Metodología de Sistemas I

Tecnicatura Universitaria en Programación

Profesor: Leandro Schmidt
Ayudante: Benjamin Wagner

Unidad 11: Administración de Proyectos de Software

La **administración del proyecto** involucra planificación, monitoreo y control del personal, procesos y acciones que ocurren, a medida que el software evoluciona desde un concepto preliminar hasta su despliegue operativo completo.

El **proyecto** debe planificarse, estimándose el esfuerzo y el cronograma necesarios para concluir las tareas.

La **comunicación** con el cliente y con otros participantes debe ocurrir de modo que los requerimientos sean comprensibles.

Unidad 11: Administración de Proyectos de Software

La administración efectiva de un proyecto de software se enfoca en las “cuatro P”:

- **Personal:** se requiere habilidad para atraer, desarrollar, motivar, organizar y conservar la fuerza de trabajo. Es necesario poner atención en áreas como:
  - Comunicación y coordinación
  - Ambiente de trabajo
  - Capacitación
  - Compensación
  - Análisis y desarrollo de competencias
  - Desarrollo profesional
  - Desarrollo de grupos de trabajo
  - Desarrollo de equipo/cultura

Unidad 11: Administración de Proyectos de Software

- **Producto:** antes de poder planear un proyecto, deben establecerse los objetivos y el ámbito del producto, considerar soluciones alternativas e identificar las restricciones técnicas y administrativas. A partir de allí logramos:
  - Estimaciones razonables (y precisas) del costo
  - Una valoración efectiva del riesgo
  - Una descomposición realista de las tareas del proyecto
  - Un calendario de proyecto manejable

Primero se deben definir los objetivos y el ámbito del producto. Los objetivos identifican las metas para el producto sin considerar cómo se lograrán estas metas (“QUE”, no “COMO”). Recién luego se plantean soluciones alternativas, teniendo en cuenta las restricciones de contexto (tiempo, presupuesto, etc).

Unidad 11: Administración de Proyectos de Software

- **Proceso:** un proceso de software proporciona el marco conceptual desde el cual puede establecerse un plan completo para el desarrollo de software. Ese plan incluye:
  - Un pequeño número de **actividades de marco** conceptual se aplica a todos los proyectos de software, sin importar su tamaño o complejidad.
  - Diferentes **tareas** (tareas, hitos, productos operativos y puntos de aseguramiento de calidad) permiten que las actividades del marco conceptual se adapten a las características del proyecto de software y a los requerimientos del equipo del proyecto.
  - Las **actividades sombrilla** (como QA, la administración de configuración del software y las mediciones) recubren el modelo de proceso. Las actividades sombrilla son independientes de cualquier actividad del marco conceptual y ocurren a lo largo del proceso.

Unidad 11: Administración de Proyectos de Software

- **Proyecto:** Los proyectos de software se planean y controlan, única forma de manejar su complejidad.

  La tasa de falla de los proyectos sigue siendo alta, superior al 50%, con problemas de cumplimiendo tanto de presupuesto como de cronograma.

  Para evitar el fracaso del proyecto, un gerente de proyecto y los ingenieros de software que construyan el producto deben:

  - Evitar un conjunto de señales de advertencia comunes (riesgos)
  - Entender los factores de éxito cruciales que conducen a una buena administración del proyecto
  - Usar el sentido común para planificar, monitorear y controlar el proyecto

Unidad 11: Administración de Proyectos de Software

Los principales participantes de un proyecto de software son:

- **Usuarios finales:** las personas que finalmente van a usar el software
- **Clientes:** que especifican los requerimientos para el software que se va a fabricar, y participantes secundarios
- **Profesionales:** que aportan las habilidades técnicas que se necesitan para crear un producto o aplicación.
- **Gerentes de proyecto (técnicos):** quienes deben planificar, motivar, organizar y controlar a los profesionales que hacen el trabajo de software.
- **Gerentes ejecutivos**: definen los temas empresariales que con frecuencia tienen una influencia significativa sobre el proyecto

Unidad 11: Administración de Proyectos de Software

Los **líderes de equipo** idealmente debieran tener una mezcla de habilidades interpersonales, además de sus competencias técnicas. De ellos se espera:

- **Motivación**: debe alentar siempre al equipo y festejar sus logros.
- **Organización**: moldear los procesos existentes (o inventar nuevos)
- **Ideas e innovación:** alentar a las personas a crear y sentirse creativas
- **Resolución de problemas:** diagnosticar los conflictos técnicos, pensar o ayudar a pensar una solución, y aplicar lecciones aprendidas
- **Buena administración:** asumir el control cuando sea necesario
- **Logros:** recompensar la iniciativa y el logro
- **Influencia y construcción de equipo:** comprender las señales y reaccionar ante necesidades de las personas

Unidad 11: Administración de Proyectos de Software

Para lograr un **equipo de alto rendimiento**, debemos asegurar que:

- Los miembros del equipo se tienen confianza entre sí.
- La distribución de habilidades debe ser adecuada para el problema
- Si no se pueden solucionar disconformidades de algún miembro del equipo, es posible que tenga que excluirlo a fin de mantener la cohesión del equipo.
- El gerente debe ayudar a crear cohesión de equipo
- No caigamos en alguno de estos escenarios:
  - Una atmósfera de trabajo frenético
  - Alta frustración que causa fricción entre los miembros del equipo
  - Un proceso de software fragmentado o pobremente coordinado
  - Una definición poco clara de los roles en el equipo de software
  - Continua y repetida exposición al fracaso.

Unidad 11: Administración de Proyectos de Software

Un **equipo ágil** es el antídoto a muchos de los problemas del trabajo en un proyecto de software. Se alienta la satisfacción del cliente y la entrega incremental temprana del software, pequeños equipos de proyecto enormemente motivados, métodos informales, mínimos productos operativos y simplicidad de desarrollo global.

Estos equipos son normalmente autogestionados, con autonomía para tomar las decisiones administrativas y técnicas del proyecto necesarias.

Para lograr esto, un equipo ágil puede realizar reuniones grupales diarias para coordinar y sincronizar el trabajo que debe realizarse en ese día.

Unidad 11: El Producto

Al iniciar un proyecto nos piden estimaciones precisas y un plan completo. Pero no hay información sólida de que valerse y requerimientos cambiantes.

¿Qué hacemos?

Una salida es examinar el **producto**: establecer y acotar el ámbito del mismo.

- Determinar el ámbito del software: contexto, objetivo, principales funciones
  - Objetos de entrada y de salida
  - Transformaciones necesarias
  - Usuarios simultáneos
  - Tiempo de respuesta máximo
- Descomponer el problema (divide y reinarás): las funciones del software se evalúan y refinan para tener más detalle **antes** de comenzar la estimación

Unidad 11: El Proceso

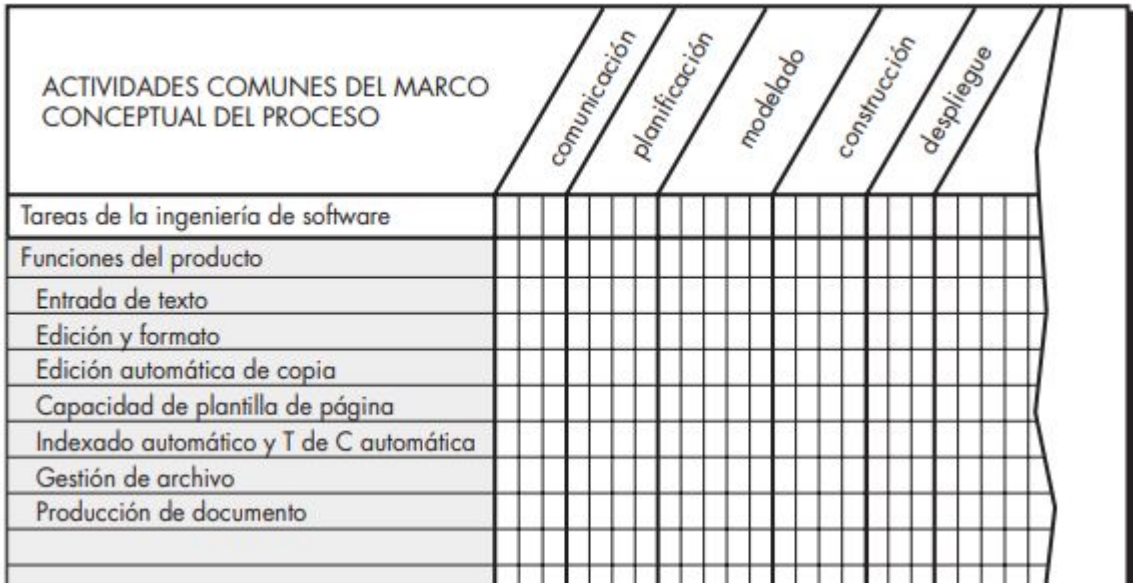
Las actividades del marco conceptual son aplicables a todos los proyectos de software. El problema es seleccionar el **modelo de proceso** que sea adecuado para el software que el equipo del proyecto intenta construir:

- Para los clientes que solicitaron el producto
- Para el personal que hará el trabajo
- Para las características del producto en sí
- Para el entorno de proyecto donde trabaja el equipo de software

La planificación del proyecto incluye la fusión de producto y proceso. Cada función debe pasar a través del conjunto de actividades de marco conceptual. Ejemplo: **comunicación, planificación, modelado, construcción y despliegue**.

Unidad 11: Fusión de Producto y Proceso

El gerente de proyecto debe estimar los requerimientos de recursos para cada celda de la matriz, fechas de inicio y término de las tareas asociadas con cada celda, y los productos operativos que se van a producir.



Unidad 11: El Proyecto

Para administrar un **proyecto** de software exitoso, se debe comprender **qué puede salir mal**, de modo que los problemas pueden evitarse. Esto es lo que se conoce como *mitigación de riesgos.*

- Top 10 de riesgos típicos en la mayoría de los proyectos:
  - El equipo no entiende las necesidades del cliente.
  - El ámbito del producto está pobremente definido
  - Los cambios se gestionan pobremente
  - Cambia la tecnología elegida.
  - Las necesidades de negocio cambian
  - Las fechas límite son irreales.
  - Los usuarios son resistentes.
  - Pérdida de patrocinio
  - El equipo del proyecto carece de personal con skills adecuadas.
  - Los gerentes y el equipo no tienen buenas prácticas
  - No se tienen en cuenta las lecciones aprendidas.

Unidad 11: El Proyecto - 5 enfoques de sentido común

- **Comenzar con el pie derecho**
  - Trabajar duro para entender el problema que debe resolverse
  - Establecer objetivos y expectativas realistas
  - Construir el equipo correcto y darle autonomía, autoridad y herramientas.
- **Mantener la cantidad de movimiento**
  - Proporcionar incentivos para mantener la rotación de personal en un mínimo
  - El equipo debe enfatizar la calidad en cada tarea que realice
  - Mantener al gerente ejecutivo lo más lejos posible del equipo
- **Siga la pista al progreso:** tener métricas confiables y valorar el progreso
- **Tome decisiones inteligentes**
  - Simplificar cada vez que se pueda
  - Utilizar componentes e interfaces estándar
  - Asignar más tiempo de lo estimado a tareas complejas o riesgosas.
- **Realice un análisis postmortem:** recopilar feedback y documentar lecciones aprendidas

Unidad 11: El Proyecto y el principio W5-HH

Siete preguntas que conducen a una definición de las características clave del proyecto y al plan de proyecto resultante:

- ¿Por qué (**why**) se desarrollará el sistema?
- ¿Qué (**what**) se hará?
- ¿Cuándo (**when**) se hará?
- ¿Quién (**who**) es responsable de cada función?
- ¿Dónde (**where**) se ubicarán en la organización?
- ¿Cómo (**how**) se hará el trabajo, técnica y organizativamente?
- ¿Cuánto (**how much**) se necesita de cada recurso?

Unidad 11: El Proyecto

Prácticas cruciales:

- Administración del proyecto basada en métrica
- Estimación empírica de costo y calendario
- Rastreo del valor ganado
- Rastreo de defecto contra metas de calidad
- Administración consciente del personal
- Gestión eficiente de la matriz de interesados
- Administración efectiva de los riesgos

Unidad 11: Metricas de proceso y de proyecto

La **medición cuantitativa** permite comprender mejor el proceso y el proyecto, al darnos un mecanismo de evaluación **objetiva.** Si no se mide, el juicio puede basarse solamente en la evaluación **subjetiva**

- **En el proceso:** se mide pensando en mejoras continuas
- **En el proyecto:** ayuda en la estimación, control de calidad, valoración de productividad y control de proyecto.
- **En el producto:** ayuda a valorar la calidad del mismo

Se debe definir un conjunto limitado de medidas de proceso, proyecto y producto, que son fáciles de recopilar. Se las normalizan usando métricas de tamaño o de función y el resultado se compara con promedios anteriores para proyectos similares. Vemos tendencias y sacamos conclusiones.

Unidad 11: Metricas de proceso

Las métricas de proceso de software aportan beneficios significativos si se trabaja para mejorar su nivel global de madurez de proceso. Sin embargo, éstas pueden tener mal uso, lo que crea más problemas de los que resuelven.

- Usar el sentido común y sensibilidad organizacional al interpretar datos de métricas.
- Dar feedback regular a los individuos y equipos que recopilan medidas y métricas.
- No usar métricas para valorar a los individuos.
- Consensuar metas y métricas claras que se usarán para lograr las primeras.
- Nunca usar métricas para amenazar a los individuos o a los equipos
- No es “negativo” que una métrica muestre un problema. Es la oportunidad de mejorar.
- No obsesionarse con una sola métrica ni excluir otras importantes

Unidad 11: Métricas de proyecto

Las métricas de proyecto tienen un uso más táctico: permiten adaptar el flujo de trabajo del proyecto y las actividades técnicas. Son útiles para adaptar el flujo de trabajo del proyecto y las actividades técnicas.

- En la etapa de **estimación**, miramos métricas de proyectos anteriores
- Durante la **ejecución** tenemos métricas para comparar *actual vs forecast*, ayudando a monitorear el avance del proyecto

La intención de las métricas de proyecto es doble:

- Se usan para gestionar el calendario y ajustarlo, evitar demoras y mitigar potenciales problemas y riesgos.
- Se usan para valorar la calidad del producto y modificar el enfoque técnico para mejorar la calidad.

Unidad 11: Métricas de calidad del Software

Un gerente de proyecto también debe evaluar la calidad conforme avanza el proyecto. Algunas mediciones de calidad son:

- **Exactitud**: grado en el cual el software realiza la función requerida, y se define en función de los errores encontrados en producción, en un periodo dado.
- **Capacidad de mantenimiento:** facilidad con la que un programa puede corregirse si se encuentra un error.
- **Integridad:** mide la habilidad del sistema para resistir ataques a su seguridad. Se basa en el concepto de amenaza (probabilidad) y la seguridad (capacidad para repeler el ataque)
- **Usabilidad:** es un intento por cuantificar la facilidad de uso, usualmente ayudado con encuestas a usuarios finales

Unidad 11: Métricas de calidad

Buenas métricas de calidad para código en desarrollo:

- Porcentaje de código que ha sido revisado (idealmente 100%).
- Porcentaje del código que tiene cobertura de tests. No menor a 70%
- (Bugs de Desarrollo / Bugs Totales) × 100
- (Bugs corregidos / Bugs Totales) × 100
- (Bugs sin solución o Next version / Bugs Totales) × 100
- Nro. de errores aceptados (sin solución)
- Nro. De Errores Stoppers identificados.

Una guia mas completa de métricas de calidad de SW

Unidad 11: Lecturas Recomendadas

Ingeniería del Software: Un Enfoque Práctico”. 7° edición. PRESSMAN, Roger S.

- Capítulo 24: CONCEPTOS DE ADMINISTRACIÓN DE PROYECTO
- Capítulo 25: MÉTRICAS DE PROCESO Y DE PROYECTO