

Olivia G. Fragoso Diaz

Sabático.pdf

 Tecnológico Nacional de México

Detalles del documento

Identificador de la entrega

trn:oid::20755:437037520

Fecha de entrega

6 mar 2025, 3:51 p.m. GMT-6

Fecha de descarga

6 mar 2025, 3:58 p.m. GMT-6

Nombre de archivo

Entregable.pdf

Tamaño de archivo

2.6 MB

49 Páginas

15,956 Palabras

84,983 Caracteres

2/1

[Handwritten signature]

6% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Fuentes principales

- 6%  Fuentes de Internet
- 2%  Publicaciones
- 5%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión



Caracteres reemplazados

54 caracteres sospechosos en N.º de páginas

Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Handwritten signature and initials in blue ink.

Fuentes principales

- 6% Fuentes de Internet
- 2% Publicaciones
- 5% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1	Internet	www.coursehero.com	<1%
2	Internet	repositorio.unas.edu.pe	<1%
3	Internet	vdocuments.com.br	<1%
4	Trabajos entregados	uvt on 2019-07-31	<1%
5	Internet	la-respuesta.com	<1%
6	Internet	www.slideshare.net	<1%
7	Trabajos entregados	Instituto Superior de Artes, Ciencias y Comunicación IACC on 2022-10-18	<1%
8	Internet	journal.comm.or.kr	<1%
9	Internet	es.unionpedia.org	<1%
10	Internet	docplayer.es	<1%
11	Internet	dl.acm.org	<1%

12	Internet	www.springerprofessional.de	<1%
13	Internet	iejemplos.com	<1%
14	Internet	tel.archives-ouvertes.fr	<1%
15	Internet	zir.nsk.hr	<1%
16	Internet	es.slideshare.net	<1%
17	Trabajos entregados	Universidad TecMilenio on 2024-03-11	<1%
18	Internet	www.researchgate.net	<1%
19	Internet	mafiadoc.com	<1%
20	Internet	mind42.com	<1%
21	Trabajos entregados	Facultad Latinoamericana de Ciencias Sociales (FLACSO) - Sede Ecuador on 2018-1...	<1%
22	Internet	files.eric.ed.gov	<1%
23	Internet	repositorio.uci.cu	<1%
24	Internet	qdoc.tips	<1%
25	Trabajos entregados	Pontificia Universidad Catolica del Peru on 2017-11-24	<1%

26	Internet	ieeexplore.ieee.org	<1%
27	Trabajos entregados	Universiti Sains Malaysia on 2024-03-19	<1%
28	Internet	repositorio.upn.edu.pe	<1%
29	Trabajos entregados	Pontificia Universidad Catolica del Ecuador - PUCE on 2021-07-22	<1%
30	Trabajos entregados	Universidad Tecnologica del Peru on 2021-09-20	<1%
31	Trabajos entregados	Universidad del Istmo de Panamá on 2024-05-23	<1%
32	Internet	www.banxico.org.mx	<1%
33	Internet	www.mindef.gob.pe	<1%
34	Trabajos entregados	Universitat Internacional de Catalunya on 2020-11-05	<1%
35	Internet	alboratorio.infeuropa.es	<1%
36	Internet	transportesynegocios.wordpress.com	<1%
37	Internet	www.di.uniovi.es	<1%
38	Internet	www.scribd.com	<1%
39	Trabajos entregados	ITESM: Instituto Tecnologico y de Estudios Superiores de Monterrey on 2023-12-15	<1%

40	Internet	cdn.slideserve.com	<1%
41	Internet	digital.csic.es	<1%
42	Internet	vdocumento.com	<1%
43	Internet	www.dspace.espol.edu.ec	<1%
44	Internet	www.icao.int	<1%
45	Internet	www.softwareadvice.com	<1%
46	Trabajos entregados	Catholic University of Peru on 2007-02-27	<1%
47	Publicación	Por Lemus. "Estudio comparado del desarrollo Embrionario de Especies ov para...	<1%
48	Trabajos entregados	Universidad Autonoma de San Luis Potosi on 2006-12-29	<1%
49	Trabajos entregados	Universidad Estatal a Distancia on 2020-09-15	<1%
50	Internet	core.ac.uk	<1%
51	Internet	documentop.com	<1%
52	Internet	fido.palermo.edu	<1%
53	Internet	ouci.dntb.gov.ua	<1%

54	Internet	viltisemann.cl	<1%
55	Internet	www.jove.com	<1%
56	Internet	pt.slideshare.net	<1%
57	Publicación	Marta Brković Dodig, Linda N. Groat. " The Routledge Companion to Games ...	<1%
58	Trabajos entregados	Universidad de Lima on 2015-06-29	<1%
59	Internet	hdl.handle.net	<1%



TÍTULO: ESTRATEGIAS PARA MEDIR PRODUCTOS DE SOFTWARE

**INFORME FINAL DEL SABÁTICO
(100%)
MODALIDAD B2 ELABORACION DE LIBRO
No. Autorización A5-1-062/2024**

Dra. Olivia Fragoso Diaz
Olivia.fd@cenidet.tecnm.mx

Tabla de contenido

Capítulo 1.....	2
Introducción a la calidad de software	2
Objetivo: Que los profesores y estudiantes relacionados con el desarrollo de software entiendan la importancia de la calidad de software.....	2
1.1 Para quien es este libro	3
1.2 Calidad	3
1.3 Proceso de desarrollo de software.....	4
1.4 Cada quien realiza a su manera las cosas.....	4
1.5 Cuesta y los clientes no lo entienden.....	5
1.6 Aprender incluyendo calidad	6
1.7 Nunca ver para atrás	6
1.8 Tener claridad y certeza para poder mejorar	6
1.9 Constancia en las actividades.....	7
1.10 Lo que este libro no es	7
1.11 Que encontramos en este libro	8
Bibliografía	10
Referencias.....	10
Capítulo 2.....	11
Definir para medir: el producto que se quiere medir	11
Objetivo: El objetivo de este capítulo es iniciar el camino hacia las mediciones de productos de software.....	11
2.1 Introducción	12
2.2 Definir	12
2.3 Asignación de valores.....	22
2.4 Definiciones.....	23
2.3 Actividades	26
Bibliografía	27
Referencias.....	27
Capítulo 3.....	28
Definir para medir el atributo de calidad que requiere medirse	28
Objetivo: El objetivo de este capítulo es mostrar la forma en que se puede mapear a un producto de software con el atributo de calidad que requiere medirse.....	28
3.1 Modelos de Calidad.....	29
3.2 Atributo de calidad y producto de software.....	31
3.2.1 Seleccionar atributo de calidad	32
3.2.2 Definir atributo de calidad	32
3.2.3 Identificar los elementos del producto de software que pueden afectar al atributo de calidad	33
3.2.3.1 Relacionar los elementos del producto de software con el atributo de calidad	33
3.2.4 Asignar importancias a los elementos relacionados	41
3.2.5 El valor de las importancias.....	42
3.2.5.1 Mejor y peor valor.....	43
3.3 Como medir	45
3.3 Actividades	47
Bibliografía	48
Referencias.....	48

[Handwritten signature]

[Handwritten signature] 1

Capítulo 1

Introducción a la calidad de software

Objetivo: Que los profesores y estudiantes relacionados con el desarrollo de software entiendan la importancia de la calidad de software.

2

turnitin

Estrategias para medir productos de software

1.1 Para quien es este libro

Este libro es para personas que se dedican al desarrollo de software, sin importar el área de su formación y para todas aquellas personas que quieren o necesitan mejorar su trabajo pero que por falta de capacitación relevante no lo pueden hacer. Por capacitación relevante se entiende que aun cuando cuentan con certificaciones o cursos diversos sobre las tecnologías de su interés, no han atendido su propio proceso de desarrollo de software, mismo que no se puede mejorar si no se mide y no se puede medir si no se conoce.

1.2 Calidad

La palabra "calidad" tiene muchos significados que dependen del contexto donde se usa. A esto se le conoce con el nombre de relatividad, es decir, la calidad es relativa al contexto en el que se usa, por esto, es importante emplearla solamente cuando se puede explicar. Existen muchos ejemplos del uso de la palabra "calidad" pero no significa que su uso sea correcto. Desafortunadamente, con frecuencia se escucha decir que algo es de calidad sin que ese algo se pueda explicar.

Mucho ha sido el trabajo de especialistas que han generado estándares en diversos dominios, pero los que interesan en este libro se refieren al dominio de Ingeniería de software, que de manera general se pueden clasificar como estándares orientados a los de procesos de desarrollo de software y estándares orientados a los productos de software.

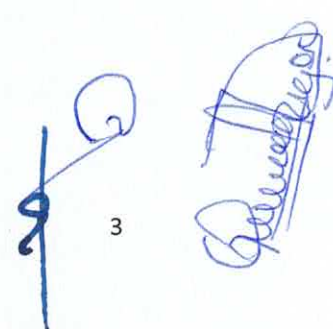
Dentro de los estándares orientados al proceso de desarrollo de software se pueden encontrar:

- ① ISO/IEC 15504- SPICE
 - ① ISO/IEC 12207
- ① IEEE-procesos, verificación, validación, pruebas, etc.
 - ① CMM-I Integrated capability maturity model
 - ① MOPROSOFT
 - ① ISO/IEC 29110

Dentro de los orientados a los productos de software se pueden encontrar los modelos y estándares

- ① McCall
- ① BOHEM
- ① FURPS
- ① DROMEY
- ① ISO-9126-25000

3



Estrategias para medir productos de software

A pesar de los esfuerzos de investigadores en el área de Ingeniería de Software, aún es complicado medir los procesos y los productos, aunque existe la filosofía de que un proceso de calidad produce un producto de calidad, lo cierto es que es importante poder decir qué está bien y qué está mal en lo que se hace y se produce. En el área de Ingeniería de software esto se vuelve complicado porque se producen cosas intangibles. Sin embargo, se puede aprender a medir esas cosas y otorgarles algún valor que permitirá identificar y corregir lo que está mal y planificar y ejecutar acciones o actividades para mejorar lo que se quiere.

El tema de la calidad es importante porque si se aprende a distinguir lo que está bien y lo que está mal, y se puede decir porqué, entonces se podrá planificar, controlar y ejecutar acciones que nos permitan llegar a la mejora continua. Mejora continua es un término que se usa cuando se puede demostrar que se conoce lo que está mal y lo que está bien en los procesos y productos y que continuamente se identifican elementos que permitan corregir cualquier defecto en lo que se hace.

Pero llegar a la mejora continua no es fácil cuando no se sabe que está mal o bien o que no se sabe cómo explicar. El saber cuándo algo está mal o bien involucra irremediamente el hacer mediciones y las mediciones es un tema difícil en lo que a Ingeniería de software respecta. En primer lugar, por la resistencia al trabajo que genera un cambio en la forma y cantidad de trabajo que se hace y, en segundo lugar, pero no menos importante es que no se sabe cómo medir lo que se hace.

Este libro no trata de cómo llegar a la mejora continua, sino de cómo se puede aprender a medir para poder iniciar un camino hacia la mejora continua que es el nivel deseable de calidad para todo lo que se hace. Por lo tanto, una recomendación muy importante es no mencionar la palabra calidad si no la podemos explicar. Para propósitos de este libro, se toma la calidad como la aplicación de las mejores prácticas de tal manera que los productos se encuentren libres de defectos, pero que también se especifican valores para los elementos que definen a un producto de software.

1.3 Proceso de desarrollo de software

En el área de Ingeniería de Software, la manera para llegar a la mejora continua es iniciar por establecer los procesos de desarrollo y medirlos todo el tiempo, desafortunadamente, en la mayoría de los cursos de Ingeniería de software las mediciones no entran en los programas de estudio ni capacitaciones. Aún existen programas en donde se enseña solamente el proceso de desarrollo, aunque no sabemos cómo se enseña, es decir, en algunos cursos se enseñaba un ejemplo diferente en cada fase del desarrollo y no se seguía un solo ejemplo durante todo el proceso. Por lo mismo no es fácil enseñar a medir. Esto se puede remediar cuando se sigue un estándar para el proceso de desarrollo.

No existe un único proceso para producir software, pero afortunadamente lo que los estándares nos ofrecen son marcos de mejores prácticas que cada quién puede implementar como lo decida.

1.4 Cada quien realiza a su manera las cosas

Si bien los estándares no obligan a realizar los procesos de desarrollo de la misma manera para todas las empresas, esto representa el beneficio de que cada quien decida

Estrategias para medir productos de software

como llevar a cabo el desarrollo de sus proyectos de software. Este beneficio representa una ventaja para empresas en donde sus trabajadores están comprometidos con su trabajo y aceptan capacitaciones continuas tanto como los cambios en la tecnología lo exigen.

Un punto importante en este aspecto es la preparación y el comportamiento ético de los líderes de las empresas. Debido a la intangibilidad del software, muchos responsables de proyectos apresuran los trabajos, no planifican las actividades considerando planeación, mediciones ni acciones de mejora, lo que los lleva a ganar tiempo para atender un requerimiento del cliente, inclusive muchas veces el cliente es quien decide el tiempo en que se desarrolla un proyecto, lo cual hace que muchas veces no se consideren los aspectos de calidad derivados de las mediciones. Esto sugiere que los clientes sean empresas o personas, son sujetos de capacitación en cuanto a los procesos de desarrollo con el propósito de no forzar el tiempo de entrega de los productos y de esta manera las empresas cumplir con sus clientes, pero cumplir también con ellas mismas en el nivel de calidad que quieren cumplir.

1.5 Cuesta y los clientes no lo entienden

La implementación de mediciones en el proceso de desarrollo implica tareas que son diferentes a las tareas de desarrollo que genéricamente las conocemos como requerimientos, análisis y diseño, implementación y pruebas. Estas tareas de mediciones también deben ser planificadas y se les debe asignar algún espacio en el tiempo dentro del proceso de desarrollo, así como asignar a los responsables de llevarlas a cabo. Al momento de ponerlas dentro de un programa de trabajo se les da tiempo y asignan responsables. Si la estimación del costo del proyecto se basa en el factor tiempo y horas/hombre, las tareas de mediciones generan un costo dentro del proyecto y deben considerarse en el tiempo de entrega del mismo. Sin embargo, en muchas ocasiones no se saben justificar, no se les otorga la importancia debida y por lo tanto los clientes no lo entienden.

Hay que entender que cuando se mide es para mejorar no para perjudicar a nadie, entender la calidad tanto en cursos de Ingeniería de Software y en las empresas que usan procesos de desarrollo de software es un tema difícil porque hay que explicarlo desde el ámbito de su utilidad y no desde el querer perjudicar a alguien. Este tema es importante porque puede generar un mal ambiente de trabajo, se puede pensar que si se mide algo es porque el jefe quiere castigar o despedir a alguien con base en las mediciones.

Es responsabilidad de la gente encargada de enseñar Ingeniería de Software y capacitación en las empresas, hacer entender que las mediciones están orientadas a mejorar lo que se hace. Y es responsabilidad de los que aprenden los procesos de desarrollo saber por qué se hacen las mediciones e inclusive tomar las mediciones como la herramienta no estándar o personalizable para mejorar en principio de manera individual y luego en equipo lo que se hace. Un ejemplo que ayuda a entender esta forma de ver las cosas en el PSP de Whats Humphrey [1].

Estrategias para medir productos de software

1.6 Aprender incluyendo calidad

Muchos han sido los avances sobre calidad en procesos de desarrollo y métricas de software. Pero la enseñanza de la Ingeniería de Software queda a deber en cuanto a calidad. Normalmente los cursos no son suficientes o no están relacionados entre sí para enseñar procesos de desarrollo de software que incluyan la preparación de la calidad. Esto no es pretexto para no aprender, en la actualidad tenemos disponible una gran cantidad de información a través de internet y que toca todos los aspectos de calidad. Si estamos en la era de la información, porqué nos cuesta aceptar que tenemos que medir?. Esto sin contar que si logramos aprender incluyendo aspectos de calidad podemos trasladar esas enseñanzas a nuestra vida diaria y por lo tanto mejorar todo nuestro contexto, en ese momento se convierte en una filosofía de vida. Si la mejora ya fuera para todo mundo una filosofía de vida, sería más fácil tener un programa de calidad en todo lo que hacemos.

1.7 Nunca ver para atrás

Cuando queremos hacer las cosas de la mejor manera, nuestra referencia es lo que existe y nos indica que sí se pueden cambiar las cosas para mejorar, voltear al pasado solamente nos pondrá obstáculos. Claro que lo que existe antes, como por ejemplo las razones que dieron origen a la famosa frase "la crisis del software" son cosas que no se deben olvidar para evitar que regresemos a la forma en cómo se hacían las cosas. Buscar hacia adelante es hacia donde debemos conducir nuestras actividades de mediciones, normalmente se trabaja sobre lo que existe, "reinventar el hilo negro" no es algo que nos ayude a mejorar, es mejor "usar el hilo negro" para mejorar lo que hacemos.

Dado que el desarrollo de software consiste muchas veces en generar soluciones a problemas de cualquier dominio, es muy importante buscar lo que esos dominios ya resolvieron para hacer uso de esas soluciones y poder generar mejores productos. Por ejemplo, si existen estudios sobre ergonomía que aportan en aspectos de usabilidad, accesibilidad, legibilidad, etc. Es importante considerar esos estudios e incluir los atributos y valores que estudia la ergonomía en la solución que queremos producir. Cuando estos atributos y valores se conocen, es muy fácil integrarlos en nuestro proceso de medición, especialmente en cuanto a los productos se refiere.

1.8 Tener claridad y certeza para poder mejorar

Cuando se quiere medir algo lo primero que se debe tener es claridad sobre lo que se requiere medir y que se le quiere medir. El primer paso es saber qué es lo que se quiere medir, es decir, cual es el objeto que se requiere o necesita medir. Este es un desafío muy grande porque no necesariamente se puede definir ese objeto, sobre todo cuando se trata de objetos relacionados con el proceso de desarrollo de software como son los requerimientos, el análisis, el diseño, las pruebas, el código, etc. Por ejemplo, ¿todas las personas que desarrollan software saben que es un documento de requerimientos?, saben que contiene?, si no se sabe que es ni que tiene difícilmente podrá medirse algo en ese documento. Después, es importante tener claridad sobre qué es lo que se le quiere medir a ese objeto. Muchas veces se escuchan o se leen términos como usabilidad, accesibilidad, mantenibilidad, disponibilidad, reusabilidad etc. Sin embargo, no necesariamente se entiende que quiere decir cada uno de estos términos, inclusive



algunas personas confunden unos con otros. Poder entender el objeto a medir y lo que se le quiere medir son tareas que no deben ser apresuradas y requieren tiempo, si no se entienden o se tiene un entendimiento equivocado sobre el objeto y lo que se le quiere medir, las mediciones que se lleven a cabo no producirán datos confiables para poder llevar a cabo mejoras sobre el objeto, mucho menos se llegará a la mejora continua.

El caso de un producto como el código tiene mayor definición y existen muchos trabajos de investigación sobre como medirlo. Específicamente, la literatura es muy amplia sobre todo en lo que respecta a código orientado a objetos. Las métricas que han sido ampliamente publicadas se basan en el conocimiento del dominio, por ejemplo, definición de clase, de objeto, de herencia, agregación, etc. Pero debido a que cada quien realiza las cosas a su manera, existen cosas que no se pueden medir con las métricas existentes. Es por eso que cuando se sigue algún estándar como MOPROSOFT, desde el inicio es importante establecer una forma de escribir el código. A partir de ese momento es importante entonces incluir la forma de escribir el código en las mediciones.

Sin embargo, en el camino para llegar al código y poder tener alguna medida de que tan bien o mal está, pasamos por varios productos, por ejemplo, los diagramas de clases. ¿Pero cómo sabemos que los diagramas están bien o mal?, también podemos medir los diagramas con base en el dominio, pero a los diagramas los acompañan elementos de otros dominios como por ejemplo el ergonómico que considera el uso de elementos humanos en las actividades que forman parte de las tareas diarias de los encargados del diseño del software. Estos elementos ergonómicos son raramente considerados o medidos. Ejemplo de los elementos humanos pueden ser los colores que se usan en los diferentes elementos de un diagrama de clases como las líneas y cantidad de clases de objetos que el cerebro puede procesar.

Tres preguntas que deben hacerse siempre que se requiera atender al propósito de la calidad de cualquier producto de software son las siguientes: ¿Cuál es el producto que se quiere medir y como se define?, que atributo de calidad se requiere medir? Y como puede llevarse a cabo la medición. Con estas tres preguntas se podrá iniciar un camino de cuidado hacia la mejora continua.

1.9 Constancia en las actividades

La constancia en las actividades que se hacen día a día es clave para poder conocer lo que se hace y en su caso poder medir. Este tipo de constancia cuesta trabajo porque primero hay que establecer como se hacen las cosas y luego mantenerse haciéndolas de la misma manera implica cierto esfuerzo. Sin embargo, la constancia en las actividades permitirá decir cómo queremos medir algo, luego nos permitirá obtener datos de lo que medimos, y finalmente, aunque muy importante, tendremos la manera de generar información suficiente para saber cuándo debemos o requerimos mejorar algo. Esta es la base de la mejora continua.

1.10 Lo que este libro no es

Este libro no es sobre procesos de desarrollo de software, ni como implementarlos, ni como medirlos. Tampoco se discuten los estándares relacionados con los procesos de desarrollo de software, aunque si se mencionan en alguna ocasión. Este libro tampoco

completo

7

Estrategias para medir productos de software

algunas personas confunden unos con otros. Poder entender el objeto a medir y lo que se le quiere medir son tareas que no deben ser apresuradas y requieren tiempo, si no se entienden o se tiene un entendimiento equivocado sobre el objeto y lo que se le quiere medir, las mediciones que se lleven a cabo no producirán datos confiables para poder llevar a cabo mejoras sobre el objeto, mucho menos se llegará a la mejora continua.

El caso de un producto como el código tiene mayor definición y existen muchos trabajos de investigación sobre como medirlo. Específicamente, la literatura es muy amplia sobre todo en lo que respecta a código orientado a objetos. Las métricas que han sido ampliamente publicadas se basan en el conocimiento del dominio, por ejemplo, definición de clase, de objeto, de herencia, agregación, etc. Pero debido a que cada quien realiza las cosas a su manera, existen cosas que no se pueden medir con las métricas existentes. Es por eso que cuando se sigue algún estándar como MOPROSOFT, desde el inicio es importante establecer una forma de escribir el código. A partir de ese momento es importante entonces incluir la forma de escribir el código en las mediciones.

Sin embargo, en el camino para llegar al código y poder tener alguna medida de que tan bien o mal está, pasamos por varios productos, por ejemplo, los diagramas de clases. ¿Pero cómo sabemos que los diagramas están bien o mal?, también podemos medir los diagramas con base en el dominio, pero a los diagramas los acompañan elementos de otros dominios como por ejemplo el ergonómico que considera el uso de elementos humanos en las actividades que forman parte de las tareas diarias de los encargados del diseño del software. Estos elementos ergonómicos son raramente considerados o medidos. Ejemplo de los elementos humanos pueden ser los colores que se usan en los diferentes elementos de un diagrama de clases como las líneas y cantidad de clases de objetos que el cerebro puede procesar.

Tres preguntas que deben hacerse siempre que se requiera atender al propósito de la calidad de cualquier producto de software son las siguientes: ¿Cuál es el producto que se quiere medir y como se define?, que atributo de calidad se requiere medir? Y como puede llevarse a cabo la medición. Con estas tres preguntas se podrá iniciar un camino de cuidado hacia la mejora continua.

1.9 Constancia en las actividades

La constancia en las actividades que se hacen día a día es clave para poder conocer lo que se hace y en su caso poder medir. Este tipo de constancia cuesta trabajo porque primero hay que establecer como se hacen las cosas y luego mantenerse haciéndolas de la misma manera implica cierto esfuerzo. Sin embargo, la constancia en las actividades permitirá decir cómo queremos medir algo, luego nos permitirá obtener datos de lo que medimos, y finalmente, aunque muy importante, tendremos la manera de generar información suficiente para saber cuándo debemos o requerimos mejorar algo. Esta es la base de la mejora continua.

1.10 Lo que este libro no es

Este libro no es sobre procesos de desarrollo de software, ni como implementarlos, ni como medirlos. Tampoco se discuten los estándares relacionados con los procesos de desarrollo de software, aunque si se mencionan en alguna ocasión. Este libro tampoco

Revisado por

7

Estrategias para medir productos de software

es acerca de métricas generales del desarrollo de software, ya existen muchos libros y artículos que proponen la medición del software en diferentes niveles de abstracción.

Este libro lo que propone es entender lo que significa calidad desde la perspectiva de lo que queremos medir y como podremos entender los objetos que requieren ser medidos. Una vez entendidos los objetos, podremos derivar de manera sencilla como medirlos. Este libro tampoco trata o explica la teoría de la medición, aquí se asume que podremos tener formas de medir siempre apegándonos a la simplicidad para poder entender los resultados de la medición y cambiar lo que resulte mal, no tiene caso tener formas de medir complejas si no se pueden entender y por lo tanto no sería fácil saber cómo mejorar lo que resulte mal. Lo que si hace este libro es mostrar cómo podemos buscar una forma de saber que está mal en un objeto que se encuentra en procesos de desarrollo de software como son los conocidos productos intermedios o el resultado de procesos de desarrollo como son los productos finales, puede ser desde una aplicación móvil, una página Web, un video juego, un sistema de video conferencia, un sistema gestor de aprendizaje, editor de texto, etc., cualquier producto de software.

1.11 Que encontramos en este libro

Para atender el propósito de este libro, se definieron dos capítulos. El capítulo 2 describe que significa definir para poder medir un producto de software, en este capítulo se describe una estrategia para poder definir qué es lo que queremos medir. Este capítulo atiende la necesidad de orientar a personas que se dedican al desarrollo de software para que establezcan una forma de conocer los productos que desarrollan ya sean intermedios o finales, a partir de un modelo de proceso de desarrollo de software y que puedan encontrar el camino hacia la mejora continua.

En el capítulo 3 se describe la asociación de la definición de productos de software que se genera en el capítulo 2 con los modelos de calidad reconocidos en el ámbito del desarrollo de software.

La Tabla 1 presenta y define los términos empleados en este libro.

Estrategias para medir productos de software

Tabla 1. Términos empleados en este libro

Ciente	Persona que solicita un producto de software
Usuario	Persona que usa el producto de software
Desarrollador	Persona que realiza tareas de Desarrollo de software y puede tomar diferentes roles: Administrador del proyecto, responsable del desarrollo, analista, diseñador, responsable de manuales, responsable de pruebas, programador, etc.
Producto	Un producto puede ser Final o Intermedio. El producto final es el resultado del proceso de desarrollo. Un producto intermedio es el resultado de alguna actividad o tarea del desarrollo.
Objeto	Elemento que se considera en la medición. No se refiere al objeto del dominio orientado a objetos.
Elemento	Es el término empleado para referirse a algo que es parte de la definición de un producto.
Sub elemento	Es un elemento que es más específico que otro elemento
Atributo	Atributo es el término empleado para referirse a una característica normalmente positiva que describe a un producto



Estrategias para medir productos de software

Bibliografía

- Norma Mexicana para el proceso de desarrollo de software -NMX-059-1-2005.
- Modelo de procesos para la industria del software MOPROSOFT.
- ISO/IEC 9126.
- ISO/IEC 25000
- Software quality attributes and trade-offs, Lars Lundberg, Michael Mattsson, Claes Wohlin (eds), Blekinge Institute of Technology June 2005.

Referencias

- [1] Humphrey W, A discipline for software engineering. SEI series in software engineering. Addison Wesley. 1995

Capítulo 2

Definir para medir: el producto que se quiere medir

Objetivo: El objetivo de este capítulo es iniciar el camino hacia las mediciones de productos de software.



11



2.1 Introducción

Antes de explicar que significa Definir en este capítulo, es necesario introducir algunos términos que se emplean muy a menudo cuando de calidad se trata. Existen evaluaciones y valores objetivos y subjetivos. Una evaluación objetiva se produce cuando en su proceso se obtienen valores objetivos, los valores objetivos son aquellos calculables y que no dependen de opiniones de personas involucradas. Este concepto coincide con el de RAE "Que no está movido o motivado por intereses, ideas o sentimientos personales" (Real Academia Española 6 agosto 24). Los valores subjetivos son aquellos que se obtienen en función de una opinión, normalmente no son calculables ni definibles. También coinciden con RAE "Que depende de la manera de pensar o sentir del individuo".

Por ejemplo, cuando se evalúa un coche se le puede otorgar un valor de malo o bueno o excelente, sin especificar como se llega a ese valor. Mientras que cuando se explica la calidad del coche en términos de eficiencia, esto significa que el coche ha sido medido conforme a la definición de eficiencia. Afortunadamente, para la mayoría de objetos tangibles como un coche existen estudios que indican cuales son los mejores valores que esos objetos pueden tener, tal es el caso de que a mayor número de kilómetros/litro, el coche es más eficiente. Número de Kilómetros y Litro son directamente calculables.

En el caso de algunos objetos como lo es el software, no hay consenso sobre las mejores medidas, pero si sobre las mejores prácticas respaldadas por algunos estándares. Lo que es importante es saber cómo podemos llegar a identificar qué elementos de los productos de software deben considerarse y tratar de otorgarles valor calculable.

Para calcular algún valor a los productos de software, se requiere definir cada producto y es de lo que se encarga la siguiente sección **Definir** de este capítulo.

2.2 Definir

La definición es un proceso de análisis acerca del objeto o producto de software que describe de manera explícita los elementos del producto que pueden ser mapeados directamente con atributos de calidad. Este proceso nos conduce a dar respuesta a las preguntas que se indican en la penúltima sección del capítulo 1: Cual es el producto que se quiere medir y como se define?, ¿qué atributo de calidad se requiere medir? Y como puede llevarse a cabo la medición. La definición de un producto de software es la tarea más difícil porque no es solamente una definición en donde se describe un concepto, sino involucra investigar y relacionar los elementos que componen al producto. Sin embargo, es importante mencionar que las definiciones conceptuales son lo que da origen en muchos casos a poder investigar los elementos del producto. Cuando la definición conceptual no incluye los elementos que componen el producto, es necesario extender esa definición. Una vez lograda la definición, esos mismos elementos son los que nos darán la pauta para establecer como medir el atributo de calidad que se requiera medir y que se explica en el siguiente capítulo.

Normalmente la definición de cualquier producto se hace de lo general a lo específico. La figura 1 es una representación de la forma de una definición.

La palabra "Elemento" se usa en este capítulo para indicar que se trata de algún objeto que tiene significado para el producto con el que se relaciona, generalmente se visualiza como una parte importante para el producto y que ayuda a definirlo.

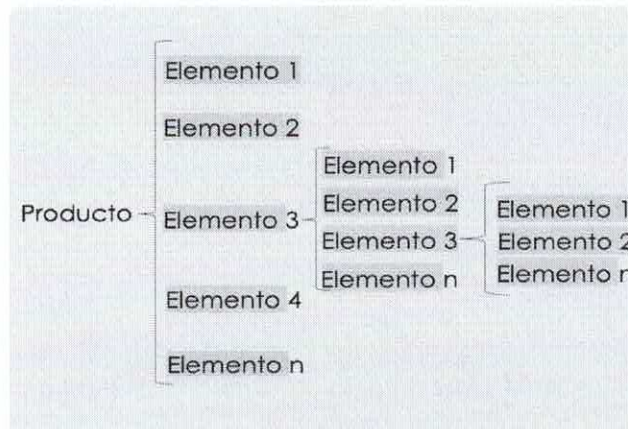


Figura 1. Estructura de la definición de un objeto o producto de software

En la Figura 1 se puede ver que un producto está compuesto de "n Elementos" entre ellos el "Elemento 3" y a su vez el "Elemento 3" también tiene sus propios elementos y así sucesivamente. Este proceso iterativo de definición de elementos puede resultar complejo dependiendo del tipo de producto. Sin embargo, es una manera ordenada de definir con precisión a un objeto. Es decir, la definición está en función de lo que se reconoce en el objeto y por eso se define identificando sus elementos. Las definiciones conceptuales o que se encuentran en un diccionario o libro son del tipo "un objeto x es un objeto que hace tal cosa o que sirve para x cosa", este tipo de definiciones casi nunca alcanzan a definir con precisión el objeto de interés, esto no quiere decir que no sirven, lo que sucede es que son muy limitadas como para definir algo con precisión, sobre todo cuando se requiere hacer una medición. La forma de definición que se describe en este libro tiene la ventaja de que cuando se llega a un nivel en donde el elemento es muy granular, este puede ser tan específico que casi nos indica como podríamos medirlo, por eso se puede decir que es una forma de definir con precisión.

Pareciera que el modelo de definición es simple, sin embargo, cuando no se conoce el objeto que se requiere medir resulta difícil entender cuáles son sus elementos. A pesar de que hay elementos comunes en algunos productos de software, no es fácil identificarlos cuando no se ha trabajado con el mismo tipo de producto. Un ejemplo son las interfaces de usuario. Las interfaces de usuario son elementos que están en todos los productos, pero sus elementos y apariencia son diferentes dependiendo del producto para el que se desarrollen, tampoco siguen necesariamente las mismas reglas de construcción o mejores prácticas. Esto se agrava porque generalmente un cliente no puede especificar sus requerimientos no funcionales como por ejemplo usabilidad, reusabilidad, requisitos de construcción, etc. Normalmente se construyen para cumplir con los requerimientos funcionales. Algunos ejemplos de interfaces diferentes se describen en la tabla 2.

Estrategias para medir productos de software

En la Tabla 2, se describen seis tipos de productos diferentes, videojuegos, sistema gestor de aprendizaje, aplicación móvil, editor de programación, página Web y sistema de videoconferencia. En la columna del medio se describen algunos elementos de los productos y en la tercera columna se presentan elementos que son específicos a los de la segunda columna. Por ejemplo: Nombre, Color y Vestimenta describen a un personaje, pero no son exhaustivos, existen más elementos que debemos identificar.

Para poder tener una definición completa de cada atributo es necesario realizar un estudio sobre cada producto e identificar todos los elementos que pertenecen a su definición, casi siempre y dependiendo del producto es difícil explicar todos sus elementos en una tabla.

Si se analizan con más detalle los ejemplos de la Tabla 2, algunos elementos de la tercera columna requieren a su vez ser definidos, por ejemplo, en el caso del videojuego, el personaje tiene Nombre, Color y Vestimenta, sin embargo, podemos otorgarle valores directamente a Nombre y Color, pero en el caso de Vestimenta, este no es específico, se debe definir a su vez y puede ser con Tipo, Color y número de Piezas. Tipo a su vez no es específico hay que analizarlo para definirlo. Tamaño puede ser otro elemento, pero Tamaño también es subjetivo porque Tamaño significa que puede ser definido en función de los elementos Alto y Ancho y nos daría la pauta para explicar algo más de un personaje. Por otro lado, Dimensión tal vez es mejor elemento para ayudar en la precisión porque es un agregado de Alto, Ancho y Largo. En el caso de que un Elemento, por ejemplo "Largo" no sea importante para el objeto que se quiera medir simplemente no se considera en la evaluación o no se le asigna algún valor. Muchos de estos elementos son considerados por aquellos que quieren evaluar la calidad de un objeto, pero hay que tomar en cuenta que cuando se quiere definir con precisión, se tienen que considerar todos los que describen al objeto.

Tabla 2. Ejemplos de elementos interfaces dependiendo del producto

Producto	Elementos de interfaz	Elementos
1.-Video juego	Organización en interfaz	Posición
	Personaje	Nombre, color, vestimenta
	Narrativa	Contexto
	Mecánica	Metas, reglas
	Logo	Tamaño, dibujo, letras
	Chat	Ícono del chat, botón de envío, indicador de recepción
	Menú	Texto, número de elementos, color de fondo
	Vista	Núm. de elementos
	Widgets	Tamaño, color
	Recompensas	Indicador numérico, tamaño, Bienes virtuales
	Premios	Tipo

Estrategias para medir productos de software

Tabla 2. Ejemplos de elementos interfaces dependiendo del producto (continuación 1)

	Barra de progreso	Indicador de progreso, tamaño, posición, color
	Pistas	Nombre
	Tutoriales	Número de elementos
	Asistencia técnica	En línea, manuales
2.-Sistemas de Gestión de aprendizaje	Recursos	Tipo de recurso, tamaño del ícono, nombre de recurso
	Registro de usuario	Texto, password, tamaño, color
	Actividades	Ícono de actividad, nombre de actividad, tamaño
	Cursos	Secciones, nombre
	Aulas	Nombre, ubicación, color de fondo
	Menú	Texto, número de elementos, color de fondo
	Organización en interfaz	Posición
	Diseño responsivo	Menú, vista
	Widgets	Tamaño, color
	Tutoriales	Número de elementos
	Asistencia técnica	En línea, manuales
3.-Aplicación móvil	Vista	Disposición, contraste, receptores, notificaciones
	Organización en interfaz	Posición, actividad
	Widgets	Tamaño, color
	Menú	Texto, número de elementos, color de fondo
4.-Editor de programación	Encabezado	Texto, color, posición
	Menú	Texto, número de elementos, color de fondo
	Área de pestañas	Posición, color
	Área de trabajo	Color, tamaño posición
	Pie de página	Color, texto, tamaño
	Widgets	Tamaño, color
	Vista	Contraste
	Tutoriales	Número de elementos
	Asistencia técnica	En línea, manuales

Estrategias para medir productos de software

Tabla 2. Ejemplos de elementos interfaces dependiendo del producto (continuación 2)

5.-Página Web	Texto	Tamaño, alineación
	Tipografía	Encabezado, pie de página, cuerpo principal
	Indicador de Navegación	Tamaño, Color, Forma
6.- Sistema de video conferencia	Organización de la estructura	Posición
	Texto	Tamaño, color
	Color de fondo	
	Diseño responsivo	Menú, vista
	Widgets	Tamaño, color
	Color de áreas	
	Tutoriales	Número de elementos
	Asistencia técnica	En línea, manuales

La tabla 2 describe algunos elementos de la interfaz de diferentes tipos de productos. La columna del extremo derecho contiene los elementos más específicos de las interfaces, algunos elementos se pueden repetir en los diferentes tipos de productos, tal es el caso de Tamaño y Color, pero los valores que se les deben asignar dependen del super elemento al que describen. Por ejemplo, para algunos tipos de videojuegos, un personaje debe tener un Color que resulte llamativo para los jugadores, en muchos casos se ven personajes vestidos de color rojo, mientras que, en un sistema gestor de aprendizaje o aplicación móvil, el color estaría asociado a las áreas en donde se escribe un texto, aquí el Color rojo es el valor más malo que pudieran tener tanto el sistema gestor de aprendizaje como la aplicación móvil y el sistema Web.

Es importante poner atención en los elementos que se repiten porque para muchos elementos ya existen investigaciones previas en donde indican cuales son los mejores valores que deben tener como el caso del Texto que puede considerar diferentes valores dependiendo en el contexto donde aparezca [6,7,8,9,10 y 11].

En el caso de elementos como Altura y Color, estos son relativamente fáciles de entender porque, en la concepción de las personas, se reconocen porque desde la infancia están presentes en cualquier proceso de enseñanza. Contraste en cambio no es lo mismo, para entender que significa contraste es necesario encontrar una definición. La definición que se puede considerar puede ser: relación entre color de fondo y color del objeto que está al frente. Hacer esta definición más precisa es entonces explicar que significa "fondo" y que significa "objeto al frente", lo que no es fácil, es por eso que se va de lo general a lo específico.

En la Tabla 2, la mayoría de los elementos de la columna del extremo derecho son específicos, pero se requiere un análisis cuidadoso para distinguir si alguno de estos tiene algo más que se deba considerar, por ejemplo, las notificaciones en las aplicaciones

Estrategias para medir productos de software

móviles deben ser definidas en mayor detalle, es decir, las notificaciones existen y usamos el término siempre que decimos que nos llega un aviso de algo, pero ¿cuáles son sus elementos que debemos considerar para poder medir?

De manera general se puede establecer que una Notificación tiene elementos como fecha y cuerpo principal. Si usamos el modelo de la Figura 1, Notificaciones se podría definir como se muestra en la Figura 2. Aunque para algunos videojuegos, las notificaciones pueden también ser definidas con un Tipo, por ejemplo, Sonoro o Luminoso. Si los tipos son importantes para describir a una notificación, entonces estos Tipos de notificación también deberían estar en la Figura 2 con la recomendación de evaluar si son lo suficientemente específicos, por ejemplo, Sonoro debería ser más ampliamente definido por medio de agregar Volumen, Forma, Tipo de Sonido, Duración etc. Y Luminoso también debería ser más específico, por ejemplo, Color, Intensidad, etc.

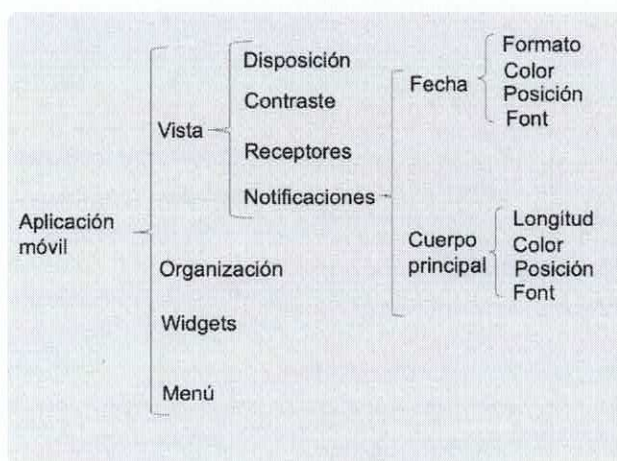


Figura 2. Descripción de notificaciones en aplicación móvil.

En la Figura 2, se muestra una definición parcial de lo que es una aplicación móvil. La aplicación móvil tiene Vistas, una Organización, tiene Widgets y Menús entre otros. A su vez, la Vista tiene Disposición, Contraste, Receptores y Notificaciones. Las Notificaciones se definen como algo que tiene Fecha, y Cuerpo principal. También se ve que Font, Color y Posición son elementos que pueden estar presentes en más de uno de los anteriores, pero los valores cambiarán dependiendo del contexto. Por ejemplo, el Font puede tener un Tipo y Tamaño en la fecha y un Tipo y Tamaño diferente en el cuerpo principal.

Entonces Tipo y Tamaño deben ser especificados como algo que hace al Font o que son parte del Font, precisando que los valores que podrá tomar cada uno de estos elementos es diferente.

completo

17

Estrategias para medir productos de software

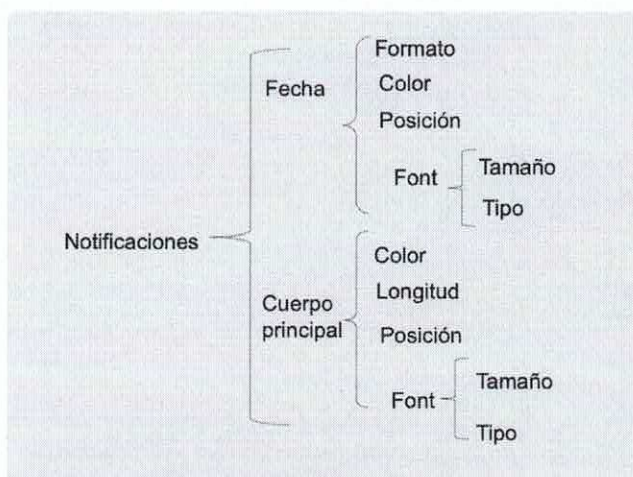


Figura 3. Definición de Notificación

Existen algunos elementos en los productos que será difícil definirlos. Los elementos como Texto, Tamaño, Font, Color, Fecha, Menú, Pie de página, Área de trabajo, etc. son elementos que con mencionarlos normalmente los entendemos, pero hay otros tal como Vestimenta en los Videojuegos, que, si bien entendemos de que se trata, no es directamente definible y hay que hacer el esfuerzo de indicar que es lo que hace una vestimenta, por ejemplo, Número de Piezas, Tipo y Color. Pero Tipo es a su vez difícil de explicar, hay que realizar una investigación para poder identificar y describir lo que hace a un tipo de vestimenta considerando que puede haber un gran número de tipos, puede ser del tipo de astronauta, granjero, guerrero, deportista, héroe, médico, constructor, etc. Estos tipos pueden considerarse valores para el elemento Tipo, pero se convierten en elementos a su vez cuando existe la necesidad de definirlos con precisión.

Los productos que se presentan en la columna 1 de la Tabla 2, se pueden considerar productos finales. Sin embargo, para poder construirlos también existen lo que se conoce como producto intermedio del proceso de desarrollo de software. La cantidad y tipo de estos productos depende en gran medida de la definición del proceso de software que se use y del nivel de madurez en el que se encuentre. Hay que recordar que, a mayor nivel de madurez del proceso, mayor definición de los productos se exige.

Para productos intermedios del desarrollo de software se puede aplicar el mismo proceso. Por ejemplo, un documento de especificación de requerimientos, si bien entendemos que se trata de un documento que contiene los requerimientos funcionales, es necesario definirlo con precisión si se quiere medir y considerar que uno de sus elementos son los requerimientos funcionales, como se presenta en la Figura 4. En la Figura 4 se presentan elementos como una Portada, un Historial de versiones, que son parte esencial de un documento cuya función principal es documentar los requerimientos funcionales.

En el caso de que lo que se requiera medir sean los requerimientos funcionales, o los no funcionales, habrá que considerar a los requerimientos como el producto a medir y por lo tanto es necesario definirlos con precisión, tanto los funcionales como los

[Handwritten signature]

Estrategias para medir productos de software

no funcionales. Para este caso un requerimiento funcional se puede definir como se describe en la Figura 5.

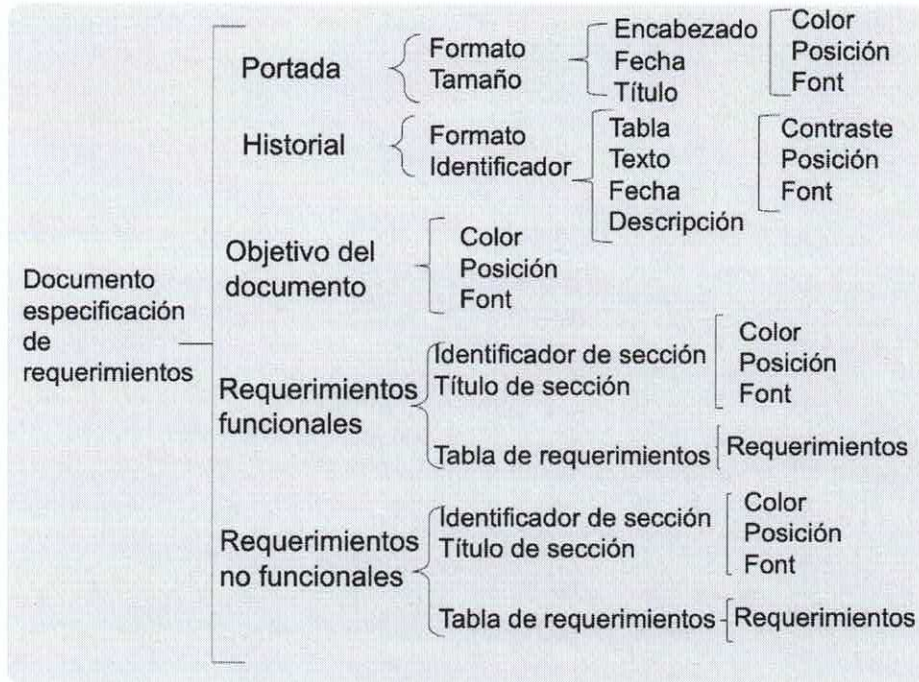


Figura 4. Definición del documento de Especificación de Requerimientos.

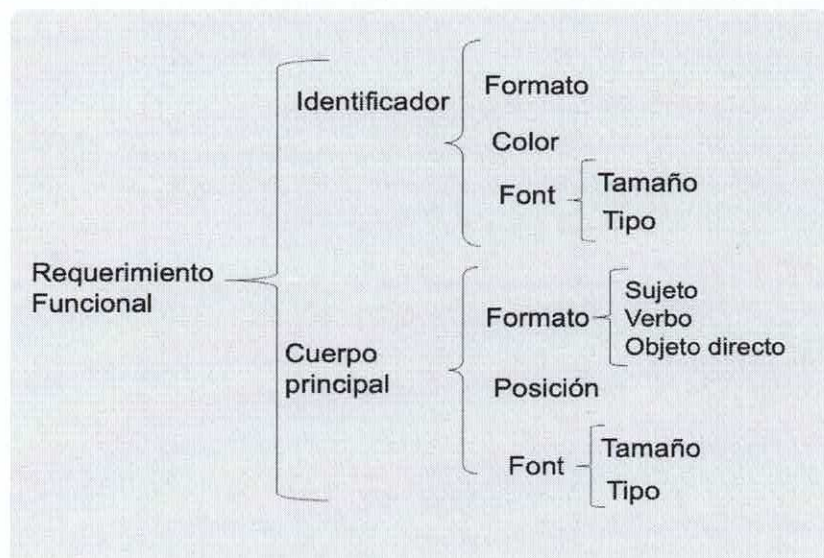


Figura 5. Definición de Requerimiento funcional.

[Handwritten signature]

Estrategias para medir productos de software

En la Figura 5, se especifica que un requerimiento funcional tiene una forma y debe cumplir con un Sujeto, un Verbo y un Objeto Directo y que también debe tener un identificador. Casi nunca se asocia el tipo de Font, Color y Formato a un elemento de un producto, pero es importante considerarlo para propósitos de consistencia, es decir, todos los documentos y texto que se manejen en un proceso de negocio que se considera en algún nivel de calidad deben cumplir con los mismos valores.

La representación de la Figura 5 es un ejemplo en donde mayormente se representan elementos de forma, pero hace falta poder indicar elementos que hacen correcta la semántica y consistencia de un escrito. En la Tabla 2.1 se presenta un ejemplo de requerimientos funcionales. Una buena práctica es explicar un requerimiento por cada identificador que se registra en la primera columna de la Tabla.

Es práctica común en encargados de los requerimientos de un sistema sin experiencia, escribir más de un requerimiento por identificador y esto puede generar problemas de entendimiento en las fases posteriores en el desarrollo de un proyecto y por lo tanto incumplimiento en el tiempo e incremento en el costo del proyecto.

La Tabla 3 presenta un ejemplo de cómo puede representarse un conjunto de requerimientos funcionales respecto al Formato descrito en la Figura 5.

Tabla 2.1. Formato de requerimientos

Requisitos de software	
No.	Descripción
R01	El sistema debe permitir al usuario autenticarse
R02	El sistema debe mostrar al responsable de proceso un formulario para llevar a cabo el registro de usuarios
R03	El responsable de proceso registrará los proyectos
R04	El sistema deberá presentar cada una de las fases del modelo de desarrollo
R05	El sistema deberá presentar cada uno de los documentos a validar
R06	El sistema mostrará los siguientes status para los documentos registrados: 1.-Registrado 2.-En revisión 3.-En corrección 4.-Verificado 5.-Validado
R07	El sistema no permitirá modificaciones a los documentos con status "Verificado" o "Validado"
R08	El sistema debe registrar el tiempo ocupado para el llenado de cada una de las etapas y al final sumarlos para reportar el tiempo final
R09	El sistema debe especificar el tiempo de llenado de cada etapa en minutos

Tabla 2.1. Formato de requerimientos (continuación)

R10	El sistema no debe permitir el proceso de validación, si aún existen documentos sin verificar
R11	El responsable de proceso podrá cerrar un proyecto aún sin haber terminado el proceso de verificación y validaciones de este, pero tendrá que ser dado de alta nuevamente como proyecto de mantenimiento
R12	El sistema debe generar un reporte al término de cada una de las etapas del proceso de desarrollo
R13	El sistema debe anexar en los reportes los siguientes datos: -Cantidad de defectos -Tiempo de corrección -Tiempo utilizado en el proceso
R14	El sistema debe guardar la información en una base de datos
R15	El sistema deberá cerrar sesión después de cada 15 minutos de inactividad

Si bien la Tabla 2.1 intenta escribir los requerimientos funcionales de un sistema de acuerdo con la definición de la Figura 5, es posible que tenga algunos defectos derivados de la falta de elementos en la definición. Por ejemplo, el R12 donde se mencionan etapas del proceso, pero no es específico al proceso ni el tipo de reporte ni siquiera los elementos del reporte.

Otros productos de software que debemos definir para poder medir son por ejemplo los Manuales de Usuario, de Instalación, de Mantenimiento, Diagramas de Clases, Reportes de pruebas, Reportes de defectos, Registros de seguimiento, etc. Para todos estos se puede sugerir iniciar la definición en términos de las reglas que los indican y que tienen que ver con el dominio que atienden, como es el caso de los principios sobre los que se construyen los diagramas del paradigma orientado a objetos, por ejemplo, un diagrama de clases que debe incluir elementos como clases abstractas, concretas, interfaces, operaciones, atributos y relaciones entre estas. A su vez las relaciones son de diferente tipo dependiendo de lo que se requiere representar, pero una clase abstracta siempre será relacionada a una clase concreta por medio de una relación de herencia, o relación de conocimiento si son clases independientes que participan juntas para lograr un propósito. En la Tabla 2.2 se presenta una lista de algunos elementos que se deben considerar para definir un diagrama de clases.

Adicionalmente se debe considerar que todos los productos tendrán en común elementos de su Formato con los Requerimientos Funcionales, por ejemplo, el Font que emplean, su Tamaño, Color y Contraste, y que para cada producto se deberán definir los mejores valores que cada elemento deba tener.

Estrategias para medir productos de software

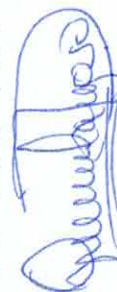
Tabla 2.2 Elementos del paradigma orientado a objetos en diagrama de clases

Elemento	Tipo
Clases	Interfaces
	Hoja
	Abstracta
	Hijas
	Superclases
Atributos	Heredados
	Públicos
	Privados
	Default
Operaciones	Heredadas
	Públicas
	Privadas
Relaciones	Herencia
	Agregación
	Dependencia
	Asociación

2.3 Asignación de valores

Cuando se logra la definición de los productos de software que interesa medir y se logra llegar a elementos lo suficientemente específicos, es decir, son tan específicos que permiten de manera automática relacionarlos con algún valor. Entonces es el momento de buscar los diferentes valores que puedan tener. Si hay elementos a los que no se les puede asignar un valor es necesario seguir analizando que elementos faltan de identificar y buscar lo que los define.

Los valores que se pueden asignar deben ser valores objetivos, es decir, calculables. Pero también existen valores de otros tipos para los cuales es necesario estudiar cómo podemos convertir esos valores en calculables para ser asignados. Por ejemplo: para un elemento puede existir un valor de "sí" o "no", o pueden ser del tipo de lista "blanco", "negro", "azul", "rojo", etc., de valores que se pueden asignar. En el caso de valores "sí" o "no" es relativamente fácil y directo el saber cuál es su mejor o peor valor ya que simplemente existen dos opciones. En cambio, cuando para un elemento existe una lista de valores que puede tomar como la de los colores, es muy importante respaldar la asignación de valores en función de estudios previos en el dominio de que se trate. Por ejemplo, en la Tabla 2.3, en cuanto al Texto hay estudios que indican que en un documento el texto debe tener un tamaño 12, entonces este tendrá el mejor valor que normalizado es 1. Si los mejores fonts son Arial, Calibrí y Times, entonces tendrán el mejor valor normalizado que es 1. Si el mejor color de texto es Negro o Azul, estos tendrán el mejor valor normalizado que es 1. Si se encuentran elementos de Texto de 10 y 14 puntos, su valor ya no es el 1, sino se reduce en un 20% y por lo tanto su valor queda como 0.8.



Estrategias para medir productos de software

Los tamaños de texto 8 y 16 se alejan aún más del mejor valor reduciendo en 40% así que su valor es solo del 0.6. Los fonts Arial, Calibrí y Times tendrán el mejor valor también tienen un valor 1 asignado y representa el mejor valor, todos los otros fonts tendrán un valor de 0.0%. La tabla 2.3 presenta un ejemplo de valores calculables y normalizados que pueden asignarse a elementos que son identificados en las definiciones de los productos.

Tabla 2.3. Ejemplo de asignación de valores

Tamaño de texto	Font	Color de Texto
8=(1-4)=0.6	Arial, Calibrí, Times=1	Negro=1
10=(1-2)=0.8	Calibrí light, Century=0.9	Azul=1
12= 1	Cambria=0.8	
14=(1-2)=0.8		
16=(1-4)=0.6		
Todos los demás tamaños=0.0	Todos los demás Fonts=0.0	Todos los demás colores=0.0

En la asignación de valores, es recomendable realizar una normalización como la que se expone en la Tabla 2.3 usando valores de 1 como valor máximo y 0.0 como valor mínimo o de menor importancia. En un proceso de definición de los elementos que permiten precisar a un producto, es posible encontrar elementos cuyos mejores valores son antagonistas, aquí la recomendación es decidir cuál es la dirección que se va a tomar en cuenta e invertir el resto de valores antagonistas hacia la misma dirección.

La asignación de valores puede documentarse directamente en los diagramas, pero resulta complicado cuando se encuentran muchos elementos y también se tienen que definir. También se pueden documentar en una tabla o nota asociada a los diagramas, esta forma es más manejable y facilita cualquier cambio.

Una vez asignados los valores ya sea directamente en diagramas o documentados a través de alguna tabla o nota asociada a los diagramas, es necesario establecer su relación con las importancias de los atributos de calidad. Esta relación se explica en la sección 3.2.5. Cuando esta relación se complete servirá para identificar qué es lo que se está haciendo mal o como puede mejorarse lo que se hace.

2.4 Definiciones

Después de explicar la forma de definir con precisión los productos de software que se requieren medir, aquí se presentan definiciones de Wikipedia (Julio 2024) de los 5 ejemplos de productos que se describen en la Tabla 2.

• Videojuego

Un videojuego o juego de video es un software o juego electrónico en el que uno o más jugadores interactúan por medio de un controlador, con un dispositivo electrónico que muestra imágenes de video.

Estrategias para medir productos de software

- Sistema de gestión de aprendizaje

Un sistema de gestión de aprendizaje es un software instalado en un servidor web que se emplea para administrar, distribuir y controlar las actividades de formación no presencial de una institución u organización.

- Aplicación móvil

Una aplicación móvil es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles.

- Editor de código

Un editor de código fuente es un editor de texto diseñado específicamente para editar el código fuente de programas informáticos. Puede ser una aplicación individual o estar incluido en un entorno de desarrollo integrado

- Página Web

Una página web es un documento en la Web que es accedida en un navegador Web.

- Sistema de videoconferencia

Un sistema de videoconferencia es un sistema interactivo que permite a varios usuarios mantener una conversación virtual por medio de la transmisión en tiempo real de video, sonido y texto a través de internet.

Si bien las definiciones anteriores se obtuvieron de Wikipedia, existen otras definiciones muy similares, lo que significa que el nivel de definición sigue siendo poco específico.

Aquí cabe la pregunta ¿se puede medir a partir de las definiciones anteriores?

Siguiendo con el ejemplo del coche, se puede medir de solo tener la definición de la RAE (Julio 2024): "automóvil destinado al transporte de personas con capacidad no superior a 7 plazas", sin embargo, una definición de "Mesa" de la misma fuente RAE (julio 2024) es: "Mueble compuesto de un tablero horizontal liso y sostenido a la altura conveniente, generalmente por una o varias patas, para diferentes usos, como escribir, comer, etcétera."

Si se requiere medir tanto el Coche como la Mesa, sería más fácil medir la Mesa porque la definición describe más elementos que la definición de Coche. Aquí Mesa ya define un Tablero horizontal y Patas. Se podría iniciar como se describe en la tabla 2.4

Para Coche es necesario distinguir el significado de Transporte, Capacidad y Plazas. Suponiendo que se entiende que Automóvil es sinónimo de Coche.

Tabla 2.4. Definición de los elementos de Mesa

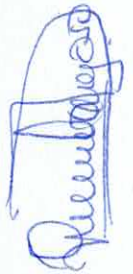
Mesa	Tablero horizontal	Ancho
		Largo
		Grosor
		Color
		Material

Estrategias para medir productos de software

Tabla 2.4. Definición de los elementos de Mesa (continuación)

Mesa	Pata	Altura
		Ancho
		Forma
		Color
		Material

Cuando se analiza la Tabla 2.4., se pueden distinguir que elementos como **Ancho**, **Largo**, **Altura**, y **Grosor** (suponiendo que es una mesa cuadrada o rectangular) en una mesa, estos elementos son directamente calculables y ya se pueden emplear para generar una medida de calidad para la mesa. En cambio, para el automóvil no es tan directa la forma de medir algún atributo porque no se ha definido.



Estrategias para medir productos de software

2.3 Actividades

En esta sección se indican algunas actividades encaminadas a reforzar el entendimiento del contenido de este capítulo. Intente realizar todas las actividades.

- 1.- Explique qué significa nivel de madurez.
- 2.- Explique que entiende por proceso de desarrollo de software.
- 3.- Describa cuales modelos de proceso de desarrollo de software conoce.
- 4.- Explique cuales son los productos del desarrollo de software más importantes para usted.
- 5.- Haga una lista de productos del proceso de desarrollo de software que usa.
- 6.- Realice un diagrama de clases en donde se presentan los elementos de la Tabla 2.2.
- 7.- Intente realizar una definición más completa de un diagrama de clases, usando los elementos de la Tabla 2.2.
- 8.- Realice una definición de algún producto final que considere importante, tal como los que se presentan en la Tabla 2.
- 9.- Realice una tabla o diagrama con los elementos de un manual de usuario
- 10.- Realice una definición de un componente de código del tipo que prefiera. Puede ser clase, servicio Web o microservicio.

46

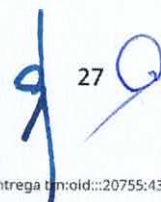
26

Bibliografía

- Norma Mexicana para el proceso de desarrollo de software -NMX-059-1-2005.
- Modelo de procesos para la industria del software MOPROSOFT.
- ISO/IEC 9126.
- Software quality attributes and trade-offs, Lars Lundberg, Michael Mattsson, Claes Wohlin (eds), Blekinge Institute of Technology June 2005.

Referencias

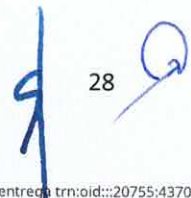
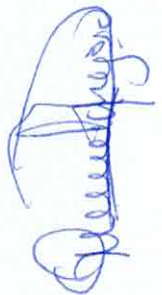
- 1.- Kirriemuir y A. Mcfarlane, Literature Review in Games and Learning, 2004. [En línea]. Available: <https://telearn.archives-ouvertes.fr/hal-00190453>.
- 2.- Medina Flores, R., & Morales Gamboa, R. (2015). Usability Evaluation by Experts of a Learning Management System. IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, 10(4), 197-203.
- 3.- Hujainah, F. (2013). Usability Guidelines of Mobile Learning Application. *Journal of Information Systems Research and Innovation*.
- 4.- A. Correa Puche. UF1460 - Composición de textos en productos gráficos. Edición 5.0, España, E-learning, 2015.
- 5.- Peña, J., Vidal, Ma. C (2000). Guía Visual de diseño de páginas Web. Primera reimpresión, Ediciones Anaya Multimedia, España.
- 6.- Darroch, I., Goodman, J., Brewster, S.A., and Gray, P.D.G. (2005) The effect of age and font size on reading text on handheld computers. *Lecture Notes in Computer Science*, 3585, pp. 253-266. (doi:10.1007/11555261_23)
- 7.- G. Alexander, K. Olga. (2012). Supervised Learning Algorithms Evaluation on Recognizing Semantic Types of Spanish Verb-Noun Collocations. *Computación y Sistemas*, Vol. 16(3), 297-308.
- 8.- J. W. Rebecca, D. Kristi, F. V. Scharff Lauren- (2005)- Effects of Typeface and Font Size on legibility for Children. *American Journal of Psychological research*, Vol.1(1) 86-102.
- 9.- H. Roman, K. Daniel (2011). Information competence and evolution of e-learning text with the fog index. *IEEE 11*, 390-394. DOI:978-1-4577-1747/11.
- 10.- D. Nancy (2008) slide:ology The Art and Science of Creating great presentations. Canada. O'Reilly Media. ISBN-13: 978-0-596-52234-6.
- 11.- AMPAS, Guía Educacional Diseño de Vestuario de Vestuario, 2014. Available: <https://www.oscars.org/sites/oscars/files/teachersguide-costumedesign-spa-2015.pdf>.


27

Capítulo 3

Definir para medir el atributo de calidad que requiere medirse

Objetivo: El objetivo de este capítulo es mostrar la forma en que se puede mapear a un producto de software con el atributo de calidad que requiere medirse.



Estrategias para medir productos de software

3.1 Modelos de Calidad

Existen varios modelos que pueden ser usados como referencia para medir la calidad de productos de software. Entre los más conocidos, aunque recientes se encuentran los modelos ISO/IEC 9126 que evolucionó a ISO 25000, el modelo de McCall, el modelo de Boehm, el modelo de FURPS, el modelo de Dromey, etc. [1]. Todos estos modelos distinguen características también conocidas como atributos de los productos de software y en su mayoría coinciden en muchos de los atributos.

La tabla 1 presenta los atributos que corresponden al ISO/IEC 9126 – Calidad Interna y Externa. En la primera columna se presentan los super atributos o atributos principales y también una definición del atributo. La segunda columna presenta los subatributos que se considera corresponden al super atributo. Es importante mencionar que los atributos a su vez están definidos básicamente en los estándares y por lo tanto se pueden emplear las definiciones cuando se requiera. De acuerdo a [1] este modelo surgió como base de las investigaciones de McCall y Boehm.

Tabla 1. Atributos de Calidad en ISO 9126

Super atributos	Sub atributos
Funcionalidad La capacidad del producto de software para proporcionar funciones que alcancen las necesidades implícitas cuando el software es usado en condiciones específicas.	Adecuabilidad, exactitud, interoperabilidad, seguridad, cumplimiento funcional
Confiabilidad La capacidad del producto de software para mantener un nivel específico de desempeño cuando se usa en condiciones específicas.	Madurez, tolerancia a fallas, recuperabilidad, cumplimiento de la confiabilidad
Usabilidad La capacidad del producto de software para ser entendido, aprendido, usado y atractivo para el usuario, cuando se usa en condiciones específicas.	Entendibilidad, aprendibilidad, operabilidad, capacidad de atraer, cumplimiento de la usabilidad
Eficiencia La capacidad del producto de software para proporcionar un desempeño apropiado, relativo a la cantidad de recursos usados, bajo las condiciones establecidas.	Comportamiento del tiempo, utilización de recursos, cumplimiento de la eficiencia

Estrategias para medir productos de software

Tabla 1. Atributos de Calidad en ISO 9126 (continuación)

Mantenibilidad La capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras, adaptaciones de software a cambios en el ambiente y en las especificaciones funcionales y de requerimientos	Analizabilidad, cambiabilidad, estabilidad, habilidad para probar, cumplimiento de la mantenibilidad
Portabilidad La capacidad del producto de software para ser transferido desde un ambiente a otro.	Adaptabilidad, instalabilidad, co-existencia, habilidad de reemplazo, cumplimiento de la portabilidad

Como se describe en la Tabla 1, el modelo define de manera general a los atributos, lo que representa una ventaja ya que podemos personalizarlos de acuerdo al producto de software que se trate. Desde luego no es lo mismo medir el atributo usabilidad de la interfaz de un sistema Web, que usabilidad de un código o de un diagrama de clases. Cada uno de estos elementos requiere ser definido primeramente como se describe en el capítulo 2 y posteriormente se deberá analizar si la definición de usable o usabilidad propuesta por el modelo que se decida aplicar, permite distinguir que elementos del producto de software pueden estar relacionados en la definición. Los modelos de McCall, Boehm and Dromey se describen en la Tabla2, Tabla 3 y Tabla 4 respectivamente.

Tabla 2. Atributos de Calidad en modelo del McCall

Super atributos	Sub atributos
Correctitud	Seguibilidad, completitud, consistencia
Confiabilidad	Consistencia, exactitud, tolerancia a fallas
Eficiencia	Eficiencia en ejecución, eficiencia de almacenamiento
Integridad	Control de acceso, auditoria de acceso
Usabilidad	Operabilidad, entrenamiento, comunicatividad
Mantenibilidad	Simplicidad, concisión, autodescriptividad, modularidad
Capacidad de prueba	Simplicidad, instrumentación, autodescriptividad, modularidad
Flexibilidad	Autodescriptividad, expandabilidad, generalidad
Portabilidad	Autodescriptividad, independencia de software, independencia de máquina
Reusabilidad	Autodescriptividad, generalidad, modularidad, independencia de software, independencia de máquina
Interoperabilidad	Modularidad, comunicación común, datos comunes

Estrategias para medir productos de software

Tabla 3. Atributos de Calidad de acuerdo a Boehm

Super atributos	Sub atributos
Portabilidad	Independencia de dispositivo, autocontención
Confiabilidad	Autocontención, exactitud, completitud, robustez/integridad, consistencia
Eficiencia	Responsabilidad, eficiencia de dispositivo, accesibilidad
Ergonomía	Robustez/integridad, accesibilidad, comunicatividad
Capacidad de prueba	Responsabilidad, comunicatividad, autodescriptividad, estructuración
Entendibilidad	Consistencia, estructuración, concisión, legibilidad
Modificabilidad	Estructuración, aumentabilidad

Tabla 4. Atributos de Calidad de acuerdo a Dromey

Super atributos	Subatributos
Correctitud	Funcionalidad, confiabilidad
Internas	Mantenibilidad, eficiencia, confiabilidad
Contextual	Mantenibilidad, reusabilidad, portabilidad, confiabilidad
Descriptividad	Mantenibilidad, reusabilidad, portabilidad, usabilidad

En la Tabla 1, Tabla 2, Tabla 3 y Tabla 4 se pueden encontrar super atributos que coinciden en los modelos como son portabilidad, reusabilidad, confiabilidad, correctitud, eficiencia, capacidad de prueba, mantenibilidad y usabilidad. Como resultado de que los super atributos coincidan, algunos de los sub atributos también se repiten. Así mismo, es válido que algunos sub atributos se relacionen con más de un super atributo.

Los modelos antes descritos resultan de gran ayuda cuando se requiere determinar la calidad de los productos de software y cada vez más existen estudios que permiten aprender otros atributos de calidad, tal como el diseño centrado en personas [ISO 9241-210:2019]. Para propósito de este capítulo se consideran los modelos de las tablas aquí presentadas.

3.2 Atributo de calidad y producto de software

Como se describe en la sección 3.1 de este capítulo, no es lo mismo medir un atributo en un producto que medir ese atributo en otro producto. Si se logra definir un producto como se establece en el capítulo 2, por ejemplo, los productos de la Tabla 2, Video juego, Sistema Gestor de Aprendizaje, Aplicación móvil, Editor de programación y Página Web, será necesario definir el atributo que se le quiere medir. La selección de cual atributo medir depende algunas veces de atender a un requerimiento específico, por ejemplo, el atributo eficiencia que es ampliamente requerido cuando se desarrolla un software. Hay

Estrategias para medir productos de software

otros atributos que se seleccionan con base en una intención de mejora o camino a la mejora continua.

Cualquiera que sea nuestro interés, es importante entender los atributos de calidad. El propósito de este libro es indicar como podemos llegar a medir lo que nos interese en el ámbito de productos de software y de atributos de calidad.

En la sección “Seleccionar atributo de calidad” siguiente, se presentan algunos ejemplos de cómo relacionamos las características que definimos en productos de software con los atributos de calidad. En particular se describe como medir el mismo atributo en varios productos. La forma de relacionar las características de los productos con los atributos de calidad es muy similar a la forma de definir productos de software en el capítulo 2. El proceso se describe en la Figura 3.1.

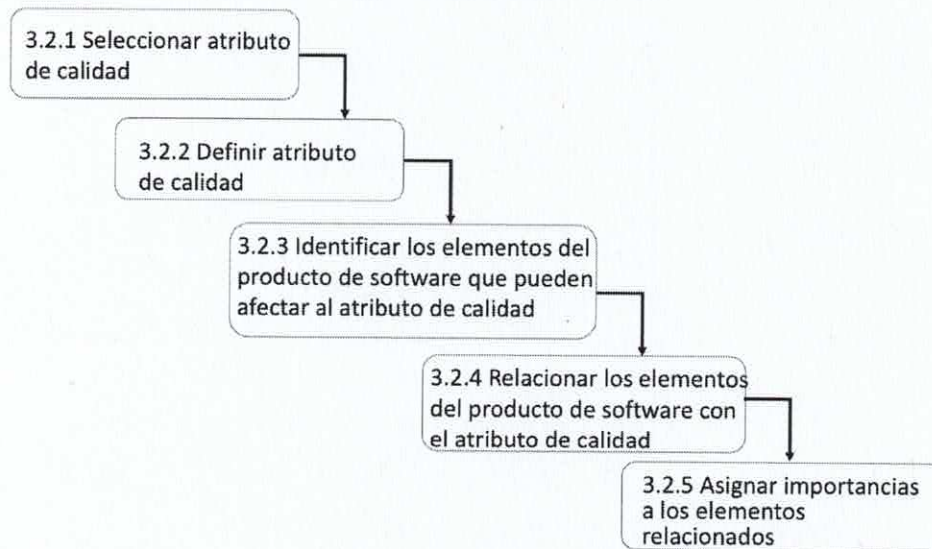


Figura 3.1 Proceso para relacionar atributos de calidad a productos de software

3.2.1 Seleccionar atributo de calidad

Para ejemplificar el proceso, en este capítulo se selecciona el atributo “Usabilidad”. Aquí se emplea el atributo de usabilidad porque es un atributo del que cada vez más se requiere tener el cumplimiento.

3.2.2 Definir atributo de calidad

Para la definición del atributo podemos buscar en diccionarios, sin embargo, la mayoría de los modelos de atributos de calidad lo definen. De tal manera que, para este ejemplo, la definición de usabilidad se puede considerar como lo establece el estándar ISO/IEC 9126. “Usabilidad es la capacidad del producto de software para ser entendido, aprendido, usado y atractivo para el usuario, cuando se usa en condiciones específicas”. De la definición también se desprenden subatributos como *entendido*, *aprendido*, *usado* y *atractivo*. Estos subatributos requieren a su vez ser definidos. Si el atributo es muy conocido o entendido por los desarrolladores de software, es posible que puedan ser

relacionados directamente con los elementos de los productos de software. En caso contrario es necesario buscar esa relación en la literatura, para esto existen muchos trabajos de investigación que nos pueden apoyar a entender el subatributo y que son tan explícitos en su definición que nos permiten relacionar directamente el subatributo con los elementos del producto. Por ejemplo, el caso de [2] que define la comprensibilidad como la facilidad, con la cual los usuarios pueden entender el contenido presentado en dispositivos móviles. En esta definición podemos ver que entendibilidad y comprensibilidad se usan de manera indistinta y por lo tanto podemos usar esta definición para establecer la relación entre los atributos. Además, esta definición ya establece el tipo de producto a medir, es decir, aplicaciones móviles.

3.2.3 Identificar los elementos del producto de software que pueden afectar al atributo de calidad

De acuerdo con [2], el entendimiento de algún objeto está relacionado con la forma en que se presenta y se organiza la información. Así que corresponde utilizar los elementos que dan la forma y orden a la información que se presenta en el producto, por lo tanto, se puede relacionar la Usabilidad con Entendibilidad y Organización de la información o contenido. Así se tiene una mayor definición del atributo de usabilidad que nos acerca a un producto de software, y ahora se puede ya iniciar con la siguiente parte del proceso que es relacionar los elementos del producto de software con el atributo de calidad.

3.2.3.1 Relacionar los elementos del producto de software con el atributo de calidad

Después de completar las actividades 3.2.1, 3.2.2 y 3.2.3, se puede iniciar la relación de elementos de los productos con el atributo o atributos de calidad de interés. Una manera de ayudar en esta actividad, es emplear una forma gráfica que permite establecer claramente la relación, por ejemplo, un árbol que inicia por la izquierda con el atributo de calidad principal y va creciendo hacia la derecha según se identifiquen los subatributos y los elementos del producto que los afectan. El ejemplo de cómo se puede relacionar la Usabilidad en aplicaciones móviles se presenta en la Figura 3.2

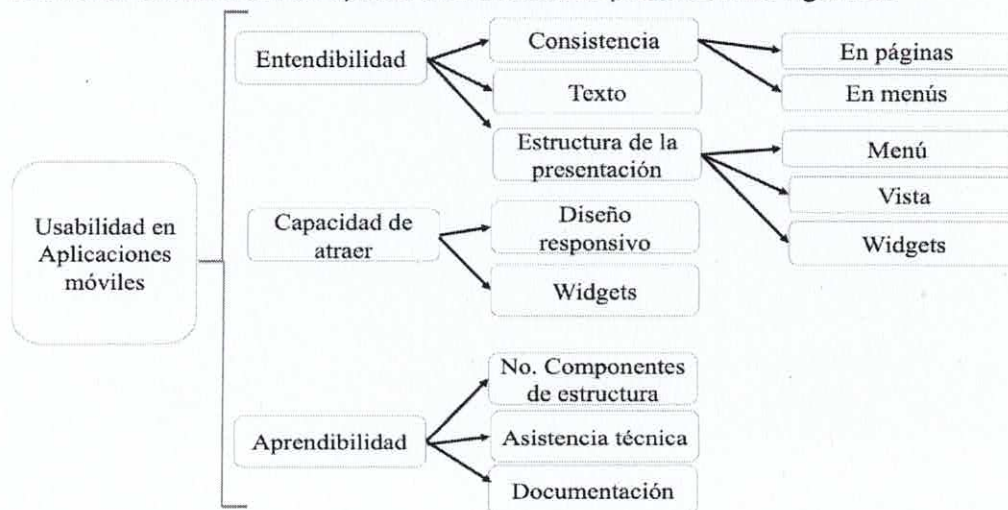


Figura 3.2 Árbol de Usabilidad en Aplicaciones Móviles

En el caso de la Figura 3.2 la Usabilidad se define por tres subatributos que son Entendibilidad, Capacidad de atraer o Atractivo y Aprendibilidad. En la tabla 2 del capítulo 2, se definieron algunos elementos de las aplicaciones móviles que corresponden a las Vistas, Organización, Widgets y Menú, se pueden relacionar estos elementos con los subatributos de calidad como se presenta en la Figura 3.2.

La Figura 3.2 no es exhaustiva, es decir, existen muchos más elementos en una aplicación móvil por lo que es necesario realizar un análisis completo de los elementos que la conforman. El propósito de la figura es servir de guía para indicar el camino de cómo podemos medir atributos de calidad. También, es importante puntualizar que se trata de atributos en el estándar ISO /IEC 9126, pero que también existen otros subatributos de usabilidad presentados en otros modelos, que en el caso de que sean de utilidad, siempre podrán integrarse a una definición del superatributo que se requiere medir, en el caso del ejemplo de la Figura 3.3 se trata del atributo Usabilidad.

En varios modelos se han presentado subatributos que se relacionan con el mismo super atributo, por lo tanto, un mismo elemento de un producto de software puede participar para evaluar diferentes atributos o subatributos de calidad, siempre cuidando no entrar en contradicción. Por ejemplo, en la Figura 3.2, Widgets se relacionan con 2 subatributos: Entendibilidad y Capacidad de atraer y que se presentan en el estándar ISO/IEC 9126. En la figura 3.3 se presenta una descripción de los elementos de Widget donde los elementos Tamaño y Color, representados en los recuadros color azul, son algunos de los elementos más específicos del producto Aplicación móvil a los que ya se les puede asignar valores como se describe en la sección 2.2 Asignación de valores.

En el modelo ISO 9241-11:2018, también se encuentra el atributo Usabilidad y en su definición se incluyen los subatributos Efectividad, Eficiencia y Satisfacción, y estos se pueden integrar en la medición de la Usabilidad de los productos de Software, algunos ejemplos se presentan en las figuras 3.4, 3.5 y 3.6

En la Figura 3.4, se puede ver la integración del subatributo Satisfacción, que no se encuentra directamente en el estándar de referencia principal, pero que es un atributo que se reconoce como importante en la Usabilidad de algún producto. El Atributo Satisfacción se asocia con Diseño responsivo porque al presentar una interfaz y que el usuario la pueda ver completa le puede generar Satisfacción. De la misma manera, la navegabilidad puede o no generar satisfacción en el usuario, sin embargo, hacer que la navegación en una interfaz facilite a un usuario lograr sus tareas puede generar satisfacción. De tal manera que es otro atributo importante y que se encuentra asociado a experiencias del usuario.

Estrategias para medir productos de software

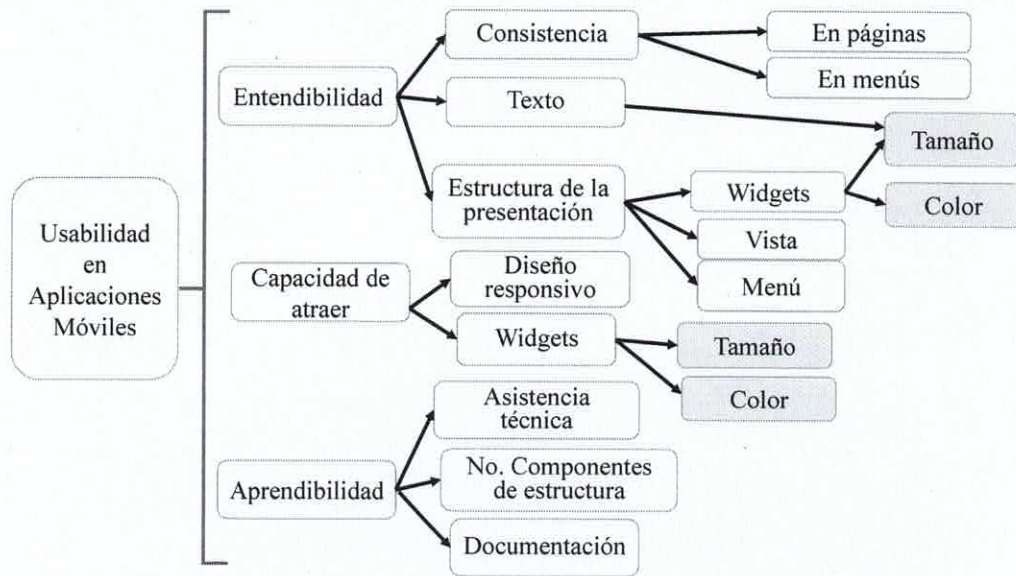


Figura 3.3 El elemento Widget se relaciona con Entendibilidad y Capacidad para atraer.

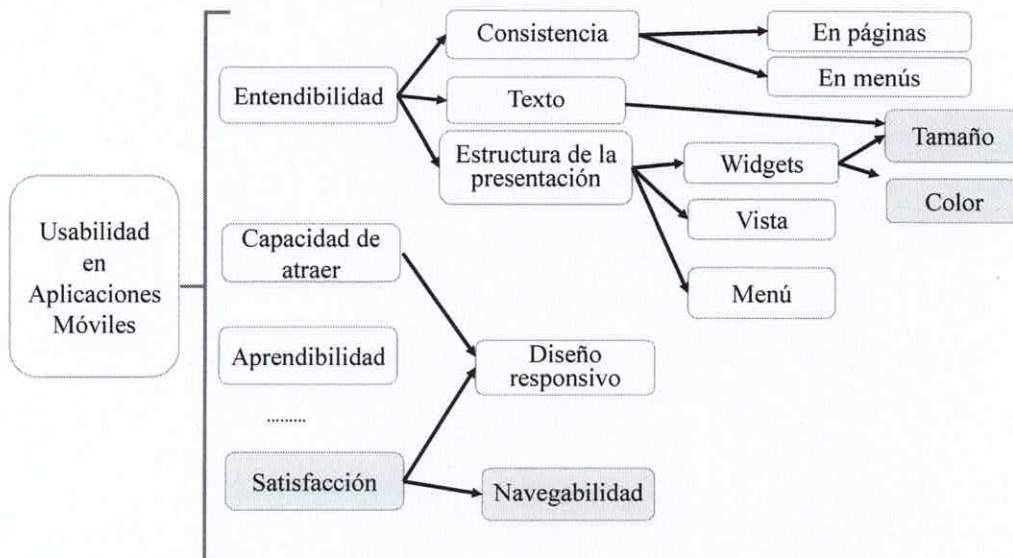


Figura 3.4. Integración del atributo Satisfacción en la Usabilidad de Aplicaciones Móviles

Los ejemplos que se describen en las Figuras 3.3 a la 3.9, describen la Usabilidad en diferentes productos de software, a primera vista se puede pensar que representan lo mismo. Esto sucede porque algunos de los elementos de los productos son comunes como Texto, Áreas de trabajo, Widgets, Documentación, Colores, etc. que en cada tipo de producto de software existen de acuerdo a su intención, por ejemplo, no es lo mismo tener un Widget para una aplicación móvil que para un videojuego porque tienen

Estrategias para medir productos de software

diferente intención, de tal manera que la diferencia se establece cuando se le asignan valores. Por ejemplo, para un editor de programación, un Widget no necesariamente se requiere que sea tan grande como para un videojuego, es decir, mientras que para un editor de programación se puede requerir para informar sobre las funciones del editor, para un videojuego se puede requerir de mayor tamaño porque es necesario que llame la atención del jugador. Así que sus mejores valores son los que realmente van a ser diferentes, por eso es necesaria la definición del producto y la asignación de sus mejores o peores valores.

Un Widget [3] es un elemento de un producto de software que principalmente brinda información, algunos conocidos son los botones, radioBoton, listas, menús, checkboxes, cuadros de dialogo, ventanas emergentes etc.

En la Figura 3.3 Tamaño y Color están relacionados con Widgets y en la Figura 3.4 también están relacionados con Widgets y con Indicadores de progreso los que finalmente pueden caer en categoría de Widget, pero al realizar el análisis del videojuego, los Indicadores de progreso resultan ser elementos importantes para la Entendibilidad así que estos se relacionan directamente con el atributo.

En la Figura 3.3 se encuentra también el subatributo Consistencia, que tiene que ver con que todos los elementos asociados a entendibilidad, sean iguales en todos los espacios del producto de software, por ejemplo, los menús deben ser todos iguales, el color del Texto debe ser el mismo para todos los elementos iguales, por ejemplo, para todos los menús el mismo Tamaño y Color de Texto, o para títulos principales el mismo Tamaño y Color, y para subtítulos de igual manera, etc. Consistencia puede ser parte de todos los productos de software, aunque es posible que su definición pueda variar dependiendo de que tan diferentes sean los productos [4]. El aspecto, la sensación y el comportamiento de la interfaz deben ser consistentes en toda la aplicación. Si la interfaz de usuario es consistente, reduce la cantidad de incertidumbre que enfrenta el usuario cuando utiliza la interfaz.

Todos los ejemplos que se describen en esta sección son meramente explicativos y los árboles de calidad de las Figuras 3.3 a la 3.9 pueden ser modificados principalmente por la experiencia de las personas encargadas de calidad. De igual manera, los nombres de los elementos de los productos pueden ser tropicalizados para evitar pensar que estos serán exactamente iguales para todos los productos.

Estrategias para medir productos de software

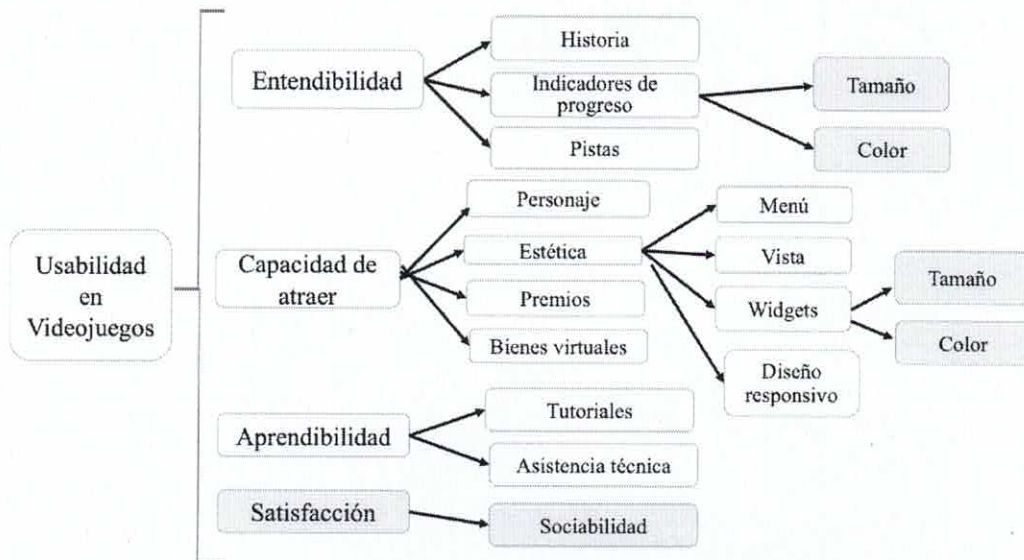


Figura 3.5 Usabilidad en videojuegos.

En el caso del atributo Entendibilidad, se puede ver que para Aplicaciones móviles, Sistemas de gestión de aprendizaje, Editores de programación, Página Web y Sistemas de videoconferencias, la Entendibilidad está fuertemente relacionada con la forma en que se presenta el producto y por eso la organización y la estructura de su contenido tiene asociados elementos similares, en cambio para un videojuego que se describe en la Figura 3.5, esa organización tiene que ver con la Historia del videojuego, pero también considera un elemento importante que debe existir para entender el avance en el juego y que son los indicadores de progreso. De igual manera las pistas se pueden considerar relevantes para poder entender el avance en el juego.

A pesar de que un videojuego es un producto muy diferente de los otros productos aquí presentados, también tiene Menú, Vista y Widgets asociados al atributo Capacidad de atraer porque su intención es precisamente tener todos los elementos necesarios para que el videojuego sea atractivo.

En el caso del atributo Satisfacción para un videojuego este se representa por un subatributo Sociabilidad [5] debido a que el tener la posibilidad de realizar el juego con otro jugador puede generar una experiencia positiva y por lo tanto satisfacción en el usuario.

Estrategias para medir productos de software

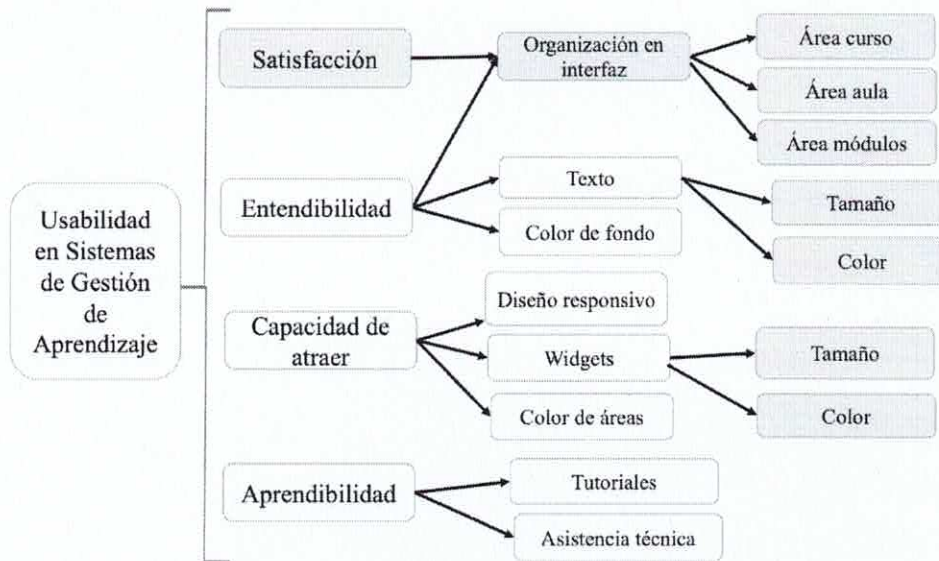


Figura 3.6 Usabilidad en sistemas de gestión de aprendizaje.

Otro elemento general que aporta a la Usabilidad, en particular porque facilita el entendimiento, es el Texto. En la Figura 3.5 a la 3.9 el Texto se hace presente porque dependiendo del valor de su subatributo Tamaño, un usuario podrá leer lo que se presente y por lo tanto entender lo que el sistema indica. Es posible que el tamaño del texto puede variar entre los sistemas de Aplicaciones móviles y páginas Web y los de Gestión de aprendizaje, Editores de programación y Sistemas de video conferencias.

Satisfacción en la Figura 3.6 se asocia con la forma en que se encuentra la información disponible en un Sistema gestor de aprendizaje por lo que se consideran las secciones generales para encontrar la información como son las de un Aula, Curso o Módulo.

Diseño responsivo [6] es otro elemento que se aparece en muchos productos de software porque permite generar una buena experiencia de usuario al permitir ver la cantidad de información que se presenta en una interfaz, y por lo tanto resulta atractivo [7] para los usuarios cuando el dispositivo que se emplea ofrece información completa, o presenta la información en pantalla completa, es decir, no corta la interfaz en pantalla.

[Firma manuscrita]

[Firma manuscrita]

Estrategias para medir productos de software



Figure 3.7 Usabilidad en editores de programación.

Respecto al atributo de Aprendibilidad que también se presentan en todos los ejemplos de este capítulo, si la definición se refiere a elementos que hacen que el producto de software de que se trate tenga elementos que hacen fácil su aprendizaje o que sirvan para aprenderlo, entonces se pueden incluir tanto la Asistencia técnica como los Tutoriales.

Dependiendo del alcance de las mediciones, se puede evaluar si existen o no, es decir, una evaluación binaria de 1 o cero, o extender las mediciones a la definición de cada uno. El alcance de las mediciones está en función de la madurez del proceso de desarrollo del producto.

Un nivel avanzado de madurez implica que la evaluación no sea binaria y que los encargados del desarrollo del producto consideren las mejores prácticas para realizar tutoriales e implementar la asistencia técnica. Por ejemplo, un Tutorial debería ser consistente con la interfaz del producto que se trate, que explique cada elemento del producto, que incluya un apartado de objetivo del tutorial, que incluya el número de versión del tutorial y a que versión del producto corresponda, pero además, deberá considerar los colores que se empleen, manejar los contrastes entre el color de fondo con los colores del texto y de las figuras, deberá considerar el Font y el Tamaño del Font, un Índice y Números de páginas, etc.



9

Estrategias para medir productos de software



Figura 3.8. Usabilidad en páginas Web.

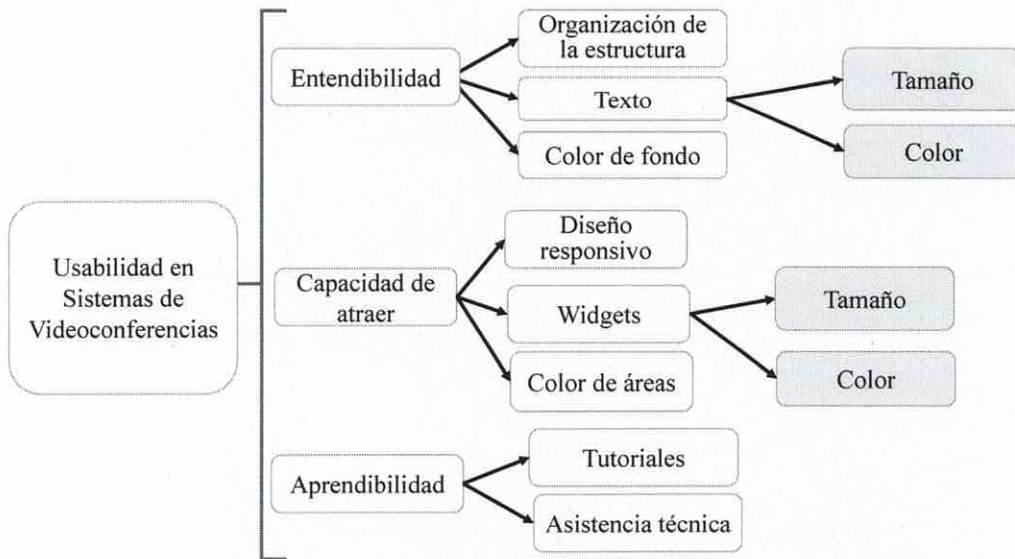


Figura 3.9 Usabilidad en sistemas de videoconferencias.

El atributo Color de áreas se hace presente también en los ejemplos de la Figura 3.5 a la 3.8. En estos, el color de áreas se relaciona con la Capacidad de atraer debido a que dependiendo de para que se requieren las áreas, es decir, hay áreas para escribir como es el caso de Editores de programación en donde un color orientado al aprendizaje sea

Estrategias para medir productos de software

el mejor valor, por ejemplo, el color azul. Mientras que para páginas Web o Aplicaciones móviles el mejor color puede ser blanco. Otra vez, los valores que Color de áreas puede tomar dependen en gran medida de la intención del producto.

La actividad de relacionar elementos de los productos de software con los atributos de los modelos de calidad es un proceso de análisis iterativo que se detiene cuando identificamos elementos tan específicos como Tamaño y Color, esto se sabe cuándo a esos elementos se les pueden asignar valores específicos, por ejemplo, a Tamaño se le puede asignar un valor de 15 * 15pp y a Color se le puede asignar el valor de "gris", asumiendo que esos puedan ser los mejores valores.

Lo importante en esta actividad es llegar a relacionar elementos tan específicos como Tamaño y Color a todos los subatributos indicados para medir el super atributo seleccionado.

3.2.4 Asignar importancias a los elementos relacionados

Las actividades de la 3.2.1 a la 3.2.4. son las que mayor desafío representan porque se requiere de capacidad de análisis para entender cuales elementos de los productos se pueden relacionar con cuales atributos o subatributos de los modelos de calidad. Una vez que ya se cumple con establecer las relaciones de elementos y atributos de software, la actividad final del proceso es básicamente asignar importancias a todos los subatributos y elementos que forman parte de un árbol de calidad, así como identificar sus mejores o peores valores como se describe en la sección 2.2 Asignación de valores.

La asignación de la importancia se establece en términos de lo que está en el árbol. Para llevar a cabo esta actividad es necesario analizar si todos los elementos del árbol tienen la misma importancia en la medición del atributo de calidad que se requiere medir. En el caso de que un subatributo es más importante que otro, o un elemento es más importante que otro, se le asignará mayor importancia. En caso de que no se tenga mayor conocimiento sobre el elemento, la sugerencia es asignar la misma importancia a todos los elementos del árbol.

La Figura 3.9 presenta el árbol de la Figura 3.3 pero incluye las importancias en todas las hojas del árbol. En el ejemplo todas las hojas tienen el mismo valor de importancia, esto se visualiza desde los tres principales subatributos de Usabilidad, en donde Entendibilidad, Capacidad de atraer y Aprendibilidad tienen una importancia de un tercio cada una. Entendibilidad tiene tres subatributos y por lo tanto cada uno vale un tercio de Entendibilidad y se representa por 33.33%. Consistencia tiene solo dos atributos y también cada uno vale 50%, es decir, cada uno vale la mitad del valor de Consistencia, en cambio Menú, Vista y Widgets valen un tercio del valor de presentación porque son tres elementos. Cuando se llegan a asignar importancias en un árbol de calidad, siempre es importante realizar una normalización como el caso de la Figura 3.9.

Existen atributos de calidad para los que solamente pueda haber un elemento del producto de software. En ese caso se considera que ese elemento tenga el total del valor de la importancia del subatributo con el que se relaciona. También puede haber elementos binarios en donde su mejor valor es si el elemento existe o cero si el elemento no existe en el producto que se está evaluando, de ahí viene la necesidad de definir a un producto de software lo más preciso que se pueda. En el caso de la Figura 3.9 Asistencia técnica y Documentación son elementos que pueden ser valorados desde el punto de vista de "si existen", entonces su importancia es el porcentaje asignado, sin embargo,

Estrategias para medir productos de software

es necesario analizar si cada uno de estos no requiere ser definido con mayor precisión. Esas son decisiones del encargado de evaluar los atributos de calidad.

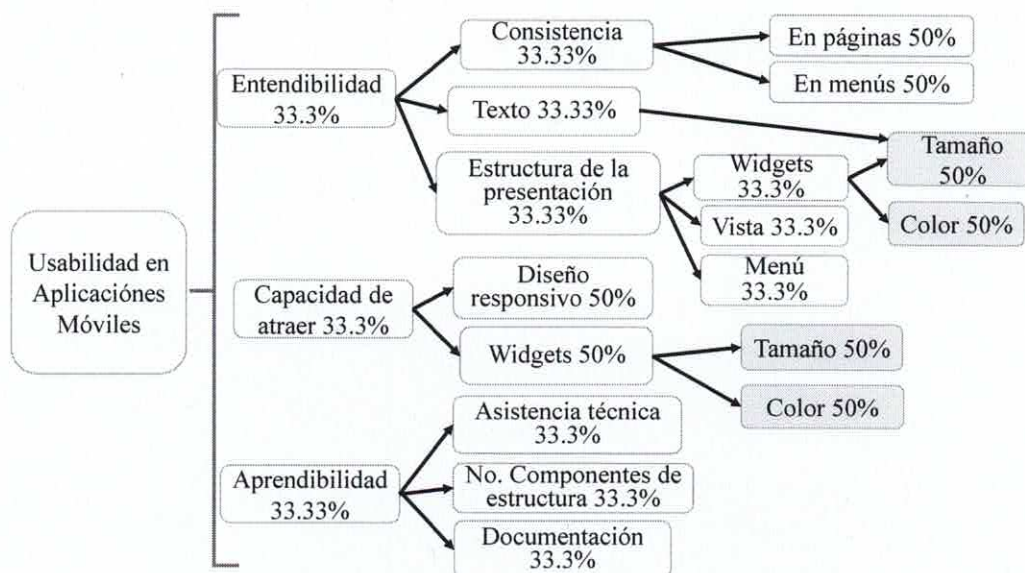


Figura 3.9 Importancias en el árbol de calidad

3.2.5 El valor de las importancias

Las importancias son valores difíciles de encontrar en la literatura porque se requieren muchos estudios para determinar que atributos o elementos son más o menos importantes, esto implicaría hacer evaluaciones por cada atributo que se presenta en los modelos de calidad asociados a un producto específico de software para que sean publicados y los desarrolladores utilicen esos estudios. Los estudios que se encuentran fácilmente son aquellos estudios ergonómicos sobre accesibilidad y legibilidad, aunque no están relacionados con productos específicos.

Cuando no se tienen importancias conocidas, la recomendación es ponderar en partes iguales todos los atributos y elementos e ir generando un histórico de las importancias asignadas, luego con base en ese histórico y observaciones de los desarrolladores ir cambiándolas para los nuevos productos de software.

Desafortunadamente, el generar un histórico no es algo que a los desarrolladores les atrae, mucho menos si no tienen algún nivel de madurez que establezca en su proceso de desarrollo el registro de datos de sus productos.

Una vez establecidas las importancias para todo el árbol de calidad, se puede ya cuantificar el valor del atributo principal o super atributo.

complementar

42

3.2.5.1 Mejor y peor valor

En la sección 2.2 se da una breve explicación de la asignación de valores que los elementos de un producto de software pueden tomar. Ahora es importante establecer la relación con los atributos de calidad que se requieren medir.

Después de relacionar todos los elementos del producto de software con los atributos y subatributos de calidad, y establecer sus importancias, es necesario identificar que valores pueden tomar los elementos más específicos descritos en los árboles de relación como el de la Figura 3.9. por ejemplo, en el caso de Widget relacionado a la Estructura de la Presentación, el mejor y el peor tamaño pueden ser como lo indica la tabla 5.

En la Tabla 5, los valores de la primera columna son calculables y en la segunda se indica como se pueden considerar para propósitos de realizar el cálculo específico. Es decir, un mejor valor siempre tendrá el máximo valor de importancia. por ejemplo, si se supone que un Widget tiene un tamaño de 15p*15p, este tendrá un valor del 100% del valor de importancia que se le asignó, es decir, el 100% del 33.33%. Si los valores son menores o mayores a 15p*15p la importancia generará un cálculo con un valor de 0, porque se asume que, si el tamaño de los widgets es diferente, entonces es de mala calidad, de tal manera que podemos ir cuantificando el valor de calidad respecto al tamaño de los widgets. Para poder asignar estos mejores o mejores valores existen estudios previos sobre ergonomía que indican los valores que algunos de los elementos presentados en las Figuras de la 3.3 a la 3.8 pueden tomar, muchos de los estudios se encuentran muy fácilmente, o son del dominio común tal como el color de fondo de una pantalla, que se sabe que el peor valor es el color rojo para el caso de aplicaciones ya sea móviles o de escritorio. Lo recomendable es buscar los estudios adecuados a lo que se requiere medir porque no es lo mismo un tamaño de texto en un .doc que en un .ppt, o el color rojo en un videojuego que en un sistema de información.

Tabla 5. Mejores y peores valores de un Widget en un sistema de videoconferencia.

Tamaño del Widget	Clasificación del valor	Porcentaje del valor de importancia	Importancia del elemento =33.33%
Tamaño <15*15	Peor valor	0%	0% del 33.33%= 0
Tamaño =15*15p	Mejor valor	100%	100% del 33.33%= 33.33
Tamaño >15*15p	Peor valor	0%	0% del 33.33%=0

Otro ejemplo muy común es el que se refiere a tamaño de Texto. En la Tabla 6 se muestra un rango de valores que puede tomar un Texto en un sistema editor de programación. Lo que la tabla indica es que cuando se escribe con un texto tamaño 6p o menor, se le debe asignar un porcentaje del valor de importancia que corresponde a un cero porque cuando se emplea un tamaño 6p es demasiado pequeño para poder leer un texto y por lo tanto genera dificultad en para ser entendido. Si el texto aumenta su valor a 12p, 14p y 16p, se le puede asignar un valor de importancia del 100 por ciento, porque ese es el tamaño en que es fácil de leer y por lo tanto entender el texto (en este

Estrategias para medir productos de software

caso el texto tiene un valor de importancia de 33% porque Entendibilidad tiene asociados tres elementos del sistema de videoconferencia). Si el tamaño es más grande de 16p, va perdiendo valor de importancia porque se aleja de los valores sugeridos para entender el texto.

Tabla 6. Mejores y peores valores para un Texto en un sistema de videoconferencia

Tamaño de texto	Clasificación del valor	Porcentaje del valor de importancia	Importancia del elemento =33.33%
6 p o menor	Peor valor	0%	0% del 33.33%= 0
8 p	Valor intermedio	60%	60% del 33.33%= 19.99
10 p	Valor intermedio	80%	80% del 33.33%=26.66
12 p	Mejor valor	100%	100% del 33.33%=33.33
14 p	Mejor valor	80%	100% del 33.33%=33.33
16 p	Mejor valor	60%	100% del 33.33%=33.33
18 p o mayor	Peor valor	0%	0% del 33.33%=0

En la Tabla 7 se presenta otro ejemplo del dominio común, es el caso del tamaño de Texto en un archivo .ppt. Si se considera que el valor de importancia de un texto en un archivo .ppt es de .20 porque el atributo de entendibilidad tiene 5 elementos del archivo .ppt asociados, entonces cada uno de ellos tiene un valor de 20%. En este caso, todos los tamaños del texto dentro del rango 18-38p, se consideran como valores que aportan a la entendibilidad de una lámina ppt. Fuera de esos valores se considera que dificultan la entendibilidad y por lo tanto su valor es 0.

Tabla 7. Mejores y peores valores en texto en un archivo .ppt

Tamaño de Texto	Clasificación del valor	Porcentaje del valor de importancia	Importancia del elemento =20%
16 P o menor	Peor valor	0%	0% de 20%=0
18 p	Mejor Valor	100 %	100% de 20%=20
20 p	Mejor valor	100%	100% de 20%=20
22 p	Mejor valor	100%	100% de 20%=20
24 p	Mejor valor	100%	100% de 20%=20
26 p	Mejor valor	100%	100% de 20%=20
28 p	Mejor valor	100%	100% de 20%=20
40 p o mayor	Peor valor	0%	0% de 20%=0

Handwritten signature in blue ink.

Handwritten signature in blue ink.

3.3 Como medir

Existe una gran cantidad de publicaciones y libros sobre métricas de software y sobre mediciones de atributos de calidad. Las métricas van desde muy sencillas hasta métricas complejas que se alimentan de cálculos y algoritmos complejos. Lo recomendable es que las métricas sean lo más simple de tal manera que para la mayoría de las personas involucradas en el desarrollo de productos de software, el resultado de los valores que se obtienen sea entendible y significativo. Si bien se puede buscar una métrica ya establecida o conocida para medir algo, a veces no resulta sencillo usarla o involucra parámetros que no se conocen, es por eso que lo recomendable es que se emplee una forma agregada de cálculo de valores de los elementos como se presenta en la ecuación 1.

$$Usabilidad = \sum_1^n (importancia)_{elemento} \quad (ecuación 1)$$

En la ecuación 1, *Usabilidad* es el atributo que se ha estado usando para propósito del ejemplo en este capítulo y representa la sumatoria de los valores de las importancias, para cada atributo, es decir, los valores de la cuarta columna en la Tabla 7.

En el caso del ejemplo de la Figura 3.9 se deben especificar 11 tablas correspondientes a las 11 hojas que se encuentran en el extremo derecho del árbol. Es decir, una tabla por cada uno de las hojas y que indique sus mejores y peores valores y además los relacione con su valor de importancia. Por eso es que las importancias juegan un papel importante en la medición. Es desde la asignación de la importancia que casi se va obteniendo un valor de calidad.

Si bien el cálculo del valor de un atributo de calidad es un agregado, es importante considerar aquellos valores que tienen diferentes direcciones, por ejemplo, los que tienen un mejor valor indicado por 1 y los que tienen el mejor valor indicado por 0. En este caso es necesario convertir ya sea los valores 1 o los valores 0 para que indiquen la misma dirección en todos los elementos. Por ejemplo, si el mejor valor es 1, la ecuación 2 se usa para convertir la dirección de los valores cero cuando cero sea el mejor valor, para que el cálculo sea real.

$$Importancia = Valor de Elemento - 1 \quad (ecuación 2)$$

Si el objetivo de la medición tiene que ver con el comportamiento de algún elemento y sus valores, y su interacción con otros elementos, es necesario buscar alguna otra ecuación donde se establezca la relación entre los elementos. Ahí se pudieran usar las técnicas de análisis de datos, tal como la correlación.

La Tabla 8 presenta un ejemplo de cómo se puede documentar y organizar la medición de algún o algunos atributos de calidad. En la Tabla 8 el ejemplo es *Usabilidad* en una aplicación móvil. En la primera columna de la tabla se describe el atributo de calidad, la segunda columna describe el valor obtenido de la medición. En esta columna el valor de Entendibilidad es 94.43% porque no en todos sus subatributos se obtuvo el mejor valor. El valor puede ser numérico o textual como en el caso de la Consistencia en páginas y en Menús cuyo valor textual es un "sí". La columna Observaciones se puede

Estrategias para medir productos de software

utilizar para explicar o describir observaciones respecto a los valores obtenidos en la medición, esta columna es muy útil porque ahí se puede documentar cual es el defecto que le da el valor obtenido al producto que se está midiendo. En el ejemplo se indica que widgets no tienen el tamaño recomendado.

Tabla 8. Plantilla para documentar la medición de los atributos de calidad

USABILIDAD	Valor Obtenido	Mejor valor	Observaciones
Entendibilidad	94.43%	100	Revisar el tamaño de los widgets en la presentación
Consistencia	33.33%		
En páginas	16.66	si	
En menús	16.66	si	
Texto	33.33%		
Tamaño	12p	12p	
Color	Azul	Azul	
Estructura de la presentación	27.77%		
Widget	16.66		
Tamaño	12p*12p	15p*15p	Los Widgets son muy pequeños
Color	Azul	Azul	El color no es el recomendado
Vista	33.33%	lqz-derecha	
Menú	33.33%	Superior	
Capacidad de atraer	24.99%		Revisar Widgets
Diseño responsivo	50%	si	
Widgets	50%		
Tamaño	12p*12p	15p*15p	
Color	Gris	Azul	
Aprendibilidad			
Asistencia técnica	33.33%	si	
No. Componentes de estructura	33.33%	10+	
Documentación	33.33%	si	

46

46

3.3 Actividades

En esta sección se indican algunas actividades encaminadas a reforzar el entendimiento del contenido de este capítulo. Intente realizar todas las actividades.

- 1.- Enliste los atributos que pueden describir a una minuta de reunión de trabajo de un equipo de desarrollo de software
 - Relacione los atributos con legibilidad.
 - Identifique las importancias de los atributos relacionados.
 - Identifique cuales atributos tienen valores con direcciones opuestas.
- 2.- Enliste los atributos que pueden describir a un documento de especificación de requerimientos.
 - Relacione los atributos con legibilidad.
 - Identifique las importancias de los atributos relacionados.
 - Identifique cuales atributos tienen valores con direcciones opuestas.
- 3.- Enliste los atributos que pueden describir a un requerimiento
 - Relacione los atributos con entendibilidad.
 - Identifique las importancias de los atributos relacionados.
 - Identifique cuales atributos tienen valores con direcciones opuestas.
- 4.- Enliste los atributos que pueden describir a un manual de operación.
 - Relacione los atributos con entendibilidad
 - Identifique las importancias de los atributos relacionados.
 - Identifique cuales atributos tienen valores con direcciones opuestas.
- 5.-Enliste los atributos que puedan describir a una aplicación de ventas en línea.
 - Relacione los atributos con usabilidad.
 - Identifique las importancias de los atributos relacionados.
 - Identifique cuales atributos tienen valores con direcciones opuestas.



Estrategias para medir productos de software

Bibliografía

- Norma Mexicana para el proceso de desarrollo de software -NMX-059-1-2005.
- Modelo de procesos para la industria del software MOPROSOFT.
- ISO/IEC 9126.
- Software quality attributes and trade-offs, Lars Lundberg, Michael Mattsson, Claes Wohlin (eds), Blekinge Institute of Technology June 2005.

Referencias

- 1.- Software quality attributes and trade-offs, Lars Lundberg, Michael Mattsson, Claes Wohlin (eds), Blekinge Institute of Technology June 2005.
- 2.- Adipat, B., & Zhang, D. (2005). Interface Design for Mobile Applications.
- 3.- Meier, R. (2012). *Professional Android 4 Application Development*.
- 4.- Chang, E., & Dillon, T. S. (2006). A Usability-Evaluation Metric Based on a Soft-Computing Approach, 36(2), 356–372.
- 5.- Emmerich, K., & Masuch, M. (2016). Game Metrics for Evaluating Social In-game Behavior and Interaction in Multiplayer Games. *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*, 20:1–20:8. <https://doi.org/10.1145/3001773.3001793>
- 6.- Olubusola, A. O. (2015). User Satisfaction in Mobile Applications.
- 7.- Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H.-J., & Zhou, H. (2003). A visual attention model for adapting images on small displays.

48