**RELACION DE PROBLEMAS 1.1 Y 1.2 (Tema 1)**

Relación 1.1

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?

la red, en el software de aplicación de gestión ha permitido a los empresas vender online y además ampliar su volumen de facturación y alcance, en el calculo distribuido ha permitido acelerar los cálculos ya que no solamente dependerá de los ordenadores locales sino que también podrá emplear otros conectados a través de la red, en los sistemas operativos ha permitido los servicios en la nube tales como OneDrive, Dropbox,… y nos ha ahorrado los viajes a la tienda de informática para comprar cd de software, ya que ahora podemos, gracias a la red, comprarlo en la misma página del desarrollador.

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

Mito: Una vez que escribimos el programa y hacemos que funcione, nuestro trabajo ha terminado

Realidad: Los datos de la industria indica que entre el 60% y el 80% del esfuerzo se realiza después de entregarle al cliente el software por primera vez

Mito: el único producto del trabajo que se entrega en un proyecto exitoso es el programa que funciona

Realidad: un programa que funciona solo es una parte de una configuración de software que incluye muchos elementos. Son varios los productos terminados que proporcionan la base de la ingeniería exitosa y que guían el apoyo para el software

7) Considere uno de los proyectos siguientes:

• Gestión de una ONG de integración social de colectivos marginados

Modelo incremental, coste alto, tiempo depende del tamaño

• Gestión de una Biblioteca

Emplearía el modelo incremental, en una primera entrega se añadiría un reducido catalogo para probar las funciones, una vez que todo este funcional, se añadiría los restantes ejemplares al catalogo.

El costo en hacerlo no seria tan alto, pero el tiempo dependerá del tamaño del catalogo

• Gestión y control de unas instalaciones deportivas

Modelo en cascada, coste alto, tiempo medio

• Aplicación de seguridad para unos grandes almacenes \

Emplearía el modelo en espiral de Boehm, coste alto, tiempo alto

• Un juego de 3 en raya para Android

Modelo de proceso evolutivo, coste bajo, tiempo de entrega corto

• Un editor de texto HTML

Modelo de proceso evolutivo, coste bajo, tiempo de entrega medio

• Cualquier otro de su elección

Un software de gestión inteligente de luces y temperaturas para mi casa, emplearía el modelo incremental, el coste seria bajo, y el tiempo alto

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.

9) La “alianza Ágil” definió una serie de 12 principios que debería tener una metodología para alcanzar niveles aceptables de agilidad. Puede encontrar estos principios en el enlace que aparece en el ejercicio 1. Comente los problemas que intenta solucionar alguno de estos principios.

Aceptamos que los requisitos cambien, incluso en etapas   
tardías del desarrollo. Los procesos Ágiles aprovechan  
el cambio para proporcionar ventaja competitiva al   
cliente.

Intenta evitar tener que rehacer el proyecto.

Entregamos software funcional frecuentemente, entre dos  
semanas y dos meses, con preferencia al periodo de   
tiempo más corto posible.

Asi el cliente puede valorar los avances y comentar lo que no le gusta, por lo tanto, evitamos una entrega final en la que el producto no es lo que esperaba el cliente

Relación 1.2

1)Nombrar todos los sistemas que fallaron por un problema en:

1. La programación o prueba del sistema software.

Mariner 1962 , at&t lines goes down, patriot fails soldier, British passport to nowhere

2. El diseño del software o del hardware.

Cia gives Soviet gas, world war iii, intel Pentium, Arianne rocket, mars climate satellite

3. El estudio del problema (análisis y especificación de requisitos o de recursos necesitados)

Wall Street crash, dot-com bubble, eds, FBI

4. Una mala documentación o uso.

Medical machine kills, cancer treatment

5. La seguridad.

Skynet, disastrous study, love virus

6. Un deficiente análisis del riesgo

Hartford Coliseum, y2k

2. De toda esta lista de desastres:

1. ¿Cuál es el intruso?

Skynet, ficticio de la película Terminator

2. ¿Cuál consideras que es el peor de todos? ¿por qué?

La falsa alarma de los sistemas de defensa soviéticos. Porque ese fallo podría haber cambiado el mundo tal y como lo vemos hoy

3. ¿Cuáles están relacionados con la industria armamentista?

Patriot fails soldier, CIA gives soviet gas, world war iii, Skynet

4. ¿Cuáles están relacionados con la industria aeroespacial?

Mariner, Arianne rocket, mars climate satellite,

5. ¿Cuáles están relacionados con la industria sanitaria?

Medical machine kills, disastrous study, cancer treatment

6. ¿Cuáles están relacionados con las finanzas?

Dot-com bubble, Wall Street crash y también metería cia gives soviet gas, ya que uno de los objetivos era cortar el suministro de gas y por lo tanto evitar la venta de gas a la CE y, por ende, la no entrada de divisas a la URSS

4. Expón, de forma breve, cuál es tu opinión respecto a todo lo aprendido ¿Cuál sería las posibles soluciones para evitar, en la medida de lo posible, los fallos en un sistema software?

Yo creo que lo mas seguro es empezar por un buen diseño, que sea escalable y por módulos, para que asi si falla una parte, no vaya todo a la basura. Y además que esos módulos se puedan probarse antes de ser implementados. Una vez terminado de implementar y que aparentemente no tenga fallos lo siguiente seria testar los casos posibles para asi evitar errores tontos, pero de perdidas millonarias como el cohete de la nasa que usa el sistema métrico imperial en vez del internacional