

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

INGENIERÍA INFORMÁTICA

Grupo de Investigación y Desarrollo en Ingeniería de Software



**Lista de Exigencias
Versión 1.0**

ELABORADO POR: Abraham Dávila edavila@pucp.edu.pe

Lima, 28 de marzo del 2005

Historial de Revisiones

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	28/03/2005	1.0	Versión inicial.	Abraham Dávila

Tabla de Contenido

1.	Introducción	4
2.	Definición	4
3.	Ubicación	5
4.	Propósito	5
5.	Aplicación	6
6.	Relación con otras herramientas	7
7.	Recomendaciones adicionales	9
8.	Ejemplo de aplicación	7
9.	Referencias Bibliográficas	9

LISTA DE EXIGENCIAS

1. Introducción

Diversas organizaciones y personalidades de la informática han señalado que una de las mayores fuentes de problemas en los proyectos informáticos, está en la correcta identificación de requerimientos. Tal ha sido la preocupación sobre el tema, que desde hace varios años se habla de Ingeniería de Requerimientos, cómo la aplicación de un enfoque de ingeniería y de manera sistemática para la obtención de requerimientos.

La Lista de Exigencias es una herramienta que puede ser utilizada en:

- La definición del alcance del proyecto.
- La derivación hacia el documento de Especificación de Requerimientos de Software (SRS de *Software Requirements Specifications*) [IEEE 1998].
- La comprobación de todos los aspectos funcionales y no funcionales que se realiza al final del proyecto (como una lista de comprobación).

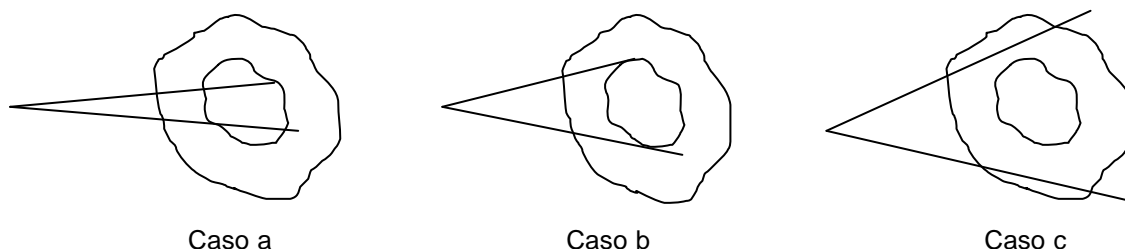
La lista de exigencias es una técnica derivada de una metodología alemana empleada para el diseño en ingeniería mecánica [BAR1985].

2. Definición

La LE se define como una herramienta que permite la recopilación de las diversas características funcionales y no funcionales que un producto podría tener para cumplir con determinados requerimientos.

Una buena lista de exigencias debe cubrir la mayor cantidad de características posibles que un producto software pudiera tener sin ningún tipo de limitación. Definitivamente todas las características que aparecen en la LE, no serán necesariamente implementadas; en cambio, la presencia (en la LE.) puede ayudar a resolver diferencias de percepción con los usuarios y sobre todo puede ayudar haciendo explícitos muchos de los requerimientos implícitos que usualmente acompañan cualquier decisión de adquisición (sea como producto o como proyecto de construcción) de un software.

Cuando una organización construye un software (bajo pedido o para un mercado determinado) se pueden presentar tres situaciones particulares de la percepción del dominio del problema donde se pretende ofrecer una solución.



El problema se representa en las figuras como la zona delimitada por la curva cerrada interior, el contexto como la zona delimitada por las curva cerrada exterior y los segmentos representan la visual del desarrollador que observa el problema.

En el primer caso (caso a) se puede apreciar que la percepción del problema no es completa y eso es frecuente cuando el desarrollador (el “analista”, que hace el análisis, la labor pensante), sólo se queda en lo que el usuario manifiesta como **necesidad** y el desarrollador cree haberlo

comprendido. Lo más probable es que haya muchos requerimientos implícitos que no han sido percibidos y que al final conlleven a una solución no satisfactoria para el usuario.

El segundo caso (caso b) representa una visión completa del problema y es muy probable que el desarrollador construya una aplicación que satisface la necesidad. También es probable, que si hay un cambio en los requerimientos, los cuales están siempre presentes en el desarrollo de software y es una situación a la que muchas veces no podemos decirles que no; el proyecto sufrirá –muy posiblemente- un retraso y la magnitud del cambio dependerá si la arquitectura elegida para la solución soporta o no fácilmente el cambio. En este caso el desarrollador solo se ha limitado a tratar de comprender el problema y ha reducido “su análisis” a cumplir con lo que el usuario necesita “en ese momento”.

El tercer caso (caso c) representa una visión completa del problema y su contexto. Un desarrollador responsable que actúa como un verdadero analista, comprende cuál es el problema, entiende cuál es la necesidad presente y futura del usuario, analiza la volatilidad del contexto, revisa la perspectiva / proyección del negocio del usuario; y tomando todo lo anterior, determina diversas alternativas de alcance del proyecto / producto a construir; de modo tal que sea el usuario –con asistencia del desarrollador- quien determine el alcance del mismo. El trabajo así realizado, merece usualmente –y por experiencia propia- el reconocimiento del usuario, porque éste comprende que el desarrollador está trabajando profesionalmente para él. La arquitectura puede ser planeada para una situación futura; pero la implementación puede atender estrictamente el problema actual, quedando para una futura etapa aquellas ideas que se obtuvieron de un análisis completo del problema.

Considerando los tres casos mencionados anteriormente, la lista de exigencias constituye una herramienta que apoya el caso c; pues busca que el desarrollador piense en todas las posibilidades o potencialidades del producto que se quiere construir y luego considerando los tiempos y presupuestos del usuario determinará con éste el alcance del proyecto.

3. Ubicación

La lista de exigencias se utiliza en la fase inicial de un proyecto y puede ser utilizada para:

- Construir la estructura de descomposición del trabajo (WBS de *Work Breakdown Structure*) [PMI 2001].
- Derivar casos de prueba de aceptación, ya que las entradas están redactadas en términos que el usuario comprenda.

4. Propósito

La LE proporciona una gran ayuda para:

Determinación del alcance

- Para asegurar que todas las características estén consideradas.
- Para ayudar a hacer explícitos, los requerimientos implícitos.

Determinación de la arquitectura a seguir.

- Determinar con el usuario, de acuerdo al presupuesto y el tiempo, los aspectos funcionales que se considerarán en la solución.
- Definir con el usuario, la posibilidad de crecimiento del producto en futuros proyectos.

Determinación de casos de prueba.

- Determinar casos de prueba de aceptación para las entradas de la lista.

5. Aplicación

Los pasos siguientes describen como se completa la LE:

Paso 1: Se recopila toda la información posible de productos similares al que se desea construir.

Paso 2: Se evalúan productos similares y se obtienen los principales características funcionales y no funcionales.

Paso 3: Se conversa con los usuarios sobre que funcionalidades considera convenientes incluir en el producto.

Paso 4: Se registran todas las entradas en la lista de exigencias (puede usarse además tormenta de ideas para completar esta parte).

Paso 5: Se clasifican las entradas de la LE en diversas características (por funcionalidades, no funcionalidades u otros clasificadores).

Paso 6: Se cuantifica la complejidad de cada características, lo recomendable es usar valores referenciales.

Paso 7: Se establece una prioridad en el desarrollo de cada característica (componente) del producto.

Paso 8: Se establece que entradas de la LE serán **Exigibles** (es decir, se convertirán en requerimientos) y cuales **Deseables** (no obligatorios para este proyecto).

El paso 4 y 5 se debe de realizar con mucho cuidado y criterio, debido a que algunas entradas se pueden rescribir, fusionar o dividir en la LE. Para los pasos 6 y 7 se pueden utilizar puntuaciones numéricas (p.e. 1..5) o de calificaciones ordinales como (alta, media, baja); de modo tal que el equipo pueda identificar la complejidad y la prioridad relativa de cada entrada de la lista de exigencias.

LISTA DE EXIGENCIAS			Pág. de	
	PROYECTO :		Equipo.	
Nro.	Características	Dific	Prior	Exig
A	Característica del tipo A			
1	Característica A1			
2	Característica A2			
3	Característica A3			
4	Característica A4			
5	Característica A5			
B	Característica del tipo B			
6	Característica B1			
7	Característica B2			
	:			
	:			

FIGURA 1. Formato de la lista de exigencia

La figura 1 presenta un formato que se puede utilizar para la LE [ABR2003]. El formato presenta las siguientes columnas: : (i) número de orden correlativo para cada agrupador de funcionalidades y características, (ii) las funcionalidades y características del producto, (iii) el nivel de dificultad en la implementación que pueda tener dicho ítem, (iv) la prioridad que tiene el ítem en el desarrollo del producto (entiéndase también como importancia), y (v) el carácter de Exigible o Deseable del ítem en el producto.

6. Ejemplo de aplicación

En la figura 2, se muestra un ejemplo reducido de un lista de exigencias de un software de operaciones con matrices [ABR2003]. El ejemplo servirá para mostrar las relaciones con las demás herramientas posteriormente. Además se debe notar que los ítems de C corresponden a características de operación (a nivel general) y que viene a ser luego un requerimiento no funcional del software.

LISTA DE EXIGENCIAS			Pág. 1 de 1	
PROYECTO : Ejemplo de Sistema de operaciones con matrices			Equipo: GIDIS	
Nro	Funcionalidades y Características	Dific	Prior	Exig
A	Ingreso/Salida de matrices			
1	Definir datos de una matriz.	3	3	E
2	Leer desde el teclado.	3	4	E
3	Importar desde una hoja de cálculo.	5	1	D
4	Generar una matriz aleatoria.	3	1	D
5	Exportar una matriz a una hoja de cálculo.	4	2	E
6	Mostrar la matriz en una grilla.	2	1	E
B	Operaciones con matrices			
1	Suma de matrices.	1	4	E
2	Multiplicación de matrices.	2	4	E
3	Inversa de una matriz.	3	3	E
4	Traspuesta.	2	3	E
5	Determinante.	2	3	E
C	Características adicionales			
1	El software deberá cargar todas las matrices en memoria.	2	2	E
2	El software funcionará sobre WEB.	2	5	E

FIGURA 2. Lista de exigencia –parcial- para un software de operaciones con matrices

7. Relación con otras herramientas

7.1. Relación con el documento inicial del proyecto.

Los proyectos tienen un documento inicial, que le da punto de partida al desarrollo del mismo. El documento inicial que puede ser el documento de visión [RSC1999], el plan de proyecto de software [PRE2002] o la definición del sistema [FAI 1988], según la metodología que use, debe ser confeccionado considerando solamente los exigibles de la L.E. (aquellas características a las que realmente se están comprometiendo). En ese sentido, la L.E. se debe completar en los primeros días de iniciado el proyecto (dentro del curso es la primera semana) de modo que pueda ser útil en la elaboración del documento de alcance inicial.

Como suele ocurrir en los proyectos, el plan de proyecto es un documento dinámico que cambia conforme transcurre el tiempo, sobretodo al inicio. La LE cumple con la misma suerte, es decir, se puede afinar el contenido de la misma conforme aparecen nuevas consideraciones para el proyecto.

7.2. Relación con los Casos de Uso.

Un caso de uso es un tipo de clasificador que representa una unidad coherente de funcionalidad provisto por un sistema, un sub-sistema o una clase manifestada por secuencias de mensajes intercambiados entre el sistema (subsistema, clase) y una o más entidades externas (denominadas actores) junto con acciones ejecutadas por el sistema (subsistema, clase) [OMG2003].

Los casos de uso se pueden obtener a partir de una L.E., como ocurre actualmente. Los alumnos elaboran los casos de uso y el catálogo de requisitos revisando constantemente la L.E., que ha sido previamente acordada con el profesor. Lo que ha venido ocurriendo en la mayoría de casos es que los alumnos construyen un caso de uso por ítem o grupo de ítems de la L.E., ha sido poco frecuente encontrar casos en los que un ítem de los casos dé lugar a dos casos de uso. Para el caso del ejemplo, se puede apreciar que solamente hay tres casos de uso: (i) el que se refiere a cargar datos en la matriz, que agrupa a los ítems A1, A2 y A6 (ii) exportar datos de la matriz se refiere a A5 y (iii) ejecutar operación con matrices agrupa a todos los del B.

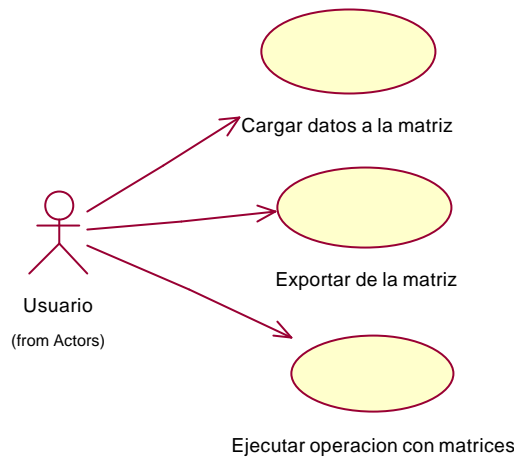


Figura 3. Caso de uso para el extracto de la L.E. del sistema de operaciones con matrices

7.3. Relación con Estructura de Descomposición del Trabajo.

La estructura de desagregación del trabajo (EDT o WBS de sus siglas en inglés de Work Breakdown Structure) es una herramienta que ha sido adoptada y es recomendada por PMI. La EDT es una agrupación de elementos de proyectos orientado al entregable que organiza y define el alcance total del trabajo del proyecto [PMI2001]. Cada sub-nivel del proyecto representa un incremento en el detalle de la definición del trabajo del proyecto y está definido por el tamaño del proyecto.

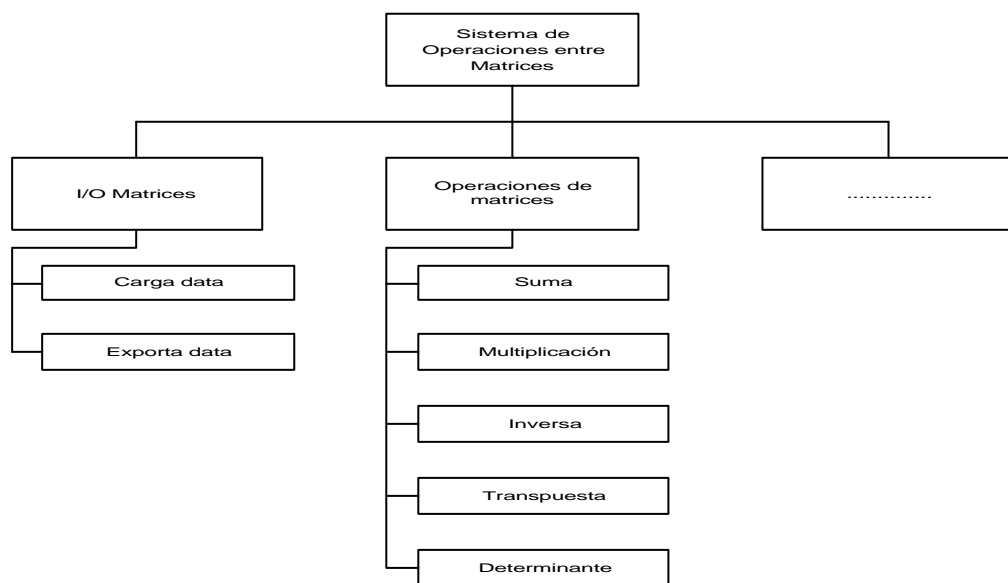


Figura 4. EDT para el extracto de la L.E. del sistema de operaciones con matrices.

Los proyectos que se desarrollan en los cursos de la universidad no son grandes, por lo que se espera que los EDT tengan dos sub-niveles y, en muy pocos casos, tres. En la figura 4, se muestra el EDT correspondiente al ejemplo mostrado en la figura 2. Se debe comentar que sólo los exigibles han sido considerados para construir el EDT y se ha dejado de lado los deseables. También se puede apreciar que los ítems A1, A2 y A6 han sido considerados como un solo paquete de trabajo ("carga data") y se ha dejado A5 como algo separado. Los primeros se han agrupado porque se pueden considerar como servicios muy relacionados. En el caso de B, cada ítem de la L.E. es un paquete de trabajo del EDT.

8. Recomendaciones adicionales

- a) La LE es un documento de trabajo inicial, en ella se pueden volcar todas las ideas, se puede discutir con el usuario que va y que no va en el proyecto; sin embargo no es un documento final por si mismo. Si desea utilizar el documento como parte del contrato, es recomendable eliminar todas las entradas que finalmente quedaron como deseables para evitar malos entendidos.
- b) La LE puede utilizarse como una herramienta para estimación de esfuerzo de manera sencilla, añadiéndole alguna columna para ese concepto.

9. Referencias Bibliográficas

- [BAR1985] Barriga, Benjamín. *Método de diseño en ingeniería mecánica*. Documento Interno PUCP, 1985.
- [DAV2003] Dávila, Abraham. Lista De Exigencias: Una Experiencia Académica para la Definición y Planificación de Proyectos de Cursos del Área de Ingeniería de Software. [disponible en: <http://gidis.inf.pucp.edu.pe/publicaciones/2003/gidis006es.pdf>].
- [FAI1988] Fairley, Richard. *Ingeniería de Software*. McGraw Hill / Interamericana de México S.A. DE C.V. 1988.
- [PRE2002] Pressman, Roger. *Ingeniería de Software- Un enfoque práctico*. McGraw Hill / Interamericana de España S.A.U. 2002.
- [RSC1999] Rational Software Corporation. *Rational Unified Process*. McGraw Hill / Interamericana de España S.A.U. 2002.
- [PMI2001] Project Management Institute, *Project Management Institute Practice Standard for Work Breakdown Structure*, www.pmi.org, 2001.
- [OMG2003] Object Management Group. *OMG Unified Modeling Language Specification Versión 1,5*. Marzo 2003, www.uml.org
- [IEEE1998] The Institute of Electrical and Electronics Engineers Inc. *IEEE Std 830-1998 Recommended Practice for Software Requirements Specifications*. IEEE – 1998.