

Contenido Guia 1 -Gestion de Proyectos-

1.Enuncie en que se centra la gestión eficaz de un proyecto de software.	2
2. Caracterice el modelo de madurez de la capacidad gestión de personal.	2
3. Caracterice producto software (objetivos y ámbito).....	3
4. Caracterice proceso software.....	3
5. Caracterice proyecto software.	3
6. Enuncie y describa categorías de participantes que componen el proceso del software.	4
7. Defina el modelo de gestión de proyectos MOI.	4
8. Enuncie opciones que pueden aplicarse a los recursos humanos de un proyecto que requiere “n” personas trabajando durante “k” años.	5
9. Enuncie los organigramas de equipo genéricos propuestos por Mantei.....	5
10.Fundamente la utilidad de cada organigrama.....	6
11.Enuncie los factores de un proyecto propuestos por Mantei que deberían considerarse cuando se planifica el organigrama de equipos de ingeniería del software.7	
12.Enuncie los paradigmas de organización para equipos de ingeniería del software propuestos por Constantine.	7
13.Enuncie premisas para conseguir un equipo de alto rendimiento.....	8
14.Enuncie factores que fomentan un entorno de equipo tóxico potencial según Jackman.....	8
15.Grafique valor y empleo de técnicas de coordinación y comunicación.....	9
16.Enuncie las categorías de técnicas de coordinación de proyectos propuestas por Kraul y Streeter.	10
17.Enuncie las cuestiones que definen el ámbito del software.....	10
18.Enuncie modelos de proceso.....	11
19.Enuncie actividades estructurales de la organización de producción de software....	11
20.Defina señales que indican que un proyecto de sistemas de información está en peligro según Reel.	12
21.Enuncie las partes que Reel sugiere como aproximación a proyectos de software. .	12
22.Enuncie las preguntas propuestas por Bohem que conducen a la definición de las características clave del proyecto y el plan del proyecto resultante.....	13
23.Enuncie las preguntas que el Concilio Airlie ha desarrollado para tener una “Visión Rápida” sobre si un proyecto ha implementado prácticas críticas.....	14

1. Enuncie en que se centra la gestión eficaz de un proyecto de software.

La gestión eficaz de un proyecto de software se centra En las cuatro p's:

- Personal
- Producto
- Proceso
- Proyecto

Dicho orden no es en vano.

Personal: El gestor que se olvida de que el trabajo de ingeniería del software es un esfuerzo humano intenso nunca tendrá éxito en la gestión de proyectos.

Producto: Un gestor que no fomenta una minuciosa comunicación con el cliente al principio de la evolución del proyecto se arriesga a construir una elegante solución para un problema equivocado

Proceso: El administrador que presta poca atención al proceso corre el riesgo de arrojar métodos técnicos y herramientas eficaces al vacío.

Proyecto: El gestor que emprende un proyecto sin un plan sólido arriesga el éxito del producto.

2. Caracterice el modelo de madurez de la capacidad gestión de personal.

El **modelo de madurez de gestión de personal (MMCGP)** define las siguientes áreas clave prácticas para el personal que desarrolla software:

- Reclutamiento
- Selección
- Gestión de rendimiento
- Entrenamiento
- Retribución
- Desarrollo de la carrera,
- Diseño de la organización y del trabajo
- Desarrollo cultural y de espíritu de equipo.

El MMCGP es compañero del modelo de madurez de la capacidad software, que guía a las organizaciones en la creación de un proceso de software maduro.

3. Caracterice producto software (objetivos y ámbito).

- **Los objetivos** identifican las metas generales del proyecto sin considerar cómo se conseguirán (desde el punto de vista del cliente).
- **El ámbito** identifica los datos primarios, funciones y comportamientos que caracterizan al producto, y, más importante, intenta *abordar* estas características de una manera cuantitativa.

Es importante antes de poder planificar un proyecto establecer ambas cosas, para así poder identificar de forma más simple soluciones alternativas, dificultades que puede presentar el proyecto tanto en técnicas como en su gestión. Sin dicha información es imposible definir costos, riesgos, división de tareas o identificar el progreso dentro del mismo proyecto.

4. Caracterice proceso software.

Un proceso de software **proporciona la estructura desde la que se puede establecer un detallado plan para el desarrollo del software.**

Un pequeño número de actividades estructurales se puede aplicar a todos los proyectos de software, sin tener en cuenta su tamaño o complejidad.

Diferentes conjuntos de tareas

- Tareas
- Hitos
- Productos del trabajo
- Puntos de garantía de calidad

permiten a las actividades estructurales adaptarse a las características del proyecto de software y a los requisitos del equipo del proyecto.

Finalmente, las actividades protectoras tales como

- Garantía de calidad del software
- Gestión de la configuración del software
- Medición

Cubren el modelo de proceso. **Las actividades protectoras son independientes de las estructurales y tienen lugar a lo largo del proceso.**

5. Caracterice proyecto software.

Dirigimos los proyectos de software planificados y controlados por una razón principal -es la Única manera conocida de gestionar la complejidad-. Y todavía seguimos esforzándonos.

Para evitar el fracaso del proyecto, un gestor de proyectos de software y los ingenieros de software que construyeron el producto deben eludir un conjunto de señales de peligro comunes; comprender los factores del éxito críticos que conducen a la gestión correcta del proyecto y desarrollar un enfoque de sentido común para planificar, supervisar y controlar el proyecto.

6. Enuncie y describa categorías de participantes que componen el proceso del software.

1. **Gestores superiores**, que definen los aspectos de negocios que a menudo tienen una significativa influencia en el proyecto.
2. **Gestores (técnicos) del proyecto**, que deben planificar, motivar, organizar y controlar a los profesionales que realizan el trabajo de software.
3. **Profesionales**, que proporcionan las capacidades técnicas necesarias para la ingeniería de un producto o aplicación.
4. **Clientes**, que especifican los requisitos para la ingeniería del software y otros elementos que tienen menor influencia en el resultado.
5. **Usuarios Finales** que interaccionan con el software una vez que se ha entregado para la producción

Para ser eficientes, el equipo del proyecto debe organizarse de manera que maximice y utilice correctamente las habilidades y capacidades de cada persona. Y este es el trabajo del jefe del equipo.

7. Defina el modelo de gestión de proyectos MOI.

Motivación. La habilidad para motivar (con un tira y afloja») personal técnico para que produzca conforme a sus mejores capacidades.

Organización. La habilidad para amoldar procesos existentes (*O* inventar unos nuevos) que permita al concepto inicial transformarse en un producto

Ideas o innovación. La habilidad para motivar al personal para crear y sentirse creativos incluso cuando deban de trabajar dentro de los límites establecidos para un producto o aplicación de software particular.

Como observación personal **un jefe de equipo no solo debe saber sobre desarrollo de software en sí, sino también como guiar al equipo.** Así como un director de orquesta puede que no sepa o domine todos los instrumentos que componen dicha orquesta, pero si sabe como dirigirla de forma eficiente. Lamentablemente en muchos ámbitos los jefes de equipo carecen de ambas habilidades.

8. Enuncie opciones que pueden aplicarse a los recursos humanos de un proyecto que requiere “n” personas trabajando durante “k” años.

1. **n individuos** son asignados a **m diferentes tareas** funcionales, tiene lugar relativamente poco trabajo conjunto; la coordinación es responsabilidad del gestor del software que puede que tenga otros seis proyectos de los que preocuparse.

2. **n individuos** son asignados a m diferentes tareas funcionales ($m < n$) de manera que se establecen «equipos ~informales; se puede nombrar un líder al efecto; la coordinación entre los equipos es responsabilidad de un gestor del software.

3. **n individuos** se organizan en **t equipos**; a cada equipo se le asignan una o más tareas funcionales; cada equipo tiene una estructura específica que se define para todos los equipos que trabajan en el proyecto; la coordinación es controlada por el equipo y por el gestor del proyecto de software.

9. Enuncie los organigramas de equipo genéricos propuestos por Mantei

Descentralizado democrático (DD).

- No tiene un jefe permanente. Se nombran coordinadores de tareas a corto plazo y se sustituyen por otros para diferentes tareas.
- Las decisiones y los enfoques se hacen por consenso.
- La comunicación entre los miembros del equipo es horizontal.

Descentralizado controlado (DC).

- Tiene un jefe definido que coordina tareas específicas y jefes secundarios que tienen responsabilidades sobre subtareas.
- La solución de problemas es una actividad del grupo, pero la implementación de soluciones se reparte entre los subgrupos por el jefe de equipo.
- La comunicación entre subgrupos e individuos es horizontal.
- También hay comunicación vertical a lo largo de la jerarquía de control.

Centralizado controlado (CC).

- El jefe de equipo se encarga de la solución de problemas a alto nivel y la coordinación interna del equipo.
- La comunicación entre el jefe y los miembros del equipo es vertical.

10. Fundamente la utilidad de cada organigrama.

Descentralizado democrático (DD).

- * es la mejor para problemas difíciles
- * más probabilidades de éxito en la resolución de problemas complejos
- * producen una moral más alta y más satisfacción por el trabajo y son, por tanto, buenos para equipos que permanecerán juntos durante mucho tiempo
- * se aplica mejor a problemas con modularidad relativamente baja, debido a la gran cantidad de comunicación que se necesita.
- * requieren mas tiempo para completar proyectos
- * son ideales cuando se necesita gran cantidad de comunicación

Descentralizado controlado (DC).

- * producen menos defectos que los equipos DD
- * mejor dirección para proyectos grandes donde se pueden crear subgrupos
- * requieren más tiempo para completar proyectos
- * son ideales cuando se necesita gran cantidad de comunicación

Centralizado controlado (CC).

- * realiza las tareas más rápidamente
- * Es mejor para problemas sencillos
- * producen menos defectos que los equipos DD
- * mejor dirección para proyectos grandes donde se pueden crear subgrupos

11. Enuncie los factores de un proyecto propuestos por Mantei que deberían considerarse cuando se planifica el organigrama de equipos de ingeniería del software.

Existen siete factores de un proyecto que deben considerarse al planificar el organigrama de equipos.

- Dificultad del problema.
- Tamaño del programa en líneas de código o puntos de función.
- Tiempo de vida del equipo.
- Grado en que el problema puede ser modularizado.
- Calidad requerida y fiabilidad del sistema que se va a construir.
- Rigidez de la fecha de entrega.
- Grado de comunicación requerido para el proyecto.

12. Enuncie los paradigmas de organización para equipos de ingeniería del software propuestos por Constantine.

Paradigmas de organización de equipos según Constantine

- Paradigma cerrado.
 - Tiene jerarquía tradicional de autoridad similar al equipo CC.
 - Trabajan bien cuando producen software similar a otros anteriores, pero son menos innovadores.
- Paradigma aleatorio.
 - El equipo se estructura libremente y depende de la iniciativa individual de los miembros.
 - Son buenos cuando se requiere innovación o avances tecnológicos.
 - Tienen problemas cuando se requiere un rendimiento ordenado.
- Paradigma abierto.
 - Estructura el equipo de forma que consiga algunos de los controles asociados con el paradigma cerrado y mucha de la innovación del paradigma aleatorio.
 - El trabajo se desarrolla en colaboración, con mucha comunicación y toma de decisiones consensuadas.
 - Son adecuados para resolver problemas complejos, pero pueden no ser tan eficientes como otros equipos.
- Paradigma sincronizado.
 - Se basa en la partición natural de un problema y organiza los miembros del equipo para trabajar en partes del problema con poca comunicación activa entre ellos.

13. Enuncie premisas para conseguir un equipo de alto rendimiento

El objetivo principal de una organización de Ingeniería del Software debe ser «convertir el caos en un equipo de alto rendimiento»

- Para conseguir un equipo de alto rendimiento.
- Los miembros del equipo deben confiar unos en otros.
- La distribución de habilidades debe adecuarse al problema.
- Para mantener la unión del equipo, los inconformistas tienen que ser excluidos del mismo

14. Enuncie factores que fomentan un entorno de equipo tóxico según Jackman.

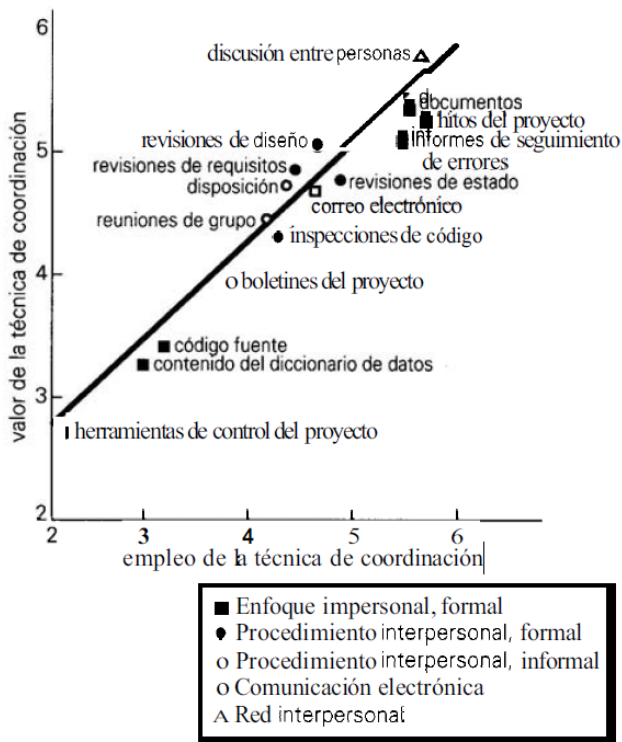
- **una atmósfera de trabajo frenética** en la que los miembros del equipo gastan energía y se descentran de los objetivos del trabajo a desarrollar;
- **alta frustración causada por factores tecnológicos**, del negocio, o personales que provocan fricción entre los miembros del equipo
- **«procedimientos coordinados pobremente o fragmentados»** o una definición pobre o impropia elegida del modelo de procesos que se convierte en un obstáculo a saltar;
- **definición confusa de los papeles a desempeñar** produciendo una falta de responsabilidad y la acusación correspondiente
- **«continua y repetida exposición al fallo»** que conduce a una pérdida de confianza y a una caída de la moral.

Jackman también propone algunos antitóxicos que vale la pena mencionar.

- la información requerida para hacer el trabajo y que los objetivos y metas principales, una vez definidos, no deberían modificarse a menos que fuese absolutamente necesario.
- las malas noticias no deberían guardarse en secreto, sino entregarse al equipo tan pronto como fuese posible
- Cuanto más control se le da al equipo para tomar decisiones técnicas y del proceso, menos frustración sentirán los miembros del equipo
- Una elección inapropiada del proceso del software puede ser evitada de dos formas:
 1. estando seguros de que las características del software a construir se ajustan al rigor del proceso elegido
 2. permitiendo al equipo seleccionar el proceso (con el reconocimiento completo de que, una vez elegido, el equipo tiene la responsabilidad de entregar un producto de alta calidad).

- refinar claramente los roles y las responsabilidades antes del comienzo del proyecto
- para eliminar una atmósfera de fallos será establecer técnicas basadas en el equipo para retroalimentar y solucionar el problema.
- cualquier fallo de un miembro del equipo debe ser considerado como un fallo del equipo. Esto lleva a un acercamiento del equipo a la acción correctiva, en lugar de culpar y desconfiar, que ocurre con rapidez en equipos tóxicos.
- Reconocer las diferencias entre los diferentes miembros del equipo para utilizarlas en favor del mismo, ya que hay gente mas introvertida, extrovertida, que entrega las cosas antes de tiempo o sobre el plazo, otros mas lógicos, etc.

15. Grafique valor y empleo de técnicas de coordinación y comunicación.



16. Enuncie las categorías de técnicas de coordinación de proyectos propuestas por Kraul y Streeter.

- **Formal, enfoque impersonal:** Incluyen documentos de ingeniería del software y entregas, memorandos técnicos, hitos del proyecto, planificaciones del programa y herramientas de control del proyecto, peticiones de cambios y documentación relativa informes de seguimiento de errores e información almacenada
- **Formal, procedimientos interpersonales:** Se centra en las actividades de garantía de calidad (Capítulo 8) aplicada a productos de ingeniería del software. Estos incluyen reuniones de revisión de estado e inspecciones de diseño y de código
- **Informal, procedimientos interpersonales:** Incluyen reuniones de para la divulgación de información y resolución de problemas, así como «definición de requisitos y del personal de desarrollo».
- **Comunicación electrónica:** Comprende correo electrónico, boletines de noticias electrónicos y, por extensión, sistemas de videoconferencia.
- **Red interpersonal:** Discusiones informales con los miembros del equipo y con personas que no están en el proyecto pero que pueden tener experiencia o una profunda visión que puede ayudar a los miembros del equipo.

17. Enuncie las cuestiones que definen el ámbito del software.

El ámbito se define respondiendo a las siguientes cuestiones:

Contexto: ¿Como encaja el software a construir en un sistema, producto o contexto de negocios mayor y qué limitaciones se imponen como resultado del contexto?

Objetivos de información. ¿Qué objetos de datos visibles al cliente se obtienen del software? ¿qué objetos de datos son requeridos de entrada?

Función y rendimiento. ¿Qué función realiza el software para transformar la información de entrada en una salida? ¿Hay características de rendimiento especiales que abordar?

Es decir, datos cuantitativos se establecen explícitamente, se anotan las limitaciones y se describen los factores de reducción de riesgos

18. Enuncie modelos de proceso.

- El modelo de prototipo
- El modelo DRA
- El modelo incremental
- El modelo en espiral
- El modelo en espiral winwin
- El modelo de desarrollo basado (ensamblaje) en componentes
- El modelo de desarrollo concurrente
- El modelo de métodos formales
- El modelo de técnicas de cuarta generación

El gestor del proyecto debe decidir qué modelo de proceso es el más adecuado para

1. Los clientes que han solicitado el producto y la gente que realizará el trabajo;
2. Las características del producto en sí, y
3. El entorno del proyecto en el que trabaja el equipo de software.

19. Enuncie actividades estructurales de la organización de producción de software.

- Comunicación con el cliente-tareas requeridas para establecer la obtención de requisitos eficiente entre el desarrollador y el cliente.
- Planificación-tareas requeridas para definir los recursos, la planificación temporal del proyecto y cualquier información relativa a él.
- Análisis del riesgo-tareas requeridas para valorar los riesgos técnicos y de gestión tareas requeridas para construir una o más representaciones de la aplicación.
- Construcción y entrega-tareas requeridas para construir, probar, instalar y proporcionar asistencia al usuario (por ejemplo: documentación y formación).
- Evaluación del cliente-tareas requeridas para obtener información de la opinión del cliente basadas en la evaluación de las representaciones de software creadas durante la fase de ingeniería e implementación durante la fase de instalación.

20. Defina señales que indican que un proyecto de sistemas de información está en peligro según Reel.

1. La gente del software no comprende las necesidades de los clientes.
2. El ámbito del producto está definido pobremente.
3. Los cambios están mal realizados.
4. La tecnología elegida cambia.
5. Las necesidades del negocio cambian [o están mal definidas].
6. Las fechas de entrega no son realistas.
7. Los usuarios se resisten.
8. Se pierden los patrocinadores [o nunca se obtuvieron adecuadamente].
9. El equipo del proyecto carece del personal con las habilidades apropiadas.
10. Los gestores [y los desarrolladores] evitan buenas prácticas y sabias lecciones.

21. Enuncie las partes que Reel sugiere como aproximación a proyectos de software.

- **Empezar con el pie derecho** para comprender el problema a solucionar y estableciendo entonces objetivos expectativas realistas para cualquiera que vaya a estar involucrado en el proyecto. **Se refuerza construyendo el equipo adecuado y dando al equipo la autonomía, autoridad y tecnología necesaria para realizar el trabajo**
- **Mantenerse.** el gestor del proyecto debe proporcionar incentivos para conseguir una rotación del personal mínima, el equipo debería destacar la calidad en todas las tareas que desarrolle y los gestores veteranos deberían hacer todo lo posible por permanecer fuera de la forma de trabajo del equipo
- **Seguimiento del Progreso.** Para un proyecto de software, el progreso se sigue mientras se realizan los productos del trabajo y se aprueban como parte de una actividad de garantía de calidad. Además, el proceso del software y las medidas del proyecto pueden ser reunidas y utilizadas para evaluar el progreso frente a promedios desarrollados por la organización de desarrollo de software.
- **Realizar un Análisis al finalizar el proyecto.** Establecer un mecanismo consistente para extraer sabias lecciones de cada proyecto. Evaluar la planificación real y la prevista, reunir y analizar métricas del proyecto de software y realimentar con datos de los miembros del equipo y de los clientes, y guardar los datos obtenidos en formato escrito.
- **Tomar decisiones inteligentes.** En esencia, las decisiones del gestor del proyecto y del equipo de software se simplifica siendo siempre que sea posible, utilice software del mismo comercial o componentes de software existentes; evite personalizar interfaces cuando estén disponibles aproximaciones estándar; identifique y elimine entonces

riesgos obvios; asigne más tiempo del que pensaba necesitar para tareas arriesgadas complejas,

22. Enuncie las preguntas propuestas por Bohem que conducen a la definición de las características clave del proyecto y el plan del proyecto resultante.

El principio WWWWWHH conduce a la definición de las características clave del proyecto y el plan del proyecto resultante:

- **¿Por qué se desarrolla el sistema?** Dicho de otra forma, ¿justifica el propósito del negocio el gasto en personal, tiempo y dinero?
- **¿Qué se realizará y cuándo?** La respuesta a estas preguntas ayuda al equipo a establecer la planificación del proyecto identificando las tareas clave del proyecto y los hitos requeridos por el cliente.
- **¿Quién es el responsable de una función?**
- **¿Dónde están situados organizacionalmente?** No todos los roles y responsabilidades residen en el equipo de software. El cliente, los usuarios, y otros directivos también tienen responsabilidades.
- **¿Cómo estará realizado el trabajo desde el punto de vista técnico y de gestión?** Una vez establecido el ámbito del producto, se debe definir una estrategia técnica y de gestión para el proyecto.
- **¿Qué cantidad de cada recurso se necesita?** La respuesta a esta pregunta se deriva de las estimaciones realizadas basadas en respuestas a las preguntas anteriores.

23. Enuncie las preguntas que el Concilio Airlie ha desarrollado para tener una “Visión Rápida” sobre si un proyecto ha implementado prácticas críticas

Estas prácticas son «utilizadas de un modo consistente por, y consideradas críticas por, organizaciones y proyectos de software de mucho éxito cuyo rendimiento “final” es más consistente que los promedios de la industria»

- **Gestión formal del riesgo** ¿Cuáles son los diez riesgos principales para este proyecto? Para cada uno de los riesgos ¿cuál es la oportunidad de que el riesgo se convierta en un problema y cuál es el impacto si lo hace?
- **Coste empírico y estimación de la planificación** ¿Cuál es el tamaño actual estimado de la aplicación de software (sin incluir el software del sistema) que será entregada en la operación? ¿Cómo se obtuvo?
- **Gestión de proyectos basada en métricas** ¿Dispone de un programa de métricas para dar una primera indicación de los problemas del desarrollo? Si es así, ¿cuál es la volatilidad de los requisitos actualmente?
- **Seguimiento del valor ganado** ¿Informa mensualmente de las métricas del valor ganado? Si es así, ¿están calculadas estas métricas desde una red de actividades de tareas para el esfuerzo total a la próxima entrega?
- **Seguimiento de defectos frente a objetivos de calidad** ¿Realiza el seguimiento e informa periódicamente del número de defectos encontrados en cada prueba de inspección [revisión técnica formal] y ejecución desde el principio del programa y del número de defectos que se corrigen y se producen en la actualidad?
- **Gestión del programa de la persona** ¿Cuál es la media de rotación de la plantilla en los tres últimos meses por cada uno de los distribuidores/desarrolladores involucrados en el desarrollo del software para este sistema?