

# Tema 4

# Extracción de Requisitos

Hipólito Guzmán Miranda  
Profesor Contratado Doctor  
Universidad de Sevilla

## Índice

- ¿Por qué escribir requisitos?
- Características de un buen requisito
- Clasificación
- Restricciones
- Niveles de importancia
- ¿Por dónde empezar?
- Verificación
- Matriz de verificación
- Plan de pruebas

## ¿Por qué escribir requisitos?

- Para que los proyectos no se eternicen
- Para no comprometernos a desarrollos imposibles
- Para traducir los objetivos del proyecto a algo *implementable y medible*
- Porque lo que no se deja por escrito no se puede revisar

## ¿Por qué escribir requisitos?

Todos los requisitos tienen que ser verificados más adelante

- requisitos deben ser medibles / verificables

Sabemos que el proyecto ha terminado porque:

- Se cumplen todos los requisitos
- No estamos encontrando fallos (bugs)

## **Características de un buen requisito:**

- Unitario
- Completo
- Consistente
- Atómico
- Trazable
- Actual
- No ambiguo
- Con importancia especificada
- Verificable

## Características de un buen requisito:

- **Unitario** (sólo refiere a una cosa)
- **Completo** (no falta información)
- **Consistente** (con el resto de la doc, sin contradicciones)
- **Atómico** (no es “1: A y B”, sino “1: A” y “2: B”)
- **Trazable** (se sabe de dónde viene y por qué es un requisito)
- **Actual** (si se vuelve obsoleto: modificar o eliminar)
- **No ambiguo** (no sujeto a interpretación)
- **Con importancia especificada** (¿es opcional u obligatorio?)
- **Verificable** (se puede comprobar)

## Classification

- Functional Req.
- Performance Req.
- Design Req.
- Operational Req.
- Electrical Req.
- Software Req.
- Mechanical Req.
- Security Req.
- Manufacturing Req.
- Test Req.
- Interface Req.
- Quality\* Req.
- Modularity Req.
- Environmental Req.

\*Also may be called: "Product Assurance Requirements"

Acknowledge: European Space Agency for the Requirements classification

## Clasificación

- Funcionales
- de Prestaciones
- de Diseño
- de Operación
- Eléctricos
- de Software
- Mecánicos
- de Seguridad
- de Fabricación
- de Test
- de Interfaz
- de Calidad
- de Modularidad
- de Entorno

## Clasificación

- **Funcionales:** describen la funcionalidad
- **Prestaciones:** describen las prestaciones de la funcionalidad anterior
- **Diseño:** imposiciones externas
- **Operación:** si el sistema es telecontrolado

## Ejemplos:

- **Funcionales**
  - F.1: El sistema será capaz de capturar imágenes
  - F.2: El sistema será capaz de reconocer señales de tráfico en ellas
- **Prestaciones**
  - P.1.1: El sistema capturará 30 imágenes / s
- **Diseño**
  - D.1: El sistema debe tener unas dimensiones menores de 60x40x20 mm
- **Operación**
  - O.1: El sistema debe poderse encender y apagar de manera remota

## Restricciones

Además de requisitos, es normal tener restricciones (en inglés, *Constraints*):

- C.1: El presupuesto máximo del proyecto será de 80€
- C.2: El proyecto debe utilizar al menos una tarjeta Raspberry Pi
- C.3: El material X, Y, sólo estará disponible para el equipo durante 12 horas presenciales

## Design Requirements vs Constraints

- **Design requirements:** aspectos de diseño que el proyecto debe cumplir para cumplir con los objetivos:
  - El peso de la prótesis debe ser menor de 2150 gr.
  - El diseño no debe causar interferencia con otros experimentos
  - El prototipo debe cumplir con la normativa XYZ
- **Constraints:** restricciones externas
  - El laboratorio donde hacer el test A sólo estará disponible en Abril
  - Uno de los miembros del equipo no estará disponible durante las fechas Y
  - Debemos usar el dispositivo/componente Z ya que el objetivo del proyecto es probarlo

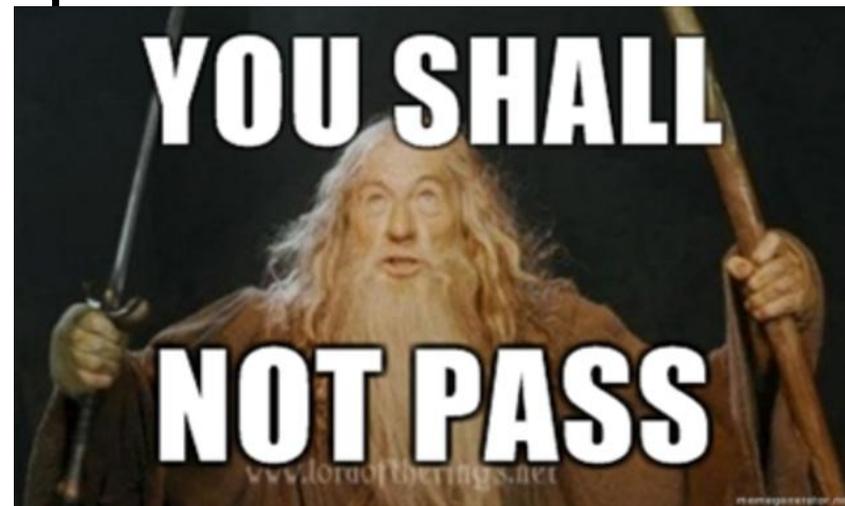
## Niveles de importancia

“Shall, should, may”. “Debe, debería, puede”:

- Shall (o Must) / tiene que (debe): obligatorio cumplimiento, es el mínimo
- Should / debería: mejora sobre el mínimo
- May / podría: a ser posible, estaría bien tenerlo

## Niveles de importancia

- Shall / Must: required
- Shall not / Must not: prohibited
- Should: recommended
- Should not: not recommended
- May: optional



## Niveles de importancia

- Obligatorio / Requerido / debe
- Opcional / Recomendado / debería
- Wishlist / puede

Sean cuales sean los términos, deben ser utilizados de manera congruente en el documento

Es buena idea que estén definidos al principio del documento

## ¿Por dónde empezar?

- Listamos las funcionalidades y las separamos
  - Obtenemos una primera versión de los requisitos funcionales
- Escribimos los requisitos de diseño
- Escribimos las restricciones (constraints)

## ¿Por dónde seguimos?

- Los requisitos de prestaciones surgen como respuesta(s) a cada requisito funcional

*F.1: El sistema debe realizar medidas de presión atmosférica*

- “¿En qué rango?” -> P.1.1
- “¿Con cuánta precisión?” -> P.1.2
- No se puede tener todo:
  - ¿Realmente necesitamos tantas prestaciones?
  - ¿Es realista alcanzarlas con los *constraints* que tenemos?

## Es un proceso iterativo

- Así que tendremos que darle varias ‘vueltas’ hasta que todo sea consistente y tenga sentido individualmente y en conjunto
- Los requisitos deben cumplir las 9 características vistas anteriormente
- Si se hace esto bien, sufriremos menos durante el proyecto ;)

## Verificación

- Es vital tener una estrategia de verificación, es decir: “¿cómo sabemos que estamos cumpliendo los requisitos?”
- Todo requisito debe verificarse
- Al menos una medida, análisis, demostración o prueba por cada requisito (I, A, D, T : Inspection, Analysis, Demonstration, Test)

## Métodos de verificación

- **Inspección**
  - Examinando de manera no destructiva, puede incluir medidas simples
- **Análisis**
  - Usando modelos o interpretando resultados (por ejemplo por simulación)
- **Demostración**
  - Observación del ítem en funcionamiento
- **Test**
  - Evaluación o ejecución del ítem bajo condiciones, configuraciones, entradas, etc, controladas

## Matriz de verificación

Req.	Nombre Req.	Verificación				Nombre prueba	Estado
		I	A	D	T		
F.1	...			X	X	test 1	pendiente
P.1	...				X	test 1, test 2	pendiente
P.2	...		X			Análisis del datasheet del sensor utilizado	pendiente
D.1	...	X				Medida de las dimensiones	ok

## Matriz de verificación

- Una forma de poner en orden el concepto “todos los requisitos deben verificarse”
- Algunas pruebas pueden servir para verificar varios requisitos
- Otros requisitos pueden necesitar de varias pruebas
- Las pruebas se describen en un apartado (o documento aparte) “Plan de pruebas”

## Plan de pruebas

- Para los procesos de verificación del tipo “T: Test”
- Para cada test:
  - Número del test
  - Tipo de test
  - Instalaciones donde se prueba
  - Item probado
  - Procedimiento y duración del test
  - Duración de la campaña de test (reservar tiempo para varios tests y no sólo uno! - margen)
  - Fecha de la campaña de test
  - Información adicional

**Table 5.2-4: Test 3**

Test number	3
Test type	Vibration and Shock
Test facility	Own Workshop
Tested item	Central Unit and <u>SN+sensor</u> Boxes
Test level/procedure and duration	Increasing drop height first 1 meter and then 3 meters. For vibration a drive in a land road for at least 20 minutes.
Test campaign duration	First drop 1 meter, check second drop. Drive for 20 minutes and check.
Test campaign date	August-September
Test completed	NO

The first part will be with some replica models (dimension and weight) and the final part with real elements working inside for shock test. As the main vibration experiment will suffer is the transport from the recovery zone to the Station, we

## Referencias

- Koen Debeule (European Space Agency): “Start Documenting a [space] Project: Requirements and Constraints”
- SPADE team, Student Experiment Documentation (BX21\_SPADE\_SED)
- Simon Mawn (ZARM), “Requirements”, RXBX08 Selection Workshop at DLR