software*

Múltiples estudios se han realizado en la industria para determinar el estado de las prácticas de la SE (PSE). Estos estudios se han llevado a cabo con el objetivo de entender esta industria, determinar sus oportunidades de mejora y fortalecer las colaboraciones entre la industria y la academia. Garousi [7] presentan un listado de encuestas realizadas en la industria sobre las prácticas de la SE abarcando todas sus áreas de conocimiento. Los estudios realizados del 2010 a la fecha se presentan en el Cuadro 1.

Cuadro 1. Resumen de encuestas sobre las prácticas de la SE

Ref.	País	Año	n	Objetivo
[9]	Italia	2010	62 pry.	Evaluar el efecto percibido.
[10]	N. Zelanda	2012	195	Entender el uso.
[11]	Holanda	2012	99	Estado actual.
[7]	Turquía	2015	202	Caracterización.
[12]	Turquía	2016	202	Análisis de factores de influencia.
[8]	Costa Rica	2016	278	Caracterización.

Este listado no es exhaustivo y no incluye encuestas que se realizan para sub temas específicos de la SE. Existe un número importante de estudios que analizan temas particulares de la SE en áreas como: la estimación, requerimientos, diseño, desarrollo, mejoramiento de procesos y calidad, pero por restricciones de espacio, muchos de los estudios no pueden ser detallados; sin embargo, la lista de estudios puede ser encontrada en el reporte técnico de este estudio, en [8]. Nuestro estudio se basa en las prácticas

generales del SWEBOK de acuerdo con el diseño propuesto por [7] y se limita a las áreas de conocimiento de los capítulos 1 al 10 de las prácticas del SWEBOK que incluyen: requerimientos, diseño, construcción, pruebas, mantenimiento, administración de la configuración, administración de la ingeniería, procesos, métodos y herramientas y calidad.

3 Descripción del estudio

Este estudio realiza una encuesta para caracterizar las prácticas de la ingeniería del *software* (SE) utilizadas por la industria. El estudio es una replicación externa, sin interacción con los investigadores originales, en la industria de *software* de Costa Rica y se basa en [7]. El estudio incluye el análisis detallado de los datos recolectados, y su comparación con [7], de acuerdo con las recomendaciones y lineamientos de [13] [14]. Este estudio es descriptivo [14].

El objetivo del estudio es caracterizar las prácticas de la SE en la industria de *software* costarricense. Con base en este objetivo, se definen las siguientes preguntas de investigación basadas en [7]:

- RQ1. ¿Cuál es el perfil profesional y las características de las organizaciones a la que pertenecen los participantes de la encuesta?
- RQ2. ¿Cuáles son las características del ciclo de vida de desarrollo de la industria de *software* de Costa Rica?
- RQ3. ¿Qué tipos de prácticas de ingeniería de requerimientos, diseño de *software*, desarrollo, pruebas, mantenimiento, gestión de la configuración, administración de proyectos y gestión de la calidad son usadas por los profesionales?
- RQ4. ¿Cuál es el nivel de investigación de las organizaciones y las colaboraciones con la academia?

Diseño de la encuesta. Para la selección de los participantes se utiliza un muestreo no probabilístico, donde los participantes son contactados por medios electrónicos y participan de manera voluntaria y anónima. El muestreo por conveniencia donde los participantes son contactados de acuerdo con las posibilidades de los investigadores es uno de los más utilizados en el área de la ingeniería del *software*.

Se diseña un cuestionario electrónico basado en la estructura del SWEBOK [1] [2] y se revisa contra los reportados en [7]. La unidad de análisis son los profesionales relacionados con la ingeniería de *software* en la industria costarricense. El cuestionario es piloteado con cinco profesionales de la industria costarricense para validar que el contenido y la terminología son entendibles. Con las recomendaciones recibidas se ajusta la versión final del cuestionario. El cuestionario cuenta con 46 preguntas basadas en las prácticas de la SE y directamente relacionadas con las preguntas de investigación del estudio. Los resultados deben ser interpretados con cautela dado el sesgo que puede introducirse por la muestra [15]. Tal como en estudios previos, los resultados deben ser interpretados bajo esta condición y no pueden ser asociados a las organizaciones o a sus proyectos de *software* [7].

Ejecución. El cuestionario electrónico fue implementado en la herramienta *LimeSurvey* (www.limesurvey.org) y se mantuvo en línea 4 meses, desde julio a octubre del 2016. Todos los participantes fueron invitados a participar de manera anónima y voluntaria por correos enviados por medio de distintas instituciones relacionadas con la industria de *software* en Costa Rica. Similar a experiencias reportadas en estudios anteriores [7], algunos participantes dejan preguntas sin responder y otros abandonan el cuestionario sin terminarlo. Todas las respuestas parciales fueron consideradas valiosas para el análisis de resultados. Después del pre-procesamiento de datos, se analizan las respuestas de 278 profesionales de ingeniería de *software*. El medio más efectivo de contacto fueron las universidades públicas, seguido por las instituciones del sector público. En promedio responden de manera completa todas las secciones de la encuesta 120 participantes, para una tasa de respuesta del 43%, similar al reportado en estudios previos.

4 Análisis y discusión de resultados

En esta sección presentamos los resultados de la encuesta y analizamos la información para cada una de la preguntas de investigación. Además comparamos los resultados con los reportados en [7]. La encuesta revela información interesante sobre las diferentes tendencias en la industria de *software* en Costa Rica. A continuación presentamos los principales resultados del estudio, el cual se encuentra extendido en el reporte técnico correspondiente [8].

4.1 Perfiles y datos demográficos

La RQ1 busca determinar los perfiles profesionales y las características de las organizaciones y los proyectos a la que pertenecen los participantes de la encuesta.

Posición de los participantes y experiencia laboral. Las posiciones reportadas por los participantes se muestran en el Cuadro 2. La pregunta fue diseñada para permitir selección múltiple por lo que se totalizan las posiciones reportadas. En nuestro estudio, las posiciones profesionales reportadas por los participantes son similares a las reportadas en [7], por lo que es interesante, a partir de la recolección de los datos comparar algunas de las tendencias obtenidas. La opinión de los participantes desde las diferentes perspectivas profesionales y puntos de vista es importante para el estudio. El 39% de los participantes reportan que desempeñan solo una posición en sus organizaciones. Sin embargo, el 30% reporta desempeñarse en 4 o más posiciones. En comparación con [7] [16] se observa que las tendencias son similares, en los tres estudios la mayoría de los profesionales se desempeña en una única posición seguida por dos posiciones. El promedio de años de experiencia es de 12.9 años y la media de 12 años. La experiencia de los participantes con igual o menos de 10 años es 47%. En comparación con los estudios previos, podemos observar que las tendencia de años de experiencia es mayor en nuestro estudio que en [16] (Pm=7.8, Md=6) y [7] (Pm=9.5, Md=8). Las tendencias muestran que la participación de profesionales jóvenes en nuestra encuesta es menor que la de los de la contraparte canadiense y turca.

Cuadro 2. Cantidad de participantes por posición

Posición	n	%	Posición	n	%
Desarrollador / programador	110	12	Analista de negocio	60	7
Analista o Ingeniero	87	10	Soporte técnico	55	6
Ingeniero de requerimientos	77	9	<i>Software tester</i>	39	4
Diseñador de <i>software</i>	74	8	Gerente de alto nivel	37	4
Líder de proyecto	72	8	Investigador	29	3
Docente	71	8	" <i>QA lead</i> "	17	2
Arquitecto de <i>software</i>	64	7	Vendedor	7	1
Administrador de proyecto	64	7	Otro	20	2

Perfil académico y formación universitaria. El 40% de los participantes cuentan con una maestría, el 22% con una licenciatura, el 26% con el bachillerato, el 0.5% con un doctorado y el restante 11.5% con un título en formación técnica no universitaria. Todos los anteriores con puestos de trabajo en la industria. En comparación con los estudios previos [7] [16], podemos observar que las tendencias se invierten entre maestrías y bachilleratos, sin embargo si se unen las licenciaturas con los bachilleratos se muestran proporciones similares. Las principales carreras universitarias de los participantes son: (a) Ingeniería en computación (33%), Sistemas de información (22%), Ciencias de la computación (18%) e Ingeniería en *software* (14%). En comparación con [7] [16] podemos observar que la tendencia se mantiene para carreras relacionadas con la computación, pero no para carreras relacionadas con negocios o ingeniería electrónica, en la cual nuestro estudio cuenta con menor número de participantes.

Productos desarrollados por las organizaciones. El sector objetivo para los cuáles los productos de *software* son desarrollados son: (a) gobierno y sector público (30%), (b) la banca y finanzas (16%) y (c) comercio electrónico en general (11%). La mayoría de las organizaciones de los participantes son de capital nacional (67%), de capital extranjero (12%) o de ambos (10%). En comparación con [7] podemos observar que la industria en Costa Rica es distinta a la de Turquía, puesto que nuestro país no se desarrolla la industria militar y de defensa. Además, la cantidad de participantes de la industria de telecomunicaciones en nuestra encuesta es solo del 3.3%. Consideramos que contamos con una buena combinación de respuestas para diferentes dominios.

Los proyectos desarrollados son en su mayoría: (a) productos de *software* a la medida a partir de las necesidades particulares de los clientes (36%), (b) desarrollo "*In-house*" (34%), (c) desarrollo de paquetes de *software* para industrias particulares (17%) y (d) de uso general para negocios (13%). La pregunta permite múltiples respuestas, donde el 50% indican una sola opción, el 30% dos opciones, el 13% tres opciones y el 7% las cuatro posibles. En comparación con [7] podemos observar una tendencia similar hacia el desarrollo basado en las necesidades particulares de los clientes y desarrollo de *software* "*In-house*". En el caso de nuestro estudio, 50% de los participantes indican que desarrollan para más de una industria, contrario al 22% indicado en la contraparte. El 67% de los participantes de nuestro estudio indican que los proyectos que desarrollan son productos de *software* a la medida a partir de las necesidades particulares de los

clientes y a la vez que es *software* desarrollado "In-house". El 34% llevan a cabo proyectos de desarrollo y mantenimiento, el 2% realizan solo proyectos de validación y verificación y el 45% indican que realizan ambos.

Tamaño de las organizaciones. El Cuadro 3 presenta el tamaño de la organización de los participantes. En comparación con los estudios previos, podemos observar que la tendencia es similar a [7] donde la mayoría de los participantes son de organizaciones grandes (64%). Los tipos de clientes para los que trabajan las organizaciones son clientes internos (27%), clientes externos (13%) y en su mayoría para ambos tipos (42%).

Cuadro 3. Tamaño de las organizaciones

Tamaño	n	%	Tamaño	n	%	Tamaño	n	%
1-5	6	5	21-30	3	3	101-200	10	8
6-10	6	5	31-50	1	1	201-500	6	5
11-20	4	3	51-100	7	6	501+	77	64

4.2 Ciclo de vida de desarrollo (SDLC) y prácticas organizacionales

La RQ2 busca determinar las características del ciclo de vida de desarrollo, los procesos generales de la ingeniería del *software* y el mejoramiento de procesos.

Nivel de involucramiento y desafío en las fases y actividades del SDLC. Los participantes reportan participación activa en todas las fases (requerimientos, diseño, desarrollo, pruebas y mantenimiento) y en administración de proyectos. En el caso de los desarrolladores y programadores la tendencia muestra un leve predominio para las fases de desarrollo y mantenimiento. En comparación con [7] podemos observar una tendencia similar hacia la fase de desarrollo y existe una baja participación en actividades de administración de la configuración, definición, evaluación y medición de los procesos de ingeniería del *software*, aseguramiento de la calidad y documentación técnica. Contrario a nuestro estudio los participantes en [7] no reportan alta participación en procesos de administración de proyectos. Al igual que los resultados en [7], los participantes reportan un nivel de desafío levemente superior para las actividades de aseguramiento de calidad y comunicación con la alta administración y con los usuarios finales. Dentro de las fases de ciclo de vida de desarrollo la etapa de pruebas es reportada como la más desafiante.

Ciclo de vida de desarrollo de *software*. El Cuadro 4 presenta los ciclos de vida de desarrollo más utilizadas por los participantes. Las metodologías ágiles son las más utilizadas por los profesionales (46.7%) y la más popular es *Scrum* (24.5%). El 19.4% indican seguir las prácticas ágiles sin adoptar una metodología específica. El ciclo de vida cascada (19.9%), prototipos (19.0%), desarrollo incremental (10.2%) y espiral (3.7%) aún se mantienen vigentes en la industria. La menos utilizada es la de "*Product-line development*" (0.5%). En el 85% de los casos, el uso de metodologías ágiles es excluyente con el uso de cascada. Los resultados son similares a [7] con respecto a la utilización de las metodologías ágiles, sin embargo la utilización de cascada (53%) y

“*Product-line development*” (12%) es superior. Estos resultados pueden ser explicados por el tipo de industria donde la mayoría de las organizaciones son del área militar y defensa [12].

Cuadro 4. Metodologías de desarrollo

Ciclo de vida	n	%	Ciclo de vida	n	%
"Scrum"	53	24.5	Incremental (linear-iterativo)	22	10.2
Desarrollo ágil o lean	42	19.4	Espiral (linear-iterativo)	8	3.7
Cascada	43	19.9	" <i>Extreme Programming</i> "	6	2.8
Prototipos (iterativo)	41	19.0	" <i>Product-line development</i> "	1	0.5

Modelos de mejoramiento de procesos y prácticas relacionadas con procesos. El 41% utilizan el estándar ISO 9000 ("*Quality management systems*") y el 36% utilizan el CMMI ("*The Capability Maturity Model Integration*"). El 23% de los participantes menciona el uso de otros modelos tales como COBIT, ITIL, y otros estándares como el ISO 27001, ISO 42010 e ISO 19510. Esta pregunta solo la responden 22 participantes. En cuanto a porcentajes, los resultados son similares a [7], pero muy inferiores en la industria costarricense. En [3] se reporta que solo el 13.2% de las organizaciones en Costa Rica han adoptado formalmente modelos o estándares de calidad y que el 21.5% de las organizaciones planean incorporar algún modelo de calidad. En relación con las prácticas de procesos, aunque la mayoría de las organizaciones no reportan la adopción formal de modelos o estándares, la tendencia se inclina hacia la aplicación de tareas basadas en un estándar de procesos de un manual de calidad. Los participantes reportan tener enfoques bien definidos para la gestión de defectos (severidad, tipos de defectos, entre otros), pero no es común la certificación en un estándar de calidad. Los resultados son similares a los reportados en [7].

4.3 Prácticas de la ingeniería del software (SE)

La RQ3 busca determinar los tipos de prácticas para cada una de las áreas de conocimiento de la SE.

Prácticas relacionadas con los requerimientos. En la fase de requerimientos, aunque no existe una diferencia marcada en la tendencia de las respuestas, las principales prácticas usadas por los participantes son: (a) la documentación y revisión formal de todas las características (*features*) del producto, (b) la validación de los requerimientos mediante inspecciones, (c) y la validación por medio de la revisión de pares y los *walk throughs*. En comparación con [7], la tendencia es similar, sin embargo con respecto a la implementación de trazabilidad de los requerimientos, nuestros resultados son menores. Al igual que en [7], las prácticas menos comunes son: (d) el análisis de requerimientos es realizado por un grupo externo y (e) los detalles de los requerimientos no se encuentran documentados y solo los encargados los conocen (“memoria”).

A partir de un análisis estadístico basado en el coeficiente de correlación de Spearman, se determina una relación fuerte, positiva y estadísticamente significativa ($\alpha = 0.01$) entre las siguientes prácticas: (a) se realiza un análisis formal y se validan los requerimientos ($r_s = .816$). (b) Todas las características (*features*) son documentadas

y revisadas formalmente y los requerimientos son documentados claramente ($r_s = .611$), se realiza un análisis formal ($r_s = .651$) y se validan los requerimientos ($r_s = .715$). En comparación con los resultados reportados en [7], se confirma la hipótesis de la relación entre el análisis formal y la validación de los requerimientos en las organizaciones. Sin embargo, aunque existe una relación negativa entre requerimientos documentados claramente y los detalles de los requerimientos no se encuentran documentados y solo los encargados los conocen ("de memoria") ($r_s = -.087$) no se puede confirmar estadísticamente esta hipótesis.

Notaciones utilizadas en documentos de requerimientos. El 27% de los participantes indica que utiliza casos de uso y escenarios, el 22.8% usa "*User stories y story boards*", el 18.6% usa diagramas de actividad UML y un 15.8% utiliza lenguaje natural. Solo el 0.9% de los participantes indica que no se documentan requerimientos o que se utilizan notaciones formales. En comparación con [7] [17] se confirma la utilización predominante de casos de uso y escenarios, y "*User stories*". Sin embargo, en nuestro estudio el uso de lenguaje natural (15.8%) es mucho menor al reportando en [7] (61%) y [17] (69%).

Prácticas relacionadas con el diseño del *software*. Las principales prácticas usadas en el diseño son: (a) la asignación de personal dedicado a la arquitectura de *software* y/o el diseño de alto nivel y (b) el equipo de trabajo realiza revisiones informales del diseño. En comparación con [7] las tendencias son similares en cuanto a las revisiones informales. Con respecto a la frecuencia en la cual se deben tomar decisiones creativas de diseño por tener documentos de requerimientos incompletos, los resultados de nuestro estudio son menores, lo cual es positivo.

Tipo de actividades de diseño y atributos de calidad en el diseño. Las principales tareas en diseño son: (a) el diseño de clases, (b) de componentes, (c) de interfaz de usuario, (d) el diseño por medio de prototipos, (e) el diseño de los servicios, (f) el diseño de la arquitectura y (g) del producto. Al igual que en [7], las actividades de diseño de clases, componentes e interfaz y el diseño de la arquitectura son comunes en la industria y el diseño por contrato es de las prácticas menos utilizadas. Tal como se reporta en [7], los resultados sobre los atributos de calidad en el diseño indican que todos los atributos (extensibilidad, tolerancia a fallos, mantenibilidad, modularidad, confiabilidad, reusabilidad, robustez y usabilidad), son considerados importantes. Contrario a [7], los atributos de seguridad y usabilidad son los reportados como los más importantes en nuestro estudio; sin embargo, las tendencias generales para todos los atributos de calidad son similares entre los estudios comparados. Consideramos de importancia para nuestra industria esta tendencia hacia la seguridad y usabilidad.

Prácticas relacionadas con el desarrollo. El Cuadro 5 presenta el promedio y la media de las frecuencias de las prácticas relacionadas con el desarrollo. Las prácticas más frecuentes son: (a) que los desarrolladores son responsables por el diseño de alto nivel y la implementación, (b) el diseño y la codificación son realizados en conjunto, y (c) el

proceso de los "builts" está documentado y se le da seguimiento. Estas prácticas coinciden con las reportadas en [7]. Las prácticas menos frecuentes son: (d) la generación automática de documentación usando herramientas como JavaDocs y (e) la medición, administración y minimización de la complejidad del código. Prácticas como la programación en pares, inspecciones y refactorización aún no han sido totalmente adoptadas en la industria, por lo que es necesario que los profesionales conozcan más de sus beneficios.

Cuadro 5. Prácticas relacionadas con el desarrollo

Práctica	Pm	Md	Práctica	Pm	Md
Programación en pares	2.54	3	Diseño y desarrollo juntos	3.45	4
Refactorización	2.54	3	<i>Builts</i> diariamente	2.71	3
Inspecciones en pares	2.69	3	<i>Builts</i> nueva funcionalidad	3.11	3
Análisis estático	2.39	3	Doc. y seg. proceso <i>builts</i>	3.12	3
Aut. de documentación	2.00	2	Encargado de los <i>builts</i>	2.80	3
Complejidad del código	2.10	2	Código fuente antes <i>built</i>	2.81	3
Responsables del diseño	3.77	4			

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

A partir del análisis de correlación de Spearman se determina una relación fuerte, positiva y estadísticamente significativa ($\alpha = 0.01$) para la hipótesis planteada en [7] entre la programación en pares, la inspección de código y las revisiones por pares ($r_s = .734$). Además se encuentran relaciones significativas entre las siguientes prácticas: (a) prácticas de refactorización y programación en pares ($r_s = .726$), e inspección de código y revisiones por pares ($r_s = .706$), (b) se realizan los *builts* diariamente cuando se ha generado una característica o funcionalidad completa ($r_s = .704$), con un proceso documentado de *builts* ($r_s = .722$) y personal dedicado a los *builts* ($r_s = .673$).

Lenguajes de programación. El 46% de los participantes reportan el uso de más de un lenguaje de programación. Los lenguajes más utilizados son: (a) .Net (C#, VB,...) (48%), (b) Java (26%), (c) C/C++ (9%), (d) PL/SQL Oracle (7.9%) y (e) PHP (5.5%). En comparación con otros estudios que reportaron el uso de lenguajes de programación [7] [16], podemos observar que Java y .Net siguen manteniendo un alto nivel de adopción.

Prácticas relacionadas con pruebas. Las prácticas reportadas como más frecuentes en la fase de pruebas son: (a) un gerente, experto en el dominio del negocio (cliente o apoyo) ayuda a probar el producto, (b) los desarrolladores prueban el producto antes de cada liberación y (c) uso de las pruebas de unidad. El 25% reporta que siempre se tiene un equipo separado para realizar las pruebas del producto y un 23% que casi siempre. Las prácticas coinciden con las reportadas en [7] donde los valores son similares con excepción (d) un gerente, experto en el dominio del negocio ayuda a probar el producto. En nuestro estudio, esta práctica es común en las pruebas de aceptación del producto. Los participantes son consultados sobre si realizan las pruebas de acuerdo a "*Test driven*

Development" (TDD) o "*Test last Development*" (TLD). La mayoría de los participantes reportan que realizan la fase de pruebas TLD (28% casi siempre y 29% siempre). Esta práctica coincide con la reportada en [7] [16] donde la tendencia es la misma.

Tipos y niveles de pruebas. El Cuadro 6 presenta el promedio y la media de las frecuencias para cada uno de los tipos y niveles de pruebas. La mayoría de los participantes reportan que realizan pruebas de aceptación de usuario, pruebas funcionales, pruebas de sistema y pruebas de integración. Estas pruebas coinciden con las reportadas en [7] [16] donde la tendencia se mantiene. La prueba menos común es la prueba de carga/estrés de igual manera que en [7].

Cuadro 6. Tipos y niveles de pruebas

Tipo	Pm	Md	Tipo	Pm	Md
Pruebas de unidad	3.65	4	Pruebas de rendimiento	3.56	4
Pruebas de integración	3.91	4	Pruebas de carga/estrés	3.14	3
Pruebas de aceptación	4.29	5	Pruebas de seguridad	3.37	4
Pruebas funcionales	4.15	5	Pruebas de usabilidad	3.50	4
Pruebas de sistema	4.07	4			

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

Automatización de pruebas. Las pruebas manuales prevalecen versus las automatizadas, sin embargo se observa algún grado de adopción de pruebas automatizadas (a veces (32%), casi siempre (18%), siempre (15%)). En comparación con [7] [16] el nivel de la automatización de las pruebas es similar.

Técnicas de diseño de casos de prueba. Las técnicas de diseño de casos de prueba basadas en modelos son las más utilizadas (22%), el análisis de valores límite (14.4%), caja blanca (14.4%), clases de equivalencia (9.8%). Aunque las técnicas de equivalencia y valores límite son de caja negra, los participantes indicaron en "otros" caja negra un 5.3%. Al igual que en [7], es alta la no utilización de una técnica específica (34.1%).

Métricas de cobertura de código y de calidad. Al igual que [7] la utilización de métricas de cobertura es limitada por lo que es necesario que los profesionales conozcan más de los beneficios de estas mediciones. La métrica más utilizada en nuestro estudio es "*Line (statement) coverage*" y las métricas de calidad más frecuentes son los casos de pruebas de aceptación aprobados y los defectos detectados por un periodo de tiempo.

Tipo de actividades de mantenimiento de software. El mantenimiento correctivo es la actividad más utilizada y el mantenimiento perfectivo la menos usada. Los valores reportados para nuestro estudio son levemente mayores, tanto para el mantenimiento correctivo como el perfectivo. Los profesionales consideran el mantenimiento perfectivo y adaptativo tareas difíciles de realizar. Uno de los principales desafíos que enfrentan son: (a) los diferentes estilos de programación usados por los desarrolladores, (b) la documentación insuficiente, y (c) la falta de herramientas de soporte para el mantenimiento. Todos estos resultados coinciden con los reportados en [7].

Medición y estimación en la administración de proyectos. El uso del juicio experto para la estimación de esfuerzo y costos es una práctica común. Se reporta la práctica de estimar el esfuerzo y el costo de forma sistemática antes de cada proyecto; sin embargo, muy pocos lo hacen mediante los métodos formales. Las metodologías reportadas para estimación son (a) uso de historias de usuario y (b) los puntos de casos de uso aplicado a historias de usuario. El 57% de los participantes indican que su organización cuenta con una oficina dedicada a la administración de proyectos.

Herramientas de la SE. Es interesante conocer el nivel de adopción de distintos tipos de herramientas en la industria, esta pregunta investiga las herramientas que soportan las actividades de las prácticas de SE. Al igual que lo reportado en [7], las herramientas más comúnmente utilizadas son para la gestión de la configuración (versionamiento, liberaciones y “*builts*”) y para la administración (planificación, seguimiento, gestión y riesgos). Las herramientas menos utilizadas son las de mantenimiento del *software* (comprensión o la reingeniería de los artefactos de diseño) y para la calidad (análisis de código estático).

4.4 Investigación y colaboración con la academia

La RQ4 busca determinar las actividades de investigación realizadas por las organizaciones, el nivel de interacción con investigadores académicos y literatura científica. En nuestro estudio, solo cinco de 136 organizaciones indican que cuentan con una unidad de investigación (4%). Los resultados distan de los reportados por [7] lo cual indica la necesidad de que las organizaciones de desarrollo puedan destinar recursos a la investigación sobre cómo mejorar en sus prácticas en SE.

Interacción con investigadores académicos y literatura científica. Tal como se observa en el Cuadro 7, los participantes reportan poca interacción con investigadores académicos. En cuanto a la lectura de artículos científicos, la tendencia mejora un poco con respecto a la pregunta anterior. Los resultados son similares a las reportadas en [7]. Al consultar a los participantes sobre cuál creen que es la mejor forma para que las universidades impulsen espacios de diálogo en donde los académicos y los profesionales compartan conocimiento científico, indican opciones como: reuniones, charlas, conversatorios, mesas redondas, grupos focales, talleres y demostraciones, publicaciones y boletines técnicos, congresos y conferencias, foros virtuales, tutoriales en video, visitas a las organizaciones, programas de intercambio de conocimiento y proyectos de colaboración, intercambio de estudiantes con las organizaciones y pasantías para los profesionales. En el caso del sector público consideran que es clave la participación de las universidades en proyectos de desarrollo de soluciones integrales. Los participantes opinan que se deben mejorar los canales de comunicación para anunciar los eventos y que la academia debe ser más abierta, con visitas a las organizaciones, y la realización de procesos de investigación aplicada que generen valor agregado. Finalmente, mencionan que la literatura científica proporciona un valor agregado en su práctica profesional y en los procesos de mejora continua; sin embargo, opinan que algunas veces los

artículos científicos no se apegan a su realidad profesional o son difíciles de poner en práctica.

Cuadro 7 Interacción con la academia

Investigadores	n	%	Artículos	n	%
Una vez al mes	13	16%	Una vez al mes	40	47%
Una vez cada seis meses	10	13%	Una vez cada seis meses	16	19%
Una vez al año	3	4%	Una vez al año	8	9%
Casi nunca	23	29%	Casi nunca	15	18%
Nunca	31	39%	Nunca	6	7%
Total	80	100%	Total	85	100%

Desafíos en la ingeniería del *software*. Se consulta a los participantes sobre cuáles son los desafíos que desearían que la academia investigue o en la cuáles se requieren colaboraciones industria academia. Las respuestas proporcionadas incluyen los siguientes desafíos relacionados con productividad y calidad: ¿cómo garantizar la calidad del código fuente?, ¿cuáles son las métricas que generan más valor en el ciclo de vida de desarrollo y cómo pueden ser automatizadas?, ¿cómo estimar el tiempo y costo de manera eficiente?, ¿cuáles son los artefactos esenciales que generan más valor en el ciclo de vida de desarrollo?, ¿cuáles son las mejores prácticas para la capacitación de personal en nuevas metodologías y herramientas? y ¿cómo desarrollar las habilidades requeridas en diseño, programación y pruebas en los nuevos profesionales?.

5 Lecciones aprendidas y limitaciones del estudio

Lecciones aprendidas. La experiencia al ejecutar la encuesta confirma las recomendaciones realizadas en estudios previos sobre este tipo de estudios [7] [16] [18]. Para el estudio se realizó una categorización de las preguntas de acuerdo al SWEBOK [2], posteriormente se validó el instrumento con profesionales para asegurar que la terminología era la común en la industria. El instrumento se aplicó en el lenguaje español y los investigadores fueron los responsables de realizar las traducciones respectivas.

Todos los participantes fueron invitados a participar de manera anónima y voluntaria por correo electrónico distribuidos por distintas instituciones relacionadas con la industria de *software* en Costa Rica. Los medios de contacto más efectivos fueron las universidades públicas, seguido por las instituciones del sector público. Similar a experiencias reportadas anteriormente, algunos participantes dejaron preguntas sin responder y otros abandonaron el cuestionario. Similar a estudios previos en el área [7] [16] [18], en promedio 120 participantes responden todas las secciones de la encuesta, para una tasa de respuesta del 43%. En este estudio, los resultados que se reportan en las secciones 4.2, 4.3 y 4.4 se basan en los participantes que responden cada una de las secciones. Analizando las tendencias de preguntas no respondidas podemos observar que la mayor cantidad de preguntas sin contestar se relacionan con las prácticas específicas en la SE. Dos factores pueden influenciar esta tendencia son: (1) la cantidad de preguntas del cuestionario y (2) el que algunos de los participantes proveen respuestas

de alto nivel, pero son reservados acerca de las prácticas específicas en sus organizaciones. La no adopción de las prácticas en la ingeniería del *software* no solo se debe a aspectos técnicos; en muchos casos, la no adopción de estas prácticas está directamente relacionada con la falta de recursos en las organizaciones.

Amenazas a la validez. Las limitaciones en nuestro estudio son las comúnmente reportadas en estudios similares en el área [7] [16] [18]. **Interna.** El sesgo de la selección de los participantes es una amenaza por ser una muestra a conveniencia no aleatorizada. Nuestros participantes fueron invitados por correo electrónico mediante las principales instituciones que tienen relación con el área de estudio. Se enviaron recordatorios periódicos a los participantes. Consideramos que fue limitada la participación del sector privado (organizaciones pequeñas y transnacionales), al igual que la participación de ingenieros jóvenes. **Constructo.** Para este estudio se utiliza el SWEBOK y procesamos la información de acuerdo a las prácticas e instrumentos recomendadas en [7]. Dado que el estudio utiliza instrumentos validados en publicaciones anteriores no se realizan validaciones como las sugeridas en [19] [20] [21]. La invitación indica explícitamente que la participación es anónima y los resultados serían publicados de manera agregada sin afectar la privacidad. **Conclusión.** Para cada pregunta de investigación se realizó un procesamiento de datos estadístico y trazable a través de la información analizada. Las preguntas abiertas son tabuladas sin realizar ninguna interpretación y se muestran tal y como las reportaron los participantes. **Externa.** Se aplicaron diferentes estrategias para minimizar el sesgo de la población. Sin embargo, los resultados de este estudio solo pueden generalizarse a los datos demográficos detallados. Aunque conocemos por experiencias previas en la industria que muchos de nuestros resultados son válidos, no se pueden generalizar sin un análisis crítico que evalúe los resultados de acuerdo con el contexto.

6 Conclusiones y trabajo futuro

De acuerdo con los resultados, la industria costarricense se encuentra en el momento justo para empezar a adoptar prácticas de aseguramiento de la calidad, mejoramiento y medición de procesos. Para ello, se deben realizar colaboraciones industria academia que generen beneficios para ambas partes. Dichas alianzas deben incluir, no solo colaboraciones para probar nuevas metodologías o herramientas, sino también, las actividades de capacitación que permitan la adopción de estas nuevas tecnologías. Las investigaciones, no solo deben dirigirse a los aspectos técnicos, sino también, a los aspectos organizacionales y sociales de la SE. Nuestro estudio confirma algunas de los resultados reportados en otros países, lo que consideramos un aporte importante para la comunidad de la ingeniería del *software*. Deben realizarse trabajos similares en distintos países, primero mapeando el estado general de adopción de las prácticas de la SE y luego analizando temas específicos de interés para la industria en distintas áreas de conocimiento de la SE. Como trabajo futuro, planeamos replicar el estudio para lograr un mayor alcance a la población de profesionales del país. Recomendamos que este estudio se

pueda realizar de manera periódica para poder determinar el nivel de avance como industria en la adopción de las prácticas de la ingeniería de *software*. Finalmente, debemos analizar cómo los factores demográficos influyen en la adopción de cada una de las prácticas de la SE, de acuerdo con las recomendaciones en [12]. Este análisis estudiará el impacto del perfil de los participantes, las organizaciones y los tipos de proyectos en la adopción de las prácticas de la SE.

Agradecimientos. Este estudio fue apoyado por el Ministerio de Ciencia, Tecnología y Telecomunicaciones (MICITT) y la Universidad de Costa Rica (No. 834-B5-A18). Agradecemos a todos (as) los (as) profesionales que participaron en la encuesta y a las instituciones que colaboraron en la difusión del cuestionario.

Referencias

1. Abran, A., Moore, J., Bourque, P., Dupuis, R. (Eds): Guide to the Software Engineering Body of Knowledge: 2004 version. IEEE Press, CA, USA (2004).
2. Bourque, P., Fairley, R. (Eds): Guide to the Software Engineering Body of Knowledge: Version 3.0. IEEE Press, CA, USA (2014). Recuperado de <http://www.swebok.org>.
3. Hewitt, J., Monge-González, R.: Mapeo del Sector de las TICs de Costa Rica. CAMTIC (2014).
4. Siles, I., González, J., Marengo, A.: ¿El “Silicon Valley latinoamericano”? La producción de tecnología de comunicación en Costa Rica (1950-2016). En: Estudios Centroamericanos, 42(1), 411-441 (2016).
5. Brenes, L., GoVaere, V.: La industria del software en Costa Rica. COMEX, 58(5), 303-311 (2008).
6. Bonilla, L., Baraya, A., Budden, M., Bermudez, L.: Revisiting the Software Industry in Costa Rica: Achievements and Challenges. Journal of Service Science, 8(1), 29 (2015).
7. Garousi, V., Coşkunçay, A., Betin-Can, A., Demirörs, O.: A survey of software engineering practices in Turkey. Journal of Systems and Software, 108, 148-177 (2015).
8. Quesada-Lopez, C., Jenkins, M.: Un estudio sobre las prácticas de la ingeniería del software en Costa Rica: Resultados preliminares (Reporte Técnico Ref. 01-01-17-001 No.834-B5-A18. Costa Rica: Universidad de Costa Rica) (2016). Recuperado de <https://goo.gl/PZIXIW>.
9. Egorova, E., Torchiano, M., y Morisio, M.: Actual vs. perceived effect of software engineering practices in the Italian industry. Journal of Systems and Software, 83(10), 1907-1916 (2010).
10. Kirk, D., Tempero, E: Software development practices in New Zealand. En: 19th Asia-Pacific Software Engineering Conference: Vol. 1, pp. 386-395 (2012).
11. Vonken, F., Brunekreef, J., Zaidman, A. E. Peeters, F.: Software Engineering in the Netherlands: The State of the Practice. Reporte Técnico TUD-SERG-2012-022. Holanda: Delft University of Technology, Software Engineering Research Group (2012).
12. Garousi, V., Coşkunçay, A., Demirörs, O., Yazici, A.: Cross-factor analysis of software engineering practices versus practitioner demographics: An exploratory study in Turkey. Journal of Systems and Software, 111, 49-73 (2016).
13. Carver, J.: Towards reporting guidelines for experimental replications: A proposal. En: International Workshop on Replication in Empirical Software Engineering Research (2010).
14. Linaker, J., Sulaman, S., Host, M., De Melo, R.: Guidelines for conducting surveys in software engineering v. 1.1. (2015).
15. Sjøberg, D., Hannay, J., Hansen, O., Kampenes, V., Karahasanovic, A., Liborg, N., Rekdal, A.: A survey of controlled experiments in software engineering. IEEE transactions on software engineering, 31(9), 733-753 (2005).
16. Garousi, V., Coşkunçay, A., Betin-Can, A., Demirörs, O.: A survey of software testing practices in Turkey. En: Turkish National Software Engineering Symposium (UYMS) (2013).
17. Aykol, M.: Software engineering and software management practices in Turkey. Tesis de Maestría. Bahçeşehir University: Turkía (2009).
18. Punter, T., Ciolkowski, M., Freimut, B., John, I.: Conducting on-line surveys in software engineering. En: International Symposium on Empirical Software Engineering (ISESE), pp. 80-88 (2003).
19. Cronbach, L.: Coefficient alpha and internal structure of tests. Psychometrika, 16(3), 297-334 (1951).
20. Dunn, T., Baguley, T., Brunsden, V.: From alpha to omega: A practical solution to the pervasive problem of internal consistency estimation. British Journal of Psychology, 105(3), 399-412 (2014).
21. Peterson, R.: A meta-analysis of Cronbach's coefficient alpha. Journal of consumer research, 21(2), 381-391 (1994).