
Relación de Problemas 1.1

Alumna: Elena Cantero Molina

DNI: 45744912M

Grupo C3

2) ¿Qué diferencias existen entre las distintas clases de Software según los derechos de autor? Proponga ejemplos de programas de las distintas clases.

- Licencia de software de código abierto permisivas: Se puede crear una obra derivada sin que ésta tenga obligación de protección alguna.

- Apache Software License v.1.1
- Python License v.2.1
- PHP License v.3.0

- Licencia de software de código abierto robustas: Estas licencias aplican algunas restricciones a las obras derivadas, haciendo que según el grado de aplicación se puedan dividir a su vez en dos subcategorías:

- Licencias de software de código abierto robustas fuertes: Las licencias de software de código abierto robustas fuertes o con copyleft fuerte, contienen una cláusula que obliga a que las obras derivadas o modificaciones que se realicen al software original se deban licenciar bajo los mismos términos y condiciones de la licencia original. Entre las licencias de esta categoría están:

- GNU General Public License v.2.0
- GNU General Public License v.3.0

- Licencias de software de código abierto robustas débiles: Las licencias de software de código abierto robustas débiles, con copyleft débil/suave o híbridas, contienen una cláusula que obliga a que las modificaciones que se realicen al software original se deban licenciar bajo los mismos términos y condiciones de la licencia original, pero que las obras derivadas que se puedan realizar de él puedan ser licenciadas bajo otros términos y condiciones distintas. Entre las licencias de esta categoría están:

- Mozilla Public License
- Open Source License.
- Apple Source License v.2.0

- Licencia de software de código cerrado: Estas licencias también se conocen con el nombre de «software propietario». En ellas los propietarios establecen los derechos de uso, distribución, redistribución, copia, modificación, cesión y en general cualquier otra consideración que se estime necesaria. Este tipo de licencias, por lo general, no permiten que el software sea modificado, desensamblado, copiado o distribuido de formas no especificadas en la propia licencia, regula el número de copias que pueden ser instaladas e incluso los fines concretos para los cuales puede ser utilizado. Algunos ejemplos de este tipo de licencias son las llamadas acuerdo de licencia de usuario final (ALUF), también llamado en algunos países contrato de licencia de usuario final (CLUF). En el mundo anglosajón se llama end-user license agreement (EULA), pero los contratos en español deben recoger el término en español.

- Software de dominio público (sin licencia): Se permite uso, copia, modificación o redistribución con o sin fines de lucro.

3) Para cada uno de los tipos de software trate de elaborar una lista comentada de las características que introduce la red y cómo influyen sobre el software estas características. Por ejemplo: ¿qué características introduce el comercio electrónico en el software de aplicación de gestión?, o ¿qué características introduce el cálculo distribuido en el software de aplicación científico?, o ¿cómo ha influido la red en los sistemas operativos?

- Software de Aplicación: En este caso la red influye poco en procesadores de texto, editores, etc, pero si tienen repercusión en los videojuegos donde puede jugar con cualquier persona del mundo.

- Software de Programación: En este caso la red tiene una alta importancia para difundir, modificar y ayudar en el desarrollo de enlazadores, intérpretes, paquetes integrados, etc.

- Software de Sistema: En este caso la importancia la tienen los servidores, que permiten conectar entre sí un grupo de gente a una misma red.

4) Ponga dos ejemplos de mito del Software y explique brevemente su significado y las diferencias entre el mito y la realidad.

Mito: Tenemos ya un libro lleno de estándares y procedimientos para construir un software ¿No les proporciona a caso todo lo que necesitan saber?

Realidad: El libro existe, pero ¿se usa?, ¿conocen los trabajadores su existencia?, ¿refleja las prácticas modernas de desarrollo del software?, ¿es completo?, en muchos casos la respuesta es ‘no’.

Mito: Si fallamos en la planificación, podemos añadir más programadores y adelantar el tiempo perdido.

Realidad: La realidad de aprender y comunicarse con el equipo nuevo hace que se reduzca la cantidad de tiempo gastado en el desarrollo productivo.

5) El cuarto principio del código de Ética y práctica Profesional de la Ingeniería del Software del ACM y del IEEE-CS dice: “Los ingenieros de software deben mantener la integridad e independencia en sus juicios profesionales”. Explique el significado que, a su juicio, puede tener este principio. Proponga un ejemplo de posible aplicación del mismo. Busque más

información sobre códigos éticos y de profesionalidad en la Ingeniería del Software. ¿Piensa que este tema tiene algún interés o utilidad?

Este principio está relacionado con el pensamiento del ingeniero profesional respecto a su integridad. Un profesional debe ser siempre leal a los pensamientos que tenga para su uso profesional.

Creo que el interés que tiene este tema es para hacer que las personas que estudian alguna ingeniería se centren en su futuro de una manera más significativa.

6) ¿Cómo piensa que puede afectar a la gestión de los proyectos software el carácter lógico del mismo?

Creo que el hecho de que el software se deteriore ofrece flexibilidad en el mantenimiento del mismo, el desarrollo del software siempre nos va a permitir, partir del punto más reciente. Será vital una buena organización en proyecto.

7) Considere uno de los proyectos siguientes:

- Gestión de una ONG de integración social de colectivos marginados
- Gestión de una Biblioteca
- Gestión y control de unas instalaciones deportivas
- Aplicación de seguridad para unos grandes almacenes
- Un juego de 3 en raya para Android
- Un editor de texto html
- Cualquier otro de su elección

Haga una estimación del esfuerzo necesario, así como de los costos y del plazo de entrega. Seleccione el modelo de proceso que mejor se adapta al proyecto. Compare sus estimaciones con las que realicen otros compañeros.

Gestión de una ONG de integración social de colectivos marginados.

Consta de 6 módulos: contabilidad, recursos humanos, compras, zona de reuniones, ventas, agenda y almacén. El proyecto era ideado. El trabajo a realizar, sigue esta línea:

- Descripción del problema.
- Obtención de requisitos.
- Modelado entidad-relación.
- Diseño base de datos.
- Implementación.
- Interfaz.

8) ¿En qué consiste la Programación Extrema (XP)?

La Programación Extrema (XP) es una metodología de desarrollo ágil que tiene como objetivo aumentar la productividad a la hora de desarrollar un proyecto software. Da prioridad a los trabajos que dan un resultado directo.

10) Haz una lista de las 3 aplicaciones que creas más importantes. Investiga y haz un resumen de todo lo relacionado con su desarrollo. Plántate la siguiente cuestión ¿qué modelo de proceso crees que se adecua más al desarrollo de cada una de ellas?

Las 3 aplicaciones que considero mas importantes son: Instagram, Facebook y Telegram y Facebook son dos redes sociales y una red de mensajería, considero que

el proceso que más se adecua más al desarrollo de cada una de ellas es el modelo incremental.