**Técnicas de análisis de requisitos**

En el documento de especificación de requisitos se describe lo que sistemas debe hacer, pero no el cómo.

Existen 5 criterios principales que se tienen en cuenta para priorizar los requerimientos como son:

1. como el valor que los usuarios le dan a la visión del producto,
2. la urgencia,
3. las limitaciones de tiempo.
4. la técnica de complejidad
5. las preferencias de las partes interesadas.

Además, muy a menudo los proyectos deben ser priorizados adecuadamente tanto para los **objetivos principales** del proyecto como para las **tareas específicas** que lograrán los objetivos. Por lo tanto, nos ocupamos de la priorización en dos niveles: **nivel de producto** y **nivel de tarea**. Especialmente teniendo en cuenta que los clientes normalmente tienden a no entender que no pueden obtener todas las características que desean en la versión 1.0 de un nuevo producto de software.

**Analiza las áreas clave que se tienen en cuenta antes de tomar una decisión importante**

**Beneficio:** una ventaja que obtiene la empresa como resultado de la implementación requerida.  
**Penalización**: una consecuencia de no implementar un requisito.  
**Costes – esfuerzo** y recursos que se requieren para implementar un requisito.  
**Riesgo:** una probabilidad de que el requisito no entregue el valor esperado.  
**Dependencias:** una relación entre los requisitos, por ejemplo, cuando el requisito requiera completar otro requisito para su implementación.  
**Sensibilidad de tiempo:** fecha de caducidad, urgencia.  
**Estabilidad:** la probabilidad de que el requisito permanezca estático.  
**Cumplimiento de políticas:** requisitos que deben implementarse para cumplir con los requisitos reglamentarios.

Técnicas para priorizar requisitos:

1. Clasificación de lista:

Cada requerimiento se prioriza con relación al resto de requerimiento, asignándole a cada requerimiento un valor único numérico del 1 hasta el ultimo requerimiento. De tal forma que no habrán 2 requerimiento con el mismo número de importancia.

1. **Mapa de dependencia**

Siempre es una buena idea crear un mapa de dependencia para comprender mejor las dependencias entre los requisitos. Especialmente teniendo en cuenta que la mayoría de los requisitos son interdependientes y difícilmente encontrarás un requisito que sea independientemente.

Entender los requisitos de dependencia es tan importante como la priorización. Sin entender la dependencia de los requisitos, es muy poco probable que llegue al orden correcto de la implementación de los requisitos. Por lo tanto, es una buena idea tener el mapa de dependencia de requisitos en tu lugar antes de priorizar los requisitos.

Por ejemplo tenemos 2 requerimiento con igual importancia, por ejemplo ambos tiene una importancia de 3:

Por ejemplo en un programa de inventario, tenemos 2 requerimientos que son:

Listado de stock mínimo,

Entrada de datos por teclado,

Ambos tienen la misma importancia 3, el cliente nos dice que sin el listado de stock minino, el no podría tomar decisiones a tiempo.

Y el otro es una entrada de datos por teclado:

Así que el listado depende de la información digitada, por lo tanto primero se debe hacer la entrada de datos y después el listado de stock.

1. **MoSCoW**

El método MoSCoW funciona mejor que el sistema de calificación numérica, ya que es mucho más fácil para las partes interesadas calificar los requisitos como Must, Should, Could or Would

El acrónimo representa lo siguiente:

Must – obligatorio  
Should – de alta prioridad  
Could – Preferido pero no necesario  
Would – puede ser pospuesto y sugerido para futura ejecución



Estimación de punto de historia.

para estimar hay que tener en cuenta el grado de dificultad,

se escoge un pivote, que es una historia de usuario que todos

la saben realizar.

y se le estima un tiempo, que vamos a utilizar como patrón de comparación.

después comparamos este patrón con cada una de las

demás historias, por ejemplo si es el doble de dificulta o

la mitad de dificultad comparándolo con el patrón (pivote)

scrum póker

Bibliografia: tomado de internet <https://apiumhub.com/es/tech-blog-barcelona/priorizacion-requerimientos-software/>