Clases 3 y 4: Ciclo de vida I, II

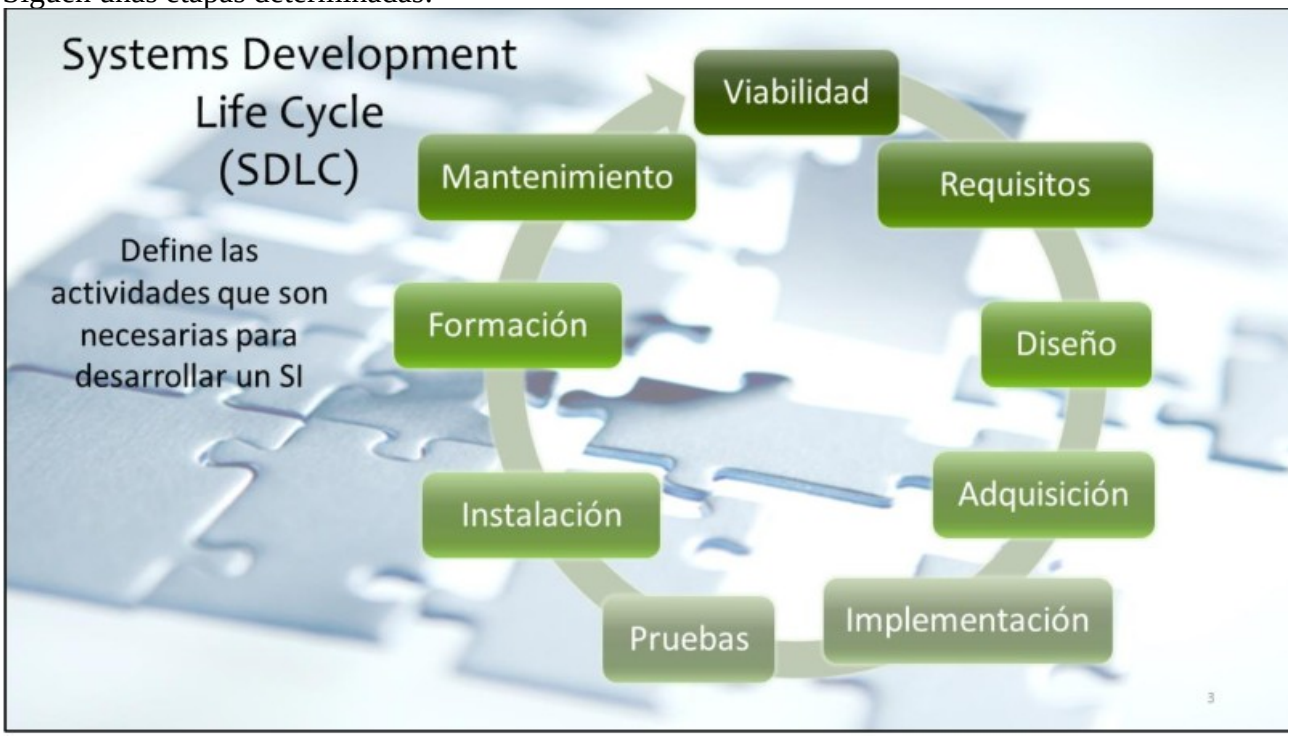
Introducción:

Definición

El ciclo de vida del desarrollo de software es el conjunto de procesos seguidos dentro de una organización para llevar a cabo todos los pasos necesarios para desarrollar un sistema de información.

Características y definición de etapas

Siguen unas etapas determinadas:



Aquí se definen cuáles son las actividades necesarias para desarrollar un sistema de información:

1. **Viabilidad:** si es adecuado, si es posible, si económicamente es viable el desarrollar el software que queremos. Tendremos que analizar los costos, de los hombres mes, es decir el esfuerzo en en personas necesarios para desarrollar este software, etc. Esto nos debe dar la respuesta a si es viable o no es viable el desarrollo del software que se nos está planteando, o sea el sistema de información que se nos está planteando.
2. **Requisitos:** aquí tenemos que ver qué es lo que tenemos que entender para desarrollar este sistema de información. Que es lo que se pretende del mismo. Entonces tenemos que analizar con detalle qué se necesita, cuál es el problema que tenemos que resolver, cuáles son los objetivos que se quieren alcanzar con este sistema de información y si los objetivos responden a los objetivos de la organización y luego tenemos que interactuar con los usuarios.
   El análisis de requisitos normalmente requiere una interacción muy fuerte con los usuarios, con los que son los usuarios funcionales de la aplicación, porque de ellos tendremos que obtener el conocimiento que nos permita entender qué tenemos que desarrollar, qué tenemos que implementar en una fase de posterior. Muchas veces podemos encontrar éstos requisitos también documentados en la literatura, pero normalmente en la mayoría de casos se requiere una fuerte interacción con los usuarios funcionales. Este análisis de requisitos nos generará una serie de documentos que nos permitirán pasar a la siguiente etapa que es la de diseño
3. **Diseño:** aquí tendremos que decidir cómo será el sistema para trabajar conjuntamente posiblemente con otros sistemas ya existentes, que componente se distingue, que hardware

se requiere o ya tenemos, que software se requiere. Tendremos también que en muchos casos este diseño del sistema de información general nos determinará también un software a lo mejor tendremos que desarrollar, o a lo mejor existe y lo podemos lo podemos adquirir.

4. **Adquisición** si se requiere tanto de software, como de de hardware. Tendremos que decidir que tenemos con comprar para diseñar o poder implementar el sistema de información que hemos diseñado en la etapa anterior, tanto el hardware si no se dispone ya él, como del software para disponer de ese sistema de información.
   Esta compra de software puede suponer también que haya que diseñar y que haya que desarrollar e implementar piezas de software a lo mejor en el mercado no se encuentran y posiblemente en la mayoría de casos, sea el software que hilvane todo el sistema de información utilizando otros componentes.
5. **Implementación:** En esta etapa lo que se hará es escribir el código, lo que constituyen los programas software que nos proporciona la funcionalidad diseñada analizada en la fase de requisitos y diseñada en la fase de diseño. Aquí lo que decimos es que implementaremos el software, los programas que nos proporcionan esa funcionalidad. Para implementar estos programas necesitamos utilizar lenguajes de programación como son java, C, C++, Python, entre algunos de los lenguajes de programación más conocidos.
6. **Pruebas:** Una vez hemos implementado esos programas, hay que ver si satisfacen los requisitos previamente establecidos. Tenemos que comprobar que efectivamente ese software que hemos desarrollado satisface esos requisitos y que realiza las funciones que debe realizar que es para lo que se diseñó. Acá se chequea que no haya errores y que el sistema no contenga defectos. Se hacen pruebas de cada una de sus unidades, pruebas de integridad, que es que todas las piezas en su conjunto marchen y pruebas de sistema. Después se realizan pruebas de integridad de toda la aplicación y se tienen que realizar también pruebas de todo sistema donde no solamente comprobaremos que todo el sistema de información realiza aquello que preveemos que tiene que realizar sino que también comprobaremos que el hardware es adecuado para poder ejecutar de forma eficiente y en unos tiempos de respuesta para entender si son adecuados los programas que hemos desarrollado.
7. **Instalación:** una vez que ya tenemos la aplicación y el sistema de información probado, hay que instalarlo en el entorno de ejecución, o sea en las máquinas donde va a ejecutarse.
8. **Formación:** En esta etapa también tenemos que formar a los usuarios sobre cómo utilizar el sistema y debemos generar un manual adecuado, un manual de funcionamiento para los usuarios del sistema
9. **Mantenimiento**: es evolutivo y normalmente se sigue durante toda la etapa de vida de ese software. Cuando tenemos un sistema de información funcionando en una organización, conforme va cambiando el tiempo, los objetivos de la organización y sus usuarios suelen cambiar, lo cual requiere incorporar nuevos objetivos de la organización en el sistema de información, o incorporar también nuevas necesidades de los usuarios en el sistema de información. Por lo tanto entramos en una etapa que es la de mantenimiento evolutivo que se da a lo largo de toda la vida de ese sistema de información. Mientras un sistema de información está vivo, es decir esta siendo utilizado, siempre se requiere un mantenimiento evolutivo, que puede ser más intenso o menos intenso. Normalmente cuando más temprano es el desarrollo del sistema, menos mantenimiento se requiere, pero conforme van evolucionando los objetivos de los usuarios y los objetivos de la organización y las necesidades de los usuarios suelen ir cambiando, mas mantenimiento hay que hacer hasta tal punto de analizar si sigue siendo viable.

Clase 5: Ciclo de vida III: Herramientas de soporte para el desarrollo de software

Herramientas CASE

CASE significa Computer Aided Software Engineering y es un software que soporta una o más actividades de ingeniería del software, dentro de todo un proceso de desarrollo de software

Se está volviendo popular para el desarrollo de sofware a medida que mejoran las capacidades y funcionalidad, y que también resultan beneficiosos para el desarrollo de software de calidad.

Las ventajas de utilizar herramientas CASE en el proceso de desarrollo de software son que reduce el costo, automatiza muchas tareas manuales repetitivas, reduce el tiempo de desarrollo del proyecto ya que apoyan la estandarización y evitan la repetición, y fomenta la la reutilización de software, desarrolla proyectos complejos de una forma más amigable y más eficiente.

Éstas herramientas nos dan soporte en las distintas etapas del ciclo de vida de desarrollo de software. Tenemos herramientas de para la parte de requisitos y diseño, como son BPM (modelado de procesos empresariales) o como es UML (lenguaje de modelado modificado) son herramientas que nos dan facilidades gráficas para poder realizar diagramas de flujo de procesos, que nos permiten el modelado a partir de los requisitos en las fases de diseño, nos permiten transformar modelos, o utilizar modelos ya existentes que son los adecuados para lo que estamos analizando y mediante estos procesos de modelado de procesos empresariales y mediante los lenguajes de modelado unificado, Nos facilitan el análisis de requisitos y diseño estas etapas.

Kits de desarrollo

Para la parte de de implementación lo que utilizamos son kits de desarrollo de software (SDK) o también entornos de desarrollo interactivo, o entornos de desarrollo integrados (IDE), nos proporciona herramientas de generación de código fuente que después podemos optimizar. Nos proporciona compiladores y nos proporciona también herramientas de depuración para comprobar la validez del software que hemos desarrollado. Se usan en la fase de implementación

Test Case Management y Source Code Analisys

Para la parte de pruebas y verificación tenemos herramientas de Test Case Management, tenemos de herramientas source code analysis, las cuales sirven para tomar medidas de performance en la ejecución de las aplicaciones.

Tambien existen métodos también para diseñar, gestionar lo que son los casos de prueba, podemos analizar el código fuente, podemos mediante estas herramientas medir el desempeño del software que hemos desarrollado y podemos especificar con el Specification o Test Cases podemos especificar los casos de prueba y determinar ahí, exactamente qué es lo que queremos que se compruebe.
Estas son herramientas que nos facilitan toda la parte de instalación.

Para instalación y mantenimiento

Para la parte de instalación y mantenimiento tenemos herramientas que nos proporcionan el sistema de control del sistema y que nos permiten también gestionar la configuración.
Aquí hay muchas herramientas que nos dan soporte para llevar las aplicaciones que hemos

desarrollado a explotar.

Clase 6: Enfoques para el desarrollo de software

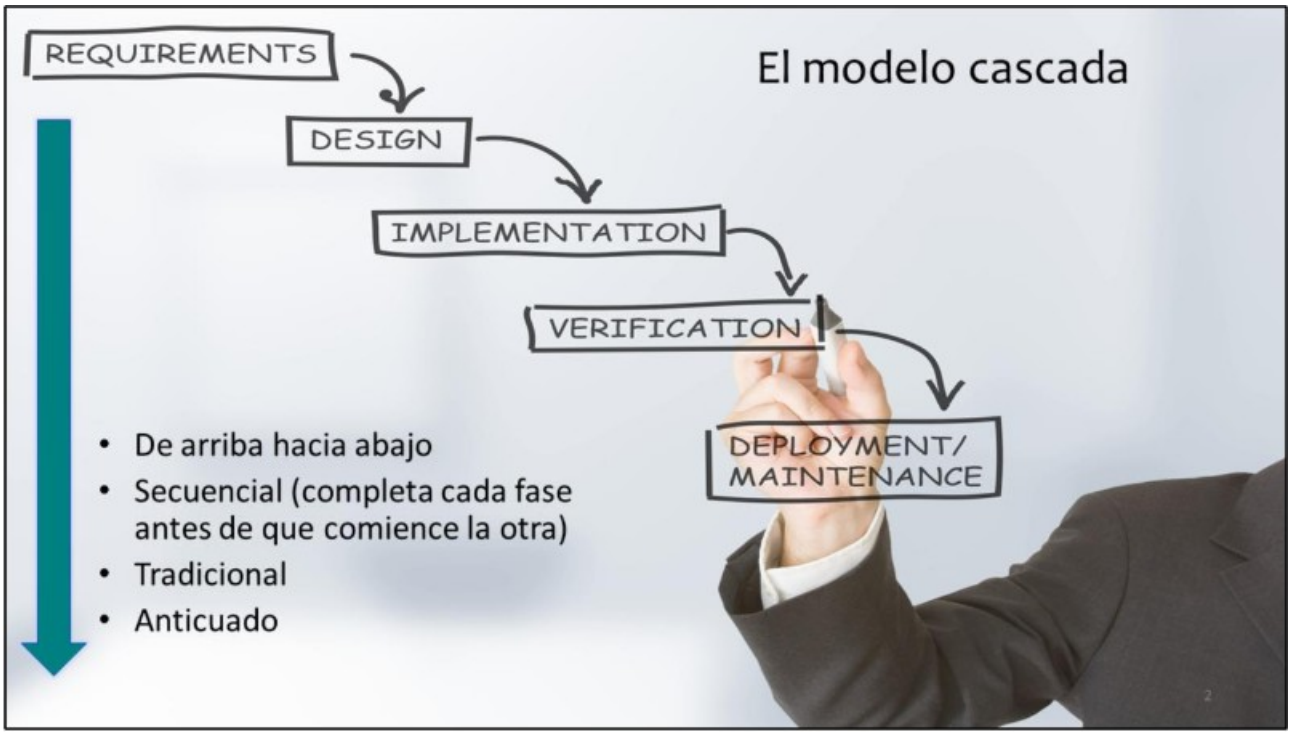
Introducción

Existen varias estrategias para desarrollo de software, algunas más obsoletas que otras.

Modelado en cascada

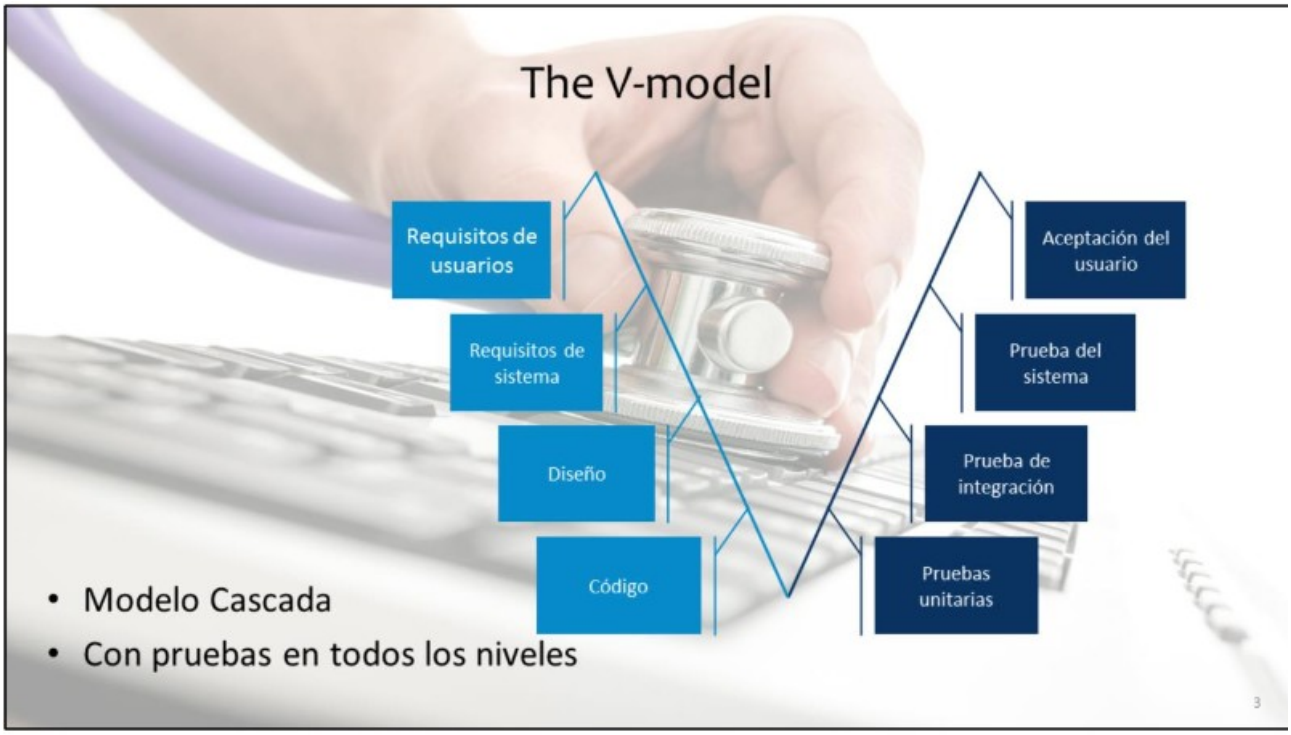
En éste modelado, la implementación es secuencial: en primer lugar se hace el análisis de requisitos, después pasamos al diseño, luego implementación y fijación, luego la validación, después la instalación y al final el mantenimiento.

Es un modelo donde vamos de arriba abajo, es totalmente secuencial y hay que desarrollar completamente una fase, una etapa antes de pasar a la siguiente. Es el modelo tradicional, pero que actualmente está bastante obsoleto.



Modelado en V:

También es un modelo en cascada pero se hacen chequeos en todas las etapas



¿Requisitos del usuario?. Comprobar la aceptación del usuario
¿Requisitos del sistema?. Pruebas del sistema
¿Requisitos de diseño?. Tiene las pruebas de integración
¿La parte de código?. Tiene sus pruebas unitarias.

Entonces es muy usual desarrollar prototipos antes desarrollar el sistema final, desarrollar prototipos que podamos comprobar que satisfacen los requisitos.

Es por todo ésto que el modelo secuencial de arriba abajo nos da una visibilidad muy tardía al usuario, pues hasta que no hemos pasado todas las etapas de requisitos diseño e implementacion y verificación, el usuario no se entera si hemos entendido lo que pretendía él que hiciese la aplicación o si nos lo ha dicho bien que las dos cosas. Nos enteramos muy tarde, con mucho esfuerzo ya invertido en todas estas etapas.

Prototipado
Son un pequeño modelo del sistema final y que con poco esfuerzo nos permita comprobar o le permita al usuario final, comprobar que hemos entendido lo que él pretende que que haga, los requisitos que pretende para la aplicación y que además, lo que nos ha dicho que necesita se cumpla.
En muchos casos cuando en ese primer prototipo, se lo enseñamos, pues él detecta que no es eso, puede ser que no lo hayamos entendido o puede ser que no se haya explicado o que el cliente se dé cuenta que en realidad faltan más cosas, añadir más requisitos.

Esto da lugar a una evaluación constante, incrementa la funcionalidad hasta que el sistema está completo, y construye relativamente el sistema, es un modo iterativo de ir desarrollando en un ciclo el sistema.

Ingenierías de desarrollo ágil de software

Se basa en desarrollo de prototipos y concentran en un tiempo corto, posiblemente unas cuatro semanas (sprint) en producir ya prototipos que podamos mostrar.
Hay menos documentación escrita pero hay más comunicación cara a cara entre los propios desarrolladores y se van produciendo pequeños lanzamientos que nos permiten comprobar con el usuario final si hemos entendido adecuadamente aquello que él pretende, es decir los requisitos que se han fijado.

Desarrollo Rapido de Aplicaciones

El desarrollo rápido de aplicaciones conocido como RAD de Rapid Application Development, consiste en desarrollar también prototipos con un conjunto de aplicaciones (lo que se conoce como JAD o Joint Application Development), donde hay una interacción continua con los usuarios en el proceso de desarrollo.

La interacción con los usuarios funcionales es contiuna y nos concentramos en el uso de herramientas CASE, en la reutilización de componentes ya preconstruidos por nosotros o porque los podemos encontrar en la red o en cualquier sitio y mediante la retroalimentación constante de los usuarios, constantemente estamos interactuando con ellos . Ésto da lugar a procesos donde se cometen menos errores y donde conseguimos con más facilidad entender y alcanzar los requisitos fijados por los usuarios.

Preguntas de test

¿Cuántas fases diferentes tiene el modelo reducido de Ciclo de Vida de Desarollo de Sistemas adoptado como estándar por el CLEP? 5
¿En qué fase del ciclo de vida de desarrollo del sistema (SDLC) necesitamos analizar el nuevo sistema para cumplir con los objetivos de la organización? Sistema de ejecución
¿Cuál de las siguientes NO es una característica de creación de prototipos? Desarrollo Secuencial
Preguntar a un grupo de usuarios para evaluar la funcionabilidad de una interfaz de usuario antes de lanzarla forma parte de ...prototipado
La gestión de la configuración es una herramienta CASE se utiliza en la fase de implementación Falso

Clase 7: Normas y gestión de proyectos

Introducción

Definición: Un proyecto es un esfuerzo temporal emprendido para lograr un producto o un servicio único.
Temporal significa que hay un inicio discreto y definible y una conclusión.

La gestión de un proyecto requiere actividades hechas a medida para dar soporte a ésta característica. Como tal un indicador clave del éxito de un proyecto es la forma en que se comporta en función de su programación, es decir se inicia y termina a tiempo que es el tiempo previsto.

Además hay un único entregable, la singularidad del entregable ya sea un producto, un servicio único o un resultado, requiere un enfoque especial en el sentido que puede que no exista un plan para la ejecución del proyecto y puede que no sea necesario repetir del proyecto una vez que que éste se complete.

La singularidad no significa que no existan similitudes con otros proyectos, pero sí requiere que el alcance de un proyecto en particular tiene entregables que deberán producirse dentro de unas restricciones riesgos, recursos específicos, lugares específicos y dentro de un proceso determinado, por lo tanto el proceso para producir el propio producto es único.

Ejemplos de proyectos

- La construcción de un puente
- Desarrollar un nuevo sistema de reservas online para una compañía aérea
- Desarrollar un nuevo sistema de información para obtener información útil de los clientes a través de una nueva tarjeta asociada a clientes en un supermercado y poder hacer Big Data sobre esta información.
- Proyectos del software IT: se refiera proyectos de hardware, software y redes, a todos estos proyectos.

Atributos de un proyecto

1. Tenemos atributos temporales que determinan el **límite del tiempo**.
2. **Propuestas y alcances únicos**. Es decir que tienen un único propósito y un proyecto ha finalizado cuando se cumplen sus metas y sus objetivos. No necesariamente único que en otros proyectos similares no estén finalizados es único en su contexto
3. **Restricciones de calidad**
4. **Restricciones de costos**
5. **Tiene incertidumbre**
6. **Tiene un actor principal**, por ejemplo, el financiador. Financiación: tenemos que tener claro de dónde viene el dinero, quien lo paga y que se pretende obtener con este producto, por lo tanto si hay una incertidumbre sobre la viabilidad esa primera etapa que tenemos en el desarrollo de software deberíamos hablar con el cliente para poder proponer o discutir con él un posible cambio de objetivos y necesidades.

¿Quiénes participan en un proyecto?.
Hay un **patrocinador** que es la gente que va a a comprar ese resultado, patrocinador que es quien promueve la el desarrollo de la aplicación.
El **financiador** que será en definitiva quien va a dar sustento económico al proyecto.
Beneficiario, que es un usuario final y un cliente final.

Entonces los recursos necesarios en un proyecto son varios pero podemos destacar las personas, las oficinas los equipos, los servicios, el material de oficina y las finanzas.
Todas las financiaciones que son necesarias para el buen desempeño del proyecto.

Por otra parte, participa **toda la gente que está involucrada** desde programadores, analistas, personal de oficina, contratistas, constructores, lo que necesitemos. Tenemos que tener en cuenta también los bienes raíces que se ven implicados en el proyecto, porque en algún sitio habrá que desarrollar esta actividad, se requerirá un edificio, una oficina o un almacén, o sea, el espacio necesario para el proyecto, la maquinaria y el equipamiento necesario, tanto el hardware como el software necesario para desarrollar el proyecto, equipos de oficina, servicios que se puedan requerir, cualquier servicio que se tenga que pagar durante el desarrollo del proyecto. Incluso el material de oficina, toner para impresora, etc.

Usuarios finales y clientes finales también son parte del proyecto.

Principio de la triple restricción de proyectos:
Son el tiempo, los costos y la calidad del producto o servicio final. Ésto es necesario para la gestión de proyectos y para poder equilibrar los objetivos del proyecto y esto no es tan sencillo, es un tema complejo.
Por ejemplo el Standish Group Chaos Manifestó en el 2013, dedujo que solamente y el 39% de los desarrollos de software acaban con éxito total, luego hay otros más con mayores fracasos o menos, pero solamente ése porcentaje, acaban adecuadamente.

Por otra parte, dedujeron que hay unos costos estimados del orden de 81 billones de dólares solamente en USA, entonces este es un tema importante, que se producen por fallos en el desarrollo de sistemas informáticos.

Por qué fallan los proyectos

- **Falta de apoyo a nivel ejecutivo.** El nivel ejecutivo no se implica bastante en el proyecto, no tiene vínculo con la estrategia empresarial.
- **Porque al elegir miembros del equipo nos hemos equivocado.** Por ende no nos dan el rendimiento adecuado
- **Mala comunicación con los usuarios funcionales** o dentro de los miembros del equipo
- **Falta de una medida útil para evaluar unas métricas adecuadas:** para evaluar el éxito del proyecto
- **Falta de gestión de riesgos:** esto es muy importante pues prever la gestión de riesgos, por incapacidad para gestionar el cambio que se va produciendo
- **Planificación deficiente del proyecto.**

Todo esto son motivo entre otros por los cuales un proyecto IT falla.
Necesitamos para la gestión de proyectos equilibrar los objetivos y llevar un proyecto al éxito, esto necesitamos incorporarlo en la gestión de los proyectos dentro de esta tripleta: tiempo, coste y calidad

Clase 7: Gestión de Proyectos

Introducción

Se basa en los principios de triple restricción de proyectos pues es necesario para la gestión de proyectos equilibrar estos tres elementos, tiempo, costo y calidad.

Definición Gestión de proyectos es el arte de organizar, dirigir, informar y completar un proyecto a través de las personas.

Los objetivos de la gestión de proyectos son

1. Lograr que se alcancen los objetivos que hemos establecido para el proyecto, en tiempo en costo y en calidad
2. Mantener a las partes interesadas satisfechas tanto a los programadores, a la gente que participa en el desarrollo del proyecto, como los usuarios finales, a los clientes para los que va a desarrollar, va va dirigido el proyecto.
3. Mantener al equipo enfocado en el objetivo.
4. Asegurar que los miembros del equipo cumplen con los objetivos de trabajo establecidos.

Ejemplo

¿Cómo elaboraríamos los planes de negocio para una empresa de biotecnología? Tendríamos que establecer los recursos, los horarios, el presupuesto, etc. Todo esto entra en la gestión de del proyecto

Gestión en proyectos IT

Ser gestor de proyectos IT es ahora una de las profesiones más demandadas, pero para esto se se requieren habilidades técnicas, hay que saber y entender sobre qué es el trabajo, sobre el problema que vamos a resolver, sobre lo que debe hacerse, la cantidad de esfuerzo que se requiere, y los riesgos que pueden surgir en el desarrollo del proyecto.

Se necesitan habilidades de planificación: planificar el trabajo que debe realizarse utilizando los recursos disponibles que tenemos, tanto materiales como humanos; se necesitan habilidades de las personas, o sea ser capaz de crear equipo de mantener al equipo focalizado, de mantener a los interesados satisfechos.

También se necesitan habilidades de comunicación, habilidades para escuchar a las partes interesadas, especialmente a los usuarios y para hablar, para dirigirnos al equipo y tener empatía con el equipo, llegar al equipo y con esto poder alcanzar las tareas necesarias; saber las necesidades del equipo, poder establecerlo y también las necesidades individuales, propias de los miembros del equipo.

Entonces en la gestión de proyectos **todo evoluciona a través o desde lo que es el plan**, el corazón de la gestión de proyectos es la planificación, pero en este plan tenemos que tener en cuenta distintas cosas como son las tareas, la ejecución, la gente y los riesgos, tenemos que analizarlo dentro de la gestión del proyecto.

Planificación

Hay que **definir las tareas y los paquetes de trabajo**, y para ello normalmente organizamos el desarrollo del proyecto por distintos paquetes de trabajo que son coherentes entre sí y estos paquetes de trabajo a su vez se organizan en tareas.

Hay que **identificar la entrega de las tareas**, hay que establecer un plan temporal de cuando deben acabarse, los hitos en el que cada tarea debe estar finalizada porque normalmente y en muchos casos los resultados de una tarea son entrada de otras tareas.

Hay que **identificar los recursos disponibles** tanto los humanos como los materiales, hay que identificar claramente los riesgos que pueden hacer peligrar el buen desarrollo del proyecto y prever planes de contingencia ante esos riesgos.

Hay que **asignar responsabilidades** a los miembros del equipo y hay que hacer el Schedule, planear el calendario en tiempo de todo el proyecto.

Puede que se requiera replanificar, por distintos motivos: porque se retrase una tarea, incluso porque se adelante alguna tarea, pero esto tiene que tener que estar muy claro y hay que intentar ser rigurosos en el cumplimiento de los plazos, sobre todo nunca, nunca retrasarlos.

En la ejecución y monitorización del proyecto lo que tenemos que hacer es **medir y monitorizar el progreso del proyecto**, o sea que estar pendientes de ver que los hitos se van cumpliendo, los hitos temporales se van cumpliendo.

Tenemos que **facilitar la comunicación entre los miembros del equipo** principalmente y también con el cliente y debemos conseguir crear esa sinergia dentro del equipo, crear ese equipo, crear el espíritu de equipo dentro del grupo de trabajo.

En la ejecución y monitorización del proyecto lo que tenemos que hacer es **medir y monitorizar el progreso del proyecto**, o sea que estar pendientes de ver que los hitos se van cumpliendo, los hitos temporales se van cumpliendo.

Tenemos que **facilitar la comunicación entre los miembros del equipo** principalmente y también con el cliente y debemos conseguir crear esa sinergia dentro del equipo, crear ese equipo, crear el espíritu de equipo dentro del grupo de trabajo.

Ejecución y monitorizción de proyectos

En la ejecución y monitorización del proyecto lo que tenemos que hacer es **medir y monitorizar el progreso del proyecto**, o sea que estar pendientes de ver que los hitos se van cumpliendo, los hitos temporales se van cumpliendo.

Tenemos que **facilitar la comunicación entre los miembros del equipo** principalmente y también con el cliente y debemos conseguir crear esa sinergia dentro del equipo, crear ese equipo, crear el espíritu de equipo dentro del grupo de trabajo.

Gestión de riesgos

En la gestión de riesgos, en primer lugar debemos identificar los riesgos, ver en aquellos puntos donde podemos fracasar, porque no lo vamos a cumplir en el tiempo o porque el objetivo que nos hemos previsto es muy complicado y podía haberse comprometido.

Por otra parte tenemos que evaluar la probabilidad de que se produzca cada fallo, que se de cada uno de esos riesgos y el impacto, que puede tener ese de ese riesgo en el desarrollo del proyecto.

Tenemos que definir también en el caso en que se de esa situación de riesgo cómo debemos actuar, tenemos que tener siempre previsto un plan b para resolver esa deficiencia y tenemos que monitorizar y renovar.

Cuando veamos que algo no se puede dar o que se ha dado ese riesgo deberemos renovar el plan para poder seguir satisfaciendo los requisitos establecidos y garantizar tanto la calidad, como el tiempo, como el costo del proyecto


Gestión de riesgos
Identificación de riesgos
Evaluar la probabilidad y el impacto
Definir acciones para cuando aparezcan
Monitorizar y renovar

Clase 8: Planificación

Introducción

La planificación de un proyecto requiere la definición de las tareas en paquetes de trabajo, identificar la entrega de las tareas, e identificar los recursos, identificar los riesgos, asignar responsabilidades y planear el calendario del tiempo.

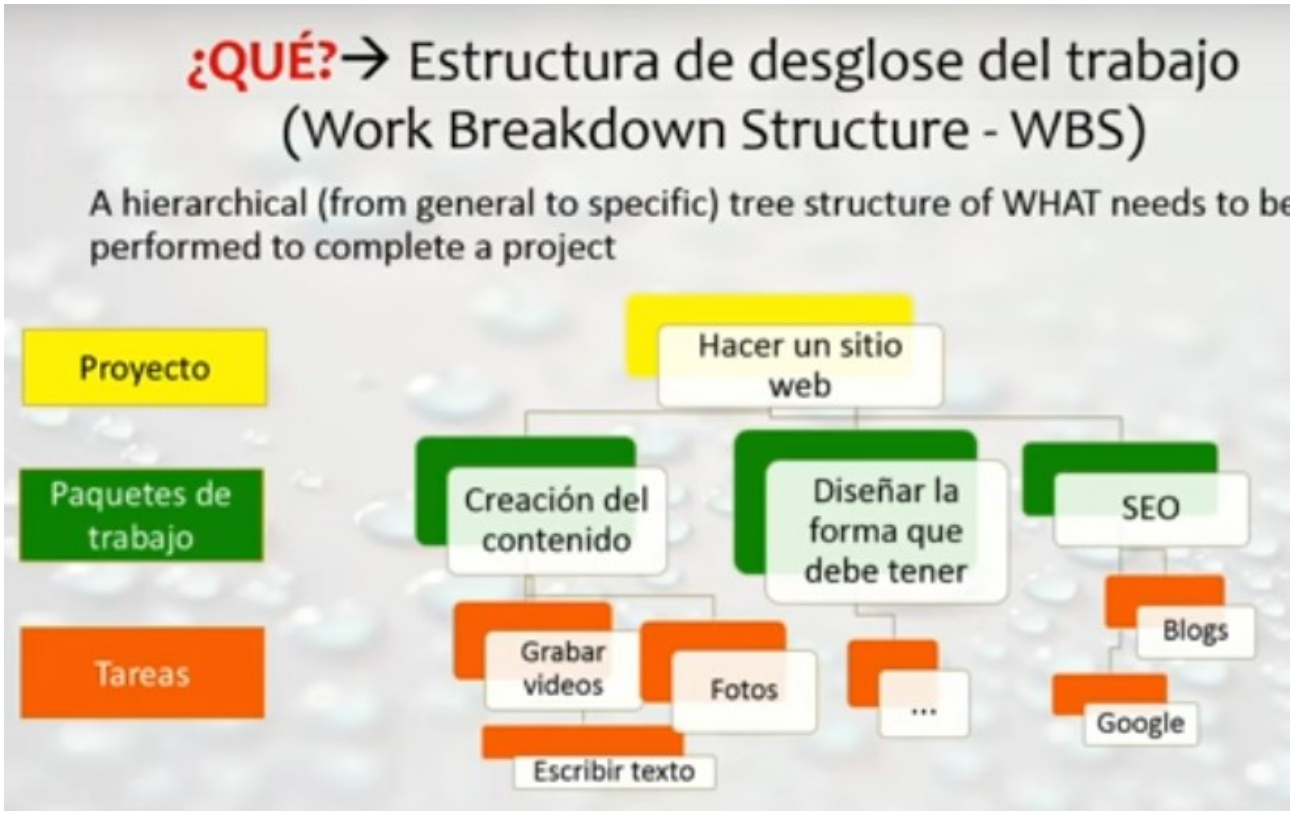
Entonces tanto las tareas y los paquetes de trabajo deben ser SMART(Specific, Measurable, Achievable, Realistic and Timed), es decir, deben ser específicos, medibles, realizables, realistas y deben de estar acotados en el tiempo.

Work Breakdown Structure:

La estructura de desglose del trabajo es una jerarquía que representamos mediante una estructura de de arbol, donde vamos de lo más general a lo más específico y que se tiene que desarrollar para desarrollar totalmente el proyecto
Por ejemplo, podemos tener un proyecto que es hacer un sitio web: establecemos distintos paquetes de trabajo como hay un paquete de trabajo que es crear el contenido, otro que es diseñar la forma que debe tener la parte visual de la página web y luego tenemos el SEO, o sea todo el sistema que lo gestiona.
Entonces la creación del contenido implica que hay que grabar vídeos hay que hacer fotos o recopilar fotos, hay que escribir texto aquí hay que a utilizar las técnicas adecuadas para diseñar la página, aquí hay que ver qué blogs o como acceder o informarnos desde desde Google.

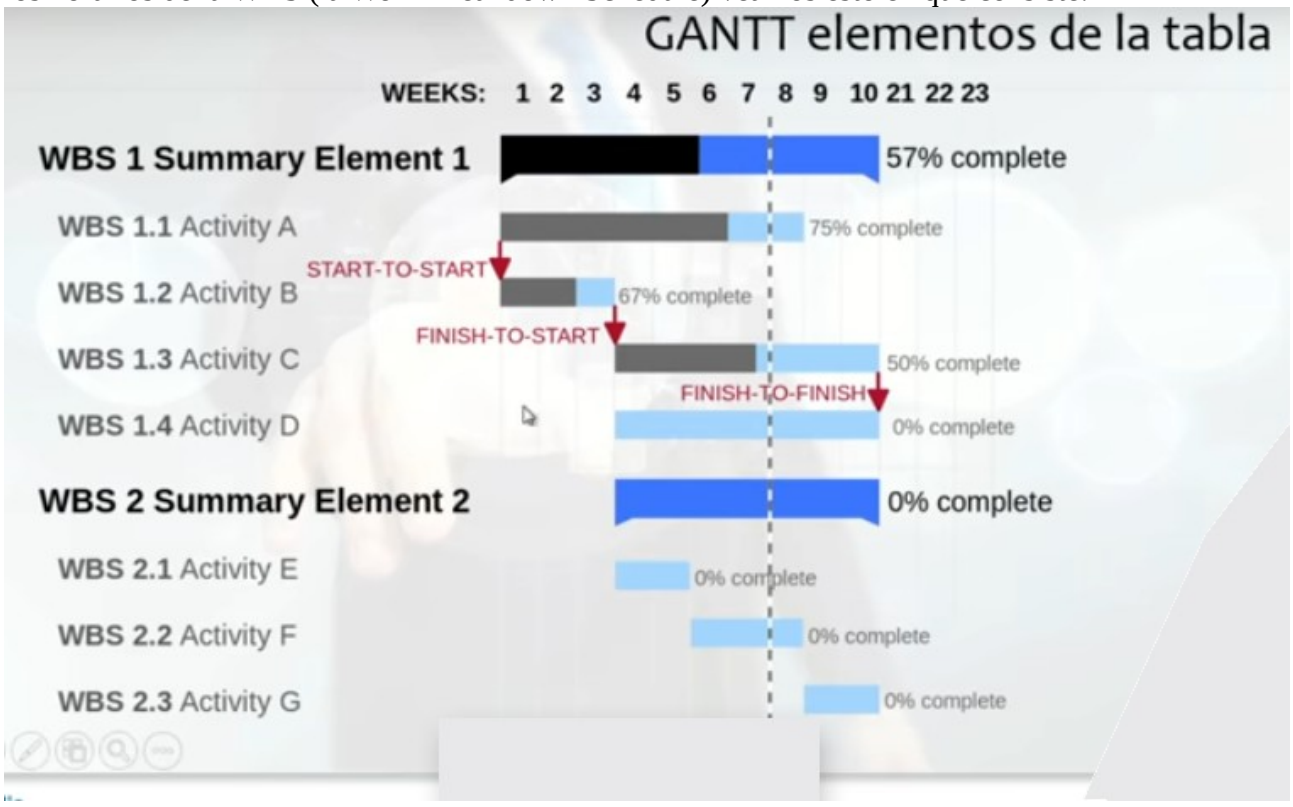
Todo esto ayuda identificar todo el trabajo que se necesita hacer, lógicamente organizar el trabajo para que pueda ser programado, tiene que ser algo realista, cuando tenemos ya estas tareas concretas, y entonses podemos asignar trabajos a los miembros del equipo y podemos asignar responsabilidad también a los work packages.

También podemos identificar los recursos necesarios y podemos comunicar que se tiene que hacer, aquí y a la gente sepa que cosas concretas tiene que hacer y podemos utilizar, organizar el trabajo mediante objetivos utilizando técnicas basadas en objetivos.



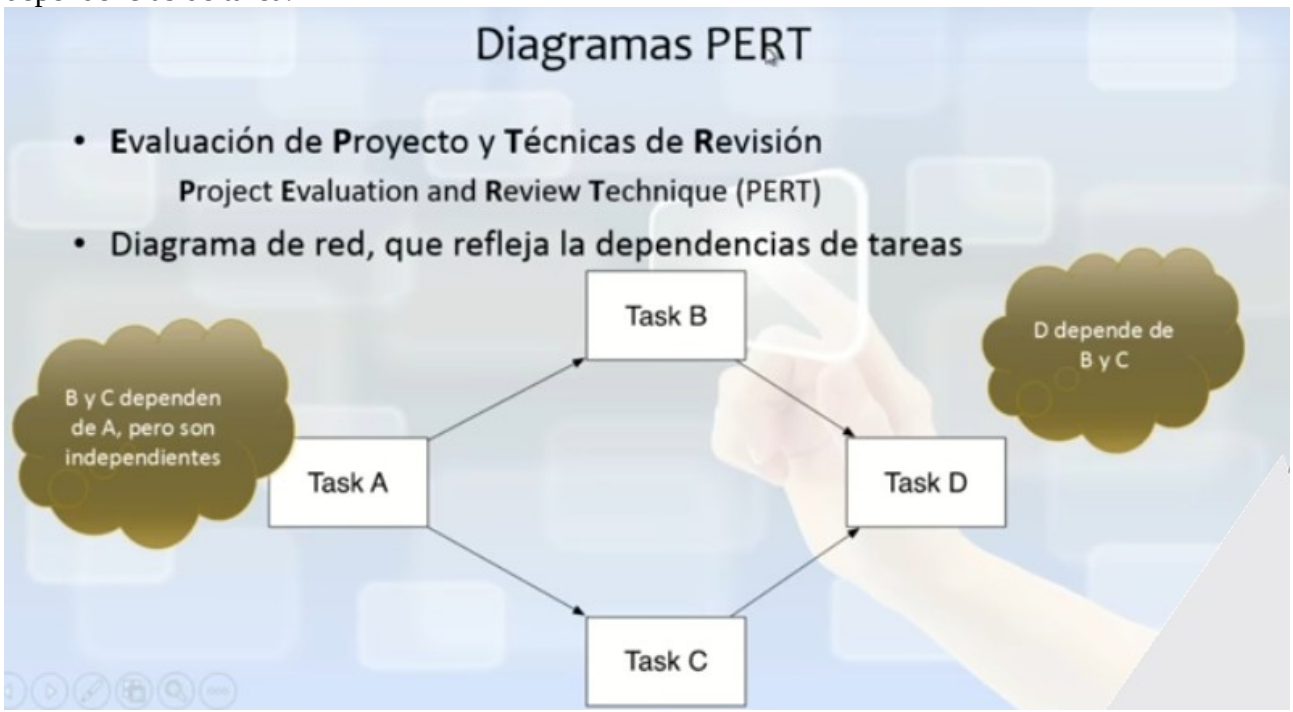
Diagramas GANTT

Los diagramas de GANTT, fueron propuestos en 1910 por Henry Gantt, y permite pues establecer los horarios de la WBS (la Work Breakdown Schedule) veamos esto en qué consiste:



Esto es un diagrama de Gantt y tenemos muchas aplicaciones que podemos utilizar para construir diagramas de Gantt, entonces esto indica que este WBS empezará en este, en el mes 1, supongamos que estos son meses, esto será lo que está desarrollado.
La linea punteada es donde estamos actualmente y, hacia la derecha es lo que nos quedaría por realizar, porque está establecido que esto acabará en este mes.
Cada work package puede estar constituido, por distintas tareas o actividades.
Las actividades A y B tienen que empezar simultáneamente porque se requiere de hacerlo así.
Por otra parte, la actividad B tiene que hacerce antes que C y D y debe eesto indica que esta tarea no puede empezar hasta que esta no ha finalizado, y estos son cosas que hemos analizado a partir del conocimiento que tenemos en el desarrollo de software y en la aplicación concreta que hemos analizado.

Diagramas PERT
En algunos casos también tenemos que establecer relaciones de precedencia, entonces podemos utilizar los diagramas PERT, que y nos permiten la evaluación de proyectos y técnicas d de revisión, de aquí viene las siglas de PERT en inglés, entonces se construye una red que refleja las dependencias de tarea.



Esto quiere decir que la tarea B y la tarea C dependen de la tarea A. Son independientes entre ellas pero ambas dependen de la tarea A, y que la tarea D depende de latarea B y de la tarea C, y por la transitividad que se da aquí, la tarea D también que dependerá de la tarea A.

Entonces podemos hacer tareas con project managment y tenemos distintas herramientas de project managment, donde se pueden detectar cuellos de botella dados por tareas críticas donde la evolución del proyecto depende de esta tarea. Si esto pudiésemos evitarlo sería mejor y también tenemos tareas, igual que estas que se pueden desarrollar en paralelo. Esas pueden desarrollarse paralelamente porque todas confluyen, y en principio no requieren ninguna ninguna sincronización.

Clase 9: Riesgos

Introducción

La gestión de riesgo forma parte de la gestión de proyectos.
Definición Los riesgos del proyecto por definición son eventos, condiciones inciertas que si ocurren tiene un efecto positivo o negativo en el objetivo del proyecto.

Evidentemente los que nos van a preocupar principalmente son aquellos que tengan un efecto negativo, como hemos visto anteriormente en la gestión de riesgos tenemos que identificar los posibles riesgos, evaluar la probabilidad y el impacto de esos riesgos, definir y qué acciones adoptar cuando aparezca, si se produce ese riesgo y monitorizar el proceso de desarrollo para intentar detectar esos riesgos y entonces actuar en consecuencia revisando el proceso.

Identificación de riesgos

Para ésto se realizan talleres de riesgos donde se reune a la gente que participa para evaluar los posibles riesgos del proyecto, siendo el resultado de la reunión, una lista donde clasificaremos los distintos riesgos que hemos identificado y los clasificaremos en probabilidad, en impacto y en gravedad para el buen cumplimiento del proyecto.

Catalogación de riesgos

Tendremos tres niveles: alto, medio y bajo.

1. En el nivel alto tendrá una probabilidad alta, es decir se puede esperar que suceda y el stateholder pierde de dinero si se produce ese riesgo
2. En un riesgo medio puede o no suceder, pero la compañía también pierde dinero si ocurre.
3. En el nivel de riesgo bajo no es probable que suceda, pero puede ser significativo y no hay pérdida monetaria, pero se producen molestias.

	Probabilidad	Impacto
Alto	Se puede esperar que suceda	El stakeholder pierde dinero
Medio	Puede o no suceder	Nuestra compañía pierde dinero
Bajo	No es probable que suceda, pero puede ser significativo	No hay pérdida de dinero sólo molestia

Ejemplos de estos riesgos son que podemos tener riesgos empresariales:

- un cambio de mercado, que altera el atractivo comercial de la aplicación, o de los resultados de la aplicación
- un mal juicio de mercado, los clientes puede que no compren, que no estén interesados en el producto
- la opinión pública de la marca de las firmas ha cambiado, cuando se ha acabado ya, o en el uso de la misma.

- hay otros proyectos que compiten con recursos, o mejores recursos
- limitaciones legales o ambientales.

También tenemos riesgos del proyecto en general:

- por la gestión de apoyo
- mala gestión o la falta de gestión
- la aceptación del usuario
- el subcontratista no hace la entrega a tiempo
- la incertidumbre acerca de los requisitos del usuario cuando no tenemos claro si el usuario nos ha dicho todos los requisitos que tiene
- el equipo que hemos constituido no tiene las habilidades requeridas
- la tecnología preveíamos que iba a darnos unas funcionalidades y no es así. Podemos estar utilizando una tecnología que está anticuada, ha quedado obsoleta, hay problemas entre las personas dentro del equipo, y muchos y muchos más que podríamos identificar.
- Y otros...

Entonces aquí tenemos un ejemplo de registro de riesgos, por ejemplo las pruebas de integración pueden fallar, probabilidad media, impacto medio, y es grave, un nivel cuatro este sería más grave

¿qué podemos hacer? Acción: hacer una revisión después de las pruebas unitarias.
¿Quién es el responsable?. El jefe del proyecto.

¿Otro riesgo? Pues los nuevos servidores no están listos antes del comienzo de desarrollo, pues esto tendría una probabilidad media, el impacto sería alto, la gravedad sería alta también y ¿qué tendremos que hacer?. Una acción sería comprar los servidores inmediatamente, ¿quién es el responsable?. Pues uno de los miembros del equipo.

Otro o riesgo que podemos identificar: la documentación de los usuarios no se imprime a tiempo. Ésta es más difícil que se produzca, tendrá una probabilidad baja, el impacto es medio y no es muy grave, pues incluir una versión online de la documentación para que los usuarios puedan tenerla es una solución.

Con eso se construye la tabla de riesgos

ID	Descripción	Probabilidad	Impacto	Gravedad	Acción	Responsabilidad
1	Las pruebas de integración pueden fallar	M	M	4	Hacer una revisión después de las pruebas unitarias	Jefe de proyecto
2	Los nuevos servidores no están listos antes del comienzo del desarrollo	M	A	6	Comprar los servidores inmediatamente	Ramón
3	La documentación de los usuarios no se imprime a tiempo	B	M	2	Incluir una versión online de la documentación	Sylvain

Clase 10: Software para Gestión de Proyectos

Introducción

El corazón de la gestión de proyectos está la planificación, entonces podemos encontrar diversas herramientas que nos ayudan en la planificación.
Para realizar diagramas GANTT o para de realizar diagramas PERT, por ejemplo en Microsoft Project de Windows o el Open Proj, que es una variante de código abierto de Microsoft Proyect, podemos encontrar calendarios electrónicos que podemos programar para que notifiquen y para recordar hitos que se tienen que producir.

Podemos encontrar software de colaboración para la comunicación, mediante videoconferencias Emails, mensajería instantánea, para compartir la información, hojas de cálculo y documentos online.
También existen herramientas colaborativos que nos permiten compartir documentación, sistemas de seguimiento de problemas y también software social que permite organizar la comunicación dentro del equipo, podemos disponer de software para monitorizar y controlar tanto la predicción del mercado WorkFlow Management del sistema.

Tenemos muchas herramientas software que nos pueden ayudar en la gestión de del proyecto, tenemos también software que nos ayuda en la gestión de riesgos, tanto en la identificación de los riesgos, como en recoger y organizar los riesgos.

Clase 11: Estandares

Introducción

En el desarrollo de software los estándares son muy importantes porque en muchos casos queremos reutilizar software, o queremos conectar piezas de sofware con otras piezas de software. Ésto requiere estandarización, que si no existiera sería imposible.
También cuando queremos conectar a un hardware determinado ciertos dispositivos, si esos dispositivos no cumplen unos estándares, pues será difícil el poder crear ese sistema de información.

Pero además comunicar a la gente de un proyecto o comunicar piezas de software, consiste en interactuar e intercambiar información, por lo tanto, si tenemos que intercambiar información necesitaremos definir estándares para poder realizar ese intercambio, por ejemplo si a través de la red tenemos que enviar una información desde un computador a otro, aquí la hemos construido de una forma, la hemos empaquetado, la hemos construido de una forma, cuando llegue al otro ordenador debe saber cómo está construido para poder de alguna forma entenderla. Para eso necesitamos establecer estándares de comunicación, o sea tcp-ip o http.

Los estándares son muy importantes, dado que son un conjunto de reglas, de condiciones de directrices, que permiten a la organización y combinaciones de hardware y de software de diferentes fabricantes y que estos puedan estar comunicados el uno con otro. Por ejemplo, el mundo real ¿cuáles son los dispositivos de entrada a un computador? el teclado, ¿en qué se convierte la info que se le introduce en el teclado?. Al final las salidas del teclado que llegan computador todos eran al final unos y ceros. O la imagen que llega de una cámara digital también serán ceros y unos, pero en un formato determinado para que luego sea comprensible por otro software por otro hardware.

En general entonces hay distintas organizaciones que se dedican a la estandarización como es el instituto IEEE, o como es ISO (International Standard Organization), o el ANSI (Instituto Nacional Americano de Estándares) entre otros.

Estandares en lenguajes de programación

Los estándares para lenguajes de programación es evidente. Es decir si luego queremos que un programa que hemos escrito en el lenguaje lo podamos trasladar a cualquier máquina compilando en esa máquina o trasladando el ejecutable a esa máquina, necesitamos estandarizar, entonces los lenguajes de programación de alto nivel como pueden ser C, C++, Java, Ada, Python, Visual Basic, .NET, tienen que estar estandarizados para poder ser utilizados en distintos ordenadores.

La necesidad de compilar el código escrito en esos lenguajes de alto nivel, en un lenguaje legible, en un lenguaje máquina legible por los ordenadores también tiene que estar estandarizada. Necesitamos estandarizar y para que el software en diferentes lenguajes pueda ser ejecutado también en distintos ordenadores depositamos estándares.

Estandares en formatos de archivos

También necesitamos estándares para los formatos de datos por ejemplo el alfanumérico, tenemos el ASCII, EBCDIC, Unicode, o formatos para imagen como puede ser JPG, GIF, PCX, TIFF, o para imágenes en movimiento como puede ser MPEG-2, Quick Time, o de sonido somo puede ser Sound Blaster, o WAV, o para gráficos y fuentes de salida como PostScript, True Type o PDF.

Entonces tenemos distintos estándares, también para protocolos de comunicación que definen el orden y el formato del dato a intercambiar como puede ser http, ftp, tcp-ip, que permiten que

distintas máquinas puedan intercambiar información entre sí. En internet tenemos un consorcio que es el W3, donde se han definido distintos estándares sobre cómo hacer las páginas en internet, esto permite luego que distintos navegadores puedan reproducir esas páginas o poder acceder a información, también está el proyecto de Web Semántica que define unos estándares para que aplicaciones que van por la red puedan tomar piezas de información que están en esa red y entender lo que esas piezas de de información significan, entonces hay distintos estándares como HTML, XHTML, CCS, XML, que se utilizan para el diseño Web.