risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

Recebido/Submission: 15/10/2019
Aceitação/Acceptance: 28/12/2019

Enseñanza de Arquitectura de Software en Modalidad Blended Learning: Análisis de la Percepción de Estudiantes de Maestría

Boris Pérez[1,2], Camilo Castellanos[1], Darío Correal[1]

borisperezg@ufps.edu.co, {br.perez41, cc.castellanos87, dcorreal}@uniandes.edu.co

[1] Univ. De los Andes, Cra 1 # 18A - 12, Bogotá, Colombia.

[2] Univ. Francisco de Paula Santander, Av. Gran Colombia # 12E - 96, Cúcuta, Colombia.

Pages: 346–358

Resumen: La arquitectura de software, como componente clave de la educación en ingeniería de software, requiere que el estudiante desarrolle un conjunto de habilidades técnicas y sociales. En el curso de maestría Arquitecturas de Software Ágiles se decidió utilizar la modalidad Blended Learning. Es relevante, por tanto, conocer la percepción de los estudiantes de esta modalidad, de quienes están tomando el curso y de quienes lo vieron hace varios meses o años. Este artículo analiza la percepción de los estudiantes sobre el curso Arquitecturas de Software Ágiles enseñado en la Univ. de los Andes. La percepción de los estudiantes se midió mediante una encuesta anónima a los estudiantes que estuvieron matriculados en el curso para los períodos 2015-2018. Los resultados mostraron que 39,7% de los encuestados consideraron que aprendieron mejor en comparación con cursos presenciales. Adicionalmente, el aspecto más valorado por los encuestados fue el uso de los problemas de la industria.

Palabras-clave: educación en ingeniería de software; arquitectura de software; metodologías ágiles; blended learning; percepción estudiantes.

Teaching Software Architecture in Blended Learning Model: Analysis of Perception of Master's Students

Abstract: Software architecture, as a key component of software engineering education, requires the student to develop a set of technical and social skills. In the master course Agile Software Architectures it was decided to use the Blended Learning modality. It is relevant, therefore, to know the students' perception of this modality, not only while taking the course but after having seen it. This article analyzes the students' perception of the Agile Software Architectures course taught at University de los Andes. The perception of the students was measured through an anonymous survey of the students who were enrolled in the course for the periods 2015-2018. The results showed that 39.7% of the respondents considered that they learned better compared to face-to-face courses. Additionally, the most valued aspect by respondents was the use of industry problems.

Keywords: software engineering education; software architecture; agile methodology; blended learning; students' perception.

1. Introducción

Como parte de la ingeniería de software, la arquitectura de software es un componente clave de la educación, a nivel de pregrado como a nivel de maestría (Van Deursen et al., 2017; Mannisto, Savolainen & Myllarniemi, 2008; Lago & Van Vliet, 2005; Urrego & Correal, 2013). Es por esto por lo que la enseñanza de la arquitectura de software ha recibido mucha atención tanto en la academia como en la industria, pero sigue siendo una tarea difícil que requiere superar una serie de retos como el trabajo en equipo, habilidades técnicas y sociales y, un contexto realista (Galster & Angelov, 2016; Angelov & de Beer, 2017).

Varias estrategias han sido sugeridas para mejorar la enseñanza de la arquitectura de software, tales como: inclusión de metodologías ágiles (Angelov & de Beer, 2017; Mahnič, 2015; Cleland-Huang, Babar & Mirakhorli, 2014), aula invertida (Cleland-Huang, Babar & Mirakhorli, 2014; Gehringer & Peddycord III, 2013), proyectos capstone (Paasivaara et al., 2018; Mahnič, 2011; Bastarrica, Perovich & Samary, 2017) y clientes reales con problemas reales (Sureka et al., 2013).

Teniendo en cuenta lo anterior, se diseñó el curso Arquitecturas de Software Ágiles (curso de Maestría) usando la metodología Blended Learning. El curso consiste en seis clases presenciales, y nueve clases virtuales. Adicionalmente, el curso hace uso de metodologías ágiles, dividiendo el contenido del curso en cinco sprints, para diseñar la arquitectura de un caso real de la industria. Con esta metodología se busca reforzar el conjunto de habilidades que se esperan del alumno en curso: autoaprendizaje, autocrítica, trabajo en equipo, resolución de problemas y desarrollo profesional.

Luego de varios semestres, se volvió indispensable conocer la percepción de los estudiantes sobre los resultados de aprendizaje de este curso. El objetivo de esta investigación fue conocer si las habilidades que se enseñan en clase consiguen permanecer con los estudiantes mucho después de haber terminado el curso. Para esto, se envió una encuesta a todos los estudiantes que han tomado el curso (227 estudiantes y exestudiantes). Un total de 63 estudiantes respondieron a la encuesta. Por tanto, la motivación detrás de esta investigación es doble: (1) Evaluar la percepción del estudiante con relación a si el curso Arquitecturas de Software Ágiles, en modalidad Blended Learning, apoya el desarrollo de las habilidades se esperan de un arquitecto de software; y (2) evaluar si esta modalidad permite a los estudiantes aprender mejor los conceptos enseñados en este curso.

El presente documento está organizado de la siguiente manera: Una revisión de los trabajos relacionados se presenta en la Sección 2. El diseño del curso se describe en la Sección 3. La metodología de este estudio se presenta en la Sección 4. Los resultados se presentados en la Sección 5. Las amenazas a la validez del presente trabajo son presentadas en la Sección 6. Y finalmente, la discusión de los resultados y las conclusiones se presentan en la Sección 7.

2. Trabajos Relacionados

La literatura presenta diferentes técnicas utilizadas para mejorar el proceso de enseñanza/aprendizaje en los cursos de Ingeniería de Software. La efectividad de estas técnicas se mide en términos de la retroalimentación de los estudiantes.

Angelov y de Beer (Angelov & de Beer, 2017) presentaron un enfoque basado en proyectos ágiles, en un contexto realista y con trabajo en equipo. La percepción de los estudiantes fue evaluada a través de un cuestionario anónimo al final del curso. Los resultados mostraron que su enfoque ayudaba a los estudiantes a percibir mejor la arquitectura de software y a comprender mejor su valor. Un enfoque similar fue presentado por Mahnič (Mahnič, 2011), el cual se enfocó en un curso de pregrado en el que expuso a los estudiantes al desarrollo ágil (SCRUM) de software. El 88% de los estudiantes sintió que el curso era mejor que otros cursos. Siguiendo con los métodos ágiles, Umphress, Hendrix y Cross (Umphress, Hendrix & Cross, 2002) descubrieron que los procesos ágiles eran los más fáciles de aceptar y ejecutar para los estudiantes, con mejores resultados para el éxito del proyecto.

Cleland-Huang, Babar y Mirakhorli (Cleland-Huang, Babar & Mirakhorli, 2014) presentaron sus experiencias en el aula en un estudio de caso utilizando personas expertas en arquitectura de software para enseñar a analizar, diseñar y evaluar sus propuestas arquitectónicas. Gehringer y Peddycord III (Gehringer & Peddycord III, 2013) utilizaron el modelo de aula invertida, en el cual los estudiantes mostraron altos niveles de compromiso. Sureka et al. (Sureka et al., 2013) presentaron un modelo de instrucción docente compuesto de aula invertida, aprendizaje basado en estudio, proyectos de clientes reales y evaluación de grupos grandes y de pares. En este estudio realizaron una encuesta detallada de cada uno de los cuatro elementos del modelo de instrucción. Los resultados de la encuesta mostraron que casi todos los estudiantes estuvieron de acuerdo en que el modelo era interesante y que tuvo un impacto positivo en la motivación.

A nuestro saber y entender, no existen estudios previos que empleen encuestas tiempo después de completar el curso para medir su efectividad en el largo plazo.

3. Diseño del Curso

En esta sección, se describe el contexto del curso Arquitecturas de Software Ágiles utilizando los factores situacionales: contexto general, contexto específico, naturaleza del sujeto y características de los alumnos (Fink, 2003).

3.1. Contexto general del estado de aprendizaje

La formación de los arquitectos de software suele ser como resultado de varios años de experiencia laboral y participación en proyectos de TI. Sin embargo, las empresas necesitan cada vez más profesionales calificados, incluidos los arquitectos de software. Es claro que no es posible sustituir la experiencia por clases universitarias, pero sí es posible acelerar la formación de futuros arquitectos, condensando el conocimiento y las mejores prácticas, y preparando a los estudiantes para trabajos en la industria. En 2011, respondiendo a los requisitos de la industria, Los Andes inició una maestría en Arquitectura de TI. A finales de 2017, más de 180 estudiantes se habían graduado

como arquitectos de TI. En 2013, se creó una Maestría en Ingeniería de Software. Desde entonces, estos dos programas comparten algunos cursos y objetivos de enseñanza, en particular, comparten el curso Arquitecturas de Software Ágiles.

3.2. Contexto específico del estado de aprendizaje

Este curso de arquitectura de software se diseñó con dos objetivos principales: Primero, seguir un enfoque ágil y, segundo, implementar blended learning. El curso tomó 2 años de diseño, comenzando en 2013, para ser lanzado por primera vez en 2015. El curso se centra en el diseño de la arquitectura de un sistema de software propuesto por una empresa real y con actores reales. Los estudiantes implementan cinco sprints durante el semestre, realizando un diseño incremental de su arquitectura. En cada sprint se centran en un atributo de calidad particular.

3.3. Naturaleza del sujeto

Durante cada sprint, los estudiantes diseñan para un atributo de calidad en particular. Esto implica que deben leer e investigar los diferentes aspectos necesarios para resolver un desafío en particular. Una vez que tienen un diseño, se prueban las decisiones críticas, implementando y desplegando prototipos para medir el cumplimiento o no de los requisitos de calidad. En el curso se utilizan varias herramientas de apoyo, tales como: Taiga, para gestionar las historias de requisitos arquitectónicos en el backlog; GitHub Classroom, para almacenar el documento arquitectónico (wiki) y el código fuente; y LucidChart, para crear modelos arquitectónicos.

3.4. Característica de los estudiantes

El público objetivo de este curso son los estudiantes de postgrado en TI de ambos másteres (Arquitectura de TI e Ingeniería de Software), generalmente de unos 20 años de edad y con 4 años de experiencia como desarrolladores o arquitectos de software.

4. Metodología

Una vez establecidas las preguntas de la investigación y teniendo en cuenta los objetivos de este estudio (Easterbrook, 2008), se decidió utilizar la encuesta como instrumento de investigación. Las fuentes de este estudio se ponen a disposición en Google Drive (https://goo.gl/T9hgc6) con (i) el formulario de la encuesta, (ii) un archivo Excel con todas las respuestas, y (iii) un archivo Excel con las respuestas codificadas.

4.1. Preguntas de investigación

El objetivo de esta investigación es descubrir si la modalidad Blended learning en este curso permite mejorar el proceso de aprendizaje en arquitectura de software. Este trabajo está guiado por las siguientes preguntas de investigación:

> **PI1:** *¿Cuál es la percepción que tienen los estudiantes del curso de maestría Arquitectura de Software Ágiles respecto a la modalidad Blended Learning?*
>
> **PI2:** *¿De qué manera la modalidad Blended Learning permite a los estudiantes aprender de mejor manera los conceptos enseñados en el curso?*

4.2. Diseño de la encuesta

Esta encuesta recoge datos subjetivos (opiniones, actitudes y preferencias del individuo) y demográficos (Kitchenham & Pfleeger, 2008). Adicionalmente, se clasifica como de investigación exploratoria y descriptiva porque se enfoca en el descubrimiento de perspectivas, fue planeada y estructurada previamente, y la información recolectada puede ser inferida estadísticamente sobre una población.

En esta encuesta, la población objetivo se limitó a los profesionales que tomaron el curso de Arquitecturas de Software Ágiles en la Universidad. Esta encuesta empleó un método de muestreo no probabilístico porque la población fue seleccionada con base en el juicio subjetivo del investigador en lugar de una selección aleatoria. A raíz de esto, no es posible establecer el margen de error de la muestra. Los encuestados fueron fácilmente accesibles para los investigadores de esta encuesta. Snowball Sampling no fue considerado (Kitchenham & Pfleeger, 2008).

4.3. Creación de la encuesta

Las preguntas de la encuesta se basaron en las dos preguntas de investigación (ver Sección 4.1.). La encuesta se realizó en Google Forms y está compuesta por 24 preguntas: Cinco preguntas demográficas (DQ1-DQ5) y diecinueve preguntas específicas, de las cuales dieciséis son preguntas cerradas relacionadas con la percepción sobre la metodología (MQ1-MQ16) y tres preguntas abiertas sobre la percepción general (PQ1-PQ3). En relación con las preguntas cerradas, catorce preguntas constan de una escala de 5 puntos tipo Likert. Una pregunta tiene una escala de 3 puntos tipo Likert y una pregunta tiene un tipo de respuestas sí/no. Las preguntas demográficas se utilizaron para recopilar información personal y laboral y se requirieron para obtener una comprensión más profunda sobre los antecedentes de los participantes.

El protocolo de esta encuesta fue refinado de forma iterativa por los autores. Luego de esto, el cuestionario fue revisado externamente por un profesor a cargo del curso, así como por dos profesionales en arquitectura de software. Los dos profesionales comprobaron el contenido y la comprensión de las preguntas, así como el tiempo necesario para completar la encuesta. Finalmente, la encuesta fue refinada con una prueba piloto con cinco participantes. Los participantes del piloto eran representativos de los encuestados potenciales de esta encuesta.

La encuesta se aplicó en el período comprendido entre el 28 de agosto de 2018 y el 12 de septiembre de 2018. La encuesta fue enviada por correo electrónico incluyendo un mensaje de bienvenida para explicar el propósito de la encuesta, afiliación de los autores e información de contacto. Para mantener la encuesta lo más simple posible, presentamos todas las preguntas en una sola página.

4.4. Análisis de los datos

Para las preguntas cerradas (respuestas cuantitativas) se utilizaron estadísticas descriptivas. Las respuestas a las preguntas abiertas se analizaron utilizando un esquema de código y extrayendo los puntos principales a una hoja de Excel.

5. Resultados

En esta sección se presentan los resultados de la percepción de los estudiantes relacionada con el curso Arquitecturas de Software Ágiles. Los principales resultados de la encuesta se presentan en las Figuras 1 a 4. Primero se presenta la información demográfica, seguido de las respuestas a las preguntas cerradas. Finalmente, se presentan los resultados de las preguntas abiertas.

5.1. Datos demográficos

Se enviaron 227 invitaciones, entre estudiantes y egresados del curso. En total se recibieron 63 respuestas, lo que equivale a 27,7% de la población. Para los autores, esta es una tasa importante teniendo en cuenta que la mayoría de encuestados no han estado vinculados a la Universidad durante uno o más años.

Respecto a los datos demográficos, el mayor porcentaje de los encuestados (20 de 63, 32%) tienen entre 31 y 36 años, seguidos por el rango entre 26 y 30 años (19 de 63, 30%). La gran mayoría de los encuestados son hombres (57 de 63, 90,5%), y casi la mitad de los encuestados (31 de 63, 49,2%) son profesionales sin posgrado. En cuanto al rol en las empresas, la mayoría de los encuestados seleccionaron "Desarrollo" (28 de 63, 44,4%) como su tarea principal, seguida por "Arquitectura de software" con 12 (19%) encuestados. Finalmente, el mayor porcentaje de los encuestados manifestaron trabajar en el sector "Tecnología" (24 de 63, 38,1%), seguidos por sector "Gobierno" (7 de 63, 11.1%) y "Finanzas" (6 de 63, 9.5%).

5.2. Percepción de los estudiantes a través de las preguntas cerradas

En términos generales, los alumnos coincidieron en la importancia y satisfacción de utilizar la modalidad Blended Learning para aprender sobre arquitectura de software (Figura 1). Esta satisfacción se evidencia con 46 respuestas positivas (*Muy de acuerdo*: 23; *De acuerdo*: 23). Sin embargo, la efectividad de esta modalidad de enseñanza fue la única pregunta con una mediana más baja que las otras respuestas. Los estudiantes prefieren estar en constante interacción con el profesor para aclarar los problemas o tener una mejor comprensión de los conceptos. Esto está también asociado a la experticia o familiaridad del estudiante con el tema.

Esta encuesta buscaba entender si la modalidad utilizada era útil para mejorar las habilidades como la autocrítica, el autoaprendizaje, el trabajo en equipo, la resolución de problemas y el desarrollo profesional.

Autoaprendizaje

En relación con esta habilidad, casi el 75% (39,7% *De acuerdo* y 34,9% *Muy de acuerdo*) de los encuestados consideraron que la modalidad Blended learning soportó el desarrollo de esta habilidad. Sólo un (1) encuestado estuvo *Muy en desacuerdo* con esta pregunta, sin embargo, este mismo encuestado manifestó que lo que más le gustó del curso fue precisamente el aprendizaje autónomo. Esto significa que el estudiante tuvo la suficiente curiosidad como para explorar los temas por sí mismo. En ese sentido, el curso sí promovió el autoaprendizaje.

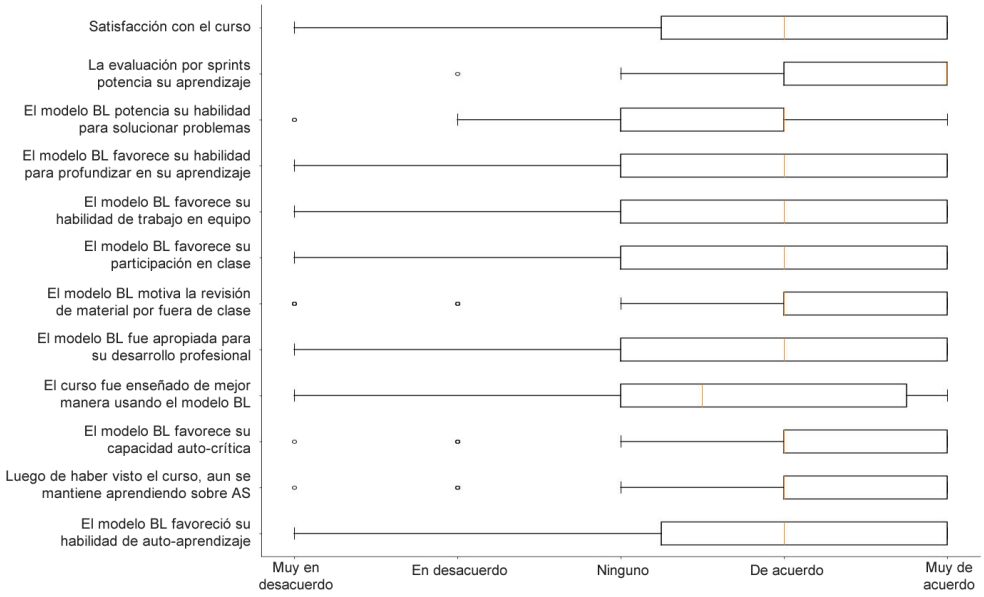

Figura 1 – Nivel de satisfacción con la modalidad Blended learning

Autocrítica

Casi el 75% (36,5% *De acuerdo* y 36,5% *Muy de acuerdo*) de los encuestados manifestaron que la modalidad Blended learning soportó el desarrollo de esta habilidad. Al preguntarles sobre lo que más les gustó del curso, la mayoría de las respuestas se centraron en la modalidad, el contenido del curso y la experiencia industrial del profesor. Esto significa que el fomento de esta habilidad fue un beneficio derivado de la modalidad de enseñanza que no estaba conscientemente presente en la mente de los estudiantes.

Trabajo en equipo

El trabajo en equipo es el esfuerzo de colaboración entre individuos interdependientes para lograr un objetivo común (Glendon & Clarke, 2015). Esta habilidad requiere tratar con el conocimiento, la actitud y la personalidad de los demás. El 36,5% y el 30,2% de los encuestados estuvieron *De acuerdo* y *Muy de acuerdo*, respectivamente, sobre el fomento a esta habilidad. Sólo 7 (11.1%) encuestados estuvieron *Muy en desacuerdo*. Adicionalmente, este último grupo de encuestados (*Muy en desacuerdo*) criticaron el aspecto semi-presencial del curso. Otro grupo de estudiantes se quejaron de la falta de compromiso de sus compañeros, como por ejemplo: "Que en algunos casos, los compañeros no trabajaron en la misma proporción y eso afecto la entrega".

Solución de problemas

La resolución de problemas requiere una comunicación efectiva y continua con otros compañeros. Si esta comunicación falla o se vuelve lenta, existe el riesgo de perder la

continuidad del problema a resolver y de no encontrar la solución óptima. Casi el 50% (31 de 63) de los encuestados estuvieron de acuerdo en que esta modalidad de enseñanza fomentó esta habilidad. *En desacuerdo* y *Muy en desacuerdo* fueron 6,3% (4 de 63) y 3,2% (2 de 63) respectivamente. Los encuestados de este grupo de percepción negativa criticaron el pequeño número de clases presenciales del curso.

Desarrollo profesional

El análisis de esta habilidad es importante porque quería establecerse si los conocimientos impartidos en clase tuvieron un impacto en la vida profesional de los estudiantes. En esta pregunta, los encuestados estuvieron *Muy de acuerdo* con el 34,9% (22 de 63) y *De acuerdo* con el 31,7% (20 de 63). *Muy en desacuerdo* con el 7,9% de los encuestados (5 de 63).

5.3. Percepción de los estudiantes a través de las preguntas abiertas

Según Reja (Reja et al., 2003), las preguntas abiertas ayudan a evitar el sesgo que puede resultar de sugerir respuestas al encuestado. Por otro lado, las preguntas abiertas requieren un sistema de codificación para organizar las respuestas y permitir su análisis.

En esta encuesta se utilizaron tres preguntas abiertas: (PQ1) ¿Qué considera que se podría mejorar del curso? (PQ2) ¿Qué no le gustó acerca del curso? Y (PQ3) ¿Qué le gustó del curso? La primera pregunta tuvo 34 respuestas válidas, la segunda tuvo 30 y la tercera tuvo 41 respuestas válidas. Es importante señalar que una respuesta podría codificarse en dos o más categorías debido a su contenido.

PQ1 "¿Qué considera que se podría mejorar del curso?"

Las respuestas se codificaron en 7 categorías: Más casos de la industria, más comunicación con el profesor, más material de trabajo, más clases presenciales, más tareas, mejor organización del tiempo de clase y otros (Figura 2).

Figura 2 – Codificación de respuestas para PQ1

La categoría con mayor número de respuestas fue "Mas ejemplos de casos reales de la industria". Estos casos reales permiten a los estudiantes entender los contextos bajo los cuales se tomaron las decisiones arquitectónicas. A veces, los arquitectos toman

decisiones que no son óptimas, pero bajo ciertas condiciones pueden resultar en decisiones exitosas. Además, está claro que la teoría y la práctica no siempre van de la mano. Algunas respuestas fueron: "... falto la experiencia sobre situaciones reales..." y "Ejemplos de arquitecturas aplicadas en casos reales".

Los encuestados también expresaron su necesidad de una mejor y más fluida comunicación con el profesor. La formulación de preguntas es parte del proceso de aprendizaje, y muchas veces, la respuesta es una pieza fundamental en la consolidación del conocimiento. Otra categoría, "Más clases en el lugar" es una categoría que se esperaba después de revisar las preguntas cerradas. Algunos estudiantes se sintieron incómodos con el aspecto semipresencial del curso porque necesitan tener más actividades cara a cara con el profesor para responder a sus preguntas. Las clases semipresenciales requieren mucha autodisciplina por parte y, según las respuestas, no todos la tienen. Algunas respuestas fueron: "El tiempo de respuesta a través de los foros hacerlo más corto" y "Más sesiones presenciales. Más contenido técnico".

PQ2 "¿Qué no le gustó del curso?"

Las respuestas se codificaron en 6 categorías: Modalidad blended, poca interacción con el profesor, fallas de los monitores del curso, metodología en general, evaluación y otros. Los resultados de la codificación se pueden ver en la Figura 3.

Fue interesante descubrir que la modalidad Blended learning fuera lo que menos gustara del curso. Algunas de las respuestas permitieron entender por qué los estudiantes se sienten incómodos con este aspecto: "Este modo de enseñanza requiere cierto nivel de compromiso y madurez profesional para aprovechar sus beneficios..." y "La modalidad blender no me gusto, en ocasiones se sentia que se perdian cosas importantes de los temas de la clase...".

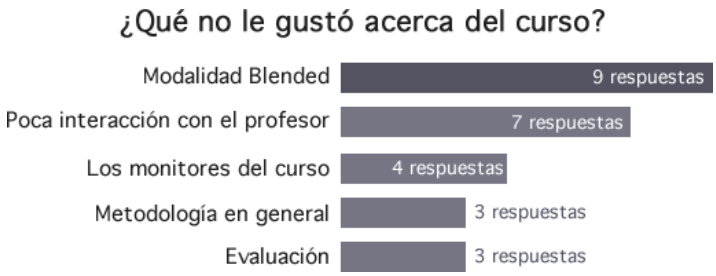

Figura 3 – Codificación de respuestas para PQ2

Otra categoría, "Fallas de los monitores del curso", fue una categoría inesperada. Esto evidencia que los estudiantes evalúan un curso desde todas las perspectivas. Además, esta categoría refuerza el concepto de que las preguntas abiertas permiten al entrevistador encontrar más de lo que anticipa. Algunas respuestas fueron: "En ese momento el profesor asistente nos enredo en unas tutorias que hizo" y "A veces el monitor no explicaba bien".

PQ3 "¿Qué le gustó del curso?"

Las respuestas se codificaron en 8 categorías: Metodología, contenido del curso, conocimientos del profesor, desarrollo de habilidades, todo, cercanía a la realidad y conceptos ágiles. Los resultados de la codificación se pueden ver en la Figura 4.

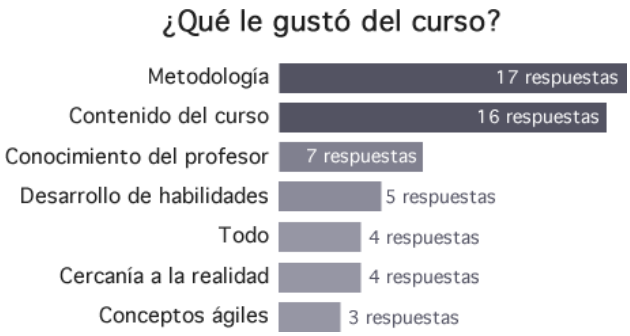

Figura 4 – Codificación de respuestas para PQ3

La modalidad blended fue el aspecto más valorado por los encuestados. En relación con esto, una de las preguntas cerradas fue qué tanto aprendió a través de esta modalidad. El resultado mostró que el 39,7% (25 de 63) de los encuestados consideraron que aprendieron más, el 34,9% que aprendieron lo mismo, y el 25,4% que aprendieron menos. Esto puede significar que les gustó la modalidad, los animó, y les permitió involucrarse más en la dinámica del curso, pero no necesariamente que los hiciera aprender más. Algunas respuestas fueron: "Auto aprendizaje, aplicar conocimientos en escenarios practicos…" y "La metodología de sprints, la temática y el material fueron muy buenos".

"El contenido del curso" fue otro aspecto apreciado por los estudiantes. En relación con esto, una de las preguntas cerradas fue si el estudiante consideraba que el contenido del curso se enseñaba mejor usando la metodología aplicada. El 27% (17 de 63) respondió *Ninguno*. Por otro lado, *De acuerdo* y *Muy de acuerdo* fueron respondidos por 23,8% (15 de 63) y 25,4% (16 de 63) de los estudiantes. Esto significaría que el contenido respondió a las necesidades de la industria, pero la modalidad de enseñanza no correspondió a una mejor manera de enseñarlo.

6. Amenazas a la validez

En este estudio, la metodología de investigación fue validada para evitar sesgos en la medida de lo posible. A continuación, se presentan las amenazas junto con sus estrategias de mitigación.

Respecto a la *validez de aspecto*, la redacción de las preguntas cerradas suele ser la amenaza más recurrente en las encuestas. Para mitigarlo, se evaluó la encuesta internamente varias veces para luego refinar el lenguaje utilizado. Respecto a la *validez interna*, se realizó un análisis de los datos utilizando estadísticas descriptivas básicas

y se realizó un análisis cruzado de las respuestas de cada participante. Respecto a la *validez externa*, se identificó como amenaza potencial la distribución demográfica de la población. Para esto, se aplicó el muestreo de conveniencia para ayudar a seleccionar a los participantes del estudio. Sin embargo, este estudio no es generalizable porque no fue replicado a otros países. Finalmente, respecto a la *confianza*, se empleó la triangulación de observadores, con la participación de todos los autores en los procesos de extracción y análisis de datos. Se evitó realizar cualquier análisis de correlación estadística porque el tamaño de la muestra no fue lo suficientemente grande.

7. Discusión y conclusiones

Este trabajo reportó las percepciones de los estudiantes sobre los resultados de aprendizaje del curso Arquitecturas de Software Ágiles usando Blended learning. La percepción se midió encuestando a los estudiantes que están tomando/tomaron el curso desde el 2015 hasta el primer semestre del 2018. De 227 invitaciones enviadas, 63 respondieron la encuesta. A partir de los resultados se estableció que la percepción de los estudiantes fue abrumadoramente positiva. También se midió la percepción en términos de un conjunto de habilidades que debe tener un graduado de este curso: autoaprendizaje, autocrítica, trabajo en equipo, resolución de problemas y desarrollo profesional.

En términos generales, los estudiantes estuvieron de acuerdo en la importancia y satisfacción de usar Blended learning para aprender sobre arquitectura de software. Sin embargo, la efectividad de esta modalidad de enseñanza para mejorar el aprendizaje de arquitectura de software fue neutra (17 de 63, 27%). Esto podría deberse a las pocas clases presenciales del curso, que fue el aspecto más criticado en las preguntas abiertas. Los estudiantes manifestaron que preferían estar en constante interacción con el profesor para aclarar los problemas o tener una mejor comprensión de los conceptos.

En relación con el conjunto de habilidades que debe tener un graduado de este curso, los encuestados estuvieron de acuerdo en que esta modalidad fomenta el autoaprendizaje y la autocrítica. Corroborado por las respuestas abiertas, fue posible establecer que la autocrítica fue un beneficio derivado de la modalidad Blended learning. El trabajo en equipo y el desarrollo profesional también fueron acordados por la mayoría de los encuestados. Sin embargo, algunos de los encuestados se quejaron de la falta de compromiso de sus pares.

A partir de las respuestas a las preguntas abiertas, se concluyó que los estudiantes solicitan más casos reales de la industria para ser incluidos en el curso. Esta es una petición importante porque los problemas en arquitectura de software sólo son visibles a escala, y desaparecen cuando se utilizan en proyectos pequeños de ejemplo (Van Deursen et al., 2017). Los encuestados también pidieron más y mejor comunicación con el profesor. El foro no es suficientemente rápido y a veces lleva a más preguntas. Esta es una preocupación clave para los estudiantes con pocos conocimientos sobre arquitectura de software en general. Ni siquiera el correo electrónico es suficiente para responder a las preguntas.

La modalidad Blended learning requiere mucha autodisciplina, y desafortunadamente, no todo el mundo la tiene. Esto depende del nivel de experiencia del estudiante con

el tema. Si el alumno tiene alguna experiencia o conocimiento sobre el tema, le será más fácil trabajar por su cuenta. Siguiendo con la modalidad, los resultados mostraron también que el 39,7% de los encuestados sintieron que aprendieron más, el 34,9% sintieron que aprendieron lo mismo, y el 25,4% sintieron que aprendieron menos. Esto puede significar que les gustó la técnica, los animó y les permitió involucrarse más en la dinámica del curso, pero no necesariamente los hizo aprender más. Sin embargo, un mejor compromiso con un tema aumenta su retención de memoria a largo plazo. Con base en esto, es posible establecer que los estudiantes aprendieron más usando esta modalidad, pero sería mucho mejor decir que los estudiantes aprendieron mejor.

Este estudio deja interesantes preguntas abiertas que requieren mayor investigación. En particular, ¿qué aspectos de la modalidad Blended learning pueden mejorarse? ¿Qué técnicas se pueden aplicar para mejorar la comunicación entre los estudiantes y el profesor? ¿Cómo se puede fomentar el autoaprendizaje en estudiantes con poco conocimiento de una asignatura en particular?

Para el trabajo futuro animamos a otros investigadores y profesores a estudiar cómo la combinación de técnicas de enseñanza puede mejorar el proceso de aprendizaje en arquitectura de software.

Referencias

Angelov, S., & de Beer, P. (2017). Designing and applying an approach to software architecting in agile projects in education. Journal of Systems and Software, 127, 78-90.

Bastarrica, M. C., Perovich, D., & Samary, M. M. (2017, May). What can students get from a software engineering capstone course?. In 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET) (pp. 137-145). IEEE.

Cleland-Huang, J., Babar, M. A., & Mirakhorli, M. (2014, May). An inverted classroom experience: engaging students in architectural thinking for agile projects. In Companion Proceedings of the 36th International Conference on Software Engineering (pp. 364-371). ACM.

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. In Guide to advanced empirical software engineering (pp. 285-311). Springer, London.

Fink, L. D. (2003). A self-directed guide to designing courses for significant learning. University of Oklahoma, 27(11).

Galster, M., & Angelov, S. (2016, May). What makes teaching software architecture difficult?. In 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C) (pp. 356-359). IEEE.

Gehringer, E. F., & Peddycord III, B. W. (2013, March). The inverted-lecture model: a case study in computer architecture. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 489-494). ACM.

Glendon, A. I., & Clarke, S. (2015). Human safety and risk management: A psychological perspective. Crc Press.

Kitchenham, B. A., & Pfleeger, S. L. (2008). Personal opinion surveys. In Guide to advanced empirical software engineering (pp. 63-92). Springer, London.

Lago, P., & Van Vliet, H. (2005, April). Teaching a course on software architecture. In 18th Conference on Software Engineering Education & Training (CSEET'05) (pp. 35-42). IEEE.

Mahnič, V. (2011). A capstone course on agile software development using Scrum. IEEE Transactions on Education, 55(1), 99-106.

Mahnič, V. (2015). Scrum in software engineering courses: an outline of the literature. Global Journal of Engineering Education, 17(2), 77-83.

Mannisto, T., Savolainen, J., & Myllarniemi, V. (2008, February). Teaching software architecture design. In Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008) (pp. 117-124). IEEE.

Paasivaara, M., Vodă, D., Heikkilä, V. T., Vanhanen, J., & Lassenius, C. (2018, May). How does participating in a capstone project with industrial customers affect student attitudes?. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training (pp. 49-57). ACM.

Reja, U., Manfreda, K. L., Hlebec, V., & Vehovar, V. (2003). Open-ended vs. close-ended questions in web questionnaires. Developments in applied statistics, 19(1), 159-177.

Umphress, D. A., Hendrix, T. D., & Cross, J. H. (2002). Software process in the classroom: The capstone project experience. IEEE software, 19(5), 78-81.

Urrego, J. S., & Correal, D. (2013, May). Archinotes: A tool for assisting software architecture courses. In 2013 26th International Conference on Software Engineering Education and Training (CSEE&T) (pp. 80-88). IEEE.

Van Deursen, A., Aniche, M., Aué, J., Slag, R., De Jong, M., Nederlof, A., & Bouwers, E. (2017, March). A collaborative approach to teaching software architecture. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (pp. 591-596). ACM.

Sureka, A., Gupta, M., Sarkar, D., & Chaudhary, V. (2013). A case-study on teaching undergraduate-level software engineering course using inverted-classroom, large-group, real-client and studio-based instruction model. arXiv preprint arXiv:1309.0714.

© 2020. This work is published under https://creativecommons.org/licenses/by-nc-nd/4.0/(the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License.