

¿Qué es el alcance del proyecto?

Sitio: Agencia de Habilidades para el Futuro

Curso: Práctica Profesional 1: Aproximación al mundo laboral 1° D

Libro: ¿Qué es el alcance del proyecto?

Imprimido por: Eduardo Moreno

Día: lunes, 19 de mayo de 2025, 01:16

Tabla de contenidos

1. ¿Qué es el alcance del proyecto?
2. ¿Por qué definir los requerimientos?
3. Problemas de Implementación
4. Clasificación de los requerimientos

1. ¿Qué es el alcance del proyecto?



¿Qué es el alcance del proyecto?

El alcance de un proyecto representa el objetivo final a donde se espera llegar. Dicho de otra forma, describe una meta por alcanzar e indica hasta dónde va el proyecto.

Aunque en la práctica el alcance del proyecto se expresa en un corto párrafo, detrás de éste está todo un análisis respecto a:

- las necesidades del cliente,
- el trabajo que se requerirá proyectar (objetivos),
- el contexto dentro del cual se desarrolla el proyecto,
- los riesgos a los cuales puede tener que hacer frente,
- las recursos con los que se cuenta (capacidad de ejecución),
- los costos en los que se incurrirá,
- el tiempo o plazo establecido, etc.

Requerimientos de un proyecto de software

- Existen diferencias entre Ingeniería de Requerimientos de Sistema y la Ingeniería de Requerimientos de Software.
- En general, los requerimientos de sistema se originan en las necesidades del usuario, mientras que la gestión de requerimientos de un proyecto de software se origina en los requerimientos y/o especificaciones del sistema.
- El cliente = experto en los procesos de su negocio, conoce sus necesidades de información.
- Proveedor o desarrollador= experto en los temas de características de los sistemas y especificaciones del software.

Factores más comunes que llevan al fracaso:

Falta de
participación
del usuario

Requerimientos
incompletos

Mal manejo del
cambio a los
requerimientos.

2. ¿Por qué definir los requerimientos?



¿Por qué definir los requerimientos?

A continuación te brindamos 7 motivos por los cuales es importante definir requerimientos.

1. Las fallas de un sistema más comunes ocurren por una mala definición de los requerimientos.
2. Lograr identificar especificaciones de software buenas, correctas, completas y medibles es crucial en el desarrollo de software.
3. No importa lo bien diseñado o codificado que esté un programa si no se ha analizado correctamente, defraudará al usuario y frustrará al desarrollador.
4. Cambios en los requisitos.
5. Suelen ocurrir.
6. No siempre se comunican a todos los grupos involucrados.
7. El impacto del cambio varía según el momento en que se introduzca:
 - Al principio del proyecto, pueden acomodarse fácilmente.
 - Al final de un proyecto, suelen necesitar modificaciones complejas y de mayor magnitud.

3. Problemas de Implementación



Problemas de Implementación

Los usuarios/as, por lo general, “saben lo que necesitan”, pero en muchas ocasiones “no saben cómo explicarlo”, mucho menos “cómo documentarlo”. Por lo tanto, **el usuario debe documentar el proceso, al menos, a nivel manual, como base para la documentación del requerimiento**. A pesar de que la escasa o nula documentación de la gestión de requerimientos suele ser el principal problema de la implementación, es algo que ni el cliente ni el proveedor del software se resisten a detallar adecuadamente. Por un lado, **el cliente en ocasiones no exige esto para ahorrar tiempo** o porque cree que sobre la marcha se harán los ajustes necesarios. Por otra parte, el **proveedor, si bien debería exigir este análisis exhaustivo, considera que eso elevaría los costos del proyecto** y en consecuencia, no se concluiría la venta. Se debe buscar un punto intermedio donde por lo menos se asegure el alcance.

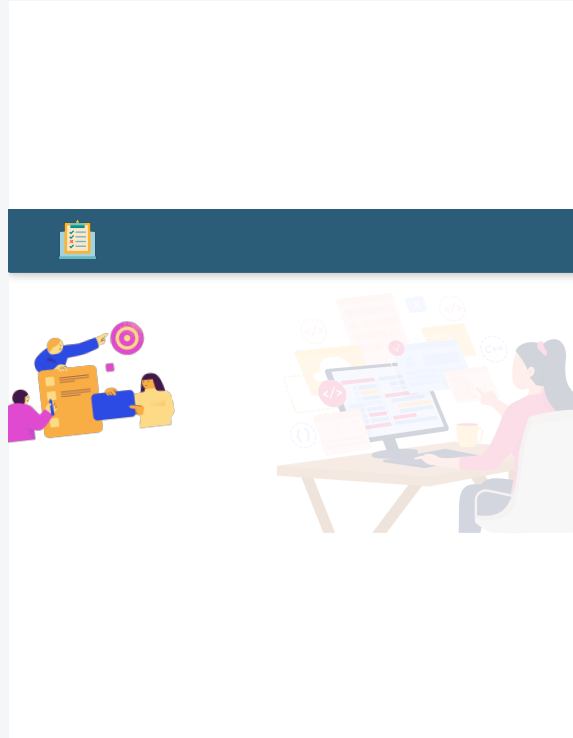
Problemática

- **Retraso de los proyectos:** suele ocurrir porque algunos aspectos están fuera del alcance original, el cual nunca fue documentado, por lo que se debe implementar un mecanismo formal para justificarlo.
- **Ajuste al alcance del proyecto:** lo más conveniente es que se maneje con un documento llamado “control de cambios” de manera que se documente el tiempo que pudiera retrasar el proyecto y también si va a tener algún costo.
- **Número de requerimientos:** es proporcional a la dimensión del proyecto. Si es muy grande, puede ser difícil controlarlos, sobre todo si no se tienen herramientas y mecanismos establecidos, considerando además que se relacionan unos con otros y se relacionan con otros artefactos (planes, código, pruebas) del proceso de ingeniería de software.
- **Cambios:** es muy caro hacer cambios a requerimientos después de que han sido acordados, ya que conllevan a costos relacionados con la especificación, el re-diseño, la re-codificación, el re-probar, la reimplantación, las acciones correctivas, los costos de desplazamiento, la documentación y el mantenimiento constante. Lo más conveniente es tener una junta o reunión de evaluación para determinar si los cambios que se requieren pueden esperar para una segunda fase o son necesarios para la puesta en marcha.

4. Clasificación de los requerimientos



Clasificación de los requerimientos



Más info sobre requisitos funcionales RF:

- Son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinadas entradas. No poner requisitos ambiguos librados a la interpretación del desarrollador. Debe ser completa y coherente.
- Cuando hablamos de las entradas, no necesariamente hablamos sólo de las entradas de los usuarios. Pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos.
- En algunos casos, los requisitos funcionales de los sistemas también establecen explícitamente lo que el sistema no debe hacer. Es importante recordar esto: un RF puede ser también una declaración negativa. Siempre y cuando el resultado de su comportamiento sea una respuesta funcional al usuario o a otro sistema, es correcto y es necesario definirlo.
- Requisitos no funcionales RNF
- Origen: necesidad del usuario, restricciones presupuestarias, políticas organizacionales, necesidad de interoperabilidad con otros sistemas de software o hardware, factores externos (regulaciones de seguridad, políticas de privacidad, por ej.)
- Clasificación según sus implicaciones:
- Requisitos de producto: especificaciones de comportamiento (rendimiento, memoria, fiabilidad, portabilidad, usabilidad)
- Requisitos organizativos (organización del cliente y el desarrollador: estándares, lenguaje, diseño, datos de entrega, documentación)
- Necesidades externas (interoperabilidad con otros sistemas, requisitos legales y éticos que deberá aceptar el usuario)

