



ReCIBE. Revista electrónica de
Computación, Informática Biomédica y

Electrónica

E-ISSN: 2007-5448

recibe@cucei.udg.mx

Universidad de Guadalajara

México

Estrada Guzmán, Elsa; Peña Pérez Negrón, Adriana

Influencia de los Roles de Equipo en las Actividades del Desarrollador de Software
ReCIBE. Revista electrónica de Computación, Informática Biomédica y Electrónica, núm.

1, mayo-octubre, 2013

Universidad de Guadalajara

Guadalajara, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=512251560002>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Influencia de los Roles de Equipo en las Actividades del Desarrollador de Software

Elsa Estrada Guzmán
Computer Science Department
CUCEI – Universidad de Guadalajara
elsa.estrada@red.cucei.udg.mx

Adriana Peña Pérez Negrón
Computer Science Department
CUCEI – Universidad de Guadalajara
adriana.ppn@red.cucei.udg.mx

Resumen: Uno de los roles básicos en el proceso del software es precisamente el de desarrollador, también denominado ingeniero de software, cuyas actividades principales son: el análisis, diseño, programación y pruebas del producto a desarrollar. Estas actividades, dependiendo generalmente del tamaño del proyecto y de la metodología, pueden estar a cargo de diferentes personas o bien de un grupo de desarrolladores que en conjunto las llevan todas a cabo; en este último caso, estaríamos hablando de trabajo en equipo entre iguales o pares. Por otro lado, de acuerdo con la teoría de roles de equipo, las personas tienden a comportarse de manera regular en forma distintiva cuando colaboran, estas formas particulares de colaborar es

probable que influyan en el desempeño del equipo de desarrolladores de software. En este documento se presenta un caso de estudio con la finalidad de entender la influencia de los roles de equipo en ciertas actividades involucradas en el proceso de desarrollo de software.

Palabras clave: desarrollador de software, ingeniero de software, roles en la ingeniería de software, roles de equipo de Belbin

Team role influences in software development activities

Abstract: One of the basic roles in the software process is precisely that of a developer, also called software engineer, whose main activities are: analysis, design, programming and product testing for said product. These activities, usually based on the project size and methodology, they can be assigned to different people or to a group of developers to take care of them; in this former case we would be talking about group work among peers. On the other hand, according to the team role theory, people tend to behave in a specific way when they collaborate, these particular collaborative behaviors probably have an influence on the software developer team's performance. This document presents a case of study, with the intention of understanding the influence of team roles in certain activities involved in the software development process.

Keywords: software developer, software engineer, software engineer roles, Belbin team roles.

1. Introducción

La ingeniería de software entendida como la aplicación sistemática, disciplinada y cuantificable para el desarrollo, operación y mantenimiento de software (Computer Society of the IEEE, 1990), es considerada como portadora de paradigmas o patrones de trabajo. Las metodologías del software especifican la serie de pasos o actividades a seguir para el desarrollo de un

sistema, que van desde la identificación de las necesidades hasta su ejecución a través de su construcción, su operación y su retiro (Dorfman, 1997).

Para llevar a cabo un proyecto, sus ejecutores suelen tener diferentes responsabilidades y derechos a fin de facilitar la organización de las actividades encaminadas al cumplimiento de los objetivos. De igual manera, para el proceso de software las actividades son asignadas con base a la ingeniería del software, de acuerdo a las funciones requeridas en el modelo implementado, estos son los llamados roles en la ingeniería de software.

Así pues, los patrones que conforman los diferentes paradigmas de la ingeniería del software se rigen por ciertos criterios esenciales que los distinguen y clasifican de acuerdo a su enfoque, sus metodologías y sus modelos. En este contexto, es importante establecer que independientemente del modelo y los roles de ingeniería del software que maneje un enfoque, el rol de desarrollador es imprescindible.

Por ejemplo, en lo que se refiere a métodos ágiles, para el modelo XP los roles de ingeniería básicos reconocidos son (Chromatic, 2003): el cliente que representa a los usuarios finales; *el rastreador* que verifica si el proyecto va en tiempo; *el entrenador* que guía al equipo para entender XP y *el desarrollo* de software sugiriendo cambios en la implementación; y el desarrollador. Mientras que en el modelo SCRUM orientado al desarrollo de software en equipo existen tres roles: *el propietario del producto* que decide lo que será construido; *el gerente de SCRUM o facilitador*, que es el líder del equipo y *el equipo*, quienes desarrollan el producto a entregar, los desarrolladores (ScrumAlliance, 2008).

Aún más, de acuerdo con un estudio para determinar los requisitos de capacitación del perfil profesional para los ingenieros de software encaminado a las necesidades de las micro y pequeñas empresas, los roles de ingeniería del software requeridos son (Montilva, Barrios, & Rivero, 2004): *líder de proyectos de software* que ejecuta la planificación y control de los proyectos; *el*

ingeniero de soporte que asegura la calidad y valida el software; *el ingeniero de operación y mantenimiento de software* que administra las aplicaciones y las bases de datos e implementa los cambios; y *el desarrollador de software*.

De tal forma que independientemente de las diferentes modalidades de los roles de la ingeniería del software, la figura del desarrollador deberá estar presente, aunado a sus actividades principales que comprenden el análisis, diseño, codificación o implementación y pruebas del producto a desarrollar.

A este respecto, el estándar 1074 de la IEEE muestra las actividades del ciclo de vida del software agrupadas en las fases: Análisis, Diseño, e Implementación. Esta última actividad se subdivide en codificación, pruebas, e instalación (IEEE Computer Society, 1997). Es importante hacer notar que aunque existe una revisión posterior de este estándar hecha en el 2006, aún no está aprobada. Utilizaremos entonces la siguiente clasificación considerando el trabajo del programador independiente del de probador, como se muestra en la Tabla 1.

Tabla 1 Responsabilidades del Ingeniero de Software de acuerdo al estándar 1074 de la IEEE

Ingeniero de Software	Responsabilidades
Analista	Definición de los requerimientos del software. Análisis de las funciones requeridas por el sistema. Desarrollo de la arquitectura funcional. Análisis de la descomposición de los requerimientos del sistema.
Diseñador	Diseño de la arquitectura de los datos. Diseño de la base de datos. Diseño de interfaces. Diseño de detalle de la arquitectura del software.
Programador	Creación código ejecutable. Creación documentación de operación.

Ingeniero de Software	Responsabilidades
	Integración del entorno. Conducir revisiones del software.
Probador	Definición de la arquitectura de las pruebas. Diseño de casos de prueba. Ejecutar pruebas del software.

Siendo que las micro y pequeñas empresas dedicadas al desarrollo de software representan un porcentaje mayoritario de este importante sector de la economía mundial (Montilva, Barrios, & Rivero, 2004), la tendencia parece moverse hacia el uso de grupos de desarrolladores con la misma jerarquía que desempeñan las diferentes responsabilidades del desarrollador, esto es el trabajo en equipo entre pares, relación en la que nos enfocamos en este documento.

Una importante propuesta a este respecto es el Team Software Process o TSP basado en el modelo Personal Software Process o PSP (Humphrey, The Personal Software Process (PSP), 2000), para el que los roles de ingeniería transforman a los miembros del equipo en co-administradores, manejando dos tipos de roles para cada miembro (Humphrey, Chick, Nichols, & Pomeroy-Huff, 2010): un *rol administrativo* con responsabilidades que se distribuyen entre los miembros del equipo y que comprenden a los diferentes administradores de planeación, del proceso, de calidad, de soporte, de interfaz del cliente, de diseño, de codificación y de pruebas; y un *rol de miembro de equipo* o ingeniero de software, el desarrollador.

Sin embargo, aún cuando hay equipos que parecen haberse hecho a medida, existen otros en los que sus miembros batallan para adaptarse o incluso les es imposible hacerlo. Esta diferencia podría explicarse por medio de la teoría de roles de equipo.

1.1 Roles de equipo de Belbin (REB)

La teoría conocida como “Belbin Team Role” o Roles de Equipo de Belbin (REB) del Dr. Meredith Belbin, basada en el estudio de los roles y responsabilidades de los miembros integrantes de un equipo, sostiene que cada individuo posee un comportamiento en el entorno laboral, al mismo tiempo que desempeña de manera más eficaz aquellas funciones que le son más naturales (Belbin M. , 2013). En esta teoría se han identificado nueve roles cuya breve descripción se muestra en la Tabla 2.

Tabla 2. Descripción de los Roles de Equipo de Belbin

REB	Descripción
Planeador/Cerebro	Creativo, imaginativo, no ortodoxo resuelve problemas difíciles.
Investigador de recursos	Extrovertido, entusiasta, comunicativo. Explora oportunidades. Hace contactos.
Coordinador	Maduro, confiente. Un buen líder, clarifica metas, promueve la toma de decisiones, delega.
Impulsor	Retador, dinámico, trabaja bajo presión, tiene el coraje para manejar y superar obstáculos.
Cohesionador	Cooperador, apacible, perceptivo y diplomático. Escucha e impide los enfrentamientos.
Monitor/Evaluador	Sobrio, estratégico y perspicaz, ve todas las opciones. Juzga con precisión.
Implementador	Disciplinado, confiable, conservador. Cambia ideas en acciones prácticas.
Finalizador	Esmerado, consciente, ansioso. Busca errores y omisiones, entrega en tiempo.
Especialista	Dedicado, aporta cualidades y conocimientos específicos.

De acuerdo con Belbin, la clave para la efectividad de los equipos está en la combinación y equilibrio entre los diferentes REB dentro del equipo. Mientras que la combinación se refiere a que un equipo debe estar formado preferentemente por REB compatibles, el equilibrio es la importancia de reconocer a todos los comportamientos como necesarios, pero sin excederse en la cantidad de miembros que se identifiquen con el mismo REB (Belbin M. , 2012).

En las relaciones colaborativas de trabajo de equipo entonces, se llegan a presentar casos de compatibilidad e incompatibilidad de REB cuando se trabaja entre pares o iguales, tal como se muestra en la Tabla 3 (Aguilar, 2008 pp. 35).

Tabla 3. Relaciones entre roles de equipo de Belbin

Rol	Compatible con	Incompatible con
Impulsor	Investigador de Recursos	Coordinador Cohesionador
Implementador	Coordinador Investigador de Recursos Monitor-Evaluador Especialista Finalizador	Implementador Planeador/Cerebro
Finalizador	Implementador	Investigador de Recursos Monitor- Evaluador
Coordinador	Planeador/Cerebro Implementador	Impulsor
Cohesionador	Cohesionador Planeador/Cerebro	Impulsor
Investigador de Recursos	Cohesionador Implementador	Finalizador Especialista

Rol	Compatible con	Incompatible con
Planeador/Cerebro	Coordinador Investigador de Recursos Cohesionador	Monitor-Evaluador Planeador/Cerebro Especialista Implementador
Monitor-Evaluador	Coordinador Implementador	Finalizador Monitor-Evaluador Planeador/Cerebro
Especialista	Implementador Cohesionador	Planeador/Cerebro Investigador de Recursos

Con la intención de mejorar la calidad en el trabajo de equipo, que a su vez se verá reflejado en la calidad del producto de software, se realizó un estudio en el que se observaron los REB para el trabajo colaborativo durante el desarrollo de software. Las preguntas que condujeron el estudio son:

1. *¿Qué combinaciones de REB producen mejores resultados en la práctica de los equipos de desarrolladores de software?*
2. *¿Existe un REB asociado a las diferentes actividades del desarrollador de software?*

2. Caso de estudio

2.1 Método experimental

Este estudio se aplicó a cuatro grupos de alumnos que llevan el curso: Taller de Estructuras de Archivos. Los 56 estudiantes que conforman los cuatro grupos pertenecen o bien a la carrera de Ingeniería en Computación o a la de Licenciatura en Informática.

Tres de estos grupos fueron experimentales y uno de control. Para los experimentales se formaron los equipos con miembros de la misma clase de manera aleatoria, al grupo control se les permitió formar los equipos con los

miembros de su elección. Se conformaron 18 equipos, doce en los grupos experimentales: once de tres integrantes y uno de cuatro; en el grupo control seis: cinco de tres integrantes y uno de cuatro.

Cada equipo recibió la descripción de un problema a ser resuelto mediante el desarrollo de una aplicación que debería incluir el manejo de archivos. El tiempo para llevar a cabo la práctica se fijó en 2 horas.

Se hicieron dos propuestas de ejercicio de entre las cuales los equipos podían elegir:

1. Hacer un programa para el registro de marcas y registro de productos utilizando multilistas en archivos.
2. Hacer un programa para el registro de derechohabientes del seguro social de salud, utilizando archivos de dispersión para resolver el problema de códigos duplicados.

Se entregó a los equipos la lista de requerimientos funcionales para cada sistema:

Caso 1: Registrar marcas de productos, registrar productos y consultar productos de una marca.

Caso 2: Registro de derechohabientes y consulta de derechohabiente.

Finalmente, para la entrega se les pidió presentar tres archivos que deberían contener:

1. El modelado de los archivos y los datos para su manipulación;
2. El modelado de las funciones/operaciones utilizando, e.g. diagramas o pseudocódigo;
3. El código fuente.

El ejercicio consistió en la aplicación de técnicas de ingeniería de software para las fases: 1) modelado del análisis, 2) modelado del diseño y 3) codificación y ejecución. Cada una de estas fases se calificó en una escala de 0 a 5. Los puntos a evaluar en cada fase se muestran en la Tabla 4.

Tabla 4. Funciones a evaluar por equipo

Análisis	Diseño	Codificación y Depuración
Definición Estructuras de Archivos	Definición de Nombres de funciones	Los tipos de datos declarados se identifican con la actividad 1 o 2
Definición de Estructuras de datos	La o las funciones utilizan las estructuras de datos modeladas	Número de funciones que se ejecutan exitosamente.
Claridad (Nombres y tipos)	La o las funciones utilizan las estructuras de archivos modeladas %	
Utilización de alguna técnica de modelado	La o las funciones usan el o los nombres de archivos identificados en la actividad 1	
	Las funciones se identifican con los requerimientos y opciones del sistema.	

Recursos

Todos los equipos contaron con al menos una computadora con la aplicación Moodle instalada, esta aplicación es una plataforma para la administración de recursos pedagógicos. En el Moodle se colocaron dos archivos con los temas: hash, técnica utilizada en archivos para el almacenamiento y recuperación de datos; y multilistas, conjunto de registros almacenados relacionados entre sí

por direcciones o cursos que los ligan. Este material se publicó con una semana de anticipación a la fecha en que se llevó a cabo la práctica.

Datos

Se aplicó un cuestionario de autoevaluación de Belbin (Lindgren & Rolf, 1997) traducido al español, para identificar el REB de cada estudiante. Dos estudiantes no contestaron el cuestionario, uno perteneciente a los grupos experimentales y uno al grupo de control, ocasionando datos faltantes. La sumatoria de los REB de los 54 alumnos en cada grupo se presenta en la Tabla 5.

Tabla 5. Frecuencia de REB en los estudiantes

REB	Experimentales				Control	
	Grupo 1	Grupo 2	Grupo 4	Grupo 3	Frecuencia	
Total	15	14	8	17	54	
Investigador	0	0	0	1	1	
Monitor	1	0	2	0	3	
Impulsor	2	2	1	3	8	
Coordinador	2	3	1	3	9	
Especialista	3	1	0	1	5	
Cohesionador	5	2	0	2	9	
Cerebro	0	1	1	2	4	
Finalizador	0	2	0	1	3	
Implementador	2	2	3	5	12	

Los equipos quedaron integrados de acuerdo a su REB como se muestra en la Tabla 6.

Tabla 6 Combinaciones de REB por equipo

Equipo	REB de los integrantes		
1	Coordinador	Finalizador	Finalizador
2	Especialista	Cohesionador	Implementador
3	Impulsor	Impulsor	Cohesionador
4	Coordinador	Coordinador	Implementador
5	Coordinador	Implementador	Implementador
6	Monitor	Impulsor	Implementador
7	Monitor	Cerebro	<i>sin dato</i>
8	Coordinador	Cohesionador	Cerebro
9	Coordinador	Cerebro	Implementador
10	Especialista	Finalizador	Implementador
11	Coordinador	Cohesionador	Implementador
12	Impulsor	Implementador	<i>sin dato</i>
13	Investigador	Impulsor	Impulsor
14	Monitor	Cohesionador	Cohesionador
15	Especialista	Cohesionador	Implementador
16	Coordinador	Cohesionador	Cohesionador
17	Impulsor	Especialista	Implementador
18	Impulsor	Coordinador	Especialista

■ Equipos del grupo de control

También se aplicó un post-cuestionario impreso en el que se pidió a cada estudiante asignar un porcentaje de participación a cada miembro de su equipo (incluido él mismo) de acuerdo al desempeño en cada una de las actividades de desarrollo de software que se llevaron a cabo en el ejercicio, entre otras preguntas (ver Anexo). Debido a la tendencia de los alumnos a estandarizar los resultados, esto es a no utilizar muy bajos o muy altos porcentajes, además de que “los porcentajes” en ocasiones no daban el entero, se decidió que cuando al menos dos integrantes del equipo asignaron el mayor porcentaje o la más alta calificación en alguna de las actividades a un alumno, a éste se le sumaría un punto para esa actividad. En caso de que dos alumnos cumplieran con este criterio, a ambos se les sumó un punto. Los resultados asociados a cada REB se presentan en la siguiente Tabla 7.

Tabla 7 Puntos acumulados del desempeño en las actividades para cada REB

Frecuencia	REB	Actividades del Desarrollador de Software		
		Análisis	Diseño	Codificación y Depuración
1	Investigador	0	2	1
3	Monitor	1	5	1
8	Impulsor	4	2	2
9	Coordinador	1	3	3
5	Especialista	3	5	2
9	Cohesionador	3	0	2
4	Cerebro	0	1	2
3	Finalizador	3	4	3
12	Implementador	0	0	9

Los resultados de la evaluación por equipo para cada una de las fases se muestran en la Tabla 8. En la primera columna está el número del equipo, seguido del número de integrantes. En las siguientes tres columnas se encuentra el número de roles compatibles, incompatibles y los neutrales dentro del equipo. Posteriormente está la calificación para cada una de las actividades, en la penúltima columna el promedio y la calificación final redondeada en la última columna. Como se puede observar, de los 18 equipos: 6 equipos obtuvieron la calificación máxima, 10 equipos obtuvieron cuatro puntos y 2 equipos la mínima calificación obtenida de tres puntos.

Tabla 8. Combinación de roles por equipo y calificación obtenida

Equipo		No. de roles			Calificación de cada Actividad				Calificación Total	
Número	Integrantes	compatibles	Incompatible	neutrales	análisis	diseño	codificación	ejecución	Promedio	redondeada
1	3	0	0	6	4	4	5	5	4.5	5
2	3	3	0	3	5	5	3	3	4.0	4
3	4	2	4	6	4	5	5	4.5	4.6	5
4	3	4	0	2	5	5	3	3	4.0	4
5	3	4	2	0	5	5	3	3	4.0	4
6	3	2	0	4	5	5	3	3	4.0	4
7	3*	0	2	0	5	5	3	3	4.0	4
8	3	4	0	2	3	5	5	4	4.3	4
9	3	3	2	1	3	3	4	3	3.3	3
10	3	4	0	2	4	5	4	4	4.3	4
11	3	3	0	3	4	4	5	5	4.5	5
12	3*	0	0	2	4	4	5	5	4.5	5
13	4	5	0	7	3	5	5	5	4.5	5
14	3	2	0	4	3	3	3	3	3.0	3
15	3	3	0	3	5	5	5	5	5.0	5
16	3	4	0	2	5	5	4	3	4.3	4
17	3	2	0	4	5	5	4	3	4.3	4
18	3	0	2	4	5	5	4	3	4.3	4
-		Equipos del grupo de control			*	Grupos con un rol sin determinar				

Promedio por REB

Los datos de calificación por REB para cada una de las actividades y el promedio final se presentan en la Tabla 9.

Tabla 9 Calificaciones al desempeño por actividad por REB y su desviación estándar

REB	Frecuencia	Calificación					Total
		Análisis	Diseño	Codificación	Ejecución		
Investigador	1	3.00	5.00	5.00	5.00	4.50	
Monitor	3	4.33	4.33	3.00	3.00	3.67	
Impulsor	8	4.13	4.88	4.50	4.13	4.41	
Coordinador	9	4.33	4.56	4.00	3.56	4.13	
Especialista	5	4.80	5.00	4.00	3.60	4.38	
Finalizador	9	4.11	4.44	4.11	3.72	4.11	
Cerebro	4	3.75	4.5	4.25	3.625	4.05	
Finalizador	3	4.00	4.33	4.67	4.67	4.43	
Implementador	12	4.42	4.67	3.92	3.75	4.20	
	54	4.10	4.63	4.16	3.89	4.21	

3. Resultados

Es importante hacer notar que la frecuencia en el número de perfiles debido al tamaño de la población generó un total poco representativo en varios casos, por ejemplo, hubo un solo caso con perfil de Investigador.

En referencia a la pregunta 1, sobre la combinación de REB con mejores resultados, en el caso del grupo de control es interesante observar que solo hubo un equipo con roles incompatibles y que es el equipo con menor calificación. Para el resto de los equipos no existe correlación entre el número de REB compatibles o incompatibles y la calificación obtenida. Sin embargo, se encontró correlación estadísticamente significativa (.567 a nivel .01 con dos colas) entre el número de miembros con relación neutral y la calificación obtenida, esto es, a mayor número de relaciones neutrales, mayor calificación.

Respecto a la segunda pregunta sobre los REB asociados a las distintas actividades del desarrollador, conforme con la autoevaluación de los

compañeros de equipo acerca de su aportación, ver Tabla 7, parece claro que el Implementador (9 estudiantes) suele tener una clara aportación para el trabajo de codificación. Mientras que el Especialista (5 estudiantes), el Monitor (3 estudiantes) y el Finalizador (4 estudiantes) hicieron una buena aportación durante el diseño. En el caso de la actividad de análisis, los alumnos con REB de Impulsor (4 estudiantes) fueron los que se distinguieron por su aportación.

La frecuencia en los roles de equipo también podría ser un indicador en la tendencia de ciertas formas características de colaborar que adoptan los desarrolladores de software; los REB con mayor frecuencia fueron los de Implementador, Coordinador y Finalizador.

4. Conclusiones y Trabajo Futuro

Particularmente en las micro y pequeñas empresas que constituyen un importante porcentaje de la industria desarrolladora de software, las actividades del desarrollador de software tienden a realizarse en equipos de trabajo con miembros de la misma jerarquía, esto es, en colaboración entre pares. En este contexto, es importante que además de las capacidades de los desarrolladores se estudien las características que deban tener los miembros del equipo para que éste funcione adecuadamente. De acuerdo con la teoría de los roles de equipo de Belbin, las personas desempeñan de manera más eficaz aquellas funciones que le son más naturales, mismas que hacen más eficiente el trabajo en equipo. En este documento se presentó un caso de estudio en el que se observaron los roles de equipo en relación a algunas de las actividades que desempeñan los desarrolladores de software: análisis, diseño y codificación de software.

A pesar de una muestra poblacional relativamente pequeña y considerando que existen 9 diferentes roles de equipo, se establecieron algunas

observaciones como el hecho de que un mayor número de miembros con roles neutrales y no el número de relaciones compatibles o incompatibles, influyeron en el desempeño del equipo. También se encontró que algunos roles corresponden a una mayor aportación con ciertas actividades como el caso del Implementador con la codificación.

Trabajos futuros

La formación de equipos de desarrollo de software es una tarea en la que aún se sigue investigando. Consideramos gratificantes los logros del trabajo en equipo y cuyos beneficios permiten alcanzar altos objetivos en la práctica de la ingeniería de software. Es por esto que es importante dar continuidad en este tipo de estudios que deseamos extender por ejemplo: realizando el experimento con alumnos de grados de estudio más avanzados para tener una mejor visión en los resultados; revisando la forma de evaluar las actividades por equipo en el ciclo de vida del software y estudiando el mecanismo que utilizan las personas para identificar sus propias preferencias en el desarrollo de software.

Bibliografía

Aguilar, R.A. (2008). Entrenamiento de Grupos: Una Estrategia Asistida por Entornos Virtuales Inteligentes. Tesis doctoral Universidad Politécnica de Madrid.

Belbin , M. (2013). *Method, Reliability & Validity,A Comprehensive Review of Belbin Team Roles.* Retrieved 04 10, 2013, from Belbin Team Roles: [http://www.belbin.com/content/page/5599/BELBIN\(uk\)-2013-A%20Comprehensive%20Review.pdf](http://www.belbin.com/content/page/5599/BELBIN(uk)-2013-A%20Comprehensive%20Review.pdf)

Belbin, M. (2012). *BELVIN.* Recuperado el 30 de 01 de 2013, de BELVIN: <http://www.belbin.com/rte.asp?id=8>

Chromatic. (2003). Team-Based Software Development Extreme Programming. In Chromatic, *Extreme Programming* (p. 108). O'Reilly Media.

Computer Society of the IEEE. (1990, 09 28). IEEE Standar Glossary of Software Engineering Terminology. Std 610.12-1990. New York, New York, USA: IEEE Standars Board.

Dorfman, M. (1997). Requirements Engineering. *IEEE*, 7-22.

Humphrey, W. S. (2000). *The Personal Software Process (PSP)*. Pittsburgh: Carnegie Mellon Software Engineering Institute.

Humphrey, W. S., Chick, T. A., Nichols, W., & Pomeroy-Huff, M. (2010). *Team Software Process (TSP) Body of Knowledge (BOK)*. Massachusetts: Carnegie Mellon University.

IEEE Computer Society. (1997). *IEEE Standar for Developing Software Life Cycle Processes IEEE std 1074-1997*. New York USA: John W. Horch.

Lindgren, B., & Rolf, M. (1997). *Rants from Rolf Marvin Bøe Lindgren*. Retrieved 01 30, 2013, from R. Meredith Belbin's Team Roles Viewed From the Perspective of The Big 5: <http://blog.grendel.no/wp-content/uploads/2002/02/hovedoppgave.pdf>

Montilva, J., Barrios, J., & Rivero, M. (2004). Requisitos de capacitación y perfiles para ingenieros de software en micro y pequeñas empresas. ecuperado el 10 de 04 de 2013, de CEISOFT: http://www.ceisoft.org/DB/Methodius/EDOCS/SRed/2009/07/T022000000150-0-Requisitos_y_perfiles_IS_para_MyPEs.pdf

ScrumAlliance. (2008). *ScrumAlliance transforming the world of work*. Retrieved 04 09, 2013, from http://www.scrumalliance.org/pages/scrum_101

Anexo

Cuestionario final

Nombre de cada uno de los miembros del equipo: _____

Nombre de la práctica que podría ser hash o multilistas: _____

Sección o grupo: _____

1. ¿Crees que realmente se trabajó en equipo? _____
2. ¿Cuál crees que fue la mayor falla en el equipo? _____
3. De las tres actividades ¿crees que se puede omitir alguna en el desarrollo de software? ¿cuál o cuáles? Y ¿por qué? _____
4. Dividir el 100% de participación entre las 3 actividades realizadas, de acuerdo al esquema siguiente:

	Análisis	Diseño	Codificación
5. % De tu participación:			
6. % De participación de compañero 1:			
7. % De participación de compañero 2:			

Notas biográficas:



Elsa Estrada Guzmán recibió el grado de Maestría en Sistemas de Información en la Universidad de Guadalajara en el 2007. Actualmente es profesora en esa Universidad en el Departamento de Ciencias Computacionales del Centro Universitario de Ciencias Exactas e Ingenierías. Sus intereses son en ingeniería de software y estructuras de archivos.



Adriana Peña Pérez Negrón recibió el grado de Doctora en Informática de la Universidad Politécnica de Madrid haciendo una estancia de investigación en la Universidad de Salford del Reino Unido para obtener la mención de “Doctor Europeo”. Su interés es en la interacción del usuario y el desarrollo de agentes en los entornos virtuales colaborativos.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México.