

Comparativa de Dos Técnicas Basadas en Roles para el Desarrollo de Software desde una Perspectiva Industrial

Silvia Rueda Pascual¹, Juan Cabotà i Soro¹, Jose Ignacio Panach Navarrete¹, Javier Martínez Plume², Mariano Pérez Martínez¹, Sergio Casas Yrurzum²

¹Departament d'Informàtica, Escola Tècnica Superior D'Enginyeria, Universitat de València, 46100, Burjassot, Valencia.

²Instituto de Robótica y de las TICs. Parque Científico, Universitat de València, C/ Catedrático José Beltrán 2, 46980, Paterna, Valencia.

{silvia.rueda,juan.cabota,j.ignacio.panach,javier.martinez-plume, mariano.perez,sergio.casas}@uv.es

Abstract

Both the degree in Telematic Engineering and the degree in Multimedia Engineering have a subject of Software Engineering. One competence of this subject is to train students as professionals for the real world. For this aim, we have applied two versions of a role-playing game where there are both clients, who ask for a system development project, and developers. In the degree in Telematic Engineering, the teacher plays the role of client, while students play the role of developers. Regarding the degree of Multimedia Engineering, students play both the role of client and developer at the same time (for different projects). In this paper, we tackle both experiences and we show a comparative of results. For Telematic Engineering, we ensure that the workload is the same for all the students. Moreover, the teacher can control directly the work of the students since he/she knows perfectly the project that needs to be developed. Regarding Multimedia Engineering, we try to enhance the students' creativity since they must devise themselves a project to develop. Moreover, the students learn how to develop both roles, which can be useful for the real world.

Keywords: role-playing, software engineering, software projects

Resumen

Dentro de los grados de Ingeniería Telemática e Ingeniería Multimedia se imparte la asignatura de Ingeniería del Software. Una de las competencias de esta asignatura es formar a los alumnos como profesionales para el mundo real. Para ello, hemos aplicado dos variantes de un juego de rol donde hay clientes, que solicitan un proyecto informático, y desarrolladores. En el Grado de Ingeniería Telemática (GIT), el rol de usuario final lo tomaba un único profesor, mientras que los usuarios jugaban el rol de desarrolladores. En el Grado de Ingeniería Multimedia (GIM), los estudiantes jugaban a la vez el rol de cliente y de desarrolladores (para proyectos distintos). En este artículo se abordan ambas experiencias y se presenta una comparativa de los resultados obtenidos con ellas. En el caso de GIT se garantiza que la carga de trabajo es para todos los estudiantes la misma. Además, el profesor puede llevar un seguimiento más directo al

conocer de primera mano el caso a desarrollar. En el caso de GIM se potencia la creatividad de los alumnos, ya que deben inventarse el proyecto a desarrollar. Además, los alumnos aprenden a desempeñar dos roles que les pueden ser útiles en el mundo real.

Palabras clave: juego de roles, ingeniería del software, proyectos software

1. Introducción

Una de las competencias del Grado de Ingeniería Telemática (GIT) y del Grado de Ingeniería Multimedia (GIM) de la Universidad de Valencia es la de formar a profesionales que se incorporen al mundo profesional como analistas de sistemas para el desarrollo software. En la asignatura de Ingeniería del Software, presente en ambas titulaciones, se imparten conocimientos sobre el desarrollo software. Normalmente, se suele partir de problemas que se presentan a los alumnos por escrito, donde vienen todos los requisitos especificados de forma inequívoca y detallada. Esta forma de proceder es cómoda tanto para los alumnos como para el profesorado porque se sabe a qué atenerse a la hora de desarrollar un sistema. Sin embargo, es muy poco realista. En el mundo real, los analistas se basan en entrevistas con los clientes a través de las cuales extraen los requisitos de forma oral. Las especificaciones de requisitos escritas son desarrolladas por los propios analistas tras varias entrevistas con el cliente.

Existen trabajos previos que han abordado estudios sobre qué técnicas son mejores para impartir la asignatura de Ingeniería del Software. Uno de estos trabajos es el desarrollado por Beecham et al. [4]. Según este trabajo, la preparación, la docencia y la evaluación del trabajo del alumno en entornos educativos de ingeniería del software requiere mucho esfuerzo y estrategia por parte del docente si los comparamos con otras asignaturas. Según Beecham et al., simulaciones del mundo real que estén bien diseñadas pueden servir para mostrar a los alumnos cómo funcionan los proyectos de desarrollo reales.

Siguiendo esta filosofía, durante el curso 2015-2016 pusimos en práctica un juego de rol en la asignatura de Ingeniería del Software en la titulación GIT. En [5] se publicaron los resultados iniciales obtenidos durante esa primera experiencia. De forma resumida, las conclusiones fueron que el juego de roles hacía que los estudiantes obtuvieran menos requisitos de los que realmente existían, pero la calidad respecto a la corrección de la especificación era mejor con el juego de roles.

Para contrastar estas ideas, durante el curso 2016-2017 hemos puesto en marcha una replicación de esta idea en GIT y GIM. Según los trabajos de Basili [3], son necesarias varias replicaciones de la misma experiencia para obtener resultados significativos. La replicación en GIT ha sido exactamente igual que la utilizada en el curso 2015-2016, mientras que en la titulación GIM se ha modificado. En ambas titulaciones la propuesta está basada en la existencia de dos roles: (1) cliente, que es la persona que solicita el desarrollo de un proyecto informático; (2) desarrollador, que es la persona que extrae los requisitos del cliente y lleva a cabo el desarrollo. Las diferencias a la hora de aplicar el juego de roles en cada una de las titulaciones es la siguiente:

- GIT: en esta titulación el rol de cliente lo ha jugado un único profesor, de forma que todos los estudiantes han desarrollado el mismo proyecto.
- GIM: en esta titulación el rol de cliente lo jugaban los mismos estudiantes, junto con el rol de desarrollador. Cada equipo de estudiantes se inventaba un proyecto informático que debían explicar a otro equipo. A su vez, esos mismos alumnos jugaban el rol de desarrollador para otro cliente que les explicaba a ellos el proyecto que debían desarrollar. De esta forma, ponían en práctica ambos roles.

En ambos casos, los estudiantes debían concertar entrevistas con el cliente y extraer los requisitos que debía tener el sistema. El trabajo de los desarrolladores consistía en extraer los requisitos del cliente y reportarlos en diagramas de Casos de Uso y en un documento de especificación de requisitos. En este artículo vamos a analizar la comparativa de ambas técnicas para aplicar el juego de roles, destacando las ventajas e inconvenientes de cada una de ellas.

Analizando los datos extraídos, podemos afirmar de forma resumida que la ventaja de la técnica usada en GIT es que los requisitos están cerrados y es sencillo hacer un seguimiento. Por el contrario, la dedicación del profesor es elevada, ya que debe concertar entrevistas con todos los alumnos. La ventaja de la técnica usada en GIM es que los alumnos viven dos experiencias distintas que les pueden ser útiles en el mundo real. Como contrapartida, el profesor tiene menos control sobre los proyectos desarrollados y tiene que evaluar varios proyectos distintos.

El artículo se compone de las siguientes secciones. En la Sección 2 se describen los objetivos que hay detrás de la propuesta del juego de roles. En la Sección 3 se explican los detalles de las dos técnicas aplicadas en GIT y GIM. En la sección 4 se hace una comparativa de resultados de ambas técnicas. Por último, en la Sección 5 se abordan las conclusiones.

2. Objetivos

El principal objetivo de la incorporación de un juego de rol en la asignatura de Ingeniería del Software es enseñar a los alumnos cómo extraer los requisitos software de los clientes a través de entrevistas reales, cómo especificarlos en un documento y cómo defender estos resultados aprendiendo a gestionar los tiempos de manera adecuada. Esta forma de trabajar es la que se encontrarán en el mercado laboral una vez finalicen los estudios. Este objetivo se enmarca dentro de las siguientes líneas de actuación estratégicas de la Universidad de Valencia:

- (L1): Fomento de la participación de los estudiantes en la mejora de los procesos de aprendizaje.
- (L2): Realización de actividades que faciliten la conexión de los egresados con el mercado de trabajo.
- (L3): Mejora de los planes de estudio con la vista puesta en la acreditación y obtención del sello de calidad.

Este objetivo se ha aplicado en la titulación GIT y GIM de la Universidad de Valencia en grupos de 43 y 40 alumnos respectivamente dentro de la asignatura de Ingeniería del Software.

3. Desarrollo de la Innovación

El juego de roles en la docencia de la Ingeniería del Software y más concretamente en la captura de requisitos, es una técnica que se ha aplicado anteriormente en otros trabajos, como el de Al-Ani y Yusop [2]. La aproximación seguida es que uno de los profesores juega el rol de cliente, mientras que los alumnos juegan el rol de analistas desarrolladores. El cliente no es informático y por tanto desconoce el lenguaje UML, por lo que los analistas no pueden basar sus preguntas o correcciones en estos diagramas. Toda la entrevista se debe hacer de forma verbal, aunque se acepta la validación de prototipos.

La innovación educativa consiste en sustituir los enunciados de un proyecto software por entrevistas reales con un profesor o con un grupo de compañeros, de forma que los requisitos del proyecto se deben extraer a partir de estas entrevistas entre clientes y analistas. A continuación, detallamos el proceso seguido en las titulaciones GIT y GIM.

3.1 GIT: Proceso usando un único profesor con el rol de cliente

En esta titulación se ha aplicado la misma técnica que se aplicó en el curso anterior (2015-2016). En este caso, el rol de cliente lo juega un único profesor y los alumnos juegan todos el rol de desarrolladores. A continuación se describen los pasos, resumidos de forma gráfica en la Fig. 1:

1. Los alumnos se agrupan en grupos de 5 a 6 personas. Estos grupos serán fijos para el desarrollo de todo el proyecto.
2. Cada uno de los grupos concierta una entrevista inicial con el profesor que juega el rol de cliente. Los estudiantes deben acudir a la entrevista teniendo preparadas una serie de preguntas para guiar la reunión. Durante la entrevista, el cliente explica las necesidades que tiene respecto al nuevo sistema software a desarrollar. Los analistas pueden preguntar en cualquier momento cualquier duda que les surja.
3. Después de esta entrevista inicial, los analistas empiezan a elaborar un diagrama de Casos de Uso de UML [1] junto con un Documento de Especificación de Requisitos para registrar los requisitos capturados.
4. Bajo demanda de los desarrolladores, el cliente se puede reunir más veces con los desarrolladores, con un máximo de 3 reuniones totales por grupo. Durante las sucesivas reuniones, el cliente puede resolver cualquier duda planteada por los desarrolladores, pero no puede corregir o completar información expresada mediante los Casos de Uso o el Documento de Especificación de Requisitos. Sí se acepta la corrección de prototipos en caso de que alguno de los grupos los desarrolle (en ningún caso se les obliga a hacer uso de prototipos).
5. Al finalizar las entrevistas, los alumnos deben entregar la siguiente información:
 - a. Diagrama de Casos de Uso, con las plantillas en formato de Alto Nivel.
 - b. Documento de Especificación de Requisitos.

- c. Plantilla en Excel con información referente al esfuerzo que les ha supuesto capturar los requisitos y elaborar la documentación.
- 6. Finalmente los alumnos hacen una presentación de resultados en público.

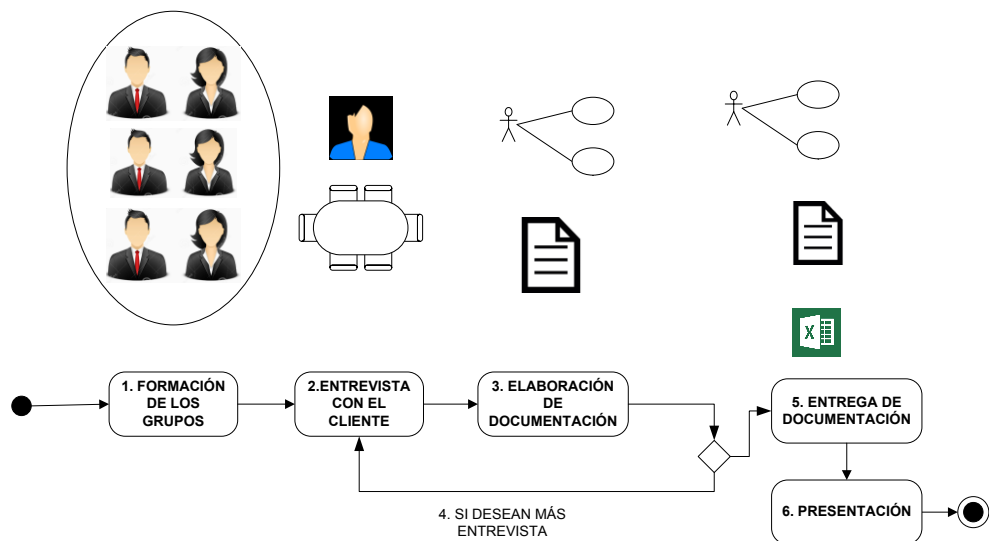


Fig. 1. Diagrama de Actividad con las actividades del juego de roles

Para llevar a cabo el juego de roles utilizamos una serie de instrumentos:

- **Hoja Excel donde capturar datos sobre el avance del trabajo de los estudiantes:** Esta hoja tenía celdas que los propios estudiantes debían rellenar sobre cómo habían hecho la captura de requisitos. En concreto se preguntaban los siguientes elementos:

Tabla 1. Hoja Excel

| | |
|-------------|--------------------------------------|
| Cantidad | Nº requisitos funcionales |
| | Nº requisitos no-funcionales |
| | Nº total requisitos |
| | Nº páginas |
| Compleitud | Nº requisitos que coinciden profesor |
| Rendimiento | Nº preguntas |
| | Nº preguntas relevantes |
| Diversidad | Nº requisitos solapados |
| Calidad | Nº requisitos nuevos |
| | Nº requisitos mal especificados |

Todos estos datos persiguen determinar de manera independiente al criterio subjetivo del profesor, en qué medida los alumnos han elaborado unos DERS adecuados, si han perdido mucho tiempo, si han dedicado un esfuerzo suficiente, etc. Además, a partir de estos datos se pretende definir una medida de la calidad de la solución obtenida, independiente de criterios subjetivos de los profesores. Esta medida de calidad Q , se obtiene comparando los datos de los estudiantes con los datos de los documentos obtenidos por los profesores. Para ellos se tuvieron en cuenta sólo los valores de completitud y calidad de los datos recogidos de los alumnos, tal y como puede verse en la ecuación (1), pues son los que más se ajustan al objetivo perseguido.

$$\begin{aligned} COMP &= \text{N}^\circ \text{ requisitos que coinciden profesor} / \text{N}^\circ \text{ total requisitos profesor} \\ NEW &= \text{N}^\circ \text{ requisitos nuevos} / \text{N}^\circ \text{ total requisitos profesor} \\ BAD &= \text{N}^\circ \text{ requisitos mal especificados} / \text{N}^\circ \text{ total requisitos profesor} \\ Q &= COMP + NEW - BAD \end{aligned} \quad (1)$$

- **Proyecto a desarrollar:** el proyecto a desarrollar era un sistema de gestión de vuelos. Los usuarios deben poder comprar billetes, crear vuelos, modificar vuelos, gestionar trayectos, etc.
- **Herramienta para modelar:** se ha utilizado Visual Paradigm como herramienta de modelado UML para construir los diagramas de Casos de Uso.
- **Repositorio SVN:** para que todos los miembros del equipo pudieran trabajar de forma simultánea y colaborativamente, hemos utilizado un repositorio SVN con el que compartir los modelos UML desde la propia herramienta Visual Paradigm.

3.2 GIM: Proceso en el que los estudiantes juegan ambos roles, el de cliente y el de desarrollador.

Los alumnos de esta titulación desempeñan tanto el rol de cliente, como el de desarrolladores. A continuación, se describe el procedimiento seguido, representado gráficamente en la Fig. 2:

Sesiones de teoría:

1. Se forman grupos de trabajo de 4 a 6 estudiantes, que serán fijos para el desarrollo de todo el proyecto. Los miembros de cada grupo de trabajo deben asistir al mismo grupo de laboratorio, de manera que puedan trabajar conjuntamente durante las horas de prácticas.
2. En las primeras sesiones de teoría de la asignatura, cada grupo desempeña el rol de cliente. Los miembros de cada grupo se reúnen y definen una aplicación software que encargarán a otro grupo de trabajo. Se describen en un documento cuáles deben ser las funcionalidades que quieren que tenga dicha aplicación desde el punto de vista del cliente.
3. El profesor de la asignatura recoge los documentos de todos los grupos, los revisa para determinar que el alcance del proyecto sea adecuado, ni demasiado extenso ni demasiado simple.

4. El profesor asigna el trabajo definido por un grupo que actúa como cliente a otro grupo del mismo laboratorio que actuará como desarrollador.
5. A continuación, los estudiantes pasan a desempeñar también el papel de analistas/desarrolladores. El profesor les encarga realizar el Documento de Especificación de Requisitos (DERS) del proyecto que se les ha asignado. La especificación del proyecto se realiza a partir del documento con la descripción funcional, y mediante entrevistas con el grupo que hace el papel de cliente.

Sesiones de laboratorio:

6. Para el desarrollo de las prácticas, se forman parejas de estudiantes que deben pertenecer al mismo grupo de trabajo.
7. Durante las sesiones de laboratorio, las parejas formadas ponen en práctica los conocimientos de Ingeniería del Software adquiridos en las clases de teoría, mediante diferentes ejercicios. Posteriormente, aplican las destrezas adquiridas al proyecto que se les ha asignado.
8. En la última sesión de laboratorio, cada uno de los grupos realiza una exposición en público, de 15 a 20 minutos de duración, en la que defiende ante el profesor de prácticas y ante el resto de compañeros, entre los que se encuentran el grupo que actuó como cliente, la solución propuesta, incluyendo todas las fases de análisis y diseño de software.

Cabe destacar que, para cada proyecto, tanto el grupo que ha desempeñado el rol de cliente, como el grupo que desempeña el rol de desarrollador, estarán en el mismo laboratorio. De esta manera se facilita que los desarrolladores puedan resolver dudas que puedan surgir durante las sesiones de prácticas.

De la misma manera, las diferentes parejas que forman parte de un grupo de trabajo, asisten al mismo laboratorio. Esto fomenta la cooperación y colaboración entre las diferentes parejas.

El hecho de que haya diferentes parejas en un mismo grupo de trabajo, implica que se desarrollarán diferentes soluciones para un mismo proyecto. En la exposición final se presentan las diferentes soluciones adoptadas para un mismo trabajo, el profesor las valora en público y se invita al resto de alumnos a comentarlas.

Para llevar a cabo el juego de roles utilizamos una serie de instrumentos:

- **Proyecto a desarrollar:** el proyecto a desarrollar dependía de cada equipo que jugaba el rol de cliente, por lo que había tanto proyectos como equipos.
- **Herramienta para modelar:** se ha utilizado Visual Paradigm como herramienta de modelado UML para construir los Casos de Uso, al igual que en GIT.

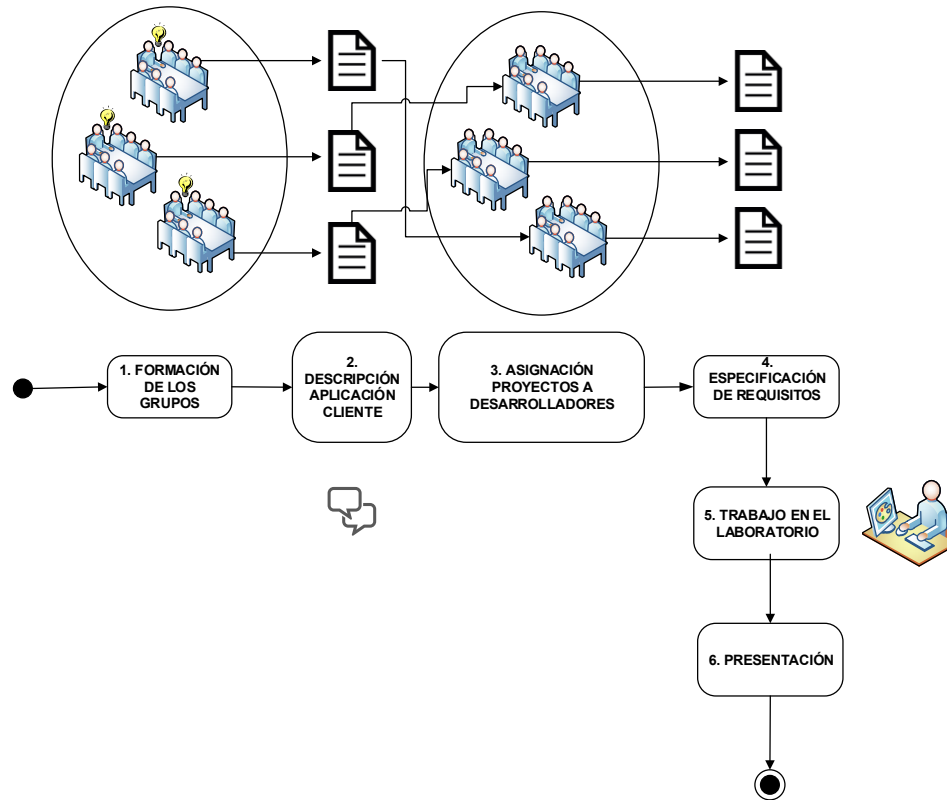


Fig. 2 Diagrama de Actividad con las actividades del juego de roles

4. Resultados

En ambas titulaciones se dispone de una nota asignada por los profesores de la asignatura tanto para el documento DERS realizado por los alumnos como para la presentación pública del trabajo que los alumnos realizaron en grupo. La medida de resultados final se calcula en base a la media de cada una de estas partes, asignando a cada una de ellas el mismo peso.

Dado que en el caso de la titulación de GIT se disponen además de los datos analizados por los alumnos que incluyen parámetros para medir la calidad (Tabla 1), hemos podido utilizar la métrica calidad (Q). La idea de introducir este parámetro es utilizar un método menos subjetivo de evaluación, ya que los profesores que evalúan no tienen siempre los mismos criterios, por mucho que se intente realizar una evaluación lo más imparcial posible y por mucho que se marquen unos criterios comunes de evaluación.

En la titulación de GIM los alumnos no realizan estos cuestionarios, pues sería impracticable que los profesores elaborasen DERS para todos los grupos de alumnos, ya que a diferencia de GIT, el proyecto sobre el que se analizan los requisitos es distinto para cada grupo. Por tanto, en el caso de GIM no disponemos de la métrica de calidad (Q).

De esta forma, la medida de resultados V en GIT se obtiene mediante la fórmula que aparece en la ecuación (2) que incluye el criterio de calidad Q , la evaluación del DERS (N_{DERS}) y la evaluación de la defensa del trabajo desarrollado para capturar requisitos (N_{PRES}); estas dos últimas, recordemos, realizadas por el profesor. Dado que la medida de calidad Q obtenida es un valor porcentual entre 0 y 1, para poder promediar con las notas de los alumnos que van en el rango 0 a 10, se debe multiplicar por 10 para poder obtener la media ponderada equitativamente.

$$V = \frac{Q * 10 + (N_{DERS} + N_{PRES})/2}{2} \quad (2)$$

Por su parte, la medida de resultados V en GIM sólo tiene en cuenta la evaluación del DERS (N_{DERS}) y la evaluación de la defensa del trabajo desarrollado para capturar requisitos (N_{PRES}); también ambas realizadas por el profesor con valores entre 0 y 10.

$$V = \frac{N_{DERS} + N_{PRES}}{2} \quad (3)$$

Una vez definido el parámetro V que permite comparar los resultados entre GIT y GIM, lo primero que se hizo fue analizar el impacto del empleo del factor de calidad Q en esta medida, únicamente usando el grupo GIT. La Fig. 1 muestra el valor promedio para el parámetro V entre todos los grupos de alumnos. En azul se muestran los resultados sin tener en cuenta el parámetro de calidad Q , lo que equivaldría a aplicar la misma fórmula empleada con los alumnos de GIM, es decir, empleando la ecuación (3). Como puede observarse, el empleo del valor de calidad supone una reducción, inferior a un punto, en la valoración media de los alumnos.

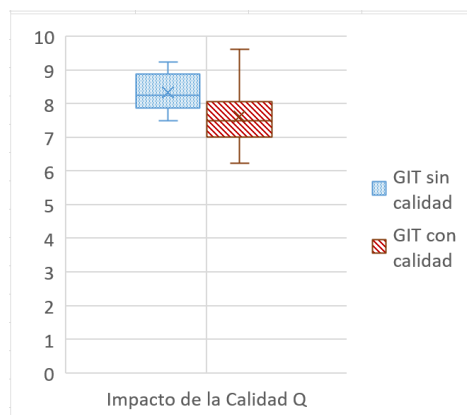


Fig. 3 Impacto de la Calidad Q en la Medida de Resultados en GIT

Posteriormente se pasó a comparar los resultados del parámetro V obtenidos entre las titulaciones analizadas. Para ello se compararon los resultados obtenidos en GIM con los obtenidos en GIT tanto con el empleo del factor de calidad Q como sin él. En la figura Fig. 4 se representan los valores promedio para el parámetro V entre todos los grupos de alumnos de GIT (en azul) frente a los de GIM (en rojo). En la figura de la izquierda se

muestran los resultados de GIT sin el factor de calidad Q , mientras que en la figura de la derecha aparecen los resultados de GIT empleando dicho factor. Como puede observarse en dichas gráficas, los resultados entre GIT y GIM son mucho más equiparables al introducir el factor de calidad Q , aunque en ambos casos son muy similares. Nótese que no disponemos de sujetos suficientes como para hacer un estudio estadístico y comprobar si hay resultados significativos entre ambas titulaciones.

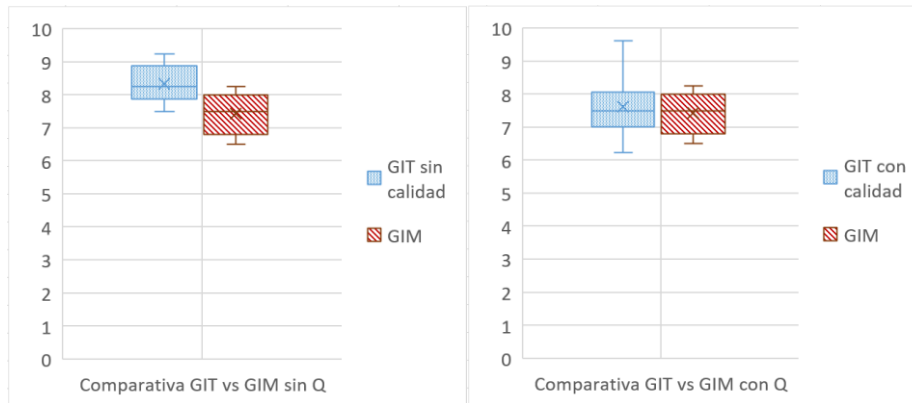


Fig. 4 Comparativa de la Medida de Resultados entre GIT y GIM sin calidad (izqda.) y con calidad (dcha.)

A continuación, vamos a comparar resultados cualitativos obtenidos durante un “focus group” realizado al final de ambos cursos.

Respecto a GIT, hemos identificado las siguientes ventajas:

- Todos los miembros del equipo han asumido distintas responsabilidades para poder desarrollar el proyecto.
- La retroalimentación que proporciona el cliente ayuda al equipo de trabajo a corregir deficiencias que hubieran sido difíciles de detectar con una especificación escrita.
- El profesor puede garantizar que los requisitos no son cambiantes a lo largo de las distintas entrevistas.
- El profesor puede guiar de forma homogénea a todos los equipos de alumnos porque conoce bien el proyecto que deben desarrollar.
- Se puede garantizar que la carga de trabajo es la misma para todos los alumnos.

Los inconvenientes detectados a través del focus group son:

- Inversión de mucho tiempo para capturar los requisitos si lo comparamos con el esfuerzo que les supone hacer el mismo trabajo si les facilitamos una descripción del trabajo a realizar por escrito.
- Dificil coordinación entre todos los miembros del equipo. Había equipos de hasta 6 personas y era difícil coincidir todos juntos.
- El profesor debe dedicar una media de 3 entrevistas por cada equipo de trabajo, lo que implica una inversión de tiempo importante.

Respecto a GIM, hemos identificado las siguientes ventajas:

- Los participantes desempeñan tanto el rol de cliente como el de desarrolladores, lo que les permite tener la visión y experiencia de ambos perfiles profesionales.
- La comunicación entre el cliente y los desarrolladores en las sesiones de laboratorio facilita al equipo de trabajo resolver las deficiencias existentes en el documento de especificación de requisitos.
- El profesor se limita a orientar a los alumnos en el desempeño de su trabajo, y a la evaluación técnica del mismo, pero no interviene en las decisiones en cuanto a los requisitos de las aplicaciones.
- Cada pareja del grupo de laboratorio comparte su solución con el resto de compañeros. Esto permite a los alumnos conocer diferentes diseños para un mismo problema, observar y comentar sus ventajas e inconvenientes.

Los inconvenientes detectados son los siguientes:

- La heterogeneidad en las aplicaciones a desarrollar puede suponer grados diferentes de dificultad para cada equipo.
- Las aplicaciones solicitadas por los alumnos, en el papel de cliente, no siempre están suficientemente detalladas en la fase de especificación de requisitos. Esto implica que surjan dudas respecto a los requisitos durante la fase de análisis y diseño.

5. Conclusiones

Este trabajo muestra una comparativa de dos puestas en práctica distintas de un juego de roles en una misma asignatura (Ingeniería del Software) impartida en dos titulaciones distintas (GIT y GIM) en la Universidad de Valencia. En GIT, el profesor juega el rol de cliente mientras que los alumnos juegan el rol de desarrolladores. En GIM, los alumnos juegan ambos roles (cliente y desarrollador) para dos proyectos distintos.

Los resultados nos muestran que los alumnos de ambas titulaciones han conseguido realizar unos DERS considerablemente buenos, ya que las notas medias de los grupos están siempre por encima de 7,5. Además, el uso del factor de calidad, ha permitido equiparar las notas entre ambas titulaciones, eliminando el factor de disparidad de los profesores a la hora de evaluar.

Centrándonos en comparar ambas técnicas, la empleada en la titulación de GIM requiere de un menor esfuerzo por parte del profesor a la hora de preparar el desarrollo de la actividad. El profesor no debe dedicar horas de tutorías grupales adicionales ya que son los alumnos los que actúan de clientes. Sin embargo, el esfuerzo a la hora de evaluar los documentos es mucho mayor, ya que el profesor debe analizar cuáles deberían ser los requisitos de cada uno de los proyectos y si coinciden con los que el grupo ha entregado, más allá de determinar si la expresión de los mismos es adecuada o no.

Por el contrario, en el caso de GIT, el esfuerzo se realiza principalmente en la fase de preparación de la actividad. Pues los profesores de la asignatura deben elaborar en común el DERS solución y es el profesor responsable el que hace de cliente en lugar de los alumnos.

El proceso de evaluación en este caso es mucho más sencillo, ya que tan solo se debe comparar el DERS de los alumnos con el DERS solución.

También se aprecian diferencias en cuanto al control que los profesores acaban teniendo sobre lo que se ajusta el DERS de los alumnos al DERS del profesor. En el caso de GIM, el control es mucho menor, ya que el profesor no está presente en todas las entrevistas (ya que se producen en el aula de forma simultánea) y por lo tanto no tienen un conocimiento real de lo que pedían los clientes y de si el DERS obtenido por los alumnos incluye o no todo lo que los clientes deseaban.

En cualquier caso, podemos afirmar que cualquiera de las dos técnicas permite alcanzar los tres objetivos básicos que se pretendía cubrir mediante esta innovación educativa:

- L1: Los estudiantes participan activamente en grupos de 5 a 6 personas para lograr un objetivo común. Esto trabaja la competencia de trabajo en equipo para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Además, también se trabaja la capacidad para saber comunicar y transmitir los conocimientos, las habilidades y las destrezas de un Ingeniero Informático.
- L2: El juego de rol permite que los estudiantes aprendan en un contexto similar al mundo real, donde los requisitos no vienen impresos en un papel, sino que deben extraerlos a través de entrevistas.
- L3: La innovación docente y la preparación de los estudiantes al mundo laboral futuro es una forma de mejorar los planes de estudio del Grado de Ingeniería Telemática y facilitar la obtención del sello de calidad.

Agradecimientos

Este trabajo se ha desarrollado dentro del proyecto de innovación educativa “Desarrollo de Software desde una Perspectiva Industrial” de la Universidad de Valencia.

Referencias

[1] UML: <http://www.uml.org/>.

[2] Al-Ani, B. and N. Yusop (2004). Role-Playing, Group Work and Other Ambitious Teaching Methods in a Large Requirements Engineering Course(ed.): *Proceedings of the 11th IEEE Intl. Conference and Workshop on the Engineering of Computer-Based Systems*

[3] Basili, V. R. and F. Lanubile (1999). Building Knowledge through Families of Experiments. *IEEE Transaction on Software Engineering* **25**(4): 456-473.

[4] Beecham, S., T. Clear, D. Damian, J. Barr, J. Noll and W. Scacchi (2017). How Best to Teach Global Software Engineering? Educators Are Divided. *IEEE Software* **34**(1): 16-19.

[5] Rueda, S., J. I. Panach, J.-B. Cabotà and F. Valverde (2016). Applying Role-Playing Game in Software Development Subjects(ed.): *ICERI2016* 3532-3538

