

**Construcción de una herramienta para evaluar la calidad de un producto
software**

Sebastián Piedrahita Mesa

**Escuela de Ingeniería
Departamento de Ingeniería de Sistemas
Universidad EAFIT**

Medellín

2007

**Construcción de una herramienta para evaluar la calidad de un producto
software**

Sebastián Piedrahita Mesa

**Trabajo de grado presentado como requisito para optar al título de Ingeniero
de Sistemas**

**Asesor metodológico y temático
Rafael David Rincón Bermúdez
Profesor Departamento de Informática y Sistemas
Universidad EAFIT**

**Escuela de Ingeniería
Departamento de Ingeniería de Sistemas
Universidad EAFIT**

MEDELLÍN

2007

Nota de Aceptación:

Presidente del jurado

Jurado

Jurado

AGRADECIMIENTOS

Realizar este trabajo de grado significó mucho esfuerzo, pero gracias a la ayuda incondicional de mi asesor Rafael David Rincón B. al cual le agradezco enormemente su gran dedicación, logré concretarla.

También quiero agradecer a varias personas que estuvieron pendientes de mi proceso como Ingeniero de Sistemas y a las cuales les debo mucho. En primera instancia quiero agradecerle a mi padre, a mi madre y a mis hermanos por estar apoyándome constantemente. En segundo término a mis tías quienes me facilitaron mucho este proceso; y finalmente a mis amigos en los cuales encontré un apoyo muy grande y quienes me brindaron muchas alegrías en momentos difíciles.

RESUMEN

Actualmente se cuenta con dos normas internacionales que permiten evaluar la calidad de un producto Software, la norma ISO/IEC 9126 e ISO/IEC 14598. La norma ISO/IEC 14598 comprende seis partes que permiten especificar el proceso a seguir para evaluar un software, mientras que la norma ISO/IEC 9126 comprende cuatro partes para evaluar la calidad de un producto de software. Esta última norma comprende características que se pueden medir con métricas.

La utilización de métricas dentro de la evaluación de la calidad de productos software es muy importante, ya que permite precisar con detalle cada una de las características del producto evaluado.

Diariamente se hacen desarrollos de software para las empresas, con el fin de ayudar a agilizar y mejorar sus procesos. El problema radica en que la calidad de lo que se está desarrollando muchas veces es poca o no es suficiente para lo que se requiere; por tanto, es necesario incluir estándares que permitan evaluar la calidad de un producto de software para lograr la entera satisfacción de los clientes.

Cuando se evalúa un producto software, un cliente puede medir los niveles de calidad que considera pertinentes para su producto y a partir de esto, realizar cambios que permitan siempre mejorar la calidad. Cuando se investigan y se implementan nuevas métricas para evaluar la calidad de un producto de software se está mejorando la calidad de los desarrollos realizados en una empresa, lo que permite tener mejores y más confiados desempeños (resultados) de los programas, evitando futuros conflictos en los procesos de las empresas.

Específicamente, la construcción de una herramienta que permita evaluar la calidad de un producto software contribuye a que los desarrollos realizados en las empresas cumplan con un nivel de calidad tal que satisfaga las expectativas del cliente para el cual fue desarrollado dicho producto.

Así se podrá evaluar la calidad de un producto software, ya sea desde la construcción del mismo, es decir, en cada una de las etapas de un software nuevo, o bien como evaluación de un producto final. Lo anterior permite llegar a tener un conocimiento pleno de las características que posee un producto software y qué tan preciso es para un cliente de acuerdo con sus necesidades.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	25
1 PRELIMINARES	27
1.1 DEFINICIÓN DEL PROBLEMA	27
1.2 OBJETIVO	29
1.2.1 Objetivo general	29
1.2.2 Objetivos específicos.....	29
2. MARCO TEÓRICO	30
2.1 PRÓLOGO	30
2.2 MODELOS DE CALIDAD	30
2.3 Norma internacional ISO/IEC 14598	41
2.4 Norma internacional ISO/IEC 9126	44
2.5 Estructura de la ISO/IEC 14598 y la ISO/IEC 9126	47
2.6 Modelo de calidad para calidad externa e interna	48
2.7 CARACTERÍSTICAS Y SUB-CARACTERÍSTICAS	51
2.7.1 Funcionalidad	51
2.7.1.1 Adecuación	51
2.7.1.2 Exactitud	51
2.7.1.3 Conformidad con la funcionalidad.....	51
2.7.2 Seguridad	52
2.7.2.1 Resistencia al acceso	52
2.7.2.2 Resistencia a la copia.....	52
2.7.2.3 Facilidad para cifrar	52
2.7.2.4 Resistencia a la falsificación	52
2.7.2.5 Robustez	52
2.7.2.6 Conformidad con la seguridad	53

2.7.3	Interoperabilidad	53
2.7.3.1	Compatibilidad de la OSI	53
2.7.3.2	Compatibilidad del software.....	53
2.7.3.3	Compatibilidad de los datos.....	53
2.7.3.4	Trazabilidad	54
2.7.3.5	Conformidad con la interoperabilidad	54
2.7.4	Fiabilidad	54
2.7.4.1	Madurez.....	54
2.7.4.2	Tolerancia a errores.....	54
2.7.4.3	Recuperabilidad.....	54
2.7.4.4	Conformidad con la fiabilidad	55
2.7.5	Usabilidad.....	55
2.7.5.1	Apropiabilidad (Comprensibilidad).....	55
2.7.5.2	Facilidad de aprendizaje.....	55
2.7.5.3	Operabilidad	55
2.7.5.4	Capacidad de ayuda (Helpfulness).....	56
2.7.5.5	Atractibilidad (Likability)	56
2.7.5.6	Conformidad de uso	56
2.7.6	Eficiencia	56
2.7.6.1	Comportamiento en el tiempo.....	56
2.7.6.2	Utilización de recursos.....	57
2.7.6.3	Conformidad con la eficiencia.....	57
2.7.7	Mantenibilidad	57
2.7.7.1	Capacidad de ser analizado	57
2.7.7.2	Facilidad de cambio.....	57
2.7.7.3	Estabilidad	58
2.7.7.4	Facilidad de prueba	58
2.7.7.5	Conformidad con la facilidad de mantenimiento	58
2.7.8	Portabilidad.....	58
2.7.8.1	Adaptabilidad	58
2.7.8.2	Facilidad de instalación	58
2.7.8.3	Coexistencia	59
2.7.8.4	Reemplazabilidad	59
2.7.8.5	Conformidad con la portabilidad	59
2.8	Modelo de calidad para la calidad del software en uso	60
2.8.1	Calidad en uso del software	63

2.8.2	Usabilidad en uso	63
2.8.2.1	Eficacia de uso	63
2.8.2.2	Productividad de uso	63
2.8.2.3	Satisfacción de uso.....	64
2.8.2.4	Conformidad con la usabilidad en uso	64
2.8.3	Contexto en uso	64
2.8.3.1	Tipos de usuario de uso	64
2.8.3.2	Tareas de uso.....	65
2.8.3.3	Ambientes de uso	65
2.8.3.4	Conformidad del contexto en uso	65
2.8.4	Seguridad en uso	65
2.8.4.1	Riesgo para el operador en uso	65
2.8.4.2	Riesgo de la corrupción del software en uso	65
2.8.4.3	Riesgos comerciales en uso	66
2.8.4.4	Seguridad en uso.....	66
2.8.4.5	Conformidad con la seguridad en uso	66
2.8.5	Adaptabilidad en uso	66
2.8.5.1	Facilidad de aprendizaje en uso	66
2.8.5.2	Flexibilidad en uso	67
2.8.5.3	Accesibilidad en uso	67
2.8.5.4	Conformidad con la adaptabilidad en uso.....	67
2.9	Ventajas de los modelos / Estándares de calidad del software	68
3	PASOS PARA REALIZAR EL PROCESO DE UTILIZACIÓN DE LAS MÉTRICAS	70
4	MÉTRICAS DE CALIDAD DE PRODUCTO SOFTWARE	72
4.1	Trabajo de campo.....	75
4.1.1	Cómo leer y usar las tablas de las métricas	80
4.1.2	Métricas investigadas	82
4.2	Plantilla para evaluar la calidad de un producto software.....	87
	CONCLUSIONES	89
	BIBLIOGRAFÍA.....	92

LISTA DE TABLAS

Tabla 1.	Métrica interna de conformidad con la seguridad - regulación de la seguridad	82
Tabla 2.	Métrica interna de la conformidad con la interoperabilidad - regulación de interoperabilidad	82
Tabla 3.	Métrica externa de resistencia a la copia - incidentes de copia.....	83
Tabla 4.	Métrica externa de resistencia a la falsificación - incidentes de falsificación	84
Tabla 5.	Métrica externa de la compatibilidad del software - interoperabilidad con el software	84
Tabla 6.	Métrica externa de la compatibilidad de los datos - intercambio de los datos.....	85
Tabla 7.	Métrica externa de la trazabilidad - iteraciones grabadas	86
Tabla 8.	Métrica interna de adecuación - suficiencia funcional	112
Tabla 9.	Métrica interna de adecuación - integridad de la implementación funcional	113
Tabla 10.	Métrica interna de adecuación - alcance de la implementación funcional	113
Tabla 11.	Métrica interna de adecuación - estabilidad (o volatilidad) de la especificación funcional.....	114
Tabla 12.	Métrica interna de exactitud - exactitud computacional	115
Tabla 13.	Métrica interna de exactitud - precisión	115
Tabla 14.	Métrica interna de conformidad con la funcionalidad - conformidad funcional	116
Tabla 15.	Métrica interna de conformidad con la funcionalidad - conformidad estándar de la interfaz	117

Tabla 16. Métrica interna de resistencia al acceso - facilidad de auditar los accesos	118
Tabla 17. Métrica interna de resistencia al acceso - uso controlado del acceso	118
Tabla 18. Métrica interna de facilidad del cifrado - cifrado de datos.....	119
Tabla 19. Métrica interna de robustez - prevención de la corrupción de datos	120
Tabla 20. Métrica interna de compatibilidad de la OSI - consistencia de la interface (protocolo)	121
Tabla 21. Métrica interna de compatibilidad de los datos - intercambiabilidad de datos (basado en el formato de datos)	122
Tabla 22. Métrica interna de madurez - detección de fallas	123
Tabla 23. Métrica interna de madurez - remoción de fallos.....	123
Tabla 24. Métrica interna de madurez - suficiencia de prueba	124
Tabla 25. Métrica interna de tolerancia a fallos - prevención de fallas	125
Tabla 26. Métrica interna de tolerancia a fallos - prevención de la operación incorrecta	126
Tabla 27. Métrica interna de recuperabilidad - facilidad de restauración	126
Tabla 28. Métrica interna de recuperabilidad - efectividad de la restauración....	127
Tabla 29. Métrica interna de conformidad con la fiabilidad - conformidad con la fiabilidad	128
Tabla 30. Métrica interna de apropiabilidad - integridad de la descripción	129
Tabla 31. Métrica interna de apropiabilidad - capacidad de la demostración	129
Tabla 32. Métrica interna de apropiabilidad - funciones evidentes	130
Tabla 33. Métrica interna de apropiabilidad - comprensibilidad de la función ...	131

Tabla 34. Métrica interna de aprendizaje - integridad de la documentación del usuario y/o de la facilidad de la ayuda	131
Tabla 35. Métrica interna de operabilidad - comprobación de la validez de la entrada	132
Tabla 36. Métrica interna de operabilidad - facilidad de cancelar la operacion del usuario	133
Tabla 37. Métrica interna de operabilidad - facilidad de anular la operación del usuario	133
Tabla 38. Métrica interna de operabilidad - uso de la personalización	134
Tabla 39. Métrica interna de operabilidad - accesibilidad física	134
Tabla 40. Métrica interna de operabilidad - capacidad de monitoreo del estado de la operación	135
Tabla 41. Métrica interna de operabilidad - consistencia operacional	136
Tabla 42. Métrica interna de operabilidad - claridad del mensaje	136
Tabla 43. Métrica interna de operabilidad - claridad del elemento de la interfaz	137
Tabla 44. Métrica interna de operabilidad - facilidad de recuperación de un error operacional	138
Tabla 45. Métrica interna de atractibilidad - interacción atractiva	138
Tabla 46. Métrica interna de atractibilidad - personalización del aspecto de la interfaz de usuario	139
Tabla 47. Métrica interna de la conformidad de uso - conformidad con la usabilidad	140
Tabla 48. Métrica interna del comportamiento en el tiempo - tiempo de respuesta	140
Tabla 49. Métrica interna del comportamiento en el tiempo - tiempo del rendimiento de procesamiento	141
Tabla 50. Métrica interna del comportamiento en el tiempo - plazo de entrega	142

Tabla 51. Métrica interna de la utilización de recursos - utilización de entrada/salida	142
Tabla 52. Métrica interna de la utilización de recursos - densidad del mensaje de la utilización de entrada/salida	143
Tabla 53. Métrica interna de la utilización de recursos - utilización de la memoria	143
Tabla 54. Métrica interna de la utilización de recursos - densidad del mensaje de la utilización de la memoria	144
Tabla 55. Métrica interna de la utilización de recursos - utilización de la transmisión	144
Tabla 56. Métrica interna de la conformidad con la eficiencia - conformidad con la eficiencia	145
Tabla 57. Métrica interna de la capacidad de ser analizado - registro de la actividad	146
Tabla 58. Métrica interna de la capacidad de ser analizado – preparación de la función de diagnóstico	146
Tabla 59. Métrica interna de la facilidad de cambio - facilidad de registrar los cambios	147
Tabla 60. Métrica interna de estabilidad - Impacto en el cambio	148
Tabla 61. Métrica interna de estabilidad - localización del impacto de la modificación	148
Tabla 62. Métrica interna de facilidad de prueba - integridad de la función de prueba predefinida.....	149
Tabla 63. Métrica interna de facilidad de prueba - autonomía de la facilidad de prueba	150
Tabla 64. Métrica interna de facilidad de prueba - facilidad de observación del desarrollo de la prueba	150
Tabla 65. Métrica interna de la conformidad de la facilidad de mantenimiento - conformidad con la facilidad de mantenimiento	151

Tabla 66. Métrica interna de adaptabilidad - adaptabilidad de las estructuras de datos	152
Tabla 67. Métrica interna de adaptabilidad - adaptabilidad del ambiente de hardware	153
Tabla 68. Métrica interna de adaptabilidad - adaptabilidad del ambiente organizacional	153
Tabla 69. Métrica interna de adaptabilidad - amigabilidad del usuario	154
Tabla 70. Métrica interna de adaptabilidad - adaptabilidad del ambiente de software del sistema	155
Tabla 71. Métrica interna de la facilidad de instalación - facilidad de reintentar el setup.....	155
Tabla 72. Métrica interna de la facilidad de instalación - esfuerzo en la instalación	156
Tabla 73. Métrica interna de la facilidad de instalación - flexibilidad en la instalación	157
Tabla 74. Métrica interna de la coexistencia - coexistencia disponible	157
Tabla 75. Métrica interna de la reemplazabilidad - uso continuo de los datos ..	158
Tabla 76. Métrica interna de la reemplazabilidad - inclusividad de la función ...	159
Tabla 77. Métrica interna de la conformidad de la portabilidad - conformidad con la portabilidad	159
Tabla 78. Métrica externa de adecuación - suficiencia funcional.....	161
Tabla 79. Métrica externa de adecuación - integridad de la implementación funcional	161
Tabla 80. Métrica externa de adecuación - alcance de la implementación funcional	162
Tabla 81. Métrica externa de adecuación - estabilidad (volatilidad) de la especificación funcional	163
Tabla 82. Métrica externa de exactitud - exactitud a la anticipación esperada...	164

Tabla 83. Métrica externa de exactitud - exactitud computacional	164
Tabla 84. Métrica externa de exactitud - precisión	165
Tabla 85. Métrica externa de la conformidad con la funcionalidad - conformidad funcional	166
Tabla 86. Métrica externa de la conformidad con la funcionalidad - conformidad estándar de la interfaz	166
Tabla 87. Métrica externa de resistencia al acceso- facilidad de auditar los accesos	167
Tabla 88. Métrica externa de resistencia al acceso- uso controlado del acceso	168
Tabla 89. Métrica externa de robustez - prevención de la corrupción de datos	169
Tabla 90. Métrica externa de la compatibilidad de la OSI - intercambiabilidad de los datos (basado en el formato de datos)	170
Tabla 91. Métrica externa de la compatibilidad de la OSI - Intercambiabilidad de los datos (éxito del usuario basado en intentos)	171
Tabla 92. Métrica externa de madurez - densidad latente estimada de los fallos	172
Tabla 93. Métrica externa de madurez - densidad de la falla contra los casos de prueba	173
Tabla 94. Métrica externa de madurez - resolución de la falla	174
Tabla 95. Métrica externa de madurez - densidad de las fallas	174
Tabla 96. Métrica externa de madurez - remoción de fallos	175
Tabla 97. Métrica externa de madurez - tiempo medio entre fallas (MTBF).....	176
Tabla 98. Métrica externa de madurez - cobertura de prueba (cobertura de pruebas especificadas del escenario de la operación)	177
Tabla 99. Métrica externa de madurez - madurez de la prueba	177
Tabla 100. Métrica externa de tolerancia a fallos - evasión de la interrupción	178

Tabla 101. Métrica externa de tolerancia a fallos - prevención de fallas	179
Tabla 102. Métrica externa de tolerancia a fallos - prevención de la operación incorrecta	179
Tabla 103. Métrica externa de recuperabilidad - disponibilidad	180
Tabla 104. Métrica externa de recuperabilidad - tiempo medio de inactividad	181
Tabla 105. Métrica externa de recuperabilidad - tiempo medio de recuperación	182
Tabla 106. Métrica externa de recuperabilidad - capacidad para recomenzar	182
Tabla 107. Métrica externa de recuperabilidad - facilidad de restauración	183
Tabla 108. Métrica externa de recuperabilidad - efectividad de la restauración ..	184
Tabla 109. Métrica externa de la conformidad con la fiabilidad - conformidad con la fiabilidad	185
Tabla 110. Métrica externa de la comprensibilidad - integridad de la descripción	185
Tabla 111. Métrica externa de la comprensibilidad - accesibilidad a la demostración	186
Tabla 112. Métrica externa de la comprensibilidad - accesibilidad a la demostración en uso	187
Tabla 113. Métrica externa de la comprensibilidad - eficacia de la demostración	187
Tabla 114. Métrica externa de la comprensibilidad - funciones evidentes	188
Tabla 115. Métrica externa de la comprensibilidad - comprensibilidad de la función	189
Tabla 116. Métrica externa de la comprensibilidad - entrada y salida comprensibles	189

Tabla 117. Métrica externa de la facilidad de aprendizaje - facilidad de aprender de la función	190
Tabla 118. Métrica externa de la facilidad de aprendizaje - facilidad de aprender a realizar una tarea en uso	191
Tabla 119. Métrica externa de la facilidad de aprendizaje - eficacia de la documentación del usuario y/o la ayuda del sistema	191
Tabla 120. Métrica externa de la facilidad de aprendizaje - eficacia de la documentación del usuario y/o la ayuda del sistema en uso	192
Tabla 121. Métrica externa de la facilidad de aprendizaje - accesibilidad de la ayuda	193
Tabla 122. Métrica externa de la facilidad de aprendizaje - frecuencia de la ayuda	193
Tabla 123. Métrica externa de la operabilidad - consistencia operacional en uso	194
Tabla 124. Métrica externa de la operabilidad - corrección de error	195
Tabla 125. Métrica externa de la operabilidad - corrección de error en uso	195
Tabla 126. Métrica externa de la operabilidad - disponibilidad del valor prefijado en uso	196
Tabla 127. Métrica externa de la operabilidad - comprensibilidad del mensaje en uso	197
Tabla 128. Métrica externa de la operabilidad - mensajes de error que se explican por sí mismos	198
Tabla 129. Métrica externa de la operabilidad - facilidad de recuperación de un error operacional en uso	199
Tabla 130. Métrica externa de la operabilidad - tiempo entre las operaciones humanas del error en uso	199
Tabla 131. Métrica externa de la operabilidad - capacidad de deshacer (corrección de error del usuario)	200
Tabla 132. Métrica externa de la operabilidad - uso de la personalización	201

Tabla 133. Métrica externa de la operabilidad - reducción del procedimiento de la operación	202
Tabla 134. Métrica externa de la operabilidad - accesibilidad física	202
Tabla 135. Métrica externa de atractivo - interacción atractiva	203
Tabla 136. Métrica externa de atractivo - personalización de la apariencia de la interfaz	204
Tabla 137. Métrica externa de la conformidad de uso - conformidad con la usabilidad	204
Tabla 138. Métrica externa del comportamiento en el tiempo - tiempo de respuesta	205
Tabla 139. Métrica externa del comportamiento en el tiempo - tiempo de respuesta (tiempo medio para responder)	206
Tabla 140. Métrica externa del comportamiento en el tiempo - tiempo de respuesta (proporción al peor caso del tiempo de respuesta).....	207
Tabla 141. Métrica externa del comportamiento en el tiempo - rendimiento de procesamiento	208
Tabla 142. Métrica externa del comportamiento en el tiempo - rendimiento de procesamiento (cantidad media de rendimiento).....	208
Tabla 143. Métrica externa del comportamiento en el tiempo - rendimiento de procesamiento (proporción del peor caso de rendimiento de procesamiento).....	209
Tabla 144. Métrica externa del comportamiento en el tiempo - plazo de entrega	210
Tabla 145. Métrica externa del comportamiento en el tiempo - plazo de entrega (tiempo medio para la respuesta)	211
Tabla 146. Métrica externa del comportamiento en el tiempo - plazo de entrega (proporción del peor caso del tiempo de respuesta).....	212
Tabla 147. Métrica externa del comportamiento en el tiempo - tiempo de espera	213

Tabla 148. Métrica externa del comportamiento en el tiempo - utilización de dispositivos de entrada-salida	213
Tabla 149. Métrica externa del comportamiento en el tiempo - límites de carga de entrada y salida	214
Tabla 150. Métrica externa del comportamiento en el tiempo - errores relacionados de entrada y salida	215
Tabla 151. Métrica externa del comportamiento en el tiempo - proporción media del cumplimiento de entrada y salida	216
Tabla 152. Métrica externa del comportamiento en el tiempo - tiempo de espera del usuario en la utilización de dispositivos de entrada y salida	216
Tabla 153. Métrica externa del comportamiento en el tiempo - utilización máxima de la memoria	217
Tabla 154. Métrica externa del comportamiento en el tiempo - ocurrencia media del error de memoria	218
Tabla 155. Métrica externa del comportamiento en el tiempo - proporción de error/tiempo de la memoria	219
Tabla 156. Métrica externa del comportamiento en el tiempo - utilización máxima de la transmisión	219
Tabla 157. Métrica externa del comportamiento en el tiempo - equilibrio de la utilización de dispositivos mediáticos	220
Tabla 158. Métrica externa del comportamiento en el tiempo - ocurrencia media del error de transmisión	228
Tabla 159. Métrica externa del comportamiento en el tiempo - media de error de transmisión por tiempo	222
Tabla 160. Métrica externa del comportamiento en el tiempo - utilización de la capacidad de transmisión	223
Tabla 161. Métrica externa de la conformidad con la eficiencia - conformidad con la eficiencia	223
Tabla 162. Métrica externa de la capacidad de ser analizado - capacidad de auditar de intervención	224

Tabla 163. Métrica externa de la capacidad de ser analizado - soporte de diagnóstico de la función	225
Tabla 164. Métrica externa de la capacidad de ser analizado - capacidad de análisis de falla	226
Tabla 165. Métrica externa de la capacidad de ser analizado - eficacia del análisis de la falla	226
Tabla 166. Métrica externa de la capacidad de ser analizado - capacidad de monitorear el estado	227
Tabla 167. Métrica externa de la facilidad de cambio - cambiar la eficacia del ciclo	228
Tabla 168. Métrica externa de la facilidad de cambio - cambiar el tiempo que transcurre en la implementación	228
Tabla 169. Métrica externa de la facilidad de cambio - complejidad de la modificación	229
Tabla 170. Métrica externa de la facilidad de cambio - modificabilidad parametrizada	230
Tabla 171. Métrica externa de la facilidad de cambio - capacidad de control de cambio en el software	231
Tabla 172. Métrica externa de la estabilidad - cambio en la proporción del éxito	231
Tabla 173. Métrica externa de la estabilidad - localización del impacto de la modificación (fallo que emerge después del cambio).....	232
Tabla 174. Métrica externa de la facilidad de prueba - disponibilidad de la función de la prueba incorporada	233
Tabla 175. Métrica externa de la facilidad de prueba - Realizar varias pruebas de eficiencia	234
Tabla 176. Métrica externa de la facilidad de prueba - capacidad para recomenzar las pruebas	234
Tabla 177. Métrica externa de la conformidad con la facilidad de mantenimiento - conformidad con la capacidad de mantenimiento	235

Tabla 178. Métrica externa de la adaptabilidad - adaptabilidad de las estructuras de datos	236
Tabla 179. Métrica externa de la adaptabilidad - adaptabilidad del ambiente de hardware	237
Tabla 180. Métrica externa de la adaptabilidad - adaptabilidad al ambiente organizacional	237
Tabla 181. Métrica externa de la adaptabilidad - amigabilidad del usuario	238
Tabla 182. Métrica externa de la adaptabilidad - adaptabilidad del ambiente de software del sistema	239
Tabla 183. Métrica externa de la facilidad de instalación - facilidad de instalación	239
Tabla 184. Métrica externa de la facilidad de Instalación - facilidad de reintentar el setup	240
Tabla 185. Métrica externa de coexistencia - coexistencia disponible	241
Tabla 186. Métrica externa de reemplazabilidad - uso continuo de datos	241
Tabla 187. Métrica externa de reemplazabilidad - inclusividad de la función	242
Tabla 188. Métrica externa de reemplazabilidad - consistencia funcional al soporte del usuario	243
Tabla 189. Métrica externa de la conformidad con la portabilidad - conformidad con la portabilidad	244

LISTA DE FIGURAS

Figura 1. Modelo de calidad de McCall (1977)	34
Figura 2. Diferentes visiones del modelo de calidad de McCall.....	35
Figura 3. Modelo de Boehm para clasificar los criterios de calidad	36
Figura 4. Relación entre los estándares internacionales ISO/IEC 9126 e ISO/IEC 14598	47
Figura 5. Modelo de calidad para la calidad externa e interna según la norma ISO/IEC 9126:2001	48
Figura 6. Modelo de calidad para la calidad externa e interna según la norma SQuaRE	49
Figura 7. Modelo de calidad para la calidad del software en uso según la norma ISO/IEC 9126:2001	60
Figura 8. Modelo de calidad para la calidad del software en uso según la norma SQuaRE	61
Figura 9. Organización de la serie de estándares de SQuaRE	97
Figura 10. Modelos del sistema y modelos de calidad	103
Figura 11. Calidad en el ciclo de vida	104
Figura 12. Modelo de referencia de la medida de calidad del producto de software	106
Figura 13. Modelo del ciclo de vida para la calidad de un producto de software.. ..	107
Figura 14. Estructura del modelo de calidad	109
Figura 15. Nota de información para el ingreso de datos	267
Figura 16. Orden de las hojas que requieren ingreso de datos por parte del usuario	267

Figura 17. Cuadro de error cuando se ingresa mal un dato	268
Figura 18. Mensaje de error cuando no se cumple la suma de las ponderaciones	268
Figura 19. Cuadro de advertencia cuando se intenta acceder a una celda diferente a las del ingreso de datos disponibles	268
Figura 20. Botón de ayuda	269
Figura 21. Ejemplo de cuadro de ayuda	269
Figura 22. Ejemplo de ingreso de datos para las ponderaciones	272
Figura 23. Nombres de las hojas de la plantilla	274
Figura 24. Hoja ponderaciones.....	275
Figura 25. Hoja precalificación	276
Figura 26. Hoja calificación detallada	277
Figura 27. Hoja calificación detallada (continuación).....	277
Figura 28. Hoja resumen calificación detallada	278
Figura 29. Muestra de la gráfica de una característica - Hoja resumen calificación detallada	279
Figura 30. Hoja gráficos calificación detallada.....	280
Figura 31. Hoja métricas.....	281

LISTA DE ANEXOS

ANEXO A. Normas SQuaRE. (Software product Quality Requirements and Evaluation) . Quality model. ISO/IEC CD 25010	96
ANEXO B. MÉTRICAS INTERNAS	112
ANEXO C. MÉTRICAS EXTERNAS	161
ANEXO D. Explicación detallada de los tipos de escala de las métricas y de los tipos de medidas	245
ANEXO D. TÉRMINOS Y DEFINICIONES	256
ANEXO E. MANUAL DE USUARIO. PLANTILLA PARA EVALUAR LA CALIDAD DE UN PRODUCTO SOFTWARE	264

INTRODUCCIÓN

La calidad de un producto software está determinada por la calidad de los procesos empleados para desarrollar y mantener dicho producto. Esto es muy importante porque un producto no se finaliza al “cerrar” el proyecto de desarrollo, sino que además debe evolucionar y pulir defectos, es decir, mantenerlo, lo que conlleva también un proceso asociado. De poco valdría que el proceso de desarrollo de una organización sea “perfecto” y proporcione productos de una calidad excelente si luego no “cuida” el producto a lo largo del tiempo y lo evoluciona.¹

“La calidad de un producto de software se debe evaluar usando un modelo definido. El modelo de calidad debe ser utilizado al fijar las metas de la calidad para los productos de software y los productos intermedios. Dado que es imposible medir la calidad de un producto de software de forma directa, éste se descompone jerárquicamente en un modelo de calidad constituido por características y sub-características.”²

Normalmente se usa un modelo de calidad para los productos de software, dividido en dos partes: Calidad interna y externa, y Calidad en uso.

La primera parte del modelo especifica ocho características para evaluar la calidad interna y externa, las cuales, a su vez, están subdivididas en sub- características. Estas características son: funcionalidad, seguridad, interoperabilidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

¹ PRESIDENCIA DEL CONSEJO DE MINISTROS (PERÚ). “Guía técnica sobre evaluación de software en la administración pública”. Mayo 05. Año 2004.

² GONZÁLEZ JARDON, Carlos. La gestión del proceso software. Primera parte. Septiembre 09. Año 2006.

La segunda parte del modelo especifica seis características para la evaluación de la calidad en uso. Estas características son: usabilidad en uso, contexto en uso, protección (safety) en uso, seguridad (security) en uso, soporte en uso y adaptabilidad en uso.

Este proyecto se realiza a partir del trabajo de grado “Automatización de los procedimientos para la recolección y consolidación de las métricas externas del producto software según la norma ISO/IEC 9126:2001”³. Se pretende, entre otras cosas, incluir un mayor número de métricas a las que ya están previamente definidas, en unas plantillas existentes⁴ con las cuales se puede evaluar la calidad de un producto software, con el propósito de tener un buen grupo de características a evaluar.

En este desarrollo se utilizaron los conceptos adquiridos en el área de Calidad de Software y estuvo basado en los estándares de calidad de las normas ISO/IEC 9126 y 14598. Además de las métricas expuestas por la norma ISO/IEC 9126-2 y ISO/IEC 9126-3, se investigaron otras métricas referentes a la Seguridad e Interoperabilidad, ajustándolas al nuevo modelo de calidad propuesto en la norma SQuaRE.

³ BUSTAMANTE BEDOYA, Andrés Camilo. “Automatización de los procedimientos para la recolección y consolidación de las métricas externas del producto software según la norma ISO/IEC 9126:2001”. Proyecto de grado. Universidad EAFIT. Año 2004.

⁴ Creación y aporte de Juan David Correa, Gerente de Nodrizza Network.

1. PRELIMINARES

1.1 DEFINICIÓN DEL PROBLEMA

En algunas de las empresas de software de nuestro entorno se usan estándares de calidad (como las normas ISO/IEC) con las cuales se pueden evaluar los productos software desarrollados. Estas normas incluyen métricas que permiten tener patrones de referencia para cumplir con la satisfacción esperada de un producto. Sin embargo, no todas las métricas son utilizadas para la evaluación; estas varían, dependiendo de las necesidades que se requieren de dicho producto.

Las empresas muchas veces miden la calidad de sus productos de manera arbitraria, es decir, realizan pruebas sencillas en donde se analiza si el software es funcional, y no se enfocan en otros aspectos que también son importantes evaluar. Normalmente las empresas no suelen tener un esquema de medición riguroso para los productos software, y la calidad de los mismos no termina siendo la mejor.

La falta de una herramienta que permita integrar la información necesaria para realizar la evaluación de los productos, hace que no se evalúen aspectos relevantes.

El propósito es considerar algunas métricas de uso continuo en algunas organizaciones de TI que permitan evaluar la calidad de un producto de software. Esto permite que se tenga un rango más amplio de características a evaluar y por tanto, se estarán considerando muchos más aspectos del producto de software.

Este trabajo es un apoyo para que los desarrollos de software tengan una evaluación más completa y así lograr que haya un mayor cumplimiento de las

expectativas que un cliente tiene sobre un producto de software. Cuando la calidad de un producto software no tiene los niveles de satisfacción esperados por un cliente, el desarrollo termina siendo en ocasiones poco útil para el propósito para el cual fue creado.

En este proyecto se definirán métricas para medir la calidad de un producto de software y se adecuarán esas métricas en un nuevo desarrollo de la plantilla que está siendo utilizada en los cursos de calidad de software de la Universidad EAFIT, con el fin de mejorar la forma de evaluar un producto software. Estas métricas serán investigadas a partir de documentación actualizada y de un estudio realizado en empresas que mantienen un contacto permanente con la calidad de los productos software.

1.2 OBJETIVO

1.2.1 Objetivo general

Construir una herramienta que permita evaluar la calidad de un producto software genérico y/o específico, con base en estándares internacionales.

1.2.2 Objetivos específicos

- 1) Desarrollar plantillas que permitan manipular las métricas utilizadas en la calificación de un producto de software.
- 2) Estudiar con mayor profundidad los estándares de calidad, los cuales servirán como guía para la investigación de las nuevas métricas.
- 3) Profundizar sobre las métricas tanto internas como externas para la evaluación de la calidad de un producto de software.
- 4) Consultar a expertos sobre métricas de calidad de productos software que se estén utilizando actualmente en el medio.

2. MARCO TEÓRICO

2.1 PRÓLOGO

La calidad del software es fundamental para las empresas y su evaluación se hace pertinente para que se cumplan los propósitos que se quieren lograr con la ayuda de esos productos software.

La evaluación de la calidad del software se hace a través de modelos y estándares de calidad del software, los cuales reúnen todas las actividades y funciones de forma tal que cada una se planee, se controle y se ejecute de un modo formal y sistemático.⁵

Cuando se aplican modelos de calidad, finalmente lo que se logra es que mejoren los procesos de software, así como la calidad del software que se desarrolla.

2.2 MODELOS DE CALIDAD

Los modelos de calidad de software sirven como una guía para el desarrollo de un proyecto, ya que los modelos de calidad ayudan para saber qué es lo que se debe hacer para llevar a cabo el desarrollo, más no dicen cómo hacerlo. Esto es por que dependiendo de la metodología establecida, se obtiene un resultado más apropiado para el proceso que se use en el desarrollo, lo cuál va enfocado a los objetivos que se quieran perseguir en la empresa. Al implementar un modelo de calidad se logra tener una mejor productividad y calidad en el producto final, satisfaciendo las necesidades y deseos de los clientes.

⁵ SCALONE, Fernanda. Software quality management. Overview sobre modelos/estándares de calidad del SW. Diciembre 02. Año 2006.

Las empresas pueden implementar la calidad a nivel del proceso y en consecuencia obtener calidad a nivel producto.

“La calidad a nivel del proceso puede ser evaluada de manera genérica o específica, según el modelo o estándar seleccionado. Todo modelo o estándar a este nivel tiene un ámbito de aplicación específico y tiene como finalidad el mejoramiento continuo, luego de realizada la implantación del mismo.”⁶ Entre los modelos y/o estándares de calidad de software a nivel proceso se pueden mencionar: (1) ISO 9001:2000, (2) CMMi, (3) ISO 90003:2004, (4) ISO 20000, y otros.

“La calidad a nivel producto plantea distintos modelos y estándares que poseen un conjunto de características, las cuales tienen asociadas sub-características y métricas. Todo equipo de desarrollo deberá evaluar la calidad de software durante las diferentes etapas de desarrollo del software y ambientes de trabajo respectivos (desarrollo, prueba y producción). Esto evita futuros problemas y contribuye a una posible disminución en los tiempos y costos. Entre los modelos y/o estándares de calidad de software a nivel producto se pueden mencionar: (1) modelo de Boehm, (2) modelo de Gilb, (3) modelo de Dromey, (4) ISO 9126, (5) modelo de McCall, (6) WebQEM, (7) ISO 25000, (8) Portal Quality Model (PQM) y otros. Los modelos y estándares a nivel producto surgen o se actualizan de acuerdo con la evolución tecnológica ocurrida.”⁷

Para llevar a cabo el proyecto “Construcción de una herramienta para evaluar un producto software”, fue necesario primero conocer cuáles son las métricas estándar que se están usando actualmente para evaluar la calidad de un producto software. Así mismo se adicionó nuevas métricas, de acuerdo con una

⁶ SCALONE, op. cit.

⁷ SCALONE, op. cit.

investigación realizada en distintas empresas y en documentos relacionados con el tema de calidad del software. También fue importante conocer los diferentes modelos de calidad que se utilizan para analizar un producto de software y que son los que permiten conocer cómo será la distribución de las métricas necesarias para dicha evaluación del producto. Para este caso el modelo de calidad de la norma SQuaRE fue el ideal pues contenía muchas características.

“Construir un modelo de calidad es bastante complejo. Es usual que estos modelos descompongan la calidad del producto software jerárquicamente en una serie de características y sub-características que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad. La mayor parte de ellos están basados en la norma ISO/IEC 9126. Esta norma define un conjunto de características de calidad que son después refinadas en sub-características, las cuales están descompuestas en atributos. Los valores de estos atributos se calculan mediante la utilización de métricas o listas de chequeo.”⁸

Existen diversos modelos de calidad que permiten evaluar un producto de software, como se mencionó anteriormente. A continuación se ampliará la información sobre algunos de ellos.

- **Modelo de McCall.** Este modelo establece tres áreas principales que intervienen en la calidad del software:
 - *Revisión de la calidad del producto de software.* Tiene como objetivo realizar revisiones durante el proceso de desarrollo para detectar los errores que afecten a la operación del producto.

⁸ MUÑOZ CALERO, Coral. “Modelos de calidad. WQM, PQM, e-commerce, portlets”. Calidad de sistemas de información. Departamento de informática. Universidad de Castilla - La Mancha. Mayo 09. Año 2005.

- *Transición del producto.* Requiere de la definición de estándares y procedimientos que sirvan como base para el desarrollo del software.
- *Calidad en la operación del producto.* Requiere que el software pueda ser entendido fácilmente, que opere eficientemente y que los resultados obtenidos sean los requeridos inicialmente por el usuario.

Con relación a la operación del producto, se sugieren las siguientes características, con sus correspondientes interrogantes:

- Corrección – ¿El producto hace lo que el usuario quiere?
- Confiabilidad – ¿El producto es seguro?
- Integridad – Lo que hace el producto, ¿lo hace en forma fiable a lo largo del tiempo?
- Usabilidad – ¿Se puede ejecutar?
- Eficiencia – ¿El comportamiento del producto será adecuado en el ambiente con el que cuenta el usuario para su ejecución?

Dentro de la revisión del producto se estudian las siguientes características, para las cuales se plantean diferentes interrogantes:

- Mantenibilidad - ¿El producto se puede arreglar?
- Flexibilidad - ¿El producto se puede cambiar?
- Facilidad de Prueba - ¿El producto se puede probar?

A la transición del producto se asocian las siguientes características:

- Portabilidad – ¿El producto puede ser usado en diferentes ambientes?
- Reusabilidad – ¿Se puede reutilizar alguna parte del producto para satisfacer una necesidad?

- Interoperabilidad – ¿El producto puede interactuar con otros sistemas?

En la figura 1 se puede observar, en resumen, la propuesta del modelo McCall.

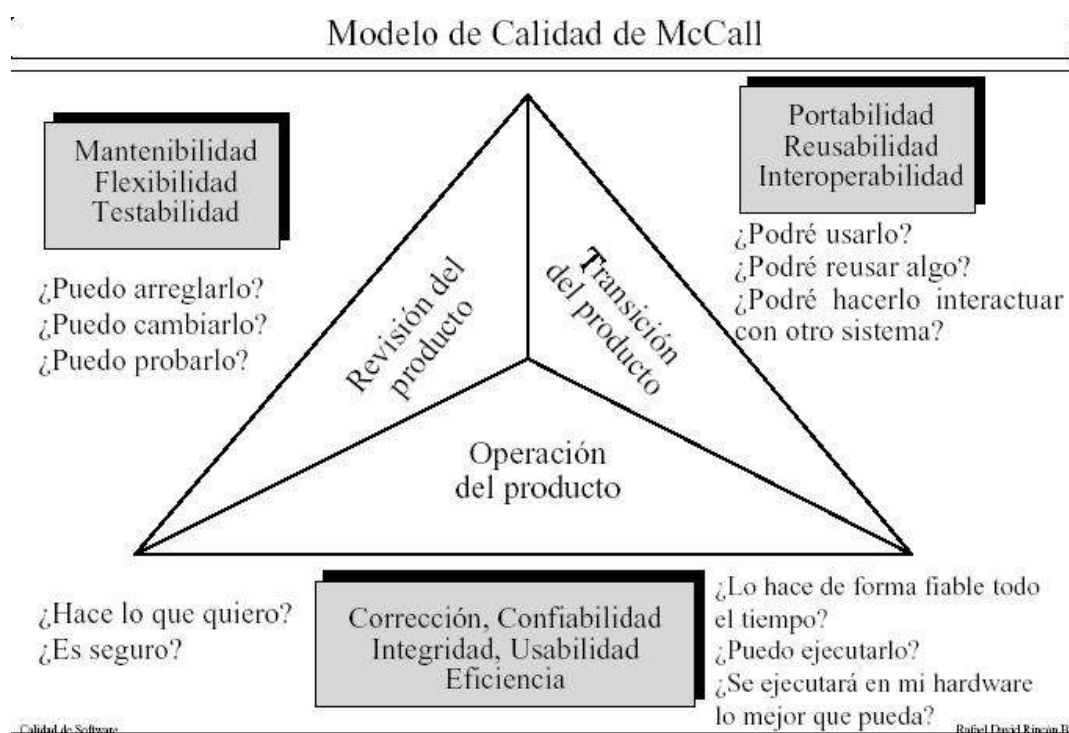


Figura 1. Modelo de calidad de McCall (1977) ⁹

Adicionalmente, el modelo de McCall se puede enmarcar dentro de tres puntos de vista: el del usuario, el de la dirección y el de la persona desarrolladora del mismo, cada uno de los cuales cumple un rol diferente dentro de la organización y observa al producto de software de una manera diferente, según la orientación de sus

⁹ JARAMILLO OSPINA, Luz Adriana y FRANCO CÓRDOBA, Ana María. Plan de calidad para la evaluación de productos de software en uso con base en la norma ISO/IEC 9126. Proyecto de grado. Universidad EAFIT. Año 2003. pág. 17

necesidades. A continuación se pueden observar los diferentes puntos de vista mencionados anteriormente y sus características asociadas.¹⁰

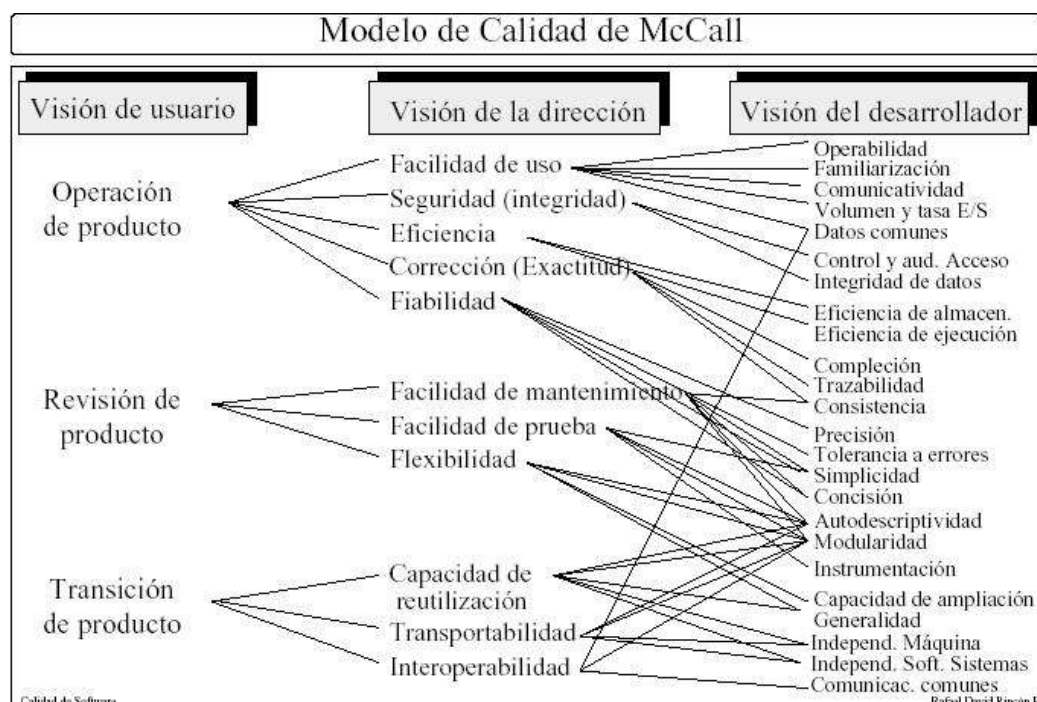


Figura 2. Diferentes visiones del modelo de calidad de McCall ¹¹

- **Modelo de Boehm (1978).** Este modelo es de naturaleza jerárquica y los criterios de calidad se presentan en tres grandes divisiones. La primer división es hecha acorde con los servicios que el sistema ofrece (Portabilidad). La segunda se hace de acuerdo con la operación del

¹⁰ Ibíd., pág. 16.

¹¹ JARAMILLO OSPINA, Luz Adriana y FRANCO CÓRDOBA, Ana María. Plan de calidad para la evaluación de productos de software en uso con base en la norma ISO/IEC 9126. Proyecto de grado. Universidad EAFIT. Año 2003. pág. 18

producto (Usabilidad) y la tercera división se hace de acuerdo con la Mantenibilidad del producto de software.¹²

En la figura 3 se presenta el modelo de Boehm, en donde se puede observar cada una de las divisiones mencionadas.

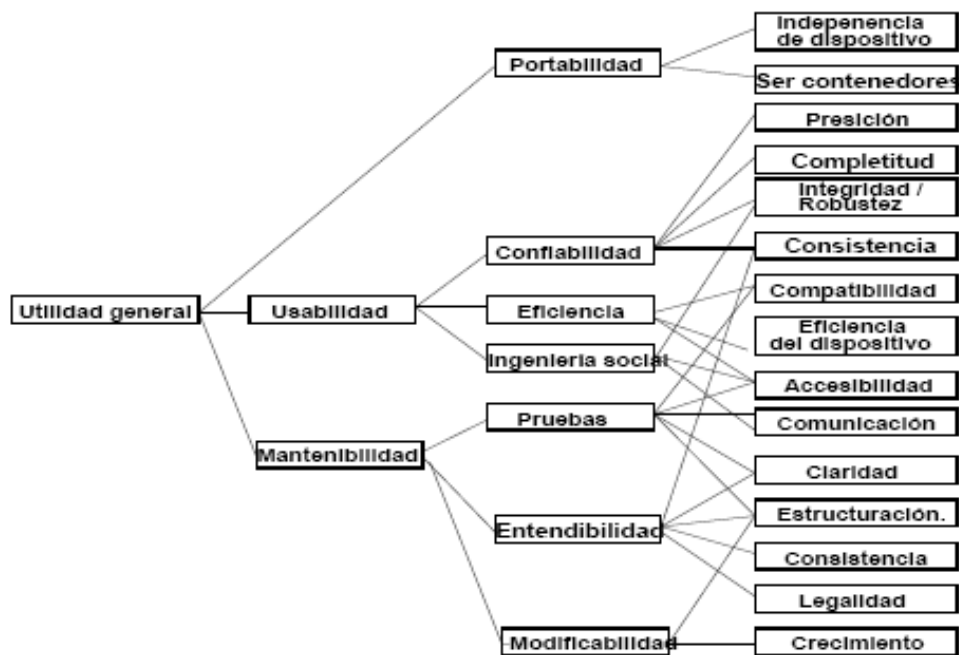


Figura 3. Modelo de Boehm para clasificar los criterios de calidad ¹³

- **Modelo de Gilb (1988).** El modelo de Gilb presenta como aspecto fundamental la definición de los atributos de calidad que realmente interesan al usuario y el nivel de calidad que debe tener cada uno de ellos

¹² DÁVILA NICANOR, Leticia y MEJÍA ÁLVAREZ, Pedro. "Evaluación de la calidad de software en sistemas de información en internet". CINVESTAV-IPN. Sección de computación. Zacatenco, México, DF. Año 2003.

¹³ DÁVILA NICANOR y MEJÍA ÁLVAREZ, op. cit., pág. 1.

para satisfacerlo, ya que no tiene sentido exigir calidad en un producto, si no se cuenta con esta base.

Mediante el método de Gilb es posible especificar los atributos de calidad de software en forma cuantitativa, incluyendo tanto tiempos de respuesta como conceptos conocidos de usabilidad y portabilidad, entre otros.

Gilb propone características como la corrección, la integridad, la facilidad de mantenimiento y la facilidad de uso, como base para proporcionar indicadores útiles para los equipos de trabajo y sugiere las definiciones, puntos de vista y medidas para cada uno de las siguientes características:

- **Corrección:** Grado en el que el software lleva a cabo su función requerida. Si un programa no opera correctamente, no dará valor agregado a sus usuarios.
- **Facilidad de mantenimiento:** Posibilidad de corregir un programa si se encuentra un error, adaptarlo si cambia su entorno, o mejorarlo si el cliente desea un cambio.
- **Integridad:** Habilidad de un sistema para resistir ataques, tanto accidentales como intencionados, contra su seguridad, a nivel de cualquiera de los tres principales componentes del software: programas, datos y documentos. Para medir la integridad, Gilb sugiere la utilización de otros dos atributos como base: **amenaza y seguridad**.
 - **Amenaza** es la probabilidad (que se puede estimar o deducir de la evidencia empírica) de que un ataque de cualquier tipo ocurra en un tiempo determinado.

- **Seguridad** es la probabilidad de que se pueda repeler un determinado ataque.
- **Facilidad de uso:** Es un intento por cuantificar “lo amigable que puede ser el producto con el usuario”.¹⁴
- **Modelo propuesto Bertoa y Vallecillo (2002).** Para componentes software en el que los autores adaptan la norma ISO/IEC 9126 a los componentes COTS (Commercial off-the-shelf).
- **Modelo de Simão y Belchior (2003).** En el que los autores han ampliado las sub-características y atributos propuestos por la norma ISO/IEC 9126, llegando a identificar 124 atributos de calidad para los componentes software.
- **Modelo de Ortega (2000).** Se basa en el modelo de calidad sistémica de Callaos y Callaos.
- **Modelo de calidad QUINT2 (Niessink, 2002).** Presenta una ampliación de la norma ISO/IEC 9126, pensada para valorar la calidad de arquitecturas software.
- **Modelo de calidad propuesto por Franch and Carvallo (2003).** Presenta una adaptación de la ISO/IEC 9126 para servidores de correo electrónico.
- **Modelo Botella (2003).** Proponen un modelo para la selección de ERP y también escogen como marco de trabajo el estándar de calidad ISO/IEC 9126-1.

¹⁴ JARAMILLO OSPINA y FRANCO CÓRDOBA, op. cit., pág. 19.

- **Modelo Cai (2000).** Proponen un modelo de calidad para componentes y sistemas basados en componentes.
- **Modelo Fernández and Rossi (2000).** Definen un modelo de calidad para software distribuido.
- **Modelo en Zo and Ramamurthy (2002).** Los autores presentan un modelo para valorar y seleccionar los sitios Web de comercio electrónico en un entorno B2C (Business-to-consumer).
- **Modelo en Webb and Webb (2002).** Presentan los factores de calidad del sitio Web que son importantes para los consumidores.
- **Modelo Parasuraman (1998).** Se describe el modelo SERVQUAL el cual contiene cinco dimensiones y 22 ítems para medir los diferentes elementos de la calidad de un servicio en general. La idea de este modelo es que puede ser adaptado a diferentes entornos en función de los servicios ofrecidos por cada uno de ellos, adaptando las dimensiones descritas en el modelo original.
- **Modelo WQM.** Pretende ser un modelo global de calidad de la web. Está caracterizado por tres elementos:
 1. La característica de calidad (basada en Quint2 y en la ISO/IEC 9126)
 2. El proceso del ciclo de vida (basado en la ISO/IEC 12207)
 3. Características (contenido, presentación y navegación)

- **Modelo de Calidad para Portales (PQM).** Utilizan Fase 1 y 2 de GQM y el modelo SERVQUAL.¹⁵
- **Modelo GQM (Goal-Question-Metric):** enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr. A continuación se presenta la estructura del modelo:
 - **Nivel Conceptual (Goal-Meta)**
Se define un objetivo (meta) para un objeto (ente), con respecto a determinado “modelo de calidad”, para un punto de vista, relativo a un contexto en particular.
 - **Nivel Operativo (Question-Pregunta)**
Se refina un conjunto de preguntas a partir de una meta, identificando el objeto de medición con respecto a características de calidad seleccionadas para un punto de vista.
 - **Nivel Cuantitativo (Metric-Métrica)**
Se asocia un conjunto de métricas para cada pregunta, de modo de responder a cada una de ellas de un modo cuantitativo.¹⁶

Más adelante se explicará el modelo con el que se realizó este proyecto, el cuál está basado en las Normas ISO/IEC 9126 e ISO/IEC 14598 y se describirán las características y sus respectivas sub-características, con el fin de señalar luego cuáles son algunas de las métricas que corresponden a dichos atributos.

¹⁵ MUÑOZ CALERO, op. cit., pág. 4.

¹⁶ OLSINA, Luis. “Ingeniería Web: Marco de medición y evaluación de calidad”. Departamento de informática. Universidad nacional de San Luis - La Rioja - Catamarca. Año 2007.

2.3 Norma internacional ISO/IEC 14598.¹⁷

La norma internacional ISO/IEC 14598-1 fue preparada por el comité técnico ISO/IEC JTC1 “Tecnología de Información”.

Las series de estándares de la ISO/IEC 14598 proveen métodos para las mediciones y evaluaciones de la calidad de un producto de software. No describen ni los métodos para evaluar los procesos de producción del software ni los métodos de predicción de costos.

Las series de la ISO/IEC 14598 dan una vista de los procesos de evaluación de un producto de software, proveen guías y requisitos de evaluación y dan una explicación de cómo el modelo de calidad puede ser aplicado en la evaluación de un producto de software.

Esta norma está diseñada para ser usada por desarrolladores, adquisidores y evaluadores independientes, particularmente para aquellos que se responsabilizan de la evaluación de productos de software. Los resultados de la evaluación producidos por la aplicación de los estándares de la ISO/IEC 14598 pueden ser usados por administradores y por desarrolladores y personas encargadas del mantenimiento para medir el “acatamiento” de los requisitos y para hacer mejoras en donde sea necesario. Los resultados de la evaluación también pueden ser usados por analistas para establecer las relaciones fundamentales entre las métricas internas y las externas. El personal de mejoras de procesos puede usar los resultados de la evaluación para determinar cómo los procesos pueden ser mejorados por medio del estudio y la evaluación de la información de la calidad del producto.

¹⁷ ISO/IEC TR 9126-3:2003. International standard “Software engineering – product quality – Part 3: internal metrics.”

La ISO/IEC 14598 consiste de las siguientes partes, bajo el título general de “Tecnología de la Información - Evaluación del producto de Software”:

Parte 1: Vista General

Parte 2: Planeación y Administración

Parte 3: Proceso para Desarrolladores

Parte 4: Proceso para Adquisidores

Parte 5: Proceso para Evaluadores

Parte 6: Documentación de módulos de evaluación

La primera parte de la ISO/IEC 14598 hace la introducción de las otras partes, define los términos técnicos usados en las demás partes, contiene los requisitos generales de la especificación y evaluación de la calidad de software y clarifica los conceptos generales. Adicionalmente, provee un marco de trabajo para la evaluación de la calidad de todos los tipos de productos de software y especifica los requisitos para los métodos de evaluación y medición de un producto de software.

En la evaluación de un producto de software los evaluadores deben seguir el proceso descrito en la norma ISO/IEC 14598: – Parte 5 (proceso para evaluadores). Esta parte de la norma provee requisitos y recomendaciones para la implementación práctica de una evaluación del producto de software cuando muchas partes necesitan entender, aceptar y creer en los resultados, se definen las actividades que se requieren para analizar los requisitos de evaluación, para especificar, diseñar y realizar acciones de evaluación y concluir la evaluación de cualquier tipo de producto de software. El proceso de evaluación puede ser usado para evaluar productos ya existentes, componentes del producto que se requieren y que ya están disponibles, o para evaluar productos que se estén desarrollando.

Esta parte de la norma la pueden usar:

- Evaluadores en un laboratorio de pruebas, cuando se proveen los servicios de evaluación de un producto de software.
- Proveedores de software, cuando se planea la evaluación de sus productos, incluyendo la evaluación que se vaya a hacer por medio de servicios independientes.
- Compradores de software, cuando un servicio de prueba o un proveedor solicita la información de evaluación.
- Usuarios, cuando evalúan productos o cuando usan reportes de evaluación proporcionados por los laboratorios de prueba.
- Cuerpos de certificación, cuando definen nuevos esquemas de certificación para productos de software.

2.4 Norma internacional ISO/IEC 9126.¹⁸

La Norma Internacional ISO/IEC 9126-1 fue realizada por el comité técnico ISO/IEC JTC 1, “Información Tecnológica”, Subcomité SC 7, “Ingeniería de Software”.

Las características de calidad definidas en la norma ISO/IEC 9126 pueden ser usadas para especificar requisitos (tanto funcionales como no funcionales) de usuarios y clientes.

La ISO/IEC 9126 consiste de las siguientes partes, bajo el título de “Ingeniería de software - calidad de producto”:

- Parte 1: Modelo de Calidad.
- Parte 2: Métricas Externas.
- Parte 3: Métricas Internas.
- Parte 4: Métricas de Calidad en uso.

La parte 1 de la norma ISO/IEC 9126 describe un modelo de dos partes para un producto de software de calidad.

La primera parte del modelo especifica seis características de calidad interna y externa, las cuales son subdivididas en sub-características. Estas sub-características se manifiestan externamente cuando el software es usado como una parte del sistema computacional, y son el resultado de unos atributos del software interno.

¹⁸ ISO/IEC 9126-1:2001. International standard “Software engineering – product quality – Part 1: Quality model.

La segunda parte del modelo especifica seis características de calidad en uso, pero no elabora el modelo para la calidad de uso del nivel inferior de las características.

Este modelo es de vital importancia, puesto que para evaluar la calidad de un producto de software a través de algunos medios cuantitativos, se requiere un conjunto de características de calidad que describan el producto y formen la base de la evaluación. Las características definidas son aplicables a cualquier clase de software, incluyendo programas de computador y los datos contenidos en un firmware. Las características y sub-características proveen terminología consistente a un producto de calidad de software. También proveen un marco de trabajo para especificar los requisitos de calidad de un software, y especifica cambios entre las capacidades de un producto de software.

Esta parte de la ISO/IEC 9126 permite que un producto de software de calidad sea especificado y evaluado desde diferentes perspectivas asociadas con la adquisición, requisitos, desarrollo, revisión, evaluación, soporte, mantenimiento y calidad confiable, asociada a un software. Este modelo, por ejemplo, puede ser usado por desarrolladores, funcionarios de calidad confiable y evaluadores independientes, particularmente aquellos responsables de especificar y evaluar la calidad de un producto de software.

Ejemplos de usos del modelo de calidad definido en esta parte de la ISO/IEC 9126, son:

- Validar la integridad de una definición de requisitos
- Identificar requisitos de Software
- Identificar objetivos de un diseño de software

- Identificar objetivos de una “evaluación” de software
- Identificar criterios de aceptación de un producto de software terminado.

2.5 Estructura de la ISO/IEC 14598 y la ISO/IEC 9126¹⁹

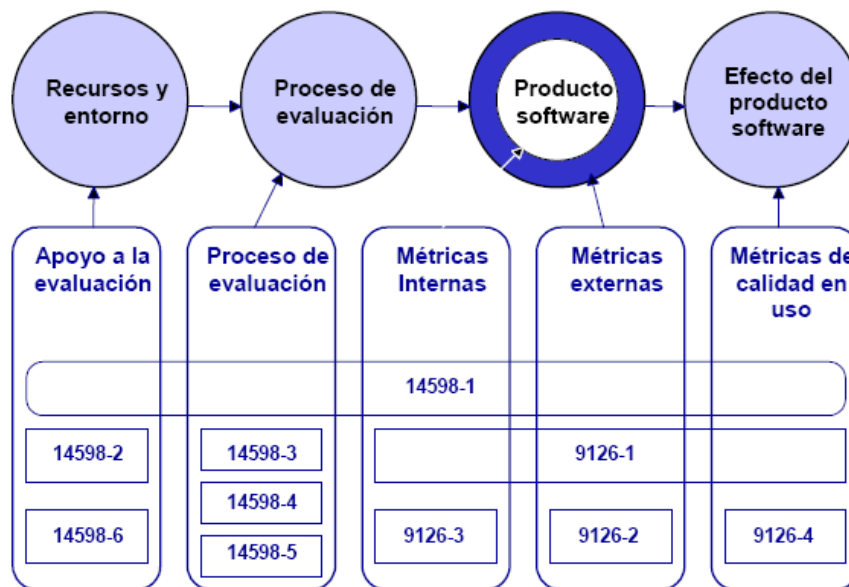


Figura 4 – Relación entre los estándares internacionales ISO/IEC 9126 e ISO/IEC 14598²⁰

Las series ISO/IEC 9126 de estándares internacionales y de informes técnicos definen un modelo de uso general de la calidad, características de la calidad y dan ejemplos de métricas.

Las series ISO/IEC 14598 de estándares internacionales dan una descripción de los procesos de evaluación de los productos de software y proporcionan la dirección y requisitos para la evaluación. Las partes 2 y 6 se relacionan con el nivel corporativo o departamental para la gerencia y la ayuda de la evaluación, mientras las partes 3, 4 y 5 proporcionan requisitos y dirección para la evaluación a nivel del proyecto.

¹⁹ ISO/IEC 25000. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE.

²⁰ ISO/IEC 9126-1:2001. International standard “Software engineering – product quality – Part 1: Quality model.

2.6 Modelo de calidad para calidad externa e interna

La norma ISO/IEC 9126-1:2001 define un modelo de calidad para evaluar la calidad de un producto software, ya sea de manera interna o externa. Esta consta de 6 características, las cuales son:

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Capacidad de mantenimiento
- Portabilidad

Éstas características se distribuyen en el modelo de la siguiente manera con sus respectivas subcaracterísticas, como se ilustra en la gráfica siguiente:

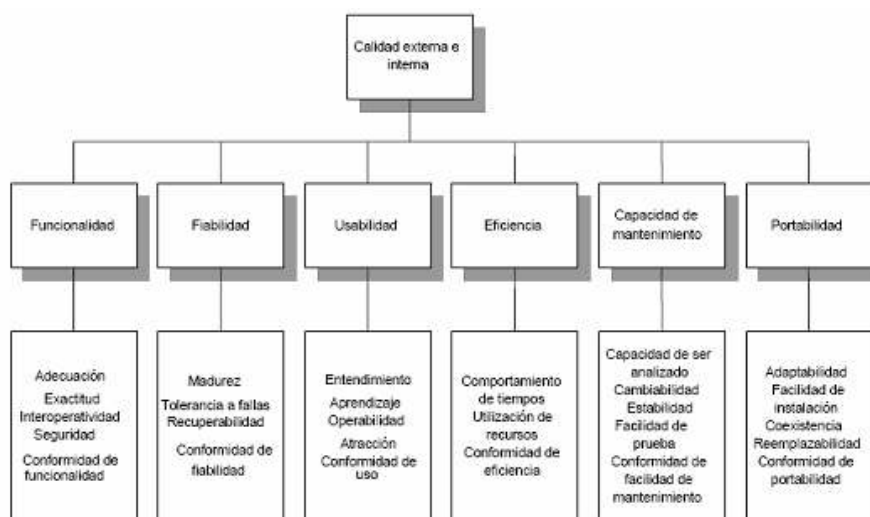


Figura 5 - Modelo de calidad para calidad externa e interna ²¹

²¹ ISO/IEC 9126-1:2001. International standard “Software engineering – product quality – Part 1: Quality model.

Dado la importancia que tiene dentro del desarrollo de un producto de software, las subcaracterísticas seguridad e interoperabilidad pasaron a ser parte de las características ya existentes, lo que re-definió el modelo de calidad propuesto por la norma, convirtiéndose en un nuevo modelo de calidad que contiene ocho características. Este modelo es propuesto en la norma SQuaRE - ISO/IEC CD 25010.

En el gráfico, a continuación se ilustra el nuevo modelo de calidad donde se incluyen las nuevas características seguridad e interoperabilidad, y sus respectivas subcaracterísticas.

Las características pueden ser medidas por métricas internas o externas.

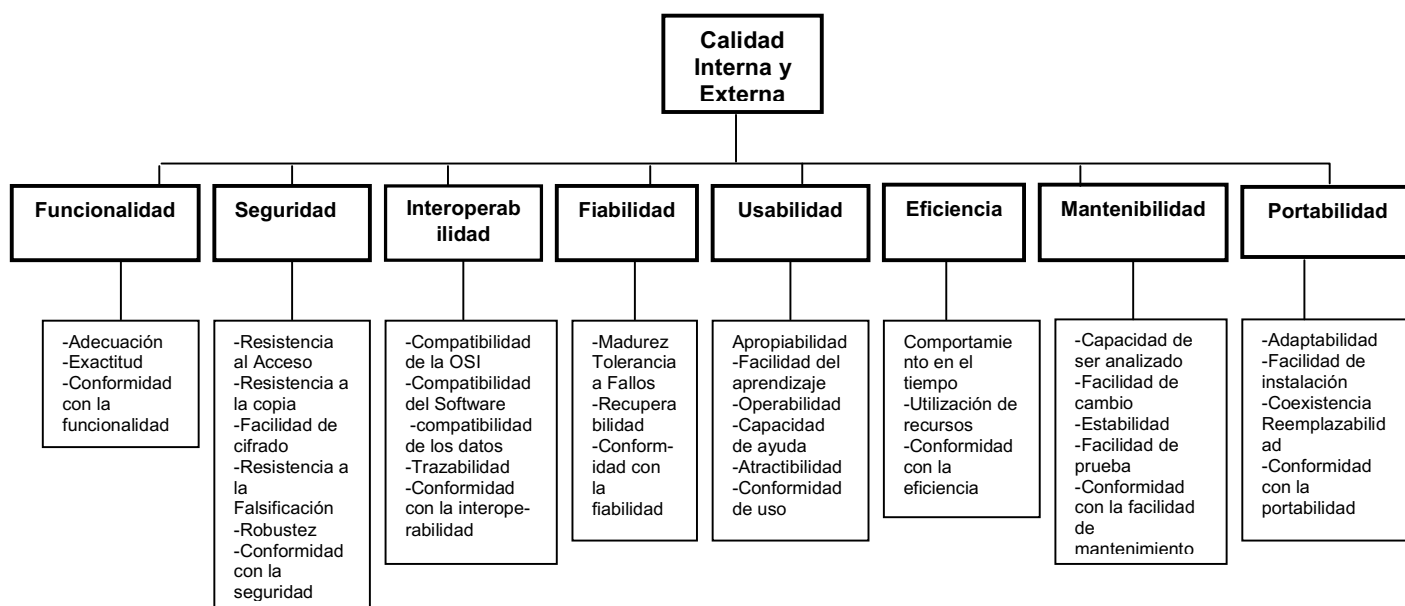


Figura 6 - Modelo de calidad para calidad externa e interna ²²

²² ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

Para cada característica y sub-característica, la capacidad del software es determinada por un conjunto de atributos internos los cuales pueden ser medidos. Las características y sub-características pueden ser medidas externamente por el grado para el cual la capacidad es proveída por el sistema que contiene el software.

2.7 CARACTERÍSTICAS Y SUB-CARACTERÍSTICAS²³

A continuación se describen cada una de las ocho características que hacen parte de la calidad que debe tener todo desarrollo de software y sus respectivas sub-características, según la norma SQuaRE 25010.

2.7.1 Funcionalidad

La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.

2.7.1.1 Adecuación

La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario.

2.7.1.2 Exactitud

La capacidad del producto de software para proveer los resultados o efectos acordados con un grado necesario de precisión.

2.7.1.3 Conformidad con la funcionalidad

La capacidad del producto de software de adherirse a los estándares, convenciones o regulaciones legales y prescripciones similares referentes a la funcionalidad.

²³ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Model.

2.7.2 Seguridad

La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o sistemas autorizados no se les niegue el acceso a ellos.

2.7.2.1 Resistencia al Acceso

La capacidad del software para proteger de accesos ilegales y no autorizados.

2.7.2.2 Resistencia a la copia

La capacidad del producto de software para protegerse de copias ilegales.

2.7.2.3 Facilidad para cifrar

La capacidad del producto de software para proteger de buscadores ilegales por encriptación.

2.7.2.4 Resistencia a la falsificación

La capacidad del producto de software para no permitir el análisis de la estructura interna y de datos almacenados.

2.7.2.5 Robustez

La capacidad del producto de software para recuperarse de entradas y situaciones anómalas.

2.7.2.6 Conformidad con la seguridad

La capacidad del producto de software para adherirse a estándares, convenciones o regulaciones en lo relacionado con seguridad.

2.7.3 Interoperabilidad

La capacidad del producto de software de interactuar con uno o más sistemas especificados. La interoperabilidad se utiliza en lugar de compatibilidad para evitar una posible ambigüedad con la reemplazabilidad.

2.7.3.1 Compatibilidad de la OSI

La capacidad del producto de software para interactuar con uno o más sistemas especificados en cada nivel de la capa de la OSI (Open Systems Interconnection)

2.7.3.2 Compatibilidad del software

La capacidad del producto de software para ser cooperativamente operable con uno o más productos de software.

2.7.3.3 Compatibilidad de los datos

La capacidad del producto de software para intercambiar los datos con uno o más sistemas especificados.

2.7.3.4 Trazabilidad

La capacidad del producto software para registrar los eventos operacionales para analizar la causa de la interacción.

2.7.3.5 Conformidad con la interoperabilidad

La capacidad del producto de software para adherirse a los estándares, convenciones o regulaciones en lo relacionado con la interoperabilidad.

2.7.4 Fiabilidad

La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo condiciones especificadas.

2.7.4.1 Madurez

La capacidad del producto de software para evitar fallas como resultado de errores en el software.

2.7.4.2 Tolerancia a errores

La capacidad del producto de software para mantener un nivel especificado de funcionamiento en caso de errores del software o de incumplimiento de su interfaz especificada.

2.7.4.3 Recuperabilidad

La capacidad del producto de software para restablecer un nivel especificado de funcionamiento y recuperar los datos afectados

directamente en el caso de una falla. Algunas veces un producto de software se “caerá” por cierto periodo de tiempo, el lapso de éste periodo es impuesto por la recuperabilidad.

2.7.4.4 Conformidad con la fiabilidad

La capacidad del producto de software para adherirse a las normas, convenciones o regulaciones relativas a la fiabilidad.

2.7.5 Usabilidad

La capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas.

2.7.5.1 Apropiabilidad (Comprensibilidad)

La capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación.

2.7.5.2 Facilidad de aprendizaje

La capacidad del producto de software para permitir al usuario aprender su aplicación. Un aspecto importante a considerar aquí es la documentación del software.

2.7.5.3 Operabilidad

La capacidad del producto de software para permitir al usuario operarlo y controlarlo. Aspectos de adaptación, facilidad de cambio e instalación

pueden afectar la operabilidad. Operabilidad corresponde a la conformidad, tolerancia a error y control que concuerdan con las expectativas del usuario.

2.7.5.4 Capacidad de ayuda (Helpfulness)

La capacidad del producto de software para proveer ayuda cuando el usuario necesita asistencia.

2.7.5.5 Atractibilidad (Likability)

La capacidad del producto de software de ser atractivo al usuario. Se refiere a los atributos del software deseados para hacer éste más atractivo al usuario, tales como el uso del color y los diseños gráficos.

2.7.5.6 Conformidad de uso

La capacidad del producto de software para adherirse a los estándares, convenciones, guías de estilo o regulaciones relacionadas con su usabilidad.

2.7.6 Eficiencia

La capacidad del producto de software para proveer un desempeño adecuado, de acuerdo con la cantidad de recursos utilizados y bajo las condiciones planteadas.

2.7.6.1 Comportamiento en el tiempo

La capacidad del producto de software para proveer tiempos adecuados de respuesta y procesamiento, y ratios de rendimiento cuando realiza su función bajo las condiciones establecidas.

2.7.6.2 Utilización de recursos

La capacidad del producto de software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo las condiciones establecidas.

2.7.6.3 Conformidad con la eficiencia

La capacidad del producto de software para adherirse a estándares o convenciones relacionados con la eficiencia.

2.7.7 Mantenibilidad

Capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, especificaciones de requerimientos funcionales.

2.7.7.1 Capacidad de ser analizado

La capacidad del producto de software para ser diagnosticado por deficiencias o causas de fallas en el software o la identificación de las partes a ser modificadas.

2.7.7.2 Facilidad de cambio

La capacidad del software para permitir que una determinada modificación sea implementada. La implementación incluye codificación, diseño y documentación de cambios.

2.7.7.3 Estabilidad

La capacidad del producto de software para evitar efectos inesperados debido a modificaciones del software.

2.7.7.4 Facilidad de prueba

La capacidad del software para permitir que las modificaciones sean validadas.

2.7.7.5 Conformidad con la facilidad de mantenimiento

La capacidad del software para adherirse a estándares o convenciones relativas a la facilidad de mantenimiento.

2.7.8 Portabilidad

La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.

2.7.8.1 Adaptabilidad

La capacidad del producto software para ser adaptado a diferentes entornos especificados sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado.

2.7.8.2 Facilidad de instalación

La capacidad del producto de software para ser instalado en un ambiente especificado.

2.7.8.3 Coexistencia

La capacidad del producto de software para coexistir con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes.

2.7.8.4 Reemplazabilidad

La capacidad del producto de software para ser utilizado en lugar de otro producto de software, para el mismo propósito y en el mismo entorno. Por ejemplo, la reemplazabilidad de una nueva versión de un producto de software es importante para el usuario cuando se vaya a actualizar.

2.7.8.5 Conformidad con la portabilidad

La capacidad del software para adherirse a estándares o convenciones relacionados con la portabilidad.

2.8 Modelo de calidad para la calidad del software en uso ²⁴

Así como el modelo de calidad para calidad externa e interna cambió, el modelo de calidad para la calidad del software en uso pasó de tener 4 componentes como se expone en la norma ISO/IEC 9126-1:2001 a tener 6 componentes como se expone en la norma SQuaRE - ISO/IEC CD 25010.

Para entender mejor lo mencionado anteriormente, se hará un comparativo entre los modelos de calidad para la calidad del Software en uso planteado en la norma ISO/IEC 9126-1:2001 y en la norma SQuaRE - ISO/IEC CD 25010.



Figura 7 - Modelo de calidad para la calidad del software en uso ²⁵

²⁴ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality Model.

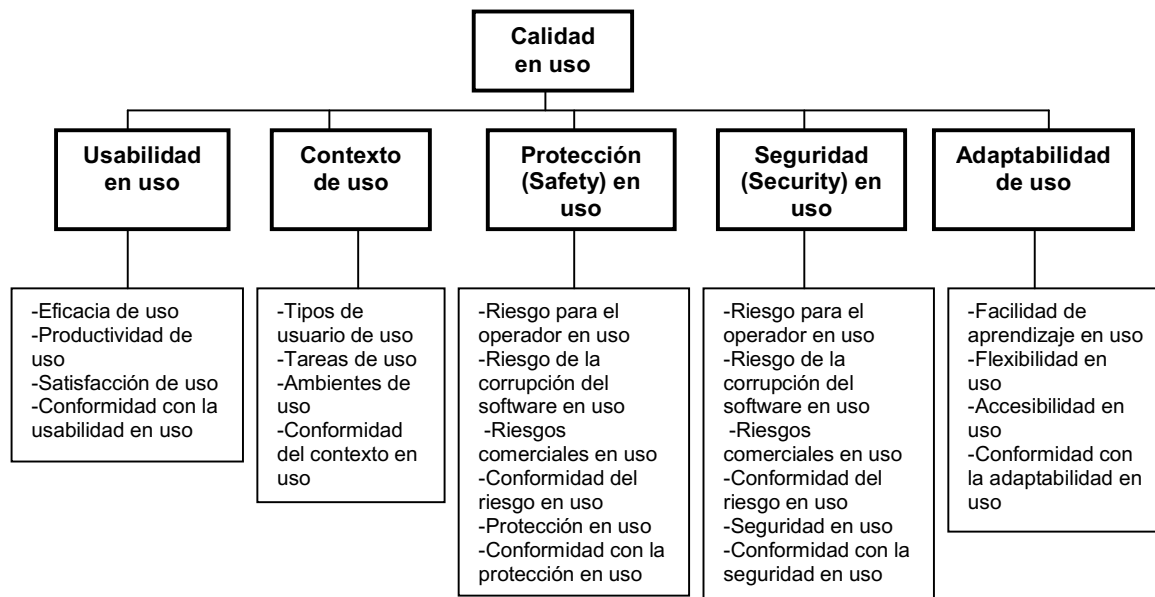


Figura 8 - Modelo de calidad para la calidad del software en uso²⁶

Calidad en uso es una medida de la calidad del sistema en su ambiente operacional. Es determinado por la naturaleza del software, del hardware, del ambiente de funcionamiento, y de las características de los usuarios, de las tareas y del ambiente social. Todos estos factores contribuyen a la calidad en uso.

La calidad en uso del software se refiere a la capacidad del software para alcanzar la calidad en uso para los contextos específicos del uso.

Alcanzar la calidad en uso depende de alcanzar la calidad externa necesaria, que alternadamente depende de alcanzar la calidad interna necesaria. Las medidas son normalmente requeridas en los tres niveles; así, resolver los criterios para las medidas internas no son generalmente suficientes para asegurar el logro de los criterios para las medidas externas, y resolver los criterios para las medidas

²⁶ ISO/IEC 9126-1:2001. International standard "Software engineering – product quality – Part 1: Quality model.

externas de las sub-características no son generalmente lo suficiente para asegurar la realización de los criterios para la calidad en uso.

2.8.1 Calidad en uso del software

La capacidad del producto de software les permite a usuarios específicos alcanzar con protección y seguridad metas específicas con la eficacia, productividad, satisfacción en contextos específicos del uso.

La calidad en uso es la vista del usuario de la calidad de un ambiente que contiene software, y es medido desde los resultados al usar el software en el ambiente, más que las características del software en sí mismo.

2.8.2 Usabilidad en uso

La capacidad del producto de software para permitir a usuarios específicos alcanzar metas específicas con eficacia, productividad y satisfacción en contextos específicos del uso.

2.8.2.1 Eficacia de uso

La capacidad del producto de software para permitir a usuarios alcanzar metas específicas con exactitud y completitud en un contexto específico del uso.

2.8.2.2 Productividad de uso

La capacidad del producto de software para permitir a usuarios consumir cantidades apropiadas de recursos en lo referente a la eficacia alcanzada en un contexto especificado del uso.

2.8.2.3 Satisfacción de uso

La capacidad del producto de software para satisfacer a usuarios en un contexto específico del uso.

La satisfacción de uso se subdivide más a fondo en las siguientes subsubcaracterísticas:

- Atractividad (satisfacción cognoscitiva)
- Placer (satisfacción emocional)
- Comodidad (satisfacción física)

2.8.2.4 Conformidad con la Usabilidad en uso

La capacidad del producto de software para adherirse a estándares o convenciones relacionadas con la usabilidad en uso.

2.8.3 Contexto de uso

La capacidad del producto de software para proveer la usabilidad en uso aceptable en todos los conceptos requeridos del uso.

2.8.3.1 Tipos de usuario de uso

La capacidad del producto de software para proveer la usabilidad de uso aceptable para todos los tipos de usuarios requeridos.

2.8.3.2 Tareas de uso

La capacidad del producto de software para proveer la usabilidad de uso aceptable para todas las tareas requeridas.

2.8.3.3 Ambientes de uso

La capacidad del producto de software para proveer la usabilidad de uso aceptable para todos los ambientes organizacionales, físicos y técnicos requeridos.

2.8.3.4 Conformidad del contexto en uso

La capacidad del producto de software para adherirse a estándares o convenciones relacionadas con el contexto en uso.

2.8.4 Seguridad en uso

2.8.4.1 Riesgo para el operador en uso

La capacidad del producto de software para proporcionar niveles de riesgo aceptables del daño al público en contextos previstos del uso.

2.8.4.2 Riesgo de la corrupción del software en el uso

La capacidad del producto de software de suministrar niveles de riesgo aceptables frente a la corrupción del software respecto de los contextos de uso previstos.

2.8.4.3 Riesgos comerciales del uso

La capacidad del producto de software de suministrar niveles de riesgo aceptables frente a una falla que redundaría en daños comerciales o daños en la reputación respecto de los contextos de uso previstos.

Nota: Los riesgos generalmente son consecuencia de las deficiencias en la funcionalidad (incluyendo la seguridad), confiabilidad, usabilidad o mantenibilidad.

2.8.4.4 Conformidad del riesgo en uso

La capacidad del producto de software para adherirse a estándares o convenciones relacionadas con la seguridad en uso.

2.8.4.5 Seguridad en uso

La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos y que a las personas o los sistemas autorizados no se les niegue el acceso a ellos.

2.8.4.6 Conformidad con la la seguridad en uso

La capacidad del producto de software para adherirse a estándares o convenciones relacionadas con la seguridad en uso.

2.8.5 Adaptabilidad de uso

2.8.5.1 Facilidad de aprendizaje en uso

La calidad en uso de los procesos de aprendizaje.

2.8.5.2 Flexibilidad en uso

La capacidad del producto de software para proporcionar software de calidad en uso en el rango más amplio de contextos del uso.

2.8.5.3 Accesibilidad en uso

La capacidad del producto de software para proporcionar software de calidad en uso para gente con el más amplio rango de capacidades.

2.8.5.4 Conformidad con la adaptabilidad en uso

La capacidad del producto de software para adherirse a estándares o convenciones relacionadas con la adaptabilidad en uso.

2.9 Ventajas de los modelos / Estándares de calidad del software²⁷

Las ventajas de implantar modelos o estándares de calidad del software son:

- *Tener una oportunidad para corregir los procesos de software que se hayan desajustado con el tiempo.*

Cuando se usa un modelo de calidad se realiza un seguimiento al proceso de desarrollo, esto permite que se puedan hacer correcciones en un momento dado y asegurar la obtención de un buen producto.

- *Clasificar a las empresas como de clase mundial.*

Las empresas tienden a ser las mejores en su campo. El uso de modelos de calidad estandar les permite diferenciar las empresas de muchas otras que no tienen un modelo establecido para evaluar la calidad de sus productos.

- *Lograr que la empresa de software sea más competitiva.*
- *Certificar la competitividad internacional requerida para competir en todos los mercados.*

El uso adecuado de los modelos de calidad permite estar a la vanguardia de las mejores empresas, haciendo que crezca su competitividad en el medio.

- *Cambiar la actitud del personal de la empresa.*
- *Desarrollar y mejorar el nivel y la calidad de vida del personal.*

Al aplicar un modelo de calidad de software las personas tienen un proceso de desarrollo/evaluación definido y claro que facilita las labores.

²⁷ SCALONE, Fernanda. Software quality management. Overview sobre modelos/estándares de calidad del SW. Diciembre 02. Año 2006.

- *Realizar una mejora continua en la calidad de los procesos de software utilizados, servicios y productos de software.*

- *Reducir los costos en todos los procesos.*

Al aplicar un modelo de calidad los procesos se vuelven mas organizados, lo que repercute en la agilización de los mismos y genera reducción de costos.

- *Aumentar la productividad, efectividad y utilidad de la empresa.*

El adoptar un modelo de calidad permite tener un proceso definido y bien documentado, hay una definición y comprensión de los roles de trabajo y hay una coherencia con el trabajo que se está realizando.

- *Asegurar la satisfacción de los clientes internos y externos.*

- *Tener aceptación total de los clientes.*

- *Tener productos de software y servicios con valor agregado.*

- *Generar una cultura organizacional enfocada a cumplir con los requisitos de los clientes.*

El uso de modelos de calidad genera confianza entre los clientes, pues esto les genera seguridad en que el producto terminado será de muy buena calidad.

- *Tener permanentemente mejores procesos, productos de software y servicios.*

El uso de un modelo de calidad permite obtener conocimiento del proceso establecido en una empresa, lo que a futuro se manifiesta en una mejora de los productos y del mismo desempeño del proceso.

- *Tener criterios de medición e indicadores congruentes que se utilizan en la empresa para comparar respecto de las mejores prácticas, para conocer fortalezas y debilidades de la empresa y establecer las estrategias necesarias para realizar mejoras.*

Al tener un modelo de calidad, las mediciones que salgan a partir del uso de este sirven como histórico para otros futuros desarrollos.

3. PASOS A REALIZAR PARA EL PROCESO DE UTILIZACIÓN DE LAS MÉTRICAS

Hemos visto anteriormente, algunos de los modelos de calidad que más se usan para evaluar la calidad de un producto software, así como las características y subcaracterísticas que se deben medir para alcanzar este propósito. Esas mediciones se hacen a través de las métricas. Seguidamente se expondrá la manera en la cual se debe seguir todo el proceso de evaluación de la calidad desde que se define y se aplica el modelo de calidad.

Una vez se aplique un modelo para evaluar la calidad, se inicia el proceso de medición, para lo cual se seleccionan las métricas a evaluar en el producto de software. Los resultados obtenidos son valores expresados en las escalas de las métricas. Luego se pasa a comparar con los criterios, es decir, se hace un comparativo de los valores medidos con los criterios predeterminados, para finalmente valorar esos resultados y realizar un reporte de la evaluación. Con esto se puede determinar el grado de satisfacción con que el modelo refleja la calidad del producto evaluado.

Lo anterior se puede detallar en 8 pasos, que identifican el proceso a seguir para realizar una adecuada utilización de las métricas. Estos pasos son:

Paso 1. Determinar el proyecto y la aplicación a evaluar

Paso 2. Seleccionar el modelo o estándar de calidad del software

Paso 3. Establecer las características, subcaracterísticas y métricas a evaluar

Paso 4. Calcular las métricas respectivas y los promedios ponderados a nivel

Subcaracterística y característica

Paso 5. Determinar el cumplimiento de las características, subcaracterísticas y métricas

Paso 6*. Evaluar y analizar los resultados

Paso 7*. Comunicar los resultados

Paso 8*. Identificar mejoras potenciales²⁸

*Ampliación de los pasos 6, 7 y 8.

Una vez que se realice la medición utilizando las métricas y se conozcan los resultados obtenidos, se procede a hacer una comparación de esos resultados a nivel de las características y subcaracterísticas. Se analiza cual característica obtuvo un mejor valor, y cuales subcaracterísticas cumplen con los propósitos establecidos inicialmente de acuerdo a una escala definida previamente por el usuario. Los resultados son confrontados con las expectativas que se tenían antes de la evaluación y son comunicados al equipo de trabajo. Finalmente se sacan conclusiones al respecto, por ejemplo: se detectan las debilidades en cada característica.

²⁸ SCALONE, Fernanda. "Estudio comparativo de los modelos y estándares de calidad del software". Maestría en ingeniería en calidad. Universidad tecnológica nacional. Facultad regional Buenos Aires. Junio. Año 2006.

4. MÉTRICAS DE CALIDAD DE PRODUCTO SOFTWARE

Una vez definidos algunos de los modelos utilizados para medir la calidad de un producto de software, es importante conocer qué son y cómo se componen las métricas de calidad.

“Las métricas son un método definido de valoración y su escala de valoración, las cuales incluyen métodos para clasificar los datos o la información cualitativa en diferentes categorías”²⁹. Las métricas pueden ser medidas de maneras diferentes; existen formas cualitativas y cuantitativas de representar los valores dados a cada una de las métricas evaluadas. Sin embargo, las calificaciones otorgadas a esas métricas deben expresarse en un rango de porcentajes, que son los que finalmente permiten obtener un valor numérico, es decir una representación numérica de la calificación. Pueden clasificarse en métricas internas (ISO/IEC 9126-2), métricas externas (ISO/IEC 9126-3) y métricas de Calidad en uso (ISO/IEC 9126-4).

“Las métricas internas pueden ser aplicadas a un producto de software no-ejecutable (como una especificación o código fuente) durante el diseño y la codificación. Estas métricas proporcionan a los usuarios, evaluadores, verificadores y desarrolladores, el beneficio de poder evaluar la calidad del producto de software y lo referido a problemas de calidad, antes que el producto de software sea puesto en ejecución.

Las métricas externas usan medidas de un producto de software, derivadas del comportamiento del mismo, a través de la prueba, operación y observación del software. Estas métricas proporcionan a los usuarios, evaluadores, verificadores y

²⁹ PRESIDENCIA DEL CONSEJO DE MINISTROS (PERÚ). “Guía técnica sobre evaluación de software en la administración pública”. Mayo 05. Año 2004.

desarrolladores, el beneficio de que puedan evaluar la calidad del producto de software durante las pruebas o el funcionamiento”.³⁰

“Las métricas de calidad en el uso miden la extensión de un producto que reúne las necesidades especificadas por los usuarios para lograr las metas propuestas, con la efectividad, productividad, seguridad y satisfacción en un contexto de uso específico. La evaluación de la calidad en uso valida la calidad del producto de software en los escenarios específicos de tareas de usuario.

La calidad en el uso es la vista del usuario sobre la calidad que el sistema de software contiene y es medida en términos de resultados de uso del software, en lugar de las propiedades del propio software. La calidad en el uso es el efecto combinado de calidad interna y externa para el usuario”.³¹

Las métricas que se han de emplear dependen del atributo de calidad a evaluar, ya que cada atributo contiene una cantidad de métricas diferentes que permiten medir características respectivas del atributo. Debido a que la calidad es difícil de definir, se ha abordado este término en ocho atributos que permiten una definición más sencilla, los cuales son definidos por la norma SQuaRE como:

- Funcionalidad
- Seguridad
- Interoperabilidad
- Fiabilidad
- Eficiencia

³⁰ SCALONE, Fernanda. “Estudio comparativo de los modelos y estándares de calidad del software”. Maestría en ingeniería en calidad. Universidad tecnológica nacional. Facultad regional Buenos Aires. Junio. Año 2006.

³¹ PRESIDENCIA DEL CONSEJO DE MINISTROS (PERÚ). “Guía técnica sobre evaluación de software en la administración pública”. Mayo 05. Año 2004.

- Usabilidad
- Mantenibilidad
- Portabilidad

A su vez, cada una de estas características del software se han subdividido en atributos aún más concretos, como se mencionó anteriormente [17] .Cada uno de esos atributos tiene distintas métricas, las cuales serán descritas más adelante. Primero se ampliará la información recolectada en el trabajo de campo realizado para la obtención de las métricas, luego se dará a entender el uso de la tabla donde se encuentra la información correspondiente a cada métrica y finalmente se proporcionarán las distintas métricas internas y externas.

4.1. Trabajo de campo

La búsqueda de información correspondiente a las métricas internas y externas de calidad se realizó por medio de documentos (normas ISO/IEC 9126-2:2003 y ISO/IEC 9126-3:2003) donde se explicaban cada una de las métricas. Como la norma SQuaRE define como características a la seguridad y a la interoperabilidad y presenta sus propias sub-características, las métricas que les correspondían según la norma ISO/IEC 9126:2003 no tenían claridad para ser asignadas a las nuevas sub-características.

Por tanto, buscar a cuál de las sub-características de la seguridad e Interoperabilidad respectivamente se podían acomodar las métricas consignadas en la norma ISO/IEC 9126:2003, fue uno de los objetivos perseguidos inicialmente en el trabajo de campo. Una vez se conocía la asignación de las métricas a dichas sub-características, se pudo pasar a investigar otras métricas que complementaran las ya establecidas para estas dos características.

En esta investigación se hizo un mayor énfasis en las métricas de seguridad e interoperabilidad, pero también se miraron otras características. Sin embargo, no fue posible conseguir la obtención de todas las métricas, pues algunas resultaron ser difíciles de plantear. Para la mayoría de las sub-características de seguridad e interoperabilidad se obtuvo tanto la métrica externa como la interna, pero en algunas solo se logró conseguir o bien la métrica interna, o bien la externa.

La investigación se realizó en tres diferentes empresas de software con el propósito de complementar la información previamente recolectada sobre las métricas. La información encontrada fue revisada con el experto entrevistado en cada una de las empresas, con el fin de que no se repitiera la información investigada con anterioridad.

Esta investigación se apoya en el modelo de calidad propuesto por la norma ISO/IEC CD 25010 (SQuaRE), la cual plantea el esquema del nuevo modelo de calidad para la evaluación de un producto software.

La búsqueda de las métricas nuevas fue hecha en dos empresas ubicadas en Medellín y la otra en Cali. Esto con el fin de que la información recogida no tuviera el mismo enfoque en las tres empresas, es decir, que tuviera distintas percepciones.

Las empresas visitadas en la ciudad de Medellín fueron Fluidsignal Group S.A.³² y Suramericana de Seguros S.A.³³; la empresa contactada en Cali fue Parquesoft³⁴.

Las entrevistas fueron realizadas a diferentes personas³⁵ que tenían un conocimiento sobre el tema de calidad de software en las respectivas empresas. En las dos primeras empresas se hicieron reuniones en contacto directo con la persona encargada, en donde se comenzó haciendo una introducción del tema a investigar y posteriormente se revisaron las métricas ya consultadas en la norma ISO/IEC 9126, con el fin de poder definir nuevas métricas de las características Seguridad e Interoperabilidad y considerar las ya definidas.

³² FluidSignal. Empresa consultora conformada con el fin de brindar asesoría en la implantación de sistemas de gestión para la seguridad de la información y en toda clase de soluciones alrededor de la seguridad de la información.

³³ Suramericana de seguros S.A. Es una compañía de seguros de vida, gerenciales y capitalización.

³⁴ GreenSQA – ParqueSoft. Es una empresa que hace parte de ParqueSoft. Ofrece servicios a pruebas de software y consultoría de clase mundial.

³⁵ Alvarez Correa, Juan Rafael. Ingeniero preventa. Fluidsignal Group S.A.

Taborda Arias, Andrés Felipe. Analista de seguridad informática. Suramericana de seguros S.A

Hoyos Carvajal, Patricia. Líder del programa de calidad de software de GreenSQA – ParqueSoft.

En la tercera empresa se hizo la entrevista por medio de una teleconferencia, realizada a través de Internet; esto debido a que la empresa no se localizaba en la misma ciudad donde se estaba realizando la investigación. De la misma manera que en las dos primeras empresas visitadas, en esta se usó el mismo esquema para la entrevista.

Las entrevistas tuvieron una duración entre 60 y 90 minutos.

Al momento de realizar las entrevistas, se hizo una introducción sobre el trabajo de grado, lo que se quería lograr con él y el aporte que se lograba al encontrar más métricas. Para esto, no se tuvo un formato de preguntas formuladas, sino que se comenzó haciendo preguntas abiertas referentes al conocimiento de la norma ISO/IEC 9126 en la empresa. Luego se le indicó a la persona entrevistada cuáles eran las métricas existentes y el cambio que proponía la norma SQuaRE para el nuevo modelo de calidad.

Una vez indicado esto, se pasó a distribuir las métricas existentes en la norma ISO/IEC 9126:2003 que necesitaban ser puestas en las características Seguridad e Interoperabilidad, según correspondían.

El análisis de esta asignación fue realizada en la primera entrevista, en la empresa Fluidsignal y en esta misma entrevista se definieron nuevas métricas para ambas características.

La nueva métrica interna que se definió para la característica seguridad fue conformidad con la seguridad, y para la característica Interoperabilidad fue conformidad con la interoperabilidad.

Las nuevas métricas externas que se definieron para la característica seguridad fueron: resistencia a la copia y resistencia a la falsificación; para la característica Interoperabilidad fueron: compatibilidad del software y trazabilidad.

Cuando fue posible hacer la segunda entrevista, ésta se realizó en Suramericana. Como ya se había realizado la asignación de las métricas de seguridad e interoperabilidad planteadas en la norma ISO/IEC 9126:2003 en la entrevista anterior, en esta se pasó a mirar el trabajo realizado en esa primera entrevista y posteriormente se logró definir una métrica más a las que ya se tenían consignadas. Esta nueva métrica externa correspondiente a la característica interoperabilidad fue: compatibilidad de los datos.

Finalmente, se debatió con el entrevistado el uso de la norma ISO/IEC 9126 dentro de la empresa, la aceptación por parte de la misma y las dificultades o beneficios obtenidos con ella.

Dos semanas después se hizo el contacto con la tercera empresa ubicada en Cali. Con la entrevistada de la empresa ParqueSoft, se usó un esquema muy parecido a las dos entrevistas anteriores, pues se comenzó realizando una introducción del trabajo de grado, el propósito que se quería lograr con la investigación y el posterior análisis de los datos recolectados en las dos anteriores entrevistas.

En esta empresa no se adicionaron nuevas métricas, pues la persona entrevistada no tenía mucho conocimiento de las métricas que manejaban a nivel de la seguridad, ya que éstas eran manejadas por outsourcing. La métrica de interoperabilidad que usaban, estaba contenida dentro de las que ya se había investigado antes.

Como la empresa se enfoca más en las características de funcionalidad y usabilidad, y sobre estas características ya se tenían las métricas

correspondientes, de todas maneras se discutió sobre las métricas de funcionalidad, pero no se hizo un aporte de nuevas métricas, ya que no era necesario.

Así se concluyó con el proceso de entrevistas a las diferentes empresas consultadas, obteniendo un total de siete métricas nuevas para las características seguridad e interoperabilidad, y complementando las ya existentes.

El aporte que tuvieron las nuevas métricas consultadas, fue de mucha utilidad para este trabajo, puesto que se complementó el compilado de métricas y a su vez, permitió una adecuada realización de la plantilla desarrollada donde se puede evaluar la calidad de un producto software.

Con esta investigación se permitió conocer el aporte que tiene la norma ISO/IEC 9126 dentro de algunas de las empresas que se dedican al desarrollo de software.

Lo que se pudo observar en estas empresas, es que la norma ISO/IEC 9126 les sirve como una base para plantear sus propias métricas de calidad, ya que ninguna contempla la totalidad de las características y subcaracterísticas propuestas por dicha norma.

A continuación se detallan las métricas que fueron investigadas en las tres empresas donde se realizaron las investigaciones. Para entender como leer las tablas, primero se describirán cada uno de los componentes de la tabla.

4.1.1 Como leer y usar las tablas de las métricas ³⁶

A continuación se explica el uso de las tablas en donde está contenida la información correspondiente a las métricas internas y externas.

La siguiente información está dada para cada métrica en la tabla:

- **Nombre:** Las métricas correspondientes en la tabla para la métrica interna y la métrica externa tienen nombres similares.
- **Propósito:** La pregunta a ser respondida con el uso de la métrica.
- **Método de aplicación:** Proporciona un esbozo de la aplicación.
- **Medición, fórmula:** Proporciona la fórmula de la medida y explica los elementos de los datos usados.
- **Interpretación:** proporciona el rango y valores válidos.
- **Tipo de escala:** Tipo de escala usada por la métrica. Los tipos de escala son: escala nominal, escala ordinal, escala intervalo, escala proporción y escala absoluta.
- **Tipo de medida:** Los tipos de medida usados son: tipo tamaño (por ejemplo, tamaño de la función, tamaño de la fuente), tipo tiempo (ejemplo,

³⁶ ISO/IEC TR 9126-3:2003. International standard “Software engineering – product quality – Part 3: Internal metrics.

tiempo transcurrido, tiempo del usuario), tipo cuenta (por ejemplo, número de cambios, número de fallos)

- **Fuente de medición:** Fuente de datos usados en la medición*.
- **ISO/IEC 12207 SLCP:** Identifica el proceso del ciclo de vida del software donde la métrica es aplicable.
- **Audiencia:** Identifica al usuario de los resultados de la medida.

* Algunos ejemplos para la fuente de medición son: especificación de requisitos, código fuente, informe de revisión, informe de pruebas, informe de resolución de problemas, informe de mantenimiento, diseño, etc.)

Nota: En el anexo D, se amplía la información correspondiente para los tipos de escala y los tipo de medidas.

4.1.2 Métricas investigadas

MÉTRICAS INTERNAS PARA LA SEGURIDAD

- CONFORMIDAD CON LA SEGURIDAD

Tabla1. Métrica interna de conformidad con la seguridad - regulación de la seguridad

Nombre:	Regulación de la seguridad
Propósito:	Cuántas regulaciones de seguridad se están cumpliendo
Método de aplicación:	Evaluar el número de regulaciones de seguridad que se cumplen y compararlas con el número de regulaciones de seguridad que se deberían cumplir según las especificaciones.
Medición, fórmula:	$X = A/B$ A = Número de regulaciones de seguridad que se cumplen. B = Número de regulaciones de seguridad que se deben cumplir según las especificaciones.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador/contador A = contador B = contador
Fuente de medición:	Especificación de requisitos Código fuente Informe de revisión
ISO/IEC 12207 SLCP:	Validación
Audiencia:	Desarrolladores Auditores

MÉTRICAS INTERNAS PARA LA INTEROPERABILIDAD

- CONFORMIDAD CON LA INTEROPERABILIDAD

Tabla 2. Métrica interna de la conformidad con la interoperabilidad - regulación de interoperabilidad

Nombre:	Regulación de interoperabilidad
Propósito:	Cuántas regulaciones de interoperabilidad se están cumpliendo

Método de aplicación:	Evaluar el número de regulaciones de interoperabilidad que se cumplen y compararlas con el número de regulaciones de interoperabilidad que se deberían cumplir según las especificaciones.
Medición, fórmula:	$X = A/B$ A = Número de regulaciones de interoperabilidad que se cumplen. B = Número de regulaciones de interoperabilidad que se deben cumplir según las especificaciones.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador/contador A = contador B = contador
Fuente de medición:	Informe de revisión
ISO/IEC 12207 SLCP:	Validación
Audiencia:	Desarrolladores Auditores

MÉTRICAS EXTERNAS PARA LA SEGURIDAD

- RESISTENCIA A LA COPIA

Tabla 3. Métrica externa de resistencia a la copia - incidentes de copia

Nombre:	Incidentes de copia
Propósito:	Cuántos incidentes de copia ilegal son realizados en el software
Método de aplicación:	Contar el número de incidentes de copia ilegal del software y compararlo con el número total de incidentes de seguridad relacionados con el software.
Medición, fórmula:	$X = A/B$ A = Número de incidentes de copia ilegal del software. B = Número total de incidentes de seguridad relacionados con el software.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 0, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador/contador A = contador B = contador
Fuente de	Informe de revisión

medición:	
ISO/IEC 12207 SLCP:	Validación
Audiencia:	Desarrolladores Auditores

- **RESISTENCIA A LA FALSIFICACIÓN**

Tabla 4. Métrica externa de resistencia a la falsificación - incidentes de falsificación

Nombre:	Incidentes de falsificación
Propósito:	Cuántos incidentes de falsificación son realizados en el software
Método de aplicación:	Contar el número de incidentes de falsificación del software y compararlo con el número total de incidentes de seguridad relacionados con el software.
Medición, fórmula:	$X = A/B$ A = Número de incidentes de falsificación del software. B = Número total de incidentes de seguridad relacionados con el software.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 0, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador/contador A = contador B = contador
Fuente de medición:	Informe de revisión
ISO/IEC 12207 SLCP:	Validación
Audiencia:	Desarrolladores Auditores

MÉTRICAS EXTERNAS PARA LA INTEROPERABILIDAD

- **COMPATIBILIDAD DEL SOFTWARE**

Tabla 5. Métrica externa de la compatibilidad del software - interoperabilidad con el software

Nombre:	Interoperabilidad con el software
Propósito:	Qué tan interoperable es el software con otros programas

Método de aplicación:	Medir la cantidad de software con el que se interactuó de forma satisfactoria y compararlo con la cantidad de software con el que se debió interactuar según los requisitos.
Medición, fórmula:	$X = A/B$ A = Cantidad de software con que se interactuó satisfactoriamente. B = Cantidad de software con que se debe interactuar según las especificaciones.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más interoperable.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador/contador A = contador B = contador
Fuente de medición:	Informe de revisión
ISO/IEC 12207 SLCP:	Validación
Audiencia:	Desarrolladores Auditores

- **COMPATIBILIDAD DE LOS DATOS**

Tabla 6. Métrica externa de la compatibilidad de los datos - intercambio de los datos

Nombre:	Intercambio de los datos
Propósito:	Cantidad de aplicaciones que pueden intercambiar datos
Método de aplicación:	Contar el número de aplicaciones con las que se puede intercambiar datos.
Medición, fórmula:	$X = 1/N$ N = Número de aplicaciones con los cuales se puede intercambiar datos.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 0, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador N = contador
Fuente de medición:	Informe de pruebas
ISO/IEC 12207 SLCP:	Validación Operación
Audiencia:	Auditores

- **TRAZABILIDAD**

Tabla 7. Métrica externa de la trazabilidad - iteraciones grabadas

Nombre:	Iteraciones grabadas
Propósito:	Cuánta trazabilidad tiene el software
Método de aplicación:	Contar el número total de iteraciones grabadas y compararlas con el número total de iteraciones entre las partes con las que se debe interoperar.
Medición, fórmula:	$X = A/B$ A = Total de interacciones grabadas B = Total de interacciones entre las partes con las que se debe interoperar.
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, mayor trazabilidad.
Tipo de escala:	Absoluta
Tipo de medida:	X = contador/contador A = contador B = contador
Fuente de medición:	Informe de revisión
ISO/IEC 12207 SLCP:	Validación
Audiencia:	Desarrolladores Auditores

4.2. Plantilla para evaluar la calidad de un producto software

La plantilla es un desarrollo sencillo creado con el fin de poder realizar la evaluación de un producto de software genérico y/o específico, ya que permite medir las características que contiene un desarrollo de software cualquiera.

Cuando el desarrollo de software es genérico, la plantilla puede utilizarse para establecer el nivel de calidad que se pretende alcanzar con ese producto. Igualmente si el desarrollo es un producto hecho a la medida (específico), es posible establecer los niveles de calidad que se pretenden tener cada vez que se van creando las etapas del desarrollo.

La plantilla fue creada en Microsoft Excel, lo cual permite una fácil manipulación por parte de los usuarios finales.

La creación de la plantilla para evaluar la calidad de un producto software, se desarrolló con las métricas investigadas a partir de la información recolectada y del trabajo de campo. Esta plantilla partió de una plantilla existente³⁷ como se mencionó anteriormente.

La plantilla existente contaba con las seis características propuestas en la norma ISO/IEC 9126:2001 (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y tenía una serie de métricas que fueron consultadas en distintas empresas del medio, pues en ese entonces no había salido el documento de la norma ISO/IEC 9126:2003 donde se proponen las métricas internas y externas para evaluar la calidad de un producto software.

³⁷ Creación y aporte de Juan David Correa, Gerente de Nodrizza Network.

El esquema propuesto en la plantilla base se mantuvo vigente para esta nueva plantilla. Se hicieron cambios en la apariencia agregando color en algunas celdas para hacer más visible los campos y hacer más agradable al usuario el uso de las mismas, pues la anterior no tenía fuerza en este aspecto; además se modificaron las métricas adaptándolas a las propuestas en la norma ISO/IEC 9126-2 e ISO/IEC 9126-3 y se incluyeron las ocho características planteadas en la norma SQuaRE (funcionalidad, seguridad, interoperabilidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad).

También se agregaron restricciones en el ingreso de los datos, con el fin de que se ingresaran valores dentro de las escalas correspondientes y se diligenciará de manera correcta cada una de las celdas.

Otro de los aspectos que se modificaron fueron las gráficas, pues se adicionaron gráficos en 3D que permiten comparar los resultados obtenidos de las calificaciones con las calificaciones ponderadas. Anteriormente, solo se mostraba una gráfica de barras donde se mostraba la calificación ponderada.

Además se agregó una hoja adicional que contiene la información completa de las métricas de donde se extrajeron los datos para el respectivo cálculo; esto con el fin de que el usuario pueda ir directamente a la información de la métrica en el momento en que le sea necesario.

Para mayor información remítase al Anexo E en donde se encuentra el manual del usuario de la plantilla.

CONCLUSIONES

El mundo globalizado exige cada vez más la aplicación de estándares internacionales que garanticen la calidad de los productos. El uso adecuado de un modelo de calidad estándar permite que los productos de software puedan tener una calidad aceptable para el cliente. La utilización de las métricas dentro de esos modelos de calidad para la evaluación de un producto de software juegan un papel determinante.

De acuerdo con la investigación realizada en el trabajo de campo para la ejecución de documento, en algunas empresas del medio dedicadas al desarrollo de productos de software o a brindar soluciones a otras empresas a través de la integración de productos de software ya existentes, se pudo concluir que hay métricas que son muy difíciles de evaluar, debido a que la fuente de datos es complicada de obtener.

Algunas de las causas por las cuales no se lleva a cabo totalmente en la práctica el modelo de calidad propuesto por la norma ISO/IEC 9126, son:

- *No hay exigencias del medio para la utilización de la norma.*

Cada empresa utiliza un modelo de calidad que se adecúe a las circunstancias del desarrollo y define sus propias métricas de calidad, muchas veces dependiendo de la experiencia de los mismos integrantes del equipo de trabajo, quienes detectan fallas en el proceso de realización/evaluación de un producto de software y a partir de esto, se define la nueva métrica. Por tal motivo, no hay una ley que exija la utilización de la norma ISO/IEC 9126 como modelo de calidad estándar para las empresas que desarrollan productos de Software.

- *Las fuentes de los datos son difíciles de obtener.*

Algunas métricas dentro del modelo de calidad propuesto por la norma ISO/IEC 9126 que exponen la forma de calcular la métrica, y esta resulta difícil de obtener por la complejidad de los datos necesarios para realizar el cálculo.

- *No se ha visto la necesidad de realizar una exhaustiva búsqueda de las fuentes de datos.*

Como en algunas empresas no han surgido problemas o situaciones relevantes que los lleven a la utilización de muchas de las métricas propuestas por la norma ISO/IEC 9126, no ven la necesidad de gastar esfuerzos en la obtención de los datos para realizar los cálculos necesarios; por ende, prefieren seguir usando las métricas que ellos mismo han desarrollado dentro de su equipo de trabajo y que además no les genera tanta dificultad a la hora de obtener los datos iniciales.

Se pudo notar que no hay un conocimiento pleno de la norma ISO/IEC 9126:2001 en algunas de las empresas del medio, y menos su consideración dentro del desarrollo de sus productos de software.

La propuesta realizada por la norma SQuaRE de adicionar dos características más al modelo de calidad ya establecido en la norma ISO/IEC 9126:2001, permite hacer una mejor evaluación del producto de software en sí, pues al considerarse como características a la seguridad e interoperabilidad, se está haciendo una evaluación total (más profunda) del mismo, ya que se consideran más aspectos.

La seguridad y la interoperabilidad fueron aspectos que pasaron de ser subcaracterísticas de la funcionalidad, a ser características que definen sus propias subcaracterísticas debido a la importancia que tienen estos dos aspectos dentro del desarrollo de un producto de software actualmente.

Se sugiere entonces, utilizar el modelo de calidad estándar de la norma ISO/IEC CD 25010 (SQuaRE) con el fin de estar a la vanguardia a nivel mundial sobre la evaluación de los productos de software como lo establece la norma mencionada. A partir de ese estándar cada empresa puede hacer su propia adaptación de la norma e incluir métricas de calidad que considere pertinentes para su negocio.

Cuando se implementaron todas las métricas investigadas en este trabajo, a una plantilla de Excel, se obtuvo una herramienta que permite realizar la evaluación de un producto de software de manera completa. Esa plantilla contiene las características, subcaracterísticas y algunas de las métricas que complementaron el trabajo investigativo realizado en papers, libros, documentos electrónicos y empresas de software, de este actual documento.

BIBLIOGRAFÍA

- [1] BUSTAMANTE BEDOYA, Andrés Camilo (2004). Automatización de los procedimientos para la recolección y consolidación de las métricas externas del producto software según la norma ISO/IEC 9126:2001. Proyecto de grado. Universidad EAFIT. Año 2004

- [2] CALVO MEDRANO, Jospe Miguel y MINGUET MILÁN, Jesús Maria. Medida de las sub-características. Capacidad de análisis y capacidad de cambio mediante la norma ISO/IEC 9126.

- [3] CARVALLO, Juan P.; FRANCH, Xavier; QUER, Carme; BURGUÉS, Xavier; GRAU, Gemma. Un método para la combinación de modelos de calidad. Universidad politécnica de Cataluña. Año 2003.

- [4] DÁVILA NICANOR, Leticia y MEJÍA ÁLVAREZ, Pedro. “Evaluación de la calidad de software en sistemas de información en internet”. CINVESTAV-IPN. Sección de computación. Av. I.P.N. 2508, Zacatenco. México, DF. 07300. Año 2003.

- [5] FRANCH, Xavier y CARVALLO, Juan P. Using quality models in software package selection. Universidad de Cataluña. IEEE Software. Año 2003.

- [6] GONZÁLEZ JARDON, Carlos. La gestión del proceso software. Primera parte. Septiembre 9. Año 2006.
<http://calidad-ti.blogspot.com/2006/09/la-gestin-del-proceso-de-software.html>

Normas ISO/IEC 9126

- [7] ISO/IEC 9126-1:2001. International standard “software engineering - product quality- Part 1: quality model
- [8] ISO/IEC TR 9126-2:2003. International standard “software engineering - product quality - part 2: external metrics
- [9] ISO/IEC TR 9126-3:2003. International standard “software engineering - product quality - part 3: internal metrics
- [10] ISO/IEC TR 9126-4:2004. International standard “software engineering - product quality - part 4: quality in use metrics

Normas ISO/IEC 14598

- [11] ISO/IEC 14598-1:1999. Information technology. Software product evaluation - part 1: general overview
- [12] ISO/IEC 14598-2:2000. Software engineering - Product evaluation - part 2: planning and management
- [13] ISO/IEC 14598-3:2000. Software engineering - Product evaluation - part 3: proccess for developers
- [14] ISO/IEC 14598-4:1999. Software engineering - Product evaluation - part 4: Proccess for acquirers
- [15] ISO/IEC 14598-5:1998. Information technology - Software product evaluation - part 5: proccess for evaluators

- [16] ISO/IEC 14598-6:2001. Software engineering - product evaluation - part 6: documentation of evaluation modules

Normas ISO/IEC 2500n

- [17] ISO/IEC 25000. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – guide to SQuaRE
- [18] ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model
- [19] JARAMILLO OSPINA, Luz Adriana y FRANCO CÓRDOBA, Ana Maria (2003). Plan de calidad para la evaluación de productos de software en uso con base en la norma ISO/IEC 9126. Proyecto de grado. Universidad EAFIT. Año 2003.
- [20] JURISTA, Natalia y MORENO, Ana M. y VEGAS, Sira. Técnicas de evaluación de software. Versión 12. Septiembre 22. Año 2004
- [21] MENA MENDOZA, Gonzalo. Estándares de calidad. Año 2006.
http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126-3/
- [22] MUÑOZ CALERO, Coral. “Modelos de calidad. WQM, PQM, e-commerce, portlets”. Calidad de sistemas de información. Departamento de informática. Universidad de Castilla-La Mancha. Mayo 09. Año 2005.
- [23] OLSINA, Luis. “Ingeniería Web: Marco de medición y evaluación de calidad”. Departamento de informática. Universidad nacional de San Luis – La Rioja - Catamarca. Año 2007.

- [24] PRESIDENCIA DEL CONSEJO DE MINISTROS (Perú). Guía técnica sobre la evaluación de software en la administración pública. Mayo 05. Año 2004.
http://www.ongei.gob.pe/Bancos/Banco_Normas/Archivos/Guia-evaluacion-sw.pdf
- [25] SCALONE, Fernanda. Software quality management. Overview sobre modelos/estándares de calidad del SW. Diciembre 02. Año 2006.
<http://softqm.blogspot.com/2006/12/overview-sobre-modelosestndares-de.html>
- [26] SCALONE, Fernanda. “Estudio comparativo de los modelos y estándares de calidad del software”. Maestría en ingeniería en calidad. Universidad tecnológica nacional. Facultad regional Buenos Aires. Junio. Año 2006.

ANEXO A

Normas SQuaRE

(Software product Quality Requirements and Evaluation) ³⁸

La serie de estándares del SQuaRE consisten de las siguientes divisiones bajo el título general Software product Quality Requirements and Evaluation (Requisitos y Evaluación de la calidad de los productos de software):

- ISO/IEC 2500n – División de la Administración de la Calidad,
- ISO/IEC 2501n – División del Modelo de Calidad,
- ISO/IEC 2502n – División de Medidas de Calidad,
- ISO/IEC 2503n – División de los Requisitos de Calidad, y
- ISO/IEC 2504n – División de la Evaluación de la Calidad

Este estándar internacional es una revisión de la ISO/IEC 9126-1: 2001, e incorpora las mismas características de la calidad del software con algunas enmiendas.

- La seguridad y la interoperabilidad son características, más que subcaracterísticas de la funcionalidad.
- La subcaracterística de la usabilidad, comprensibilidad ha sido renombrada como appropriateness (apropiabilidad), y atractibilidad como likability.
- Calidad en uso ha sido subdividida en usabilidad en uso y seguridad en uso, la cuales incorporan las características previas, y tiene nuevas

³⁸ Traducción literal de la norma ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

subcaracterísticas que son: contexto en uso, seguridad en uso y adaptabilidad en uso.

Este estándar internacional es entendido para ser usado en conjunción con las otras partes de los estándares de las series del SQuaRE (ISO/IEC 25000 - ISO/IEC 25050), y con la ISO/IEC 14598 hasta ser reemplazado por las series de estándares de la ISO/IEC 25000.

Este estándar internacional se conforma con los procesos técnicos definidos en la ISO/IEC 15288:2002 relacionado con la definición y el análisis de los requisitos de calidad. La figura a continuación ilustra la organización de las series de SQuaRE, representado en familias de estándares, llamadas además divisiones.

Las divisiones dentro del modelo SQuaRE, son:

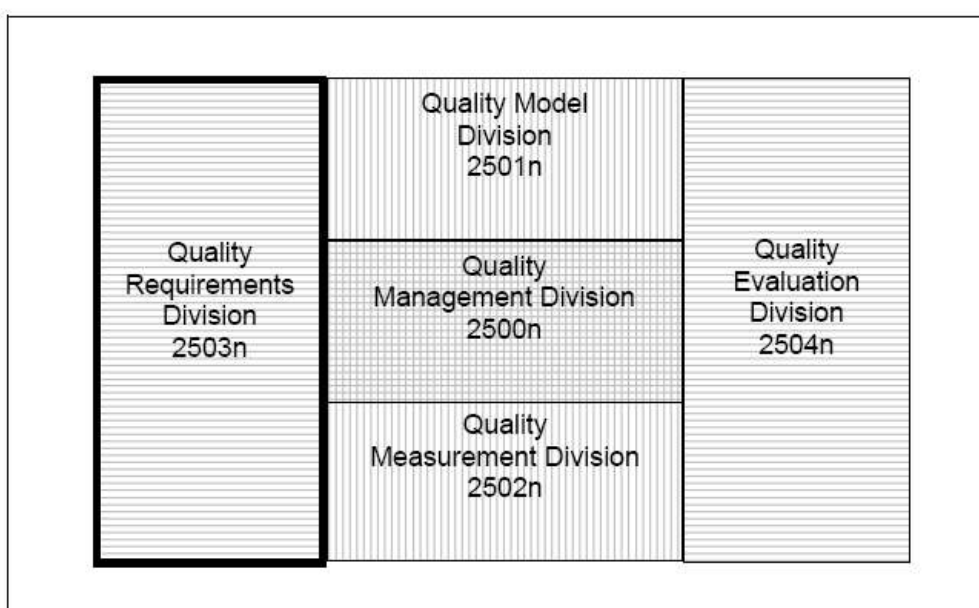


Figura 9 - Organización de la serie de estándares de SQuaRE ³⁹

³⁹ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

- **ISO/IEC 2500n – División de la Administración de la Calidad.** Los estándares que conforman esta división definen todos los modelos comunes, términos y definiciones referidas más allá de todas las otras series de estándares de la SQuaRE.

La división proporciona también requisitos y la dirección para una función de soporte, la cual es responsable de la administración de la especificación y de la evaluación de requisitos del producto de software.

- **ISO/IEC 2501n – División del Modelo de Calidad.** El estándar que forma esta división presenta un modelo detallado de calidad incluyendo características para la calidad interna, externa y calidad en uso.

Además, las características internas y externas de la calidad del software se descomponen en sub-características.

La dirección práctica en el uso del modelo de la calidad es también proporcionada.

- **ISO/IEC 2502n – División de Medidas de Calidad.** Los estándares que forman esta división incluyen un modelo de referencia de la medida de calidad del producto de software, definiciones matemáticas de las medidas de la calidad, y dirección práctica para su aplicación. Las actuales medidas se aplican a la calidad interna del software, a la calidad externa del software y a la calidad en uso.

- **ISO/IEC 2503n – División de los Requisitos de Calidad.** El estándar que forma esta división ayuda a especificar requisitos de calidad. Estos requisitos de calidad pueden ser usados en el proceso de elicitación de requisitos de calidad para un producto de software a ser desarrollado o como entrada para un proceso de evaluación. El proceso de definición de requisitos es proyectado para procesos técnicos definidos en la ISO/IEC

15288 – Tecnología de información – Administración del ciclo de vida – Procesos del ciclo de vida del sistema.

- **División de la Evaluación de la Calidad.** Los estándares que forman esta división proporcionan requisitos, recomendaciones y las pautas para la evaluación del producto de software, son realizados por los evaluadores, los adquirentes o los desarrolladores. El soporte para documentar una medida como un módulo de evaluación está también presente.

Modelo de Calidad

El modelo de calidad se divide en 2 partes: calidad interna y calidad externa, y calidad en uso. La primera parte del modelo especifica ocho (8) características, las cuales se subdividen más a fondo en sub-características. Estas características son manifestadas externamente cuando el software es usado como parte de un sistema informático, y son un resultado de las cualidades internas del software.

La segunda parte del modelo especifica seis (6) características de calidad en uso.

Las características definidas son aplicables a cada tipo de software. Las características proveen la constante terminología para la calidad del producto de software. También proporcionan un marco para especificar los requisitos de calidad para el software, y hacer compensaciones entre las capacidades del producto de software.

El modelo de calidad puede ser usado para soportar la especificación y la evaluación del software desde diferentes perspectivas asociadas con la adquisición, los requisitos, el desarrollo, el uso, la evaluación, la ayuda, el mantenimiento, la garantía de calidad y la intervención del software.

Puede por ejemplo ser usado por los desarrolladores, los adquirentes, los aseguradores de la calidad y evaluadores independientes, particularmente esos responsables de especificar y de evaluar la calidad del producto de software.

Marco del Modelo de Calidad

Modelo de calidad del software

La calidad de un sistema es el resultado de la calidad de sus elementos y su interacción.

La calidad del software es la capacidad del producto de software para satisfacer necesidades establecidas e implícitas cuando es usado bajo condiciones especificadas.

El modelo de calidad del producto de software define ocho características de calidad del software: funcionalidad, seguridad, interoperabilidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

El modelo de calidad en uso define seis características a nivel del sistema: usabilidad en uso, contexto en uso, protección (safety) en uso, seguridad (security) en uso, soporte en uso y adaptabilidad en uso.

El modelo define tres (3) diferentes vistas de calidad:

- Calidad en uso
- Calidad externa del software
- Calidad interna del software

La calidad en uso es una medida de la calidad del sistema en su ambiente operacional para usuarios, para las tareas específicas que realizan.

La calidad del software en uso es la capacidad del software para permitir la calidad en uso en su ambiente operacional, para las tareas específicas que realizan los usuarios.

La calidad externa del software proporciona una vista de “caja negra” del software y trata las características relacionadas con la ejecución del software en el hardware y el sistema operativo.

La calidad externa del software proporciona una vista de “caja blanca” y trata las características del producto de software que están típicamente disponibles durante el desarrollo. La calidad interna del software está principalmente relacionada con las propiedades estáticas del software.

La calidad interna del software tiene un impacto en la calidad externa del software, que tiene a su vez un impacto en la calidad en uso.

La figura a continuación muestra la interacción entre los diferentes modelos de calidad y los modelos del sistema.

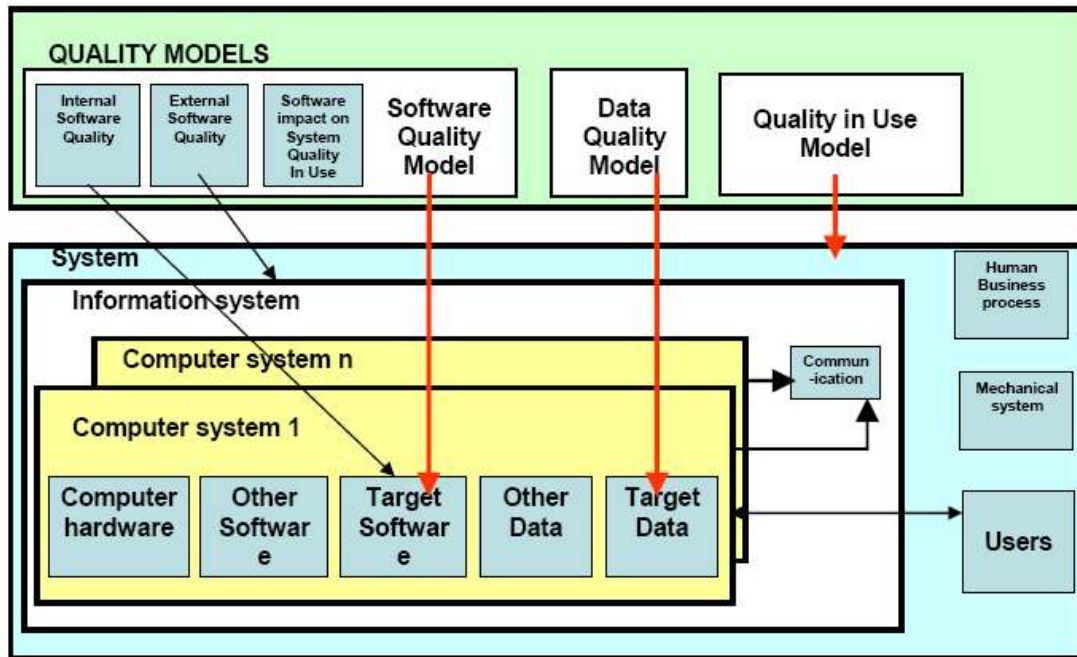


Figura 10 – Modelos del sistema y modelos de calidad ⁴⁰

El modelo de calidad sirve para asegurar que todos los aspectos de calidad son considerados desde el punto de vista interna, externa y de calidad en uso.

⁴⁰ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

Acercamientos a la Calidad

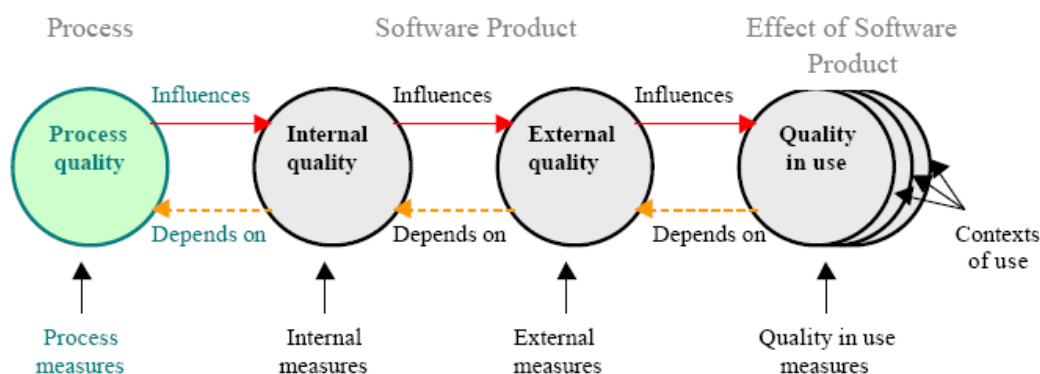


Figura 11 – Calidad en el ciclo de vida ⁴¹

Las necesidades de la calidad del usuario incluyen requisitos para la calidad en uso en contextos específicos del uso. Estas necesidades identificadas pueden ser utilizadas al especificar la calidad externa e interna usando características y sub-características de la calidad del producto de software.

La calidad de los productos de software pueden ser evaluados midiendo las atributos internos (medidas típicamente estáticas de productos intermedios), o por medición de atributos externos (típicamente midiendo el comportamiento del código cuando está ejecutado), o por medición de los atributos de la calidad en uso (cuando el producto está en uso simulado o real)

El proceso de calidad contribuye a mejorar la calidad del producto, y la calidad del producto contribuye a mejorar la calidad en uso. Por lo tanto, determinar y mejorar un proceso es un medio para mejorar la calidad del producto, y evaluar y mejorar la calidad del producto es un medio para mejorar la calidad del uso. Similarmente, la evaluación de la calidad en uso puede proveer la retroalimentación para mejorar

⁴¹ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

un producto, y la evaluación de un producto puede proporcionar la retroalimentación para mejorar un proceso.

Los atributos apropiados internos del software son un pre-requisito para alcanzar el comportamiento externo requerido, y el comportamiento externo apropiado es un pre-requisito para alcanzar la calidad en uso.

Modelo de la medida de Calidad del Software

Las propiedades inherentes al software, que pueden ser distinguidas cuantitativamente o cualitativamente, son llamadas atributos. Los atributos de calidad son inherentes a las propiedades del software que contribuyen a la calidad. Los atributos de calidad son categorizados entre una o más sub-características.

Los atributos de calidad son medidos aplicando un método de la medida. Un método de la medida es una secuencia lógica de operaciones usadas para cuantificar un atributo con respecto a una escala específica. El resultado de aplicar un método de la medida es llamado una medida base. Las características y sub-características de calidad pueden ser cuantificadas para aplicar funciones de la medida. Una función de la medida es un algoritmo usado para combinar elementos de la medida de calidad. El resultado de aplicar una función de la medida es llamada una medida de la calidad del software. De esta manera las medidas de la calidad del software llegan a ser cuantificaciones de las características y sub-características de la calidad.

Más de una medida de la calidad del software puede ser usada para medir una característica o sub-característica de calidad.

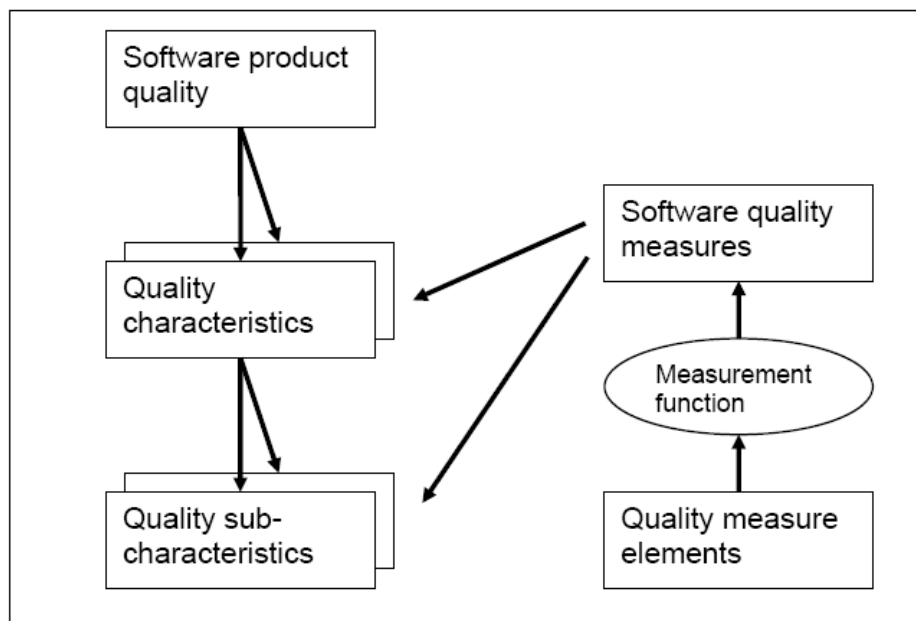


Figura 12 – Modelo de referencia de la medida de calidad del producto de software ⁴²

Modelo del ciclo de vida de calidad de un producto de software

El Modelo del ciclo de vida de calidad de un producto de software trata la calidad de un producto de software en tres fases principales del ciclo de vida del producto de software: producto bajo desarrollo, producto en operación y producto en uso.

- La fase de un producto bajo desarrollo es el tema de la calidad interna del software
- La fase de un producto en operación es el tema de la calidad externa del software
- La fase de un producto en uso es el tema de la calidad en uso

⁴² ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

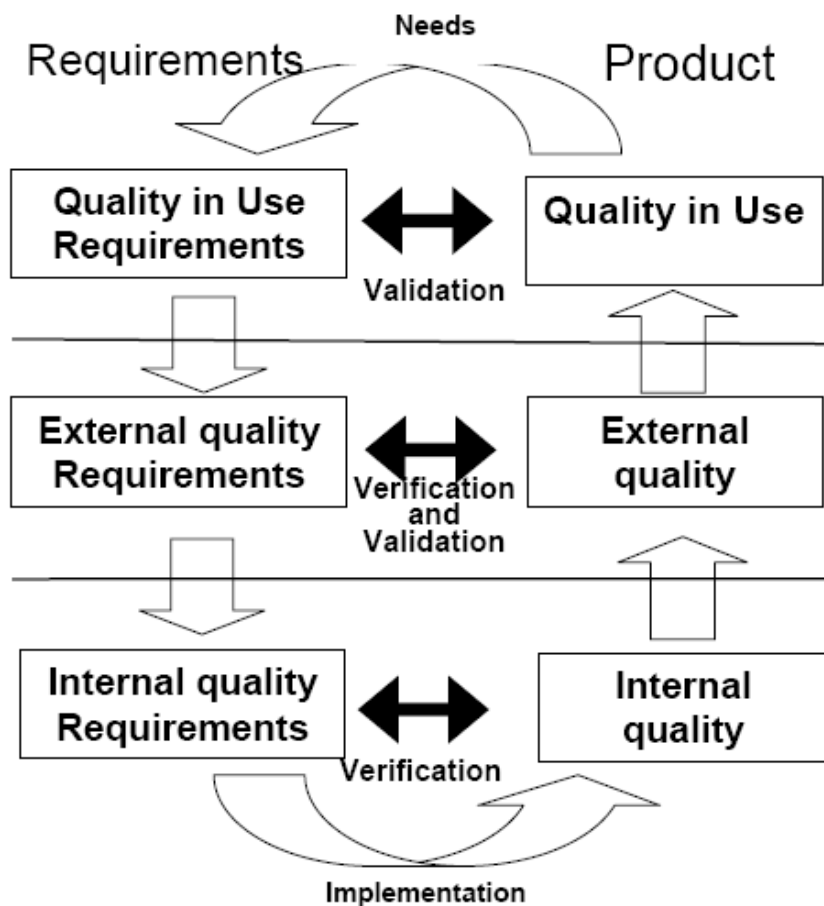


Figura 13 - Modelo del ciclo de vida para la calidad de un producto de software ⁴³

El modelo del ciclo de vida de la calidad de un producto de software también indica que la implementación de la calidad del software requiere de un proceso similar al proceso de desarrollo del software para cada tipo de calidad: requisitos, implementación y validación de los resultados.

Los Requisitos de la calidad en uso especifican el nivel de calidad desde el punto de vista de los usuarios finales. Esos requisitos son derivados de las necesidades

⁴³ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

de cada contexto del uso. Los requisitos de la calidad en uso son usados como punto para la validación del producto de software por el usuario.

Los requisitos de la calidad externa del software especifican el nivel de calidad requerido desde una vista externa. Ellos incluyen requisitos derivados desde los requisitos de calidad del usuario, incluyendo requisitos de la calidad en uso. Los requisitos de la calidad externa del software son usados como punto para la verificación y validación técnica del producto de software. Los requisitos para las características de la calidad externa del software deben ser indicados en la especificación de calidad de requisitos, usando medidas externas y utilizarlos como criterios cuando un producto es evaluado.

Los requisitos de la calidad interna del software especifican el nivel de calidad requerido desde la vista interna del producto. Ellos incluyen requisitos derivados de los requisitos de la calidad externa del software. Los requisitos de la calidad interna del software son usados para especificar propiedades de productos de software intermedios (especificaciones, código fuente, etc). Los requisitos de la calidad interna del software pueden también ser usados para especificar propiedades de los productos de software entregables, no ejecutables, tales como la documentación y los manuales.

Los requisitos de la calidad interna del software pueden ser usados como puntos para la verificación de varios estados del desarrollo. Ellos también pueden ser usados para definir estrategias de desarrollo y criterios para la evaluación y la verificación durante el desarrollo. Los requisitos de calidad internos deben ser especificados cuantitativamente en términos de las medidas internas.

Estructura del modelo de Calidad

El modelo de calidad SQuaRE categoriza la calidad del software entre características, las cuales son subdivididas más a fondo entre subcaracterísticas y atributos de calidad.

El modelo de calidad SQuaRE consiste de dos partes, el modelo para la calidad externa e interna del software y el modelo para la calidad en uso.

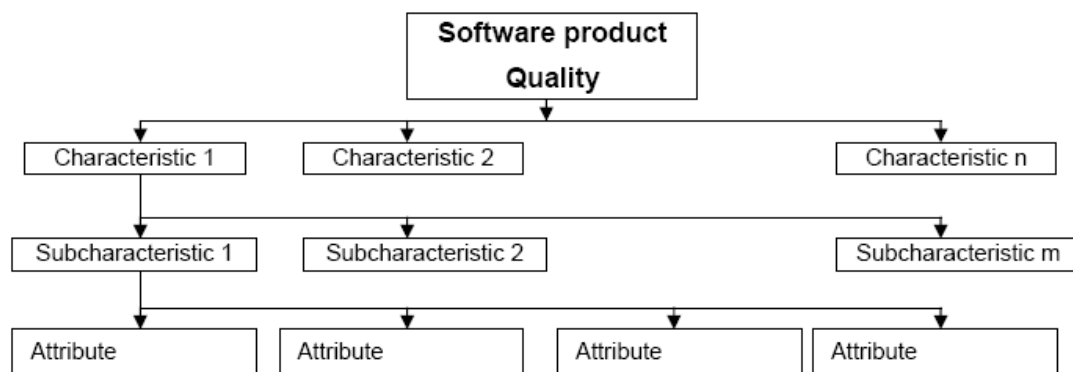


Figura 14 – Estructura del modelo de calidad ⁴⁴

Diferencias entre las medidas de calidad internas, externas y en uso.

Las medidas de calidad interna del software pueden ser usadas tempranamente en el proceso de desarrollo del sistema para predecir medidas de la calidad externa del software. Hay a menudo medidas internas y externas para la misma propiedad, por ejemplo, una medida interna para estimar la respuesta prevista para predecir el tiempo medido externamente.

⁴⁴ ISO/IEC CD 25010:2007. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – quality model.

Las medidas de calidad en uso relativas a los usuarios completando tareas realistas (por las pruebas de usuario o en uso actual)

La relación de la calidad en uso respecto a las otras características de la calidad del producto de software depende del tipo de usuario:

- El usuario final para quien la calidad en uso es principalmente un resultado de la funcionalidad, fiabilidad, usabilidad y eficiencia.
- La persona que mantiene el software para quien la calidad en uso es un resultado de la mantenibilidad.
- La persona que conecta el software para quien la calidad en uso es un resultado de la portabilidad.

Usando un modelo de calidad

La calidad de un producto de software debería ser evaluada usando un modelo de calidad definido. El modelo de calidad debería ser usado cuando se fijan las metas de la calidad para los productos de software y los productos intermedios.

La calidad de un producto de software debería ser descompuesta jerárquicamente entre un modelo de calidad integrado por las características y sub-características, las cuales pueden ser usadas como una lista de comprobación de los problemas relacionados con la calidad.

No es prácticamente posible medir todas las sub-características internas y externas para todas las partes de un producto de software grande. Similarmente, no es usualmente práctico medir la calidad en uso para todos los posibles

escenarios de las tareas del usuario. Los recursos para la evaluación necesitan ser asignados entre los diversos tipos de medida, dependiente de los objetivos del negocio y la naturaleza del producto y diseño de procesos.

ANEXO B

MÉTRICAS INTERNAS⁴⁵

FUNCIONALIDAD

- ADECUACIÓN

Tabla 8. Métrica interna de adecuación - suficiencia funcional

Nombre:	Functional Adequacy
Propósito:	How adequate are the checked functions?
Método de aplicación:	Count the number of functions that are suitable for performing the specified tasks, then measure the ratio of it to functions implemented. The following may be measured; • All or parts of design specifications. • Completed modules/parts of software products
Medición, fórmula:	$X = 1 - A/B$ A = Number of functions in which problems are detected in evaluation B = Number of functions checked
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more adequate
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

⁴⁵ ISO/IEC TR 9126-3:2003. International standard “software engineering – product quality – part 3: internal metrics.

Tabla 9. Métrica interna de adecuación - integridad de la implementación funcional

Nombre:	Functional implementation completeness
Propósito:	How complete is the functional implementation?
Método de aplicación:	Count the number of missing functions detected in evaluation and compare with the number of function described in the requirement specifications.
Medición, fórmula:	$X = 1 - A/B$ A = Number of missing functions detected in evaluation B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Req. Spec Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 10. Métrica interna de adecuación - alcance de la implementación funcional

Nombre:	Functional implementation coverage
Propósito:	How correct is the functional implementation?
Método de aplicación:	Count the number of incorrectly implemented or missing functions and compare with the number of functions described in the requirement specifications.
Medición, fórmula:	$X = 1 - A/B$ A = Number of incorrectly implemented or missing functions detected B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more correct
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count

	B = count
Fuente de medición:	Req. Spec Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 11. Métrica interna de adecuación - estabilidad (o volatilidad) de la especificación funcional

Nombre:	Functional specification stability (volatility)
Propósito:	How stable is the functional specification during the development life cycle?
Método de aplicación:	Count the number of functions changed (added, modified, or deleted) during development life cycle phase, then compare with the number of functions described in the requirement specifications.
Medición, fórmula:	$X = 1 - A/B$ A = Number of functions changed during development life cycle phases B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more stable
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirements specifications Review Report
ISO/IEC 12207 SLCP:	Validation Quality Assurance Qualification testing Problem Resolution Operation
Audiencia:	Developers Maintainers

- **EXACTITUD**

Tabla 12. Métrica interna de exactitud - exactitud computacional

Nombre:	Computational Accuracy
Propósito:	How completely have the accuracy requirements been implemented?
Método de aplicación:	Count the number of functions that have implemented the accuracy requirements and compare with the number of functions with specific accuracy requirements.
Medición, fórmula:	$X = A/B$ A = Number of functions in which specific accuracy requirements had been implemented, as confirmed in evaluation. B = Number of functions for which specific accuracy requirements need to be implemented.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Requirers Developers

Tabla 13. Métrica interna de exactitud - precisión

Nombre:	Precision
Propósito:	How complete was the implementation of specific levels of precision for the data items?
Método de aplicación:	Count the number of data items that meet the requirements of specific levels of precision and compare to the total number of data items with specific level of precision requirements.
Medición, fórmula:	$X = A/B$ A = Number of items implemented with specific B = Number of functions for which specific accuracy requirements need to be

	implemented.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ $A = \text{count}$ $B = \text{count}$
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

• CONFORMIDAD DE LA FUNCIONALIDAD

Tabla 14. Métrica interna de conformidad con la funcionalidad - conformidad funcional

Nombre:	Functional compliance
Propósito:	How compliant is the functionality of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.
Medición, fórmula:	$X = A/B$ $A = \text{Number of correctly implemented items related to functionality compliance confirmed in evaluation.}$ $B = \text{Total number of compliance items.}$
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more compliant.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ $A = \text{count}$ $B = \text{count}$
Fuente de medición:	Specification of compliance and related standards, conventions or regulations. Design Source code Review report

ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 15. Métrica interna de conformidad con la funcionalidad - conformidad estándar de la interfaz

Nombre:	Intersystem Standard compliance
Propósito:	How compliant are the interfaces to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of interfaces that meet required compliance and compare with the number of interfaces requiring compliance as in the specifications.
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented interfaces as specified, confirmed in review. B = Total number of interfaces requiring compliance.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more compliant.
Tipo de escala:	Absolute.
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

SEGURIDAD

• RESISTENCIA AL ACCESO

Tabla 16. Métrica interna de resistencia al acceso - facilidad de auditar los accesos

Nombre:	Access auditability
Propósito:	How auditability is access login?
Método de aplicación:	Count the number of access types that are being logged correctly as un the specifications and compare with the number of access types that are required to be logged in the specifications.
Medición, fórmula:	$X = A/B$ A = Number of access types that are being logged as in the specifications. B = Number of access types required to be logged in the specifications.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more auditable.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 17. Métrica interna de resistencia al acceso - uso controlado del acceso

Nombre:	Access controllability
Propósito:	How controllable is access to the system?
Método de aplicación:	Count the number of access controllability requirements implemented correctly as in the specifications and compare with the number of access controllability requirements in the specifications.
Medición, fórmula:	$X = A/B$ A = Number of access controllability requirements implemented correctly as in the specifications. B = Number of access controllability requirements in the specifications.

Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more controllable.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ $A = \text{count}$ $B = \text{count}$
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- **RESISTENCIA A LA COPIA**

No se tiene una métrica definida para esta subcaracterística.

- **FACILIDAD DEL CIFRADO**

Tabla 18. Métrica interna de facilidad del cifrado - cifrado de datos

Nombre:	Data encryption
Propósito:	How complete is the implementation of data encryption?
Método de aplicación:	Count the number of implemented instances of encryptable/decryptable data items as specified and compare with the number of instances of data items requiring data encryption/decryption facility as in specifications.
Medición, fórmula:	$X = A/B$ $A =$ Number of implemented instances of encryptable/decryptable data items as specified confirmed in review. $B =$ Number of items requiring data encryption/decryption facility as in specifications.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ $A = \text{count}$ $B = \text{count}$
Fuente de	Requirement specification

medición:	Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification
Audiencia:	Developers

- **RESISTENCIA A LA FALSIFICACIÓN**

No se tiene una métrica definida para esta subcaracterística.

- **ROBUSTEZ**

Tabla 19. Métrica interna de robustez - prevención de la corrupción de datos

Nombre:	Data corruption prevention
Propósito:	How complete is the implementation of data corruption prevention?
Método de aplicación:	Count the number of implemented instantes of data corruption prevention as specified and compare with the number of instantes of operations/access specified in requirements as capable of corrupting/destroying data.
Medición, fórmula:	$X = A/B$ A = Number of implemented instances of data corruption prevention as specified confirmed in review. B = Number of instances of operation/access identified in requirements as capable of corrupting/destroying data.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers

- **CONFORMIDAD CON LA SEGURIDAD**

Se definió una métrica en la investigación realizada. Ver tabla 1, Sección 4.1.2 – Métricas Investigadas.

INTEROPERABILIDAD

- **COMPATIBILIDAD DE LA OSI**

Tabla 20. Métrica interna de compatibilidad de la OSI - consistencia de la interface (protocolo)

Nombre:	Interface consistency (protocol)
Propósito:	How correctly have the interface protocols been implemented?
Método de aplicación:	Count the number of interface protocols that were implemented correctly as in the specifications and compare with the number of interface protocols to be implemented as in the specifications.
Medición, fórmula:	$X = A/B$ A = Number of interface protocols implementing consistent format as in the specification confirmed in review. B = Number of interface protocols to be implemented as in the specifications.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more consistent.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- **COMPATIBILIDAD DEL SOFTWARE**

No se tiene una métrica definida para esta subcaracterística.

- **COMPATIBILIDAD DE LOS DATOS**

Tabla 21. Métrica interna de compatibilidad de los datos - intercambiabilidad de datos (basado en el formato de datos)

Nombre:	Data exchangeability (Data format based)
Propósito:	How correctly have the interface data formats been implemented?
Método de aplicación:	Count the number of interface data formats that have been implemented correctly as in the specifications and compare to the number of data formats to be exchanged as in the specifications.
Medición, fórmula:	$X = A/B$ A = Number of interface data formats that have been implemented correctly as in the specifications. B = Number of data formats to be exchanged as in the specifications.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more correct.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Design Source code Review Report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- **TRAZABILIDAD**

No se tiene una métrica definida para esta subcaracterística.

- **CONFORMIDAD CON LA INTEROPERABILIDAD**

Se definió una métrica en la investigación realizada. Ver tabla 2, Sección 4.1.2 – Métricas Investigadas.

FIABILIDAD

- MADUREZ

Tabla 22. Métrica interna de madurez - detección de fallas

Nombre:	Fault detection
Propósito:	How many faults were detected in reviewed product?
Método de aplicación:	Count the number of detected faults in review and compare it to the number of estimated faults to be detected in this phase.
Medición, fórmula:	$X = A/B$ A = Absolute number of faults detected in review B = Number of estimated faults to be detected in review(using past history or reference model)
Interpretación:	$0 \leq X$ A high value for X implies good product quality, while A=0 does not necessarily imply fault free status of the reviewed item.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Value A comes from review report. Value B comes from the organization database.
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 23. Métrica interna de madurez - remoción de fallos

Nombre:	Fault renoval
Propósito:	How many faults have been corrected? What is the proportion of faults removed?
Método de aplicación:	Count the number of faults removed during design/coding and compare it to the number of faults detected in review during design/coding.
Medición, fórmula:	$X = A$ A = Number of corrected faults in design/coding

	$Y = A/B$ A = Number of corrected faults in design/coding. B = Number of faults detected in review.
Interpretación:	$0 \leq X$ A high value for X implies, that less faults remain $0 \leq Y \leq 1$ The closer to 1, the better (more faults removed)
Tipo de escala:	Ratio. Absolute.
Tipo de medida:	$Y = \text{count/count}$ $B = \text{count}$
Fuente de medición:	Value A comes from review fault removal report. Value B comes from review report.
ISO/IEC 12207	Verification
SLCP:	Joint review
Audiencia:	Developers Requirers

Tabla 24. Métrica interna de madurez - suficiencia de prueba

Nombre:	Test adequacy
Propósito:	How much of the required test cases are covered by the test plan?
Método de aplicación:	Count the number of test cases planned and compare it to the number of test cases required to obtain adequate test coverage.
Medición, fórmula:	$X = A/B$ A = Number of test cases designed in test plan and confirmed in review. B = Number of test cases required.
Interpretación:	$0 \leq X$ Where X is greater the better adequacy.
Tipo de escala:	Absolute.
Tipo de medida:	$X = \text{count/count}$ $A = \text{count}$ $B = \text{count}$
Fuente de medición:	Value A comes from test plan Value B comes from requirements
ISO/IEC 12207	QA

SLCP:	Problem resolution Verification
Audiencia:	Developers Requirers

• TOLERANCIA A FALLOS

Tabla 25. Métrica interna de tolerancia a fallos - prevención de fallas

Nombre:	Failure avoidance
Propósito:	How many fault patterns were brought under control to avoid critical and serious failures?
Método de aplicación:	Count the number of avoided fault patterns and compare it to the number of fault patterns to be considered.
Medición, fórmula:	$X = A/B$ A = Number of fault patterns having avoidance in design/code B = Number of fault patterns to be considered <i>Comment(s)</i> Fault pattern examples out of range data deadlock <i>Comment(s)</i> Fault tree analysis technique may be used to detected fault patterns.
Interpretación:	$0 \leq X$ Where X is greater the better failure avoidance.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Value A comes from review report Value B comes from requirement specification document.
ISO/IEC 12207 SLCP:	Verification Validation Joint review Problem resolution
Audiencia:	Developers Requirers Maintainers

Tabla 26. Métrica interna de tolerancia a fallos - prevención de la operación incorrecta

Nombre:	Incorrect operation avoidance
Propósito:	How many functions are implemented with incorrect operations avoidance capability?
Método de aplicación:	Count the number of implemented functions to avoid critical and serious failures caused by incorrect operations and compare it to the number of incorrect operation patterns to be considered. <i>Comment(s)</i> Also data damage in addition to system failure.
Medición, fórmula:	$X = A/B$ A = Number of functions implemented to avoid incorrect operation patterns. B = Number of incorrect operation patterns to be considered <i>Comment(s)</i> Incorrect operation patterns Incorrect data types as parameters Incorrect sequence of data input Incorrect sequence of operation <i>Comment(s)</i> Fault tree analysis technique may be used to detected incorrect operation patterns.
Interpretación:	$0 \leq X$ Where X is greater the better incorrect operation avoidance.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Value A comes from review report Value B comes from requirement specification document.
ISO/IEC 12207 SLCP:	Verification Validation Joint review Problem resolution
Audiencia:	Developers Requirers Maintainers

• RECUPERABILIDAD

Tabla 27. Métrica interna de recuperabilidad - facilidad de restauración

Nombre:	Restorability
Propósito:	How capable is the product in restoring itself after abnormal event or at request?

Método de aplicación:	Count the number of implemented restoration requirements and compare it to the number of restoration requirements in the specifications. Restoration requirement examples: database checkpoint, transaction checkpoint, redo function, undo function.
Medición, fórmula:	$X = A/B$ A = Number of implemented restoration requirements confirmed in review B = Number of restoration requirements in the specifications
Interpretación:	$0 \leq X \leq 1$ Where X is greater, the better restorability
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	A comes from review document B comes from requirements or design document
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 28. Métrica interna de recuperabilidad - efectividad de la restauración

Nombre:	Restoration Effectiveness
Propósito:	How effective is the restoration capability?
Método de aplicación:	Count the number of implemented meeting target restoration time (by calculations or simulations) and compare it to the number of restoration requirements with specified target time.
Medición, fórmula:	$X = A/B$ A = Number of implemented restoration requirements meeting target restore time B = Number of restoration requirements with specified target times
Interpretación:	$0 \leq X \leq 1$ Where X is greater, the better effectiveness
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	A comes from review document B comes from requirements or design document

ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- **CONFORMIDAD DE LA FIABILIDAD**

Tabla 29. Métrica interna de conformidad con la fiabilidad - conformidad con la fiabilidad

Nombre:	Reability compliance
Propósito:	How compliant is the reliability of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented items related to reliability compliance confirmed in evaluation B = Total number of compliance items
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more compliant.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Specification of compliance and related standards, conventions or regulations. Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

USABILIDAD

- APROPIABILIDAD

Tabla 30. Métrica interna de apropiabilidad - integridad de la descripción

Nombre:	Completeness of description
Propósito:	What proportion of functions (or types of function) are described in the product description?
Método de aplicación:	Count the number of functions which are adequately described and compare with the total number of functions in the product.
Medición, fórmula:	$X = A/B$ A = Number of functions (or types of functions) described in the product description B = Total number of functions (or types of functions)
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 31. Métrica interna de apropiabilidad - capacidad de la demostración

Nombre:	Demonstration capability
Propósito:	What proportion of functions requiring demonstration have demonstration capability?
Método de aplicación:	Count the number of functions that are adequately demonstrable and compare with the total number of functions requiring demonstration capability.
Medición, fórmula:	$X = A/B$ A = Number of functions demonstrated and confirmed in review B = Total number of functions requiring demonstration capability
Interpretación:	$0 \leq X \leq 1$

	The closer to 1, the more capable.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 32. Métrica interna de apropiabilidad - funciones evidentes

Nombre:	Evident functions
Propósito:	What proportion of the product functions are evident to the user?
Método de aplicación:	Count the number of functions that are evident to the user and compare with the total number of functions.
Medición, fórmula:	X = A/B A = Number of functions (or types of functions) evident to the user B = Total number of functions (or types of functions)
Interpretación:	0 <= X <= 1 The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 33. Métrica interna de apropiabilidad - comprensibilidad de la función

Nombre:	Function understandability
Propósito:	What proportion of the product functions will the user be able to understand correctly?
Método de aplicación:	Count the number of user interface functions where purposes is understood by the user and compare with the number of user interface functions.
Medición, fórmula:	$X = A/B$ A = Number of user interface functions whose purpose is understood by the user B = Number of user interface functions
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- FACILIDAD DE APRENDIZAJE**

Tabla 34. Métrica interna de aprendizaje - integridad de la documentación del usuario y/o de la facilidad de la ayuda

Nombre:	Completeness of user documentation and/or help facility
Propósito:	What proportion of functions are described in the user documentation and/or help facility?
Método de aplicación:	Count the number of functions implemented with help facility and/or documentation and compare with the total number of functions in product.
Medición, fórmula:	$X = A/B$ A = Number of functions described B = Total of number of functions provided
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete

Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

• OPERABILIDAD

Tabla 35. Métrica interna de operabilidad - comprobación de la validez de la entrada

Nombre:	Input validity checking
Propósito:	What proportion of input items provide check for valid data?
Método de aplicación:	Count the number of input items, which check for valid data and compare with the number of input items, which could check for valid data.
Medición, fórmula:	$X = A/B$ A = Number of inputs items which check for valid data B = Number of input items which could check for valid data
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 36. Métrica interna de operabilidad - facilidad de cancelar la operacion del usuario

Nombre:	User operation cancellability
Propósito:	What proportion of functions can be cancelled prior to completion?
Método de aplicación:	Count the number of implemented functions, which can be cancelled by the user prior to completion and compare it with the number of functions requiring the precancellation capability.
Medición, fórmula:	$X = A/B$ A = Number of implemented functions which can be cancelled by the user B = Number of functions requiring the precancellation capability
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better cancellability
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 37. Métrica interna de operabilidad - facilidad de anular la operación del usuario

Nombre:	User operation Undoability
Propósito:	What proportion of functions can be undone?
Método de aplicación:	Count the number of implemented functions, which can be undone by the user after completion and compare it with the number of functions.
Medición, fórmula:	$X = A/B$ A = Number of implemented functions which can be undone by the user B = Number of functions
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better undoability
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count

Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 38. Métrica interna de operabilidad - uso de la personalización

Nombre:	Customisability
Propósito:	What proportion of functions can be customized during operation?
Método de aplicación:	Count the number of implemented functions, which can be customized by the user during operation and compare it with the number of functions requiring the customization capability.
Medición, fórmula:	$X = A/B$ A = Number of functions which can be customized during operation B = Number of functions requiring the customization capability
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better customizability
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 39. Métrica interna de operabilidad - accesibilidad física

Nombre:	Physical accessibility
Propósito:	What proportion of functions can be customized for access by users with physical handicaps?
Método de	Count the number of implemented functions, which can be customized and

aplicación:	compare it with the number of functions.
Medición, fórmula:	$X = A/B$ A = Number of functions which can be customized B = Number of functions
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better physical accessibility
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 40. Métrica interna de operabilidad - capacidad de monitoreo del estado de la operación

Nombre:	Operation status monitoring capability
Propósito:	What proportion of functions have operations status monitoring capability?
Método de aplicación:	Count the number of implemented functions, which status can be monitored and compare it with the number of functions requiring the monitoring capability.
Medición, fórmula:	$X = A/B$ A = Number of functions having status monitoring capability B = Number of functions that are required to have monitoring capability
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better monitoring capability
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207	Verification

SLCP:	Joint review
Audiencia:	Developers Requirers

Tabla 41. Métrica interna de operabilidad - consistencia operacional

Nombre:	Operational consistency
Propósito:	What proportion of operations behave the same way to similar operations in other parts of the system?
Método de aplicación:	Count the number of instances of operations with inconsistent behavior and compare it with the total number of operations.
Medición, fórmula:	$X = 1 - A/B$ A = Number of instances of operations with inconsistent behavior B = Total number of operations
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more consistent
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207	Verification
SLCP:	Joint review
Audiencia:	Developers Requirers

Tabla 42. Métrica interna de operabilidad - claridad del mensaje

Nombre:	Message Clarity
Propósito:	What proportion of messages are self-explanatory?
Método de aplicación:	Count the numbers of implemented messages with clear explanations and compare it with the total number of messages implemented.
Medición, fórmula:	$X = A/B$ A = Number of implemented messages with clear explanations B = Number of messages implemented
Interpretación:	$0 \leq X \leq 1$

	The closer to 1, the more clear
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 43. Métrica interna de operabilidad - claridad del elemento de la interfaz

Nombre:	Interface element clarity
Propósito:	What proportion of interface elements are self-explanatory?
Método de aplicación:	Count the numbers of interface elements which are self explanatory and compare it with the total number of interface elements.
Medición, fórmula:	X = A/B A = Number of interface elements which are self-explanatory. B = Total number of interface elements.
Interpretación:	0 <= X <= 1 The closer to 1, the more clear.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 44. Métrica interna de operabilidad - facilidad de recuperación de un error operacional

Nombre:	Operational error recoverability
Propósito:	What proportion of functions can tolerate user error?
Método de aplicación:	Count the number of functions implemented with user error tolerance and compare it to the total number of functions requiring the tolerance capability
Medición, fórmula:	$X = A/B$ A = Number of functions implemented with user error tolerance B = Total number of functions requiring the tolerance capability
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more recoverable.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- **CAPACIDAD DE AYUDA**

No se tiene una métrica definida para esta subcaracterística.

- **ATRACTIBILIDAD**

Tabla 45. Métrica interna de atractibilidad - interacción atractiva

Nombre:	Attractive interaction
Propósito:	How attractive is the interface to the user?
Método de aplicación:	Questionnaire to users
Medición, fórmula:	Questionnaire to assess the attractiveness of the interface to users, taking account of attributes such as color and graphical design. <i>Comment(s)</i> Issues that potentially contribute to attractiveness include: Alignment of items (Vertical and Horizontal), Grouping, Use of colours,

	Appropriate and reasonable sized graphics, Use of whitespace/separators/borders, Animation, Typography, and 3D interface.
Interpretación:	Assessment classification
Tipo de escala:	Ordinal
Tipo de medida:	X = count/count (Count is a score)
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 46. Métrica interna de atractibilidad - personalización del aspecto de la interfaz de usuario

Nombre:	User interface appearance customizability
Propósito:	What proportion of user interface elements can be customized in appearance?
Método de aplicación:	Inspection (by expert)
Medición, fórmula:	$X = A/B$ A = Number of types of interface elements that can be customized B = Total number of types of interface elements
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

- **CONFORMIDAD DE USO**

Tabla 47. Métrica interna de la conformidad de uso - conformidad con la usabilidad

Nombre:	Usability compliance
Propósito:	How compliant is the product to applicable regulations, standards and conventions for usability?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented items related to usability compliance confirmed in evaluation B = Total number of compliance items
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more compliant.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Specification of compliance and related standards, conventions or regulations. Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

EFICIENCIA

- **COMPORTAMIENTO EN EL TIEMPO**

Tabla 48. Métrica interna del comportamiento en el tiempo - tiempo de respuesta

Nombre:	Response time
Propósito:	What is the estimated time to complete a specified task?
Método de aplicación:	Evaluate the efficiency of the operating system and the application system calls. Estimate the response time based on this.

	The following may be measured, - all or parts of design specifications - test complete transaction path - test complete modules/parts of software product - complete software product during test phase
Medición, fórmula:	$X = \text{time (calculated or simulated)}$
Interpretación:	The shorter the better
Tipo de escala:	Ratio
Tipo de medida:	$X = \text{time}$
Fuente de medición:	Known operating system. Estimated time in system calls.
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 49. Métrica interna del comportamiento en el tiempo - tiempo del rendimiento de procesamiento

Nombre:	Throughput time
Propósito:	What is the estimated number of tasks that can be performed over a unit of time?
Método de aplicación:	Evaluate the efficiency of handling resources in the system. Make a factor based upon the application calls to the system in handling the resources.
Medición, fórmula:	$X = \text{Number of tasks per unit of time}$
Interpretación:	The greater the better
Tipo de escala:	Ratio
Tipo de medida:	$X = \text{count}$
Fuente de medición:	Known operating system. Estimated time in system calls.
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

Tabla 50. Métrica interna del comportamiento en el tiempo - plazo de entrega

Nombre:	Turnaround time
Propósito:	What is the estimated time to complete a group of related tasks as job lot?
Método de aplicación:	Evaluate the efficiency of the operating system and the application system calls. Estimate the response time to complete a group of related tasks based on this. The following may be measured, - all or parts of design specifications - test complete transaction path - test complete modules/parts of software product - complete software product during test phase
Medición, fórmula:	X = time (calculated or simulated)
Interpretación:	The shorter the better
Tipo de escala:	Ratio
Tipo de medida:	X = time
Fuente de medición:	Known operating system. Estimated time in system calls.
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Developers Requirers

• UTILIZACIÓN DE RECURSOS

Tabla 51. Métrica interna de la utilización de recursos - utilización de entrada/salida

Nombre:	I/O Utilization
Propósito:	What is the estimated I/O utilization to complete a specified task?
Método de aplicación:	Estimate the I/O utilization requirement for the application
Medición, fórmula:	X = number of buffers (calculated or simulated)
Interpretación:	The shorter the better
Tipo de escala:	Ratio
Tipo de medida:	X = size

Fuente de medición:	Source code
ISO/IEC 12207 SLCP:	Verification
Audiencia:	Developers

Tabla 52. Métrica interna de la utilización de recursos - densidad del mensaje de la utilización de entrada/salida

Nombre:	I/O Utilization Message Density
Propósito:	What is the density of messages relating to I/O utilization in the lines of code responsible in making system calls?
Método de aplicación:	Count the number of errors pertaining to I/O failure and warnings and compare it to the estimated number of lines of code responsible in system calls.
Medición, fórmula:	$X = A/B$ A = Number of I/O related error messages. B = Number of lines of code directly related to system calls.
Interpretación:	The greater the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Source code
ISO/IEC 12207 SLCP:	Verification
Audiencia:	Developers

Tabla 53. Métrica interna de la utilización de recursos - utilización de la memoria

Nombre:	Memory utilization
Propósito:	What is the estimated memory size that the product will occupy to complete a specified task?
Método de aplicación:	Estimate the memory requirement
Medición, fórmula:	$X = \text{size in bytes (calculated or simulated)}$

Interpretación:	The lesser the better
Tipo de escala:	Ratio
Tipo de medida:	X = size
Fuente de medición:	Estimated size of memory utilization.
ISO/IEC 12207 SLCP:	Verification
Audiencia:	Developers

Tabla 54. Métrica interna de la utilización de recursos - densidad del mensaje de la utilización de la memoria

Nombre:	Memory utilization message density
Propósito:	What is the density of messages relating to memory utilization in the lines of code responsible in making system calls?
Método de aplicación:	Count the number of error messages pertaining to memory failure and warning and compare it to the estimated number of lines of code responsible in system calls.
Medición, fórmula:	X = bits/time (calculated or simulated) A = Number of memory related error messages B = Number of lines of code directly related to system calls
Interpretación:	The greater the better
Tipo de escala:	Ratio
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Source code
ISO/IEC 12207 SLCP:	Verification
Audiencia:	Developers

Tabla 55. Métrica interna de la utilización de recursos - utilización de la transmisión

Nombre:	Transmission Utilization
Propósito:	What is the estimated amount of transmission resources utilization?
Método de	Estimate the Transmission resource utilization requirements by estimating

aplicación:	the transmission volumes.
Medición, fórmula:	$X = \text{bits/time}$ (calculated or simulated)
Interpretación:	The lesser the ratio better
Tipo de escala:	Ratio
Tipo de medida:	$X = \text{time}$
Fuente de medición:	Known operating system Estimated time in system calls
ISO/IEC 12207 SLCP:	Verification
Audiencia:	Developers

• CONFORMIDAD DE EFICIENCIA

Tabla 56. Métrica interna de la conformidad con la eficiencia - conformidad con la eficiencia

Nombre:	Efficiency compliance
Propósito:	How compliant is the efficiency of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented items related to efficiency compliance confirmed in evaluation B = Total number of compliance items
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more compliant.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count/count}$ A = count B = count
Fuente de medición:	Specification of compliance and related standards, conventions or regulations Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint review

Audiencia:	Requirers Developers
-------------------	-------------------------

MANTENIBILIDAD

• CAPACIDAD DE SER ANALIZADO

Tabla 57. Métrica interna de la capacidad de ser analizado - registro de la actividad

Nombre:	Activity recording
Propósito:	How thorough is the recording of the system status?
Método de aplicación:	Count the number of items logged in the activity log as specified and compare it to the number of items required to be logged.
Medición, fórmula:	$X = A/B$ A = Number of implemented data login items as specified confirmed in review B = Number of data items to be logged defined in the specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1, more data provided to record system status. <i>Comment(s)</i> It is necessary to convert this value to the <0,1> interval if making summarization of characteristics.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Value A comes from review report. Value B comes from requirement specifications
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Maintainers Users

Tabla 58. Métrica interna de la capacidad de ser analizado – preparación de la función de diagnóstico.

Nombre:	Readiness of diagnostic function
Propósito:	How thorough is the provision of the diagnostic functions?
Método de	Count the number of implemented diagnostic functions as specified and

aplicación:	compare it to the number of diagnostic functions required in specifications.
Medición, fórmula:	$X = A/B$ A = Number of implemented diagnostic functions as specified confirmed in review B = Number of diagnostic functions required
Interpretación:	$0 \leq X$ The closer to 1, the better implementation of diagnostic Comment(s) It is necessary to convert this value to the $<0,1>$ interval if making summarization of characteristics.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Value A comes from review report Value B comes from requirement specifications
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Maintainers Users

• FACILIDAD DE CAMBIO

Tabla 59. Métrica interna de la facilidad de cambio - facilidad de registrar los cambios

Nombre:	Change recordability
Propósito:	Are changes to specifications and program modules recorded adequately in the code with comment lines?
Método de aplicación:	Record ratio of module change information.
Medición, fórmula:	$X = \text{bits}/\text{time}$ (calculated or simulated)
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more recordable. The charge control 0 indicates poor change control or little changes, high stability.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Configuration control system

	Version logs Specifications
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

- **ESTABILIDAD**

Tabla 60. Métrica interna de estabilidad - impacto en el cambio

Nombre:	Change Impact
Propósito:	What is the frequency of adverse impacts after modification?
Método de aplicación:	Count the number of detected adverse impacts after modification and compare it to the number of modifications performed.
Medición, fórmula:	$X = 1 - A/B$ A = Number of detected adverse impacts after modifications. B = Number of modifications made
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	A comes from review report B comes from review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

Tabla 61. Métrica interna de estabilidad - localización del impacto de la modificación

Nombre:	Modification impact localization
Propósito:	How large is the impact of the modification on the software product?
Método de	Count the number of affected variables from a modification and compare it to

aplicación:	the total number of variables in the product. Comment(s) Impacted variable is a) All variables in the instruction which was changed b) Variable which is in the same instruction with the variable defined by a)
Medición, fórmula:	$X = A/B$ A = Number of affected variable data by modification, confirmed in review. B = Total number of variables.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the lesser impact of modification
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	A comes from review report B comes from review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

• FACILIDAD DE PRUEBA

Tabla 62. Métrica interna de facilidad de prueba - integridad de la función de prueba predefinida

Nombre:	Completeness of built-in test function
Propósito:	How complete is the built-in test capability?
Método de aplicación:	Count the number of implemented built-in test functions as specified and compare it to the number of built-in test functions in the requirements.
Medición, fórmula:	$X = A/B$ A = Number of implemented built-in test function as specified confirmed in review B = Number of built-in test function required
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count

Fuente de medición:	A comes from review document B comes from requirements or design document
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

Tabla 63. Métrica interna de facilidad de prueba - autonomía de la facilidad de prueba

Nombre:	Autonomy of testability
Propósito:	How independently can the software be tested?
Método de aplicación:	Count the number of dependencies on other systems for testing that have been simulated with stubs and compare it with the total number of test dependencies on other systems.
Medición, fórmula:	$X = A/B$ A = Number of dependencies on other systems for testing that have been simulated with stubs B = Total number of test dependencies on other systems
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	A comes from review document B comes from requirements or design document
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

Tabla 64. Métrica interna de facilidad de prueba - facilidad de observación del desarrollo de la prueba

Nombre:	Test progress observability
Propósito:	How complete are the built in test result displays during testing?

Método de aplicación:	Count the number of implemented checkpoints as specified and compare it to the number specified checkpoints required by design.
Medición, fórmula:	$X = A/B$ A = Number of implemented checkpoints as specified confirmed in review B = Number of designed checkpoints
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	A comes from review document B comes from design document
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

• CONFORMIDAD CON LA FACILIDAD DE MANTENIMIENTO

Tabla 65. Métrica interna de la conformidad de la facilidad de mantenimiento - conformidad con la facilidad de mantenimiento

Nombre:	Maintainability compliance
Propósito:	How compliant is the maintainability of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented items related to maintainability compliance confirmed in evaluation B = Total number of compliance items
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more compliant.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count

Fuente de medición:	Specification of compliance and related standards, conventions or regulations Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Requirers

PORTABILIDAD

- **ADAPTABILIDAD**

Tabla 66. Métrica interna de adaptabilidad - adaptabilidad de las estructuras de datos

Nombre:	Adaptability of data structures
Propósito:	How adaptable is the product to the data structure changes?
Método de aplicación:	Count the number of data structures, which are operable and has no limitation after adaptation and compare it to the total number of data structures requiring adaptation capability
Medición, fórmula:	$X = A/B$ A = Number of data structures which are operable and has no limitation after adaptation, confirmed in review B = Total number of data structures requiring adaptation capability
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

Tabla 67. Métrica interna de adaptabilidad - adaptabilidad del ambiente de hardware

Nombre:	Hardware environmental adaptability
Propósito:	How adaptable is the product to the H/W related environmental change?
Método de aplicación:	Count the number of implemented functions which are capable of achieving required results in specified multiple H/W environments as specified and compare it to the number of functions with H/W environment adaptation capability requirements
Medición, fórmula:	$X = A/B$ A = Number of implemented functions which are capable of achieving required results in specified multiple H/W environment as specified, confirmed in review B = Total number of functions with H/W environment adaptation capability requirements
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

Tabla 68. Métrica interna de adaptabilidad - adaptabilidad del ambiente organizacional

Nombre:	Organizational environment adaptability
Propósito:	How adaptable is the product to organizational change?
Método de aplicación:	Count the number of implemented functions which are capable of achieving required results in specified multiple organizational and business environments as specified and compare it to the number of functions with organizational environment adaptation capability requirements.
Medición, fórmula:	$X = A/B$ A = Number of implemented functions which are capable of achieving required results in specified multiple organizational and business environment as specified, confirmed in review B = Total number of functions with organizational environment adaptation

	capability requirements
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

Tabla 69. Métrica interna de adaptabilidad - amigabilidad del usuario

Nombre:	Porting user friendliness
Propósito:	How effortless is it to perform porting operations on the product?
Método de aplicación:	Count the number of implemented functions which are capable of supporting ease-of-adaptation by user as specified and compare it to the number of functions with easy-to-adapt capability requirements.
Medición, fórmula:	$X = A/B$ A = Number of functions supporting ease-of-adaptation by user as specified, confirmed in review B = Total number of functions with ease-of-adaptation capability requirements.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more friendly.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers

	Maintainers Requirers
--	--------------------------

Tabla 70. Métrica interna de adaptabilidad - adaptabilidad del ambiente de software del sistema

Nombre:	System software environmental adaptability
Propósito:	How adaptable is the product to system software related environmental changes?
Método de aplicación:	Count the number of implemented functions which are capable of achieving required results in specified and compare it to the number of functions with system software environment adaptation capability requirements.
Medición, fórmula:	$X = A/B$ A = Number of implemented functions which are capable of achieving required results in specified multiple system software environment as specified, confirmed in review B = Total number of functions with system software environment adaptation capability requirements
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirements specifications Design Review report
ISO/IEC 12207 SLCP:	Verification Joint Review
Audiencia:	Developers Maintainers Requirers

- FACILIDAD DE INSTALACION**

Tabla 71. Métrica interna de la facilidad de instalación - facilidad de reintentar el setup

Nombre:	Ease of Setup Re-try
Propósito:	How easy is it to repeat setup operation?

Método de aplicación:	Count the number of implemented setup retry operations and compare it to the number of setup retry operations required.
Medición, fórmula:	$X = A/B$ A = Number of implemented retry operations for setup, confirmed in review B = Total number of setup operations required
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the easier
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Review report
ISO/IEC 12207 SLCP:	Validation
Audiencia:	Developers

Tabla 72. Métrica interna de la facilidad de instalación - esfuerzo en la instalación

Nombre:	Installation effort
Propósito:	What level of effort is required for installation?
Método de aplicación:	Count the number of implemented installation automated steps and compare it to the number of prescribed installation steps.
Medición, fórmula:	$X = A/B$ A = Number of automated installation steps confirmed in review B = Number of installation steps required
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Review report
ISO/IEC 12207 SLCP:	Validation
Audiencia:	Developers

Tabla 73. Métrica interna de la facilidad de instalación - flexibilidad en la instalación

Nombre:	Installation flexibility
Propósito:	How flexible and customizable is the installation capability?
Método de aplicación:	Count the number of implemented customizable installation operations as specified and compare it to the number of installation operations with customization capability requirements.
Medición, fórmula:	$X = A/B$ A = Number of implemented customizable installation operation as specified confirmed in review B = Number of customizable installation operation required <i>Comment(s)</i> Customizable: e.g., nesting depth, number of panels.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more flexible
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Review report
ISO/IEC 12207 SLCP:	Validation
Audiencia:	Developers

- COEXISTENCIA**

Tabla 74. Métrica interna de la coexistencia - coexistencia disponible

Nombre:	Available co-existence
Propósito:	How flexible is the product in sharing its environment with other products without adverse impacts on other products?
Método de aplicación:	Count the number of entities with which product can co-exist as specified and compare it to the number of entities in production environment that require co-existence.
Medición, fórmula:	$X = A/B$ A = Number of entities with which product can co-exist as specified B = Number of entities in production environment that require co-existence
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better

Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Review report Test report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Requirers Developers Maintainers

- **REEMPLAZABILIDAD**

Tabla 75. Métrica interna de la reemplazabilidad - uso continuo de los datos

Nombre:	Continued use of data
Propósito:	What is the amount of original data that remain unchanged after replacement with this product?
Método de aplicación:	Count the number of data items, that continue to be used after replacement as specified, and compare it to the total number of data items required to be used from the old data after software replacement.
Medición, fórmula:	X = A/B A = Number of software data items that continue to be used as specified after replacement, confirmed in evaluation B = Number of old data items required to be used from old software.
Interpretación:	0 <= X <= 1 The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Design Source code Review report Test report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Requirers

	Developers Maintainers
--	---------------------------

Tabla 76. Métrica interna de la reemplazabilidad - inclusividad de la función

Nombre:	Function inclusiveness
Propósito:	What's the amount of functions that remain unchanged?
Método de aplicación:	Count the number of functions covered by new software that produces similar results and compare it to the number of function in the old software.
Medición, fórmula:	$X = A/B$ A = Number of functions covered by new software that produces similar results confirmed in review. B = Number of functions in old software.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the better
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Design Source code Review report Test report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Requirers Developers Maintainers

• CONFORMIDAD DE LA PORTABILIDAD

Tabla 77. Métrica interna de la conformidad de la portabilidad - conformidad con la portabilidad

Nombre:	Portability compliance
Propósito:	How compliant is the portability of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the

	specification.
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented items related to portability compliance confirmed in evaluation. B = Total number of compliance items.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more complete.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Specification of compliance and related standards, conventions or regulations Design Source code Review report
ISO/IEC 12207 SLCP:	Verification Joint review
Audiencia:	Requirers Developers

ANEXO C

MÉTRICAS EXTERNAS⁴⁶

FUNCIONALIDAD

- ADECUACIÓN

Tabla 78. Métrica externa de adecuación - suficiencia funcional

Nombre:	Functional Adequacy
Propósito:	How adequate are the evaluated functions?
Método de aplicación:	Number of functions that are suitable for performing the specified tasks comparing to the number of function evaluated.
Medición, fórmula:	$X = 1 - A/B$ A = Number of functions in which problems are detected in evaluation B = Number of functions evaluated
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, the more adequate
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Evaluation Report
ISO/IEC 12207 SLCP:	Validation Quality Assurance Qualification testing
Audiencia:	Developer SQA

Tabla 79. Métrica externa de adecuación - integridad de la implementación funcional

Nombre:	Functional implementation complétense
Propósito:	How complete is the implementation according to requirement specifications?

⁴⁶ ISO/IEC TR 9126-2:2003. International standard “software engineering – product quality – part 2: external metrics.

Método de aplicación:	Do functional tests (black box test) of the system according to the requirement specifications. Count the number of missing functions detected in evaluation and compare with the number of function described in the requirement specifications.
Medición, fórmula:	$X = 1 - A/B$ A = Number of missing functions detected in evaluation B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification Evaluation report
ISO/IEC 12207 SLCP:	Validation Quality Assurance Qualification testing
Audiencia:	Developer SQA

Tabla 80. Métrica externa de adecuación - alcance de la implementación funcional

Nombre:	Functional implementation coverage
Propósito:	How correct is the functional implementation?
Método de aplicación:	Do functional tests (black box test) of the system according to the requirement specifications. Count the number of incorrectly implemented or missing functions and compare with the number of functions described in the requirement specifications. Count the number of functions that are complete versus the ones that are not.
Medición, fórmula:	$X = 1 - A/B$ A = Number of incorrectly implemented or missing functions detected in evaluation B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count

	B = count
Fuente de medición:	Requirement specification Evaluation report
ISO/IEC 12207 SLCP:	Validation Quality Assurance Qualification testing
Audiencia:	Developer SQA

Tabla 81. Métrica externa de adecuación - estabilidad (volatilidad) de la especificación funcional

Nombre:	Functional specification stability (volatility)
Propósito:	How stable is the functional specification after entering operation?
Método de aplicación:	Count the number of functions described in functional specifications that had to be changed after the system is put into operation and compare with the number of functions described in the requirement specifications.
Medición, fórmula:	$X = 1 - A/B$ A = Number of functions changed after entering operation starting from entering operation B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirements specifications Evaluation Report
ISO/IEC 12207 SLCP:	Problem Resolution Operation
Audiencia:	Maintainer SQA

- **EXACTITUD**

Tabla 82. Métrica externa de exactitud - exactitud a la anticipación esperada

Nombre:	Accuracy to expectation
Propósito:	Are differences between the actual and reasonable expected results acceptable?
Método de aplicación:	Do input vs output test cases and compare the output to reasonable expected results. Count the number of cases encountered by the users with an unacceptable difference from reasonable expected results.
Medición, fórmula:	$X = A/T$ A = Number of cases encountered by the users with a difference against to reasonable expected results beyond allowable. T = Operation time.
Interpretación:	$0 \leq X$ The closer to 0 is the better.
Tipo de escala:	Ratio.
Tipo de medida:	$X = \text{count}/\text{time}$ A = count T = time
Fuente de medición:	Requirement specification User operation manual Hearing to users Test report
ISO/IEC 12207 SLCP:	Validation Quality Assurance
Audiencia:	Developer User

Tabla 83. Métrica externa de exactitud - exactitud computacional

Nombre:	Computational Accuracy
Propósito:	How often do the end users encounter inaccurate results?
Método de aplicación:	Record the number of inaccurate computations based on specifications.
Medición, fórmula:	$X = A/T$ A = Number of inaccurate computations encountered by users T = Operation time
Interpretación:	$0 \leq X$ The closer to 0 is the better.

Tipo de escala:	Ratio.
Tipo de medida:	X = count/time A = count T = time
Fuente de medición:	Requirement specification Test report
ISO/IEC 12207 SLCP:	Validation Quality Assurance
Audiencia:	Developer User

Tabla 84. Métrica externa de exactitud - precisión

Nombre:	Precision
Propósito:	How often do the end users encounter results with inadequate precision?
Método de aplicación:	Record the number of results with inadequate precision.
Medición, fórmula:	$X = A/T$ A = Number results encountered by the users with level precision different from required. T = Operation time
Interpretación:	$0 \leq X$ The closer to 0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/time A = count T = time
Fuente de medición:	Requirement specification Test report
ISO/IEC 12207 SLCP:	Validation Quality Assurance
Audiencia:	Developer User

- **CONFORMIDAD DE LA FUNCIONALIDAD**

Tabla 85. Métrica externa de la conformidad con la funcionalidad - conformidad funcional

Nombre:	Functional compliance
Propósito:	How compliant is the functionality of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance in the specification. Design test cases in accordance with compliance items. Conduct functional testing for these test cases. Count the number of compliance items that have been satisfied.
Medición, fórmula:	$X = 1 - A/B$ A = Number of functionality compliance items specified that have not been implemented during testing B = Total number of functionality compliance items specified.
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Product description (User manual or Specification) of compliance and related standards, conventions or regulations. Test specification and report
ISO/IEC 12207 SLCP:	Validation Qualification testing
Audiencia:	Supplier User

Tabla 86. Métrica externa de la conformidad con la funcionalidad - conformidad estándar de la interfaz

Nombre:	Interface Standard compliance
Propósito:	How compliant are the interfaces to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of interfaces that meet required compliance and compare with the number of interfaces requiring compliance as in the specifications.
Medición, fórmula:	$X = A/B$ A = Number of correctly implemented interfaces as specified.

	B = Total number of interfaces requiring compliance.
Interpretación:	0 <= X <= 1 The closer to 1.0 is the better.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Product description of compliance and related standards, conventions or regulations. Test specification and report
ISO/IEC 12207 SLCP:	Validation
Audiencia:	Developer

SEGURIDAD

• RESISTENCIA AL ACCESO

Tabla 87. Métrica externa de resistencia al acceso- facilidad de auditar los accesos

Nombre:	Access auditability
Propósito:	How complete is the audit trail concerning the user access to the system and data?
Método de aplicación:	Evaluate the amount of accesses that the system recorded in the access history database.
Medición, fórmula:	X = A/B A = Number of “user accesses to the system and data” recorded in the access history database. B = Number of “user accesses to the system and data” done during evaluation.
Interpretación:	0 <= X <= 1 The closer to 1.0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Test specification Test Report
ISO/IEC 12207	Validation

SLCP:	
Audiencia:	Developer

Tabla 88. Métrica externa de resistencia al acceso- uso controlado del acceso

Nombre:	Access controllability
Propósito:	How controllable is access to the system?
Método de aplicación:	Count the number of detected illegal operations with comparing to number of illegal operations as in the specification.
Medición, fórmula:	$X = A/B$ A = Number of detected different types of illegal operations. B = Number of types of illegal operations as in the specification.
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Test specification Test report Operation report
ISO/IEC 12207 SLCP:	Validation Quality Assurance
Audiencia:	Developer

- **RESISTENCIA A LA COPIA**

Se definió una métrica en la investigación realizada. Ver tabla 3, Sección 4.1.2 – Métricas Investigadas.

- **FACILIDAD DEL CIFRADO**

No se tiene una métrica definida para esta subcaracterística.

- **RESISTENCIA A LA FALSIFICACIÓN**

Se definió una métrica en la investigación realizada. Ver tabla 4, Sección 4.1.2 – Métricas Investigadas.

- **ROBUSTEZ**

Tabla 89. Métrica externa de robustez - prevención de la corrupción de datos

Nombre:	Data corruption prevention
Propósito:	What is the frequency of data corruption events?
Método de aplicación:	Count the occurrences of major and minor data corruption events.
Medición, fórmula:	a) $X = 1 - A/N$ A = Number of time that a major data corruption event occurred. N = Number of test cases tried data to cause data corruption event. b) $Y = 1 - B/N$ A = Number of times that a minor data corruption event occurred. c) $Z = A/T$ or B/T A = Period of operation time (during operation testing)
Interpretación:	a) $0 \leq X \leq 1$ The closer to 1.0, is the better. b) $0 \leq Y \leq 1$ The closer to 1.0, is the better. c) $0 \leq Z$ The closer to 0, is the better.
Tipo de escala:	a) Absolute b) Absolute c) Ratio
Tipo de medida:	X = count/count A = count B = count a) X = count/count A = count N = count b) Y = count/count B = count c) Z = count/time T = time
Fuente de medición:	Requirement specification Design

	Source code Review Report
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	Developer Maintainer

- **CONFORMIDAD CON LA SEGURIDAD**

No se tiene una métrica definida para esta subcaracterística.

INTEROPERABILIDAD

- **COMPATIBILIDAD DE LA OSI**

Tabla 90. Métrica externa de la compatibilidad de la OSI - intercambiabilidad de los datos (basado en el formato de datos)

Nombre:	Data exchangeability (Data format based)
Propósito:	How correctly have the exchange interface functions for specified data transfer been implemented?
Método de aplicación:	Test each downstream interface function output record format of the system according to the data fields specifications. Count the number of data formats that are approved to be exchanged with other software or system during testing on data exchanges in comparing with the total number.
Medición, fórmula:	$X = A/B$ A = Number of data formats which are approved to be exchanged successfully with other software or system during testing on data exchanges. B = Total number of data formats to be exchanged.
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0 is the better.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirement specification (User manual) Test Report

ISO/IEC 12207 SLCP:	Validation
Audiencia:	Developer

Tabla 91. Métrica externa de la compatibilidad de la OSI - intercambiabilidad de los datos (éxito del usuario basado en intentos)

Nombre:	Data exchangeability (User's success attempt based)
Propósito:	How often does the end user fail to exchange data between target software and other software? How often are the data transfers between target software and other software successful? Can user usually succeed in exchanging data?
Método de aplicación:	Count the number of cases that interface functions were used and failed.
Medición, fórmula:	a) $X = 1 - A/B$ A = Number of cases in which user failed to exchange with other software or systems. B = Number of cases in which user attempted to exchange data. b) $Y = A/T$ T = Period of operation time
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better. $0 \leq Y$ The closer to 0 is the better.
Tipo de escala:	a) Absolute b) Absolute
Tipo de medida:	X = count/time A = count T = time Y = count/time T = time
Fuente de medición:	Requirement specification (User manual) Test report
ISO/IEC 12207 SLCP:	Operation
Audiencia:	Maintainer

- **COMPATIBILIDAD DEL SOFTWARE**

Se definió una métrica en la investigación realizada. Ver tabla 5, Sección 4.1.2 – Métricas Investigadas.

- **COMPATIBILIDAD DE LOS DATOS**

Se definió una métrica en la investigación realizada. Ver tabla 6, Sección 4.1.2 – Métricas Investigadas.

- **TRAZABILIDAD**

Se definió una métrica en la investigación realizada. Ver tabla 7, Sección 4.1.2 – Métricas Investigadas.

- **CONFORMIDAD CON LA INTEROPERABILIDAD**

No se tiene una métrica definida para esta subcaracterística.

FIABILIDAD

- **MADUREZ**

Tabla 92. Métrica externa de madurez - densidad latente estimada de los fallos

Nombre:	Estimated latent fault density
Propósito:	How many problems still exist that may emerge as future faults?
Método de aplicación:	Count the number of faults detected during a defined trial period and predict potential number of future faults using a reliability growth estimation model.
Medición, fórmula:	$X = \{ABS(A1 - A2)\}/B$ (X: Estimated residuary latent fault density) ABS() = Absolute Value A1 = Total number of predicted latent faults in a software product A2 = Total number of actually detected faults B = product size
Interpretación:	$0 \leq X$ It depends on stage of testing At the later stages, smaller is better.

Tipo de escala:	Absolute.
Tipo de medida:	X = count/size A1 = count A2 = count B = size
Fuente de medición:	Test report Operation report Problem report
ISO/IEC 12207 SLCP:	Integration Qualification Testing Operation Validation Quality Assurance
Audiencia:	Developer Tester SQA User

Tabla 93. Métrica externa de madurez - densidad de la falla contra los casos de prueba

Nombre:	Failure density against test cases
Propósito:	How many failures were detected during defined trial period?
Método de aplicación:	Count the number of detected failures and performed test cases.
Medición, fórmula:	$X = A1/A2$ A1 = Number of detected failures A2 = Number of performed test cases
Interpretación:	$0 \leq X$ It depends on stage of testing At the later stages, smaller is better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/time A = count T = time
Fuente de medición:	Test report Operation report Problem report
ISO/IEC 12207 SLCP:	Integration Qualification Testing Operation Quality Assurance

Audiencia:	Developer Tester SQA
-------------------	----------------------------

Tabla 94. Métrica externa de madurez - resolución de la falla

Nombre:	Failure resolution
Propósito:	How many failure conditions are resolved?
Método de aplicación:	Count the number of failures that did not reoccur during defined trial period under the similar conditions. Maintain a problem resolution report describing status of all the failures.
Medición, fórmula:	$X = A1/A2$ A1 = Number of resolved failures A2 = Total number of actually detected failures.
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better as more failures are resolved.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A1 = count A2 = count A3 = count
Fuente de medición:	Test report Operation (test) report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation
Audiencia:	User SQA Maintainer

Tabla 95. Métrica externa de madurez - densidad de las fallas

Nombre:	Fault density
Propósito:	How many faults were detected during defined trial period?
Método de aplicación:	Count the number of detected faults and compute density
Medición, fórmula:	$X = A/B$ A = Number of detected faults B = product size

Interpretación:	0 ≤ X It depends on stage of testing At the later stages, smaller is better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/size A = count B = size
Fuente de medición:	Test report Operation report Problem report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Quality Assurance
Audiencia:	Developer Tester SQA

Tabla 96. Métrica externa de madurez - remoción de fallos

Nombre:	Fault removal
Propósito:	How many faults have been corrected?
Método de aplicación:	Count the number of faults removed during testing and compare with the total number of faults detected and total number of faults predicted
Medición, fórmula:	a) $X = A1/A2$ A1 = Number of corrected faults A2 = Total number of actually detected faults b) $Y = A1/A3$ A3 = Total number of predicted latent faults in the software product.
Interpretación:	a) $0 \leq X \leq 1$ The closer to 1.0, is better as fewer faults remain. b) $0 \leq Y$ The closer to 1.0, is better as fewer faults remain.
Tipo de escala:	a) Absolute. b) Absolute.
Tipo de medida:	Y = count/count B = count
Fuente de medición:	Test report Organization Database

ISO/IEC 12207 SLCP:	Integration Validation Qualification testing Quality Assurance
Audiencia:	Developer SQA Maintainer

Tabla 97. Métrica externa de madurez - tiempo medio entre fallas (MTBF)

Nombre:	Mean time between failures (MTBF)
Propósito:	How frequently does the software fail in operation?
Método de aplicación:	Count the number of failures occurred during a defined period of operation and compute the average interval between the failures.
Medición, fórmula:	a) $X = T1/A$ b) $Y = T2/A$ T1 = operation time T2 = sum of time intervals between consecutive failure occurrences A = total number of actually detected failures (Failures occurred during observed operation time)
Interpretación:	$0 < X, Y$ The longer is the better as longer time can be expected between failures.
Tipo de escala:	a) Ratio. b) Ration
Tipo de medida:	A = Count T1 = Time T2 = Time X = time/count Y = time/count
Fuente de medición:	Test report Operation (test) report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation testing Operation
Audiencia:	Maintainer User

Tabla 98. Métrica externa de madurez - cobertura de prueba (cobertura de pruebas especificadas del escenario de la operación)

Nombre:	Test coverage (Specified operation scenario testing coverage)
Propósito:	How much of required test cases have been executed during testing?
Método de aplicación:	Count the number of test cases performed during testing and compare the number of test cases required to obtain adequate test coverage.
Medición, fórmula:	$X = A/B$ A = Number of actually performed test cases representing operation scenario during testing B = Number of test cases to be performed to cover requirements.
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better test coverage.
Tipo de escala:	Absolute
Tipo de medida:	X = count/time A = count T = time
Fuente de medición:	Requirement specification, Test specification or User manual Test report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Validation Quality Assurance
Audiencia:	Developer Tester SQA

Tabla 99. Métrica externa de madurez - madurez de la prueba

Nombre:	Test maturity
Propósito:	Is the product well tested? <i>Comment(s)</i> This is to predict the success rate the product will achieve in future testing.
Método de aplicación:	Count the number of passed test cases which have been actually executed and compare it to the total number of test cases to be performed as per requirements.
Medición, fórmula:	$X = A/B$ A = Number of passed test cases during testing or operation B = Number of test cases to be performed to cover requirements.
Interpretación:	$0 \leq X \leq 1$

	The closer to 1 is the better.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Requirement specification, Test specification or User manual Test report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Quality Assurance
Audiencia:	Developer Tester SQA

• TOLERANCIA A FALLOS

Tabla 100. Métrica externa de tolerancia a fallos - evasión de la interrupción

Nombre:	Breakdown avoidance
Propósito:	How often the software product causes the break down of the total production environment?
Método de aplicación:	Count the number of breakdowns occurrence with respect to number of failures. If it is under operation, analyze log of user operation history.
Medición, fórmula:	$X = 1 - A/B$ A = Number of breakdowns B = Number of failures
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation
Audiencia:	User

	Maintainer
--	------------

Tabla 101. Métrica externa de tolerancia a fallos - prevención de fallas

Nombre:	Failure avoidance
Propósito:	How many fault patterns were brought under control to avoid critical and serious failures?
Método de aplicación:	Count the number of avoided fault patterns and compare it to the number of fault patterns to be considered.
Medición, fórmula:	$X = A/B$ A = Number of avoided critical and serious failure occurrences against test cases of fault pattern. B = Number of executed test cases of fault pattern (almost causing failure) during testing.
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is better as the user can more often avoid critical or serious failure.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Validation
Audiencia:	User Maintainer

Tabla 102. Métrica externa de tolerancia a fallos - prevención de la operación incorrecta

Nombre:	Incorrect operation avoidance
Propósito:	How many functions are implemented with incorrect operations avoidance capability?
Método de aplicación:	Count the number of test cases of incorrect operations which were avoided to cause critical and serious failures and compare it to the number of executed test cases of incorrect operation patterns to be considered.
Medición,	$X = A/B$

fórmula:	A = Number of avoided critical and serious failures occurrences B = Number of executed test cases of incorrect operation patterns (almost causing failure) during testing
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is better as more incorrect user operation is avoided
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation
Audiencia:	User Maintainer

• RECUPERABILIDAD

Tabla 103. Métrica externa de recuperabilidad - disponibilidad

Nombre:	Availability
Propósito:	How available is the system for use during the specified of time?
Método de aplicación:	Test system in a production like environment for a specified period of time performing all user operations. Measure the repair time period each time the system was unavailable during the trial. Compute mean time to repair.
Medición, fórmula:	a) $X = \{ To / (To + Tr) \}$ b) $Y = A1 / A2$ b) To = operation time Tr = time to repair A1 = total available cases of user's successful software use when user attempt to use A2 = Total number of cases of user's attempt to use the software during observation time. This is from the user callable function operation view.
Interpretación:	$0 \leq X \leq 1$ The larger and closer to 1 is the better, as the user can use the software for

	more time. $0 \leq Y \leq 1$ The larger and closer to 1 is the better.
Tipo de escala:	(a),(b) Absolute
Tipo de medida:	To = Time Tr = Time X = Time/Time A1 = Count A2 = Count Y = count/count
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation
Audiencia:	User Maintainer

Tabla 104. Métrica externa de recuperabilidad - tiempo medio de inactividad

Nombre:	Mean down time
Propósito:	What is the average time the system stays unavailable when a failure occurs before gradual start up?
Método de aplicación:	Measure the down time each time the system is unavailable during a specified trial period and compute the mean time.
Medición, fórmula:	$X = T/N$ T = Total down time N = Number of observed breakdowns The worst case or distribution of down time should be measured.
Interpretación:	$0 < X$ The smaller is the better, system will be down for shorter time.
Tipo de escala:	Ratio
Tipo de medida:	T = Time N = Count X = time/count
Fuente de medición:	Test report Operation report
ISO/IEC 12207	Integration

SLCP:	Qualification testing Operation Validation
Audiencia:	User Maintainer

Tabla 105. Métrica externa de recuperabilidad - tiempo medio de recuperación

Nombre:	Mean recovery time
Propósito:	What is the system takes to complete recovery from initial partial recovery?
Método de aplicación:	Measure the full recovery times for each of the time the system was brought down during the specified trial period and compute the mean time.
Medición, fórmula:	$X = \text{Sum}(T)/B$ T = Time to recovery downed software system at each opportunity N = Number of cases which observed software system entered into recovery
Interpretación:	$0 < X$ The smaller is the better.
Tipo de escala:	Ratio.
Tipo de medida:	T = Time N = Count X = Time/Count
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Validation
Audiencia:	User Maintainer

Tabla 106. Métrica externa de recuperabilidad - capacidad para recomenzar

Nombre:	Restartability
Propósito:	How often the system can restart providing service to users within a required time?
Método de aplicación:	Count the number of times the system restarts and provides service to users within a target required time and compare it to the total number of restarts, when the system was brought down during the specified trial period.

Medición, fórmula:	$X = A/B$ A = Number of restarts which met to required time during testing or user operation support B = Total number of restarts during testing or user operation support
Interpretación:	$0 \leq X \leq 1$ The larger and closer to 0 is the better, as the user can restart easily.
Tipo de escala:	Absolute
Tipo de medida:	A = Count B = Count $X = \text{count}/\text{count}$
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Validation
Audiencia:	User Maintainer

Tabla 107. Métrica externa de recuperabilidad - facilidad de restauración

Nombre:	Restorability
Propósito:	How capable is the product in restoring itself after abnormal event or at request?
Método de aplicación:	Count the number of successful restorations and compare it to the number of tested restoration required in the specifications. Restoration requirement examples: database checkpoint, transaction checkpoint, redo function, undo function, etc.
Medición, fórmula:	$X = A/B$ A = Number of restoration cases successfully done B = Number of restoration cases tested as per requirements
Interpretación:	$0 \leq X \leq 1$ The larger and closer to 1.0 is better, as the product is more capable to restore in defined cases.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Requirement specification Test specification or User manual

	Test Report Operation Report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Validation
Audiencia:	User Maintainer

Tabla 108. Métrica externa de recuperabilidad - efectividad de la restauración

Nombre:	Restoration Effectiveness
Propósito:	How effective is the restoration capability?
Método de aplicación:	Count the number of tested restoration meeting target restoration time and compare it to the number of restorations required with specified target time.
Medición, fórmula:	$X = A/B$ A = Number of cases successfully restored meeting the target restore time B = Number of cases performed
Interpretación:	$0 \leq X \leq 1$ The larger and closer to 1.0, is the better, as the restoration process in product is more effective.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Validation
Audiencia:	User Maintainer

- **CONFORMIDAD DE LA FIABILIDAD**

Tabla 109. Métrica externa de la conformidad con la fiabilidad - conformidad con la fiabilidad.

Nombre:	Reability compliance
Propósito:	How compliant is the reliability of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification
Medición, fórmula:	$X = 1 - A/B$ A = Number of reliability compliance items specified that have not been implemented during testing B = Total number of reliability compliance items specified
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0, is the better.
Tipo de escala:	Absolute.
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Product description (User manual or Specification) of compliance and related standards, conventions or regulations. Test specification and report
ISO/IEC 12207 SLCP:	Qualification testing Validation
Audiencia:	Supplier User

USABILIDAD

- **APROPIABILIDAD**

Tabla 110. Métrica externa de la apropiabilidad - integridad de la descripción

Nombre:	Completeness of description
Propósito:	What proportion of functions (or types of function) is understood after reading the product description?
Método de aplicación:	Conduct user test and interview user with questionnaires or observe user behavior. Count the number of functions which are adequately understood and compare

	with the total number of functions in the product.
Medición, fórmula:	X = A/B A = Number of functions (or types of functions) understood. B = Total number of functions (or types of functions)
Interpretación:	0 ≤ X ≤ 1 The closer to 1.0 is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	User Manual Operation (test) report
ISO/IEC 12207 SLCP:	Qualification testing Operation
Audiencia:	User Maintainer

Tabla 111. Métrica externa de la apropiabilidad - accesibilidad a la demostración

Nombre:	Demonstration accessibility
Propósito:	What proportion of the demonstrations/tutorials can the user access?
Método de aplicación:	Conduct user test and observe user behaviour. Count the number of functions that are adequately demonstrable and compare with the total number of functions requiring demonstration capability.
Medición, fórmula:	X = A/B A = Number of cases in which user successfully sees demonstration when user attempts to see demonstration. B = Number of cases in which user attempts to see demonstration during observation period.
Interpretación:	0 ≤ X The closer to 0 is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = Count B = Count X = Count/Count
Fuente de medición:	User Manual Operation (test) report
ISO/IEC 12207 SLCP:	Qualification testing Operation

Audiencia:	User Maintainer
-------------------	--------------------

Tabla 112. Métrica externa de la apropiabilidad - accesibilidad a la demostración en uso

Nombre:	Demonstration accessibility in use
Propósito:	What proportion of the demonstrations / tutorials can the user access whenever user actually needs to do during operation?
Método de aplicación:	Observe the behaviour of the user who is trying to see demonstration/tutorial. Observation may employ human cognitive action monitoring approach with video camera.
Medición, fórmula:	$X = A/B$ A = Number of cases in which user successfully sees demonstration when user attempts to see demonstration B = Number of cases in which user attempts to see demonstration during observation period.
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = Count B = Count X = Count/Count
Fuente de medición:	User Manual Operation (test) report User Monitoring record (video tape and action record)
ISO/IEC 12207 SLCP:	Qualification testing Operation
Audiencia:	User Maintainer

Tabla 113. Métrica externa de la apropiabilidad - eficacia de la demostración

Nombre:	Demonstration effectiveness
Propósito:	What proportion of functions can the user operate successfully after a demonstration or tutorial?
Método de aplicación:	Observe the behavior of the user who is trying to see demonstration/tutorial. Observation may employ human cognitive action monitoring approach with video camera.
Medición, fórmula:	$X = A/B$

	A = Number of functions operated successfully B = Number of demonstrations/tutorials accessed
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = Count B = Count X = Count/Count
Fuente de medición:	User Manual Operation (test) report
ISO/IEC 12207 SLCP:	Qualification testing Operation
Audiencia:	User Maintainer

Tabla 114. Métrica externa de la apropiabilidad - funciones evidentes

Nombre:	Evident functions
Propósito:	What proportion of functions (or types of function) can be identified by the user based upon start up conditions?
Método de aplicación:	Conduct user test and interview user with questionnaires or observe user behavior. Count the number of functions that are evident to the user and compare with the total number of functions.
Medición, fórmula:	$X = A/B$ A = Number of functions (or types of functions) identified by the user B = Total number of actual functions (or types of functions)
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0 is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	User Manual Operation (test) report
ISO/IEC 12207 SLCP:	Qualification testing Operation
Audiencia:	User Maintainer

Tabla 115. Métrica externa de la apropiabilidad - comprensibilidad de la función

Nombre:	Function understandability
Propósito:	What proportion of the product functions will the user be able to understand correctly?
Método de aplicación:	Conduct user test and interview user with questionnaires. Count the number of user interface functions where purposes are easily understood by the user and compare with the number of functions available for user.
Medición, fórmula:	$X = A/B$ A = Number of user interface functions whose purpose is correctly described by the user B = Number of functions available from the interface.
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0 the better.
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	User Manual Operation (test) report
ISO/IEC 12207 SLCP:	Qualification testing Operation
Audiencia:	User Maintainer

Tabla 116. Métrica externa de la apropiabilidad - entrada y salida comprensibles

Nombre:	Understandable input and output
Propósito:	Can users understand what is required as input data and what is provided as output by software system?
Método de aplicación:	Conduct user test and interview user with questionnaires or observe user behaviour. Count the number of input and output data items understood by the user and compare with the total number of them available for user.
Medición, fórmula:	$X = A/B$ A = Number of input and output data items which user successfully understands B = Number of input and output data items available from the interface

Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	User Manual Operation (test) report
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Maintainer

• FACILIDAD DE APRENDIZAJE

Tabla 117. Métrica externa de la facilidad de aprendizaje - facilidad de aprender de la función

Nombre:	Ease of function learning
Propósito:	How long does the user take to learn to use a function?
Método de aplicación:	Conduct user test and observe user behaviour
Medición, fórmula:	$T = \text{Mean time taken to learn to use a function correctly}$
Interpretación:	$0 < T$ The shorter is the better.
Tipo de escala:	Ratio.
Tipo de medida:	T = time
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Maintainer

Tabla 118. Métrica externa de la facilidad de aprendizaje - facilidad de aprender a realizar una tarea en uso

Nombre:	Ease of learning to perform a task in use
Propósito:	How long does the user take to learn how to perform the specified task efficiently?
Método de aplicación:	Observe user behaviour from when they start to learn until they begin to operate efficiently
Medición, fórmula:	$T = \text{Sum of user operation time until user achieved to perform the specified task within a short time}$
Interpretación:	$0 < T$ The shorter is the better.
Tipo de escala:	Ratio
Tipo de medida:	$T = \text{time}$
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Maintainer

Tabla 119. Métrica externa de la facilidad de aprendizaje - eficacia de la documentación del usuario y/o la ayuda del sistema

Nombre:	Effectiveness of the user documentation and/or help system
Propósito:	What proportion of task can be completed correctly after using the user documentation and/or help system?
Método de aplicación:	Conduct user test and observe user behaviour. Count the number of tasks successfully completed after accessing online help and/or documentation and compare with the total number of tasks tested.
Medición, fórmula:	$X = A/B$ $A = \text{Number of tasks successfully completed after accessing online help and/or documentation}$ $B = \text{Total of number of tasks tested.}$
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	$A = \text{count}$

	B = count X = count/count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 120. Métrica externa de la facilidad de aprendizaje - eficacia de la documentación del usuario y/o la ayuda del sistema en uso

Nombre:	Effectiveness of the user documentation and/or help systems in use
Propósito:	What proportion of functions can be used correctly after reading the documentation or using help systems?
Método de aplicación:	Observe user behaviour. Count the number of functions used correctly after reading the documentation or using help systems and compare with the total number of functions.
Medición, fórmula:	$X = A/B$ A = Number of functions that can be used B = Total of number of functions provided
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	User manual Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 121. Métrica externa de la facilidad de aprendizaje - accesibilidad de la ayuda

Nombre:	Help accessibility
Propósito:	What proportion of the help topics can the user locate?
Método de aplicación:	Conduct use test and observe user behaviour. Count the number of tasks for which correct online help is located and compare with the total number of tasks tested.
Medición, fórmula:	$X = A/B$ A = Number of tasks for which correct online help is located B = Total of number of tasks tested
Interpretación:	$0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 122. Métrica externa de la facilidad de aprendizaje - frecuencia de la ayuda

Nombre:	Help frequency
Propósito:	How frequently does a user have to access help to learn operation to complete his/her work task?
Método de aplicación:	Conduct user test and observe user behaviour. Count the number of cases that a user accesses help to complete his/her task.
Medición, fórmula:	$X = A$ A = Number of accesses to help until a user completes his/her task.
Interpretación:	$0 \leq X$ The closer to 0 is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count A = count
Fuente de medición:	Operation (test) report

	User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

• OPERABILIDAD

Tabla 123. Métrica externa de la operabilidad - consistencia operacional en uso

Nombre:	Operational consistency in use
Propósito:	How consistent are the component of the user interface?
Método de aplicación:	Observe the behaviour of the user and ask the opinion
Medición, fórmula:	a) $X = 1 - A/B$ A = Number of messages or functions which user found unacceptably inconsistent with the user's expectation. b) $Y = N/UOT$ N = Number operations which user found unacceptably inconsistent with user's expectation UOT = user operating time (during observation period)
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better. $0 \leq Y$ The smaller and closer to 0, is the better.
Tipo de escala:	a) Absolute b) Ratio
Tipo de medida:	A = count B = count X = count/count UOT = Time N = Count Y = Count/Time
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing

	Operation
Audiencia:	User Human interface designer

Tabla 124. Métrica externa de la operabilidad - corrección de error

Nombre:	Error correction
Propósito:	Can user easily correct error on task?
Método de aplicación:	Conduct user test and observe behaviour.
Medición, fórmula:	$T = T_c - T_s$ T_c = Time of completing correction of specified type errors of performed task T_s = Time of starting correction of specified type errors of performed task.
Interpretación:	$0 \leq T$ The shorter is the better.
Tipo de escala:	Ratio
Tipo de medida:	T_s, T_c = Time T = Time
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 125. Métrica externa de la operabilidad - corrección de error en uso

Nombre:	Error correction in use
Propósito:	Can user easily recover his/her error or retry task? Can user easily recover his/her input?
Método de aplicación:	Observe the behaviour of the user who is operating software. Observe the behaviour of the user who is operating software.
Medición, fórmula:	a) $X = A / UOT$ A = Number of times that the user succeeds to cancel their error operation UOT = user operating time during observation period

	Comment(s) When function is tested one by one, the ratio be also calculated, that is the ratio of number of functions which user succeeds to cancel his/her operation to all functions.
Interpretación:	$0 \leq X$ The higher the better. $0 \leq X \leq 1$ The closer to 1 is the better.
Tipo de escala:	Ratio Absolute
Tipo de medida:	A = Count UOT = Time X = Count/Time A = count B = count X = count/count
Fuente de medición:	Operation (test) report User monitoring record Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation Validation Qualification testing Operation
Audiencia:	User Human interface designer User Human interface designer

Tabla 126. Métrica externa de la operabilidad - disponibilidad del valor prefijado en uso

Nombre:	Default value availability in use
Propósito:	Can user easily select parameter values for his/her convenient operation?
Método de aplicación:	Observe the behaviour of the user who is operating software.

	Count how many times user attempts to establish r to select parameter values and fails, (because user can not use default values provided by the software)
Medición, fórmula:	$X = 1 - A/B$ A = The number of times that the user fail to establish or to select parameter values in a short period (because user can not use default values provided by the software) B = Total number of times that the user attempt to establish or to select parameter values
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 127. Métrica externa de la operabilidad - comprensibilidad del mensaje en uso

Nombre:	Message understandability in use
Propósito:	Can user easily understand messages from software system? Is there any message which caused the user a delay in understanding before starting the next action? Can user easily memorise important message?
Método de aplicación:	Observe user behaviour who is operating software.
Medición, fórmula:	$X = A/UOT$ A = Number of times that the user pauses for a long period or successively and repeatedly fails at the same operation, because of the lack of message comprehension. UOT = User operating time (observation period)
Interpretación:	$0 \leq X$ The smaller and closer to 0, is the better.
Tipo de escala:	Ratio.
Tipo de medida:	X = count/count

	A = count B = count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 128. Métrica externa de la operabilidad - mensajes de error que se explican por sí mismos

Nombre:	Self-explanatory error messages
Propósito:	In what proportion of error conditions does the user propose the correct recovery action?
Método de aplicación:	Conduct user test and observe user behaviour.
Medición, fórmula:	$X = A/B$ A = Number of error conditions for which the user proposes the correct recovery action B = Number of error conditions tested
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 129. Métrica externa de la operabilidad - facilidad de recuperación de un error operacional en uso

Nombre:	Operational error recoverability in use
Propósito:	Can user easily recover his/her worse situation?
Método de aplicación:	Observe the behaviour of the user who is operating software.
Medición, fórmula:	$X = 1 - A/B$ A = Number of unsuccessfully recovered situation (after a user error or change) in which user was not informed about a risk by the system B = Number of user errors or changes
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count $X = \text{count}/\text{count}$
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 130. Métrica externa de la operabilidad - tiempo entre las operaciones humanas del error en uso

Nombre:	Time between human error operations in use
Propósito:	Can user operate the software long enough without human error?
Método de aplicación:	Observe the behaviour of the user who is operating software.
Medición, fórmula:	$X = T/N$ (at time t during [t-T,t]) T = Operation time period during observation (or The sum operating time between user's human error operations) N = Number of occurrences of user's human error operation
Interpretación:	$0 < X$ The higher is the better.
Tipo de escala:	Ratio

Tipo de medida:	T = Time N = Count X = Time/Count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 131. Métrica externa de la operabilidad - capacidad de deshacer (corrección de error del usuario)

Nombre:	Undoability (User error correction)
Propósito:	How frequently does the user successfully correct input errors? How frequently does the user correctly undo errors?
Método de aplicación:	Conduct user test and observe user behaviour. Conduct user test and observe user behaviour.
Medición, fórmula:	a) $X = A/B$ A = Number of input errors which the user successfully corrects B = Number of attempts to correct input errors b) $Y = A/B$ A = Number of error conditions which the user successfully corrects B = Total number of error conditions tested
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better. $0 \leq Y \leq 1$ The closer to 1, is the better.
Tipo de escala:	a) Absolute b) Absolute
Tipo de medida:	A = count B = count X = count/count A = count B = count Y = count/count
Fuente de medición:	Operation (test) report

	User monitoring record
	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation Validation Qualification testing Operation
Audiencia:	User Human interface designer User Human interface designer

Tabla 132. Métrica externa de la operabilidad - uso de la personalización

Nombre:	Customisability
Propósito:	Can user easily customize operation procedures for his/her convenience? Can a user, who instructs end users, easily set customized operation procedure templates for preventing their errors? What proportion of functions can be customized?
Método de aplicación:	Conduct user test and observe user behavior.
Medición, fórmula:	$X = A/B$ A = Number of functions successfully customized B = Number of attempts to customizability
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0 is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	User Manual Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation

Audiencia:	User Human interface designer
-------------------	----------------------------------

Tabla 133. Métrica externa de la operabilidad - reducción del procedimiento de la operación

Nombre:	Operation procedure reduction
Propósito:	Can user easily reduce operation procedures for his/her convenience?
Método de aplicación:	Count user's strokes for specified operation and compare them between before and after customizing operation.
Medición, fórmula:	$X = 1 - A/B$ A = Number of reduced operation procedures after customizing operation B = Number of operation procedures before customizing operation
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 134. Métrica externa de la operabilidad - accesibilidad física

Nombre:	Physical accessibility
Propósito:	What proportion of functions can be accessed by users with physical handicaps?
Método de aplicación:	Conduct user test and observe user behavior.
Medición, fórmula:	$X = A/B$ A = Number of functions successfully accessed B = Number of functions
Interpretación:	$0 \leq X \leq 1$ The closer to 1.0 is the better.
Tipo de escala:	Absolute

Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Operation (test) report User monitoring record
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

- **CAPACIDAD DE AYUDA**

No se tiene una métrica definida para esta subcaracterística.

- **ATRACTABILIDAD**

Tabla 135. Métrica externa de atractibilidad - Interacción atractiva

Nombre:	Attractive interaction
Propósito:	How attractive is the interface to the user?
Método de aplicación:	Questionnaire to users
Medición, fórmula:	Questionnaire to assess the attractiveness of the interface to users, after experience of usage.
Interpretación:	Depende on its questionnaire scoring method.
Tipo de escala:	Absolute
Tipo de medida:	Count
Fuente de medición:	Questionnaire result
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

Tabla 136. Métrica externa de atractibilidad - personalización de la apariencia de la interfaz

Nombre:	Interface appearance customizability
Propósito:	What proportion of interface elements can be customized in appearance to the user's satisfaction?
Método de aplicación:	Conduct user test and observe user behaviour.
Medición, fórmula:	$X = A/B$ A = Number of interface elements customized in appearance to user's satisfaction B = Number of interface elements that the user wishes to customise
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count $X = \text{count}/\text{count}$
Fuente de medición:	User's requests Operation (test) report
ISO/IEC 12207 SLCP:	Validation Qualification testing Operation
Audiencia:	User Human interface designer

- **CONFORMIDAD DE USO**

Tabla 137. Métrica externa de la conformidad de uso - conformidad con la usabilidad

Nombre:	Usability compliance
Propósito:	How completely does the software adhere to the standards, conventions, style guides or regulations relating to usability?
Método de aplicación:	Specify required compliance items based on standards, conventions, style guides or regulations relating to usability. Design test cases in accordance with compliance items. Conduct functional testing for these test cases.
Medición, fórmula:	$X = 1 - A/B$ A = Number of usability compliance items specified that have not been implemented during testing B = Total number of usability compliance items specified

Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Product description (User manual or Specification) of compliance and related standards, conventions, style guides or regulations. Testing specification and report
ISO/IEC 12207 SLCP:	Qualification testing Validation
Audiencia:	Supplier User

EFICIENCIA

• COMPORTAMIENTO EN EL TIEMPO

Tabla 138. Métrica externa del comportamiento en el tiempo - tiempo de respuesta

Nombre:	Response time
Propósito:	What is the time taken to complete a specified task? How long does it take before the system response to a specified operation?
Método de aplicación:	Start a specified task. Measure the time it takes for the sample to complete its operation. Keep a record of each attempt.
Medición, fórmula:	$T = (\text{time of gaining the result}) - (\text{time of command entry finished})$
Interpretación:	$0 < T$ The sooner is the better.
Tipo de escala:	Ratio
Tipo de medida:	T = time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User

	Developer Maintainer SQA
--	--------------------------------

Tabla 139. Métrica externa del comportamiento en el tiempo - tiempo de respuesta (tiempo medio para responder)

Nombre:	Response time (Mean time to response)
Propósito:	What is the average wait time the user experiences after issuing a request until the request is completed within a specified system load in terms of concurrent tasks and system utilization?
Método de aplicación:	Execute a number of scenarios of concurrent task. Measure the time it takes to complete the selected operation(s). Keep a record of each attempt and compute the mean time for each scenario.
Medición, fórmula:	$X = T_{mean} / TX_{mean}$ $T_{mean} = \sum(T_i) / N$, (for $I = 1$ to N) TX_{mean} = required mean response time T_i = response time for i-th evaluation (shot) N = Number of evaluations (sampled shots)
Interpretación:	$0 \leq X$ The nearer to 1 and less than 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	T_{mean} = Time TX_{mean} = Time T_i = Time N = Count X = Time/Time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 140. Métrica externa del comportamiento en el tiempo - tiempo de respuesta (proporción al peor caso del tiempo de respuesta)

Nombre:	Response time (Worst case response time ratio)
Propósito:	What is the absolute limit on time required in fulfilling a function? In the worst case, can user still get response within the specified time limit? In the worst case, can user still get reply from the software within a time short enough to be tolerable for user?
Método de aplicación:	Calibrate the test. Emulate a condition where by the system reaches a maximum load situation. Run application and monitor result(s).
Medición, fórmula:	$X = T_{max} / R_{max}$ $T_{max} = \text{MAX}(T_i) = \text{maximum response time among evaluations}$ $N = \text{number of evaluations (sampled shots)}$ $T_i = \text{response time for } i\text{-th evaluation (shot)}$ Comment(s) Distribution may be calculated as illustrated below. Statistical maximal ratio $Y = T_{dev} / R_{max}$ $T_{dev} = T_{mean} + K (DEV)$ T_{dev} is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation. K : coefficient (2 or 3) $DEV = \text{SQRT} \{ \sum ((T_i - T_{mean})^2) / (N-1) \}$ (for $i = 1$ to N) $T_{mean} = \sum(T_i) / N$, (for $I = 1$ to N) $TX_{mean} = \text{required mean response time}$
Interpretación:	$0 < X$ The nearer to 1 and less than 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	$T_{mean} = \text{Time}$ $R_{mean} = \text{Time}$ $T_i = \text{Time}$ $N = \text{Count}$ $X = \text{Time/Time}$
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 141. Métrica externa del comportamiento en el tiempo - rendimiento de procesamiento

Nombre:	Throughput
Propósito:	How many tasks can be successfully performed over a given period of time?
Método de aplicación:	Calibrate each task according to the intended priority given. Start several job tasks. Measure the time it takes for the measured task to complete its operation. Keep a record of each attempt.
Medición, fórmula:	$X = A/T$ A = Number of completed task. T = Observation time period.
Interpretación:	$0 < X$ The larger is the better.
Tipo de escala:	Ratio
Tipo de medida:	A = count T = time X = count/time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 142. Métrica externa del comportamiento en el tiempo - rendimiento de procesamiento (cantidad media de rendimiento)

Nombre:	Throughput (Mean amount of throughput)
Propósito:	What is the average number of concurrent task the system can handle over a set unit of time?
Método de aplicación:	Calibrate each task according to intended priority. Execute a number of concurrent tasks. Measure the time it takes to complete the selected task in the given traffic. Keep a record of each attempt.
Medición, fórmula:	$X = X_{\text{mean}} / R_{\text{mean}}$ $X_{\text{mean}} = \sum(T_i)/N$

	Rmean = required mean throughput $X_i = A_i / T_i$ A_i = Number of concurrent tasks observed over set period of time for i-th evaluation T_i = Set period of time for i-th evaluation N = Number of evaluations
Interpretación:	$0 < X$ The larger is the better.
Tipo de escala:	Absolute
Tipo de medida:	Xmean = Count Rmean = Count A_i = Count T_i = Time X_i = Count/Time N = Count X = Count/Count
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 143. Métrica externa del comportamiento en el tiempo - rendimiento de procesamiento (proporción del peor caso de rendimiento de procesamiento)

Nombre:	Throughput (Worst case throughput ratio)
Propósito:	What is the absolute limit on the system in terms of the number and handling of concurrent tasks as throughput?
Método de aplicación:	Calibrate the test. Emulate the condition whereby the system reaches a situation of maximum load. Run job tasks concurrently and monitor result(s).
Medición, fórmula:	$X = X_{max} / R_{max}$ $X_{max} = \text{MAX}(X_i)$ (for $i = 1$ to N) R_{max} = Required maximum throughput $\text{MAX}(X_i)$ = Maximum number of job tasks among evaluations. $X_i = A_i / T_i$ A_i = Number of concurrent tasks observed over set period of time for i-th

	evaluation T_i = Set period of time for i-th evaluation N = Number of evaluations
Interpretación:	$0 < X$ The larger is the better.
Tipo de escala:	Absolute
Tipo de medida:	X_{max} = Count R_{max} = Count A_i = Count T_i = Time X_i = Count/Time N = Count X_{dev} = Count X = Count/Count
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 144. Métrica externa del comportamiento en el tiempo - plazo de entrega

Nombre:	Turnaround time
Propósito:	What is the wait time the user experiences after issuing an instruction to start a group of related tasks and their completion?
Método de aplicación:	Calibrate the test accordingly. Start the job task. Measure the time it takes for the job task to complete its operation. Keep a record of each attempt.
Medición, fórmula:	T = time between user's finishing getting output results and user's finishing request.
Interpretación:	$0 < T$ The shorter the better
Tipo de escala:	Ratio
Tipo de medida:	T = time
Fuente de medición:	Testing report Operation report showing elapse time

ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 145. Métrica externa del comportamiento en el tiempo - plazo de entrega (tiempo medio para la respuesta)

Nombre:	Turnaround time (Mean time for turnaround)
Propósito:	What is the average wait time the user experiences after issuing an instruction to start a group of related tasks and their completion within a specified system load in terms of concurrent tasks and system utilization?
Método de aplicación:	Calibrate the test. Emulate a condition where a load is placed on the system by executing a number of concurrent tasks (sampled shots) Measure the time it takes to complete the selected job task in the given traffic. Keep a record of reach attempt.
Medición, fórmula:	$X = T_{mean}/TX_{mean}$ $T_{mean} = \sum(T_i)/N$, (for $i = 1$ to N) TX_{mean} = required mean turnaround time T_i = Turnaround time for i -th evaluation (shot) N = Number of evaluations (sampled shots)
Interpretación:	$0 < X$ The shorter is the better
Tipo de escala:	Absolute
Tipo de medida:	T_{mean} = Time TX_{mean} = Time T_i = Time N = Count X = Time/Time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User Developer

	Maintainer SQA
--	-------------------

Tabla 146. Métrica externa del comportamiento en el tiempo - plazo de entrega (proporción del peor caso del tiempo de respuesta)

Nombre:	Turnaround time (Worst case turnaround time ratio)
Propósito:	What is the absolute limit on time required in fulfilling a job task? In the worst case, how long does it take for software system to perform specified tasks?
Método de aplicación:	Calibrate the test. Emulate a condition where by the systems reaches maximum load in terms of tasks performed. Run the selected job task and monitor result(s).
Medición, fórmula:	$X = T_{max}/T_{Xmax}$ $T_{max} = \text{MAX}(T_i)$ (for $i = 1$ to N) R_{max} = Required maximum turnaround time $\text{MAX}(T_i)$ = Maximum turnaround time among evaluations. N = Number of evaluations (sampled shots) T_i = Turnaround time for i -th evaluation (shot)
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X_i = Time/Time T_{max} = Time R_{max} = Time T_i = Time N = Count T_{dev} = Time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 147. Métrica externa del comportamiento en el tiempo - tiempo de espera

Nombre:	Waiting time
Propósito:	What proportion of the time do users spend waiting for the system to respond?
Método de aplicación:	Execute a number of scenarios of concurrent tasks. Measure the time it takes to complete the selected operation(s). Keep a record of each attempt and compute the mean time for each scenario.
Medición, fórmula:	$X = Ta / Tb$ Ta = Total time spent waiting Tb = Task time
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	Ta = Time Tb = Time X = Time/Time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Sys./Sw. Integration Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

• UTILIZACIÓN DE RECURSOS

Tabla 148. Métrica externa del comportamiento en el tiempo - utilización de dispositivos de entrada-salida

Nombre:	I/O devices utilization
Propósito:	Is the I/O device utilization too high, causing inefficiencies?
Método de aplicación:	Execute concurrently a large number of tasks, record I/O device utilization, and compare with the design objectives.
Medición, fórmula:	$X = A/B$ A = Time of I/O devices occupied

	B = Specified time which is designed to occupy I/O devices
Interpretación:	$0 \leq X \leq 1$ The less than and nearer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = Time B = Time X = Time/Time
Fuente de medición:	Testing report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer SQA

Tabla 149. Métrica externa del comportamiento en el tiempo - límites de carga de entrada y salida

Nombre:	I/O loading limits
Propósito:	What is the absolute limit on I/O utilization in fulfilling a function?
Método de aplicación:	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run application and monitor result(s).
Medición, fórmula:	$X = A_{max} / R_{max}$ $A_{max} = \text{MAX}(A_i)$, (for $I = 1$ to N) R_{max} = required maximum I/O messages $\text{MAX}(A_i)$ = Maximum number of I/O messages from 1 st to i-th evaluation N = Number of evaluations.
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	A_{max} = count R_{max} = count A_i = Count N = Count X = count/count
Fuente de medición:	Testing report Operation report showing elapse time

ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 150. Métrica externa del comportamiento en el tiempo - errores relacionados de entrada y salida

Nombre:	I/O related errors
Propósito:	How often does the user encounter problems in I/O device related operations?
Método de aplicación:	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum I/O load. Run the application and record number of errors due to I/O failure and warnings.
Medición, fórmula:	$X = A/T$ A = Number of warning messages or system failures T = User operating time during user observation
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Ratio
Tipo de medida:	A = count T = time X = count/time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Maintainer SQA

Tabla 151. Métrica externa del comportamiento en el tiempo - proporción media del cumplimiento de entrada y salida

Nombre:	Mean I/O fulfillment ratio
Propósito:	What is the average number of I/O related error messages and failures over a specified length of time and specified utilization?
Método de aplicación:	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and record number of errors due to I/O failure and warnings.
Medición, fórmula:	$X = A_{mean} / R_{mean}$ $A_{mean} = \sum(A_i) / N$ R_{mean} = Required mean number of I/O messages A_i = Number of I/O error messages for i-th evaluation N = Number of evaluations
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	A_{mean} = count R_{mean} = count A_i = count N = count X = count/count
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 152. Métrica externa del comportamiento en el tiempo - tiempo de espera del usuario en la utilización de dispositivos de entrada y salida

Nombre:	User waiting time of I/O devices utilization
Propósito:	What is the impact of I/O device utilization on the user wait times?
Método de	Execute concurrently a large amount of tasks and measure the user wait

aplicación:	times as a result of I/O device operation.
Medición, fórmula:	T = Time spent to wait for finish of I/O devices operation.
Interpretación:	$0 < T$ The shorter is the better.
Tipo de escala:	Ratio
Tipo de medida:	T = Time
Fuente de medición:	Testing report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 153. Métrica externa del comportamiento en el tiempo - utilización máxima de la memoria

Nombre:	Maximum memory utilization
Propósito:	What is the absolute limit on memory required in fulfilling a function?
Método de aplicación:	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and monitor result(s).
Medición, fórmula:	$X = A_{max} / R_{max}$ $A_{max} = \text{MAX}(A_i)$, (for $i = 1$ to N) R_{max} = Required maximum memory related error messages $\text{MAX}(A_i)$ = Maximum number of memory related error messages from 1 st to i -th evaluation. N = Number of evaluations.
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	A_{max} = count R_{max} = count A_i = count N = count X = count/count

Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 154. Métrica externa del comportamiento en el tiempo - ocurrencia media del error de memoria

Nombre:	Mean occurrence of memory error
Propósito:	What is the average number of memory related error messages and failures over a specified length of time and a specified load on the system?
Método de aplicación:	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and record number of errors due to memory failure and warnings.
Medición, fórmula:	$X = A_{mean} / R_{mean}$ $A_{mean} = \sum(A_i) / N$ R_{mean} = Required mean number of memory related error messages $MAX(A_i)$ = Maximum number of memory related error messages from 1 st to i-th evaluation. N = Number of evaluations.
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	A_{mean} = count R_{mean} = count A_i = count N = count X = count/count
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User

	Developer Maintainer SQA
--	--------------------------------

Tabla 155. Métrica externa del comportamiento en el tiempo - proporción de error/tiempo de la memoria

Nombre:	Ratio of memory error/time
Propósito:	How many memory errors were experienced over a set period of time and specified resource utilization?
Método de aplicación:	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and record number of errors due to memory failure and warnings.
Medición, fórmula:	$X = A/T$ A = Number of warning messages or system failures T = User operating time during user observation
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Ratio
Tipo de medida:	A = Count T = Time X = count/time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Maintainer SQA

Tabla 156. Métrica externa del comportamiento en el tiempo - utilización máxima de la transmisión

Nombre:	Maximum transmission utilization
Propósito:	What is the absolute limit of transmissions required to fulfill a function?

Método de aplicación:	Evaluate what is required for the system to reach a situation of maximum load. Emulate this condition. Run application and monitor result(s).
Medición, fórmula:	$X = A_{max} / R_{max}$ $A_{max} = \text{MAX}(A_i)$, (for $i = 1$ to N) R_{max} = required maximum number of transmission related error messages and failures $\text{MAX}(A_i)$ = Maximum number of transmission related error messages and failures from 1 st to i -th evaluation N = Number of evaluations
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	A_{max} = count R_{max} = count A_i = count N = count X = count/count
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 157. Métrica externa del comportamiento en el tiempo - equilibrio de la utilización de dispositivos mediáticos

Nombre:	Media device utilization balancing
Propósito:	What is the degree of synchronisation between different media over a set period of time?
Método de aplicación:	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum transmission load. Run the application and record the delay in the processing of different media types.
Medición, fórmula:	$X = \text{SyncTime} / T$ SyncTime = Time devoted to a continuous resource

	T = Required time period during which dissimilar media are expected to finish their tasks with synchronization
Interpretación:	The smaller is the better.
Tipo de escala:	Ratio
Tipo de medida:	SyncTime = Time T = Time X = Time/Time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Maintainer SQA

Tabla 158. Métrica externa del comportamiento en el tiempo - ocurrencia media del error de transmisión

Nombre:	Mean occurrence of transmission error
Propósito:	What is the average number of transmission-related error messages and failures over a specified length of time and specified utilization?
Método de aplicación:	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum transmission load. Run the application and record number of errors due to transmission failure and warnings.
Medición, fórmula:	$X = A_{mean} / R_{mean}$ $A_{mean} = \sum(A_i)/N$ R_{mean} = Required mean number of transmission related error messages and failure A_i = Number of transmission related error messages and failures for i-th evaluation N = Number of evaluations
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Absolute
Tipo de medida:	A_{mean} = count R_{mean} = count A_i = count

	N = count X = count/count
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer SQA

Tabla 159. Métrica externa del comportamiento en el tiempo - media de error de transmisión por tiempo

Nombre:	Mean of transmission error per time
Propósito:	How many transmission-related error messages were experienced over a set period of time and specified resource utilization?
Método de aplicación:	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum transmission load. Run the application and record number of errors due to transmission failure and warnings.
Medición, fórmula:	$X = A/T$ A = Number of warning messages or system failures T = User operating time during user observation
Interpretación:	$0 \leq X$ The smaller is the better.
Tipo de escala:	Ratio
Tipo de medida:	A = Count T = Time X = count/time
Fuente de medición:	Testing report Operation report showing elapse time
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Maintainer SQA

Tabla 160. Métrica externa del comportamiento en el tiempo - utilización de la capacidad de transmisión

Nombre:	Transmission capacity utilization
Propósito:	Is software system capable of performing tasks within expected transmission capacity?
Método de aplicación:	Execute concurrently specified tasks with multiple users, observe transmission capacity and compare specified one.
Medición, fórmula:	$X = A/B$ A = Transmission capacity B = Specified transmission capacity which is designed to be used by the software during execution
Interpretación:	$0 \leq X \leq 1$ The less than and nearer to the 1 is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = size B = size X = size/size
Fuente de medición:	Testing report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer SQA

- **CONFORMIDAD DE EFICIENCIA**

Tabla 161. Métrica externa de la conformidad con la eficiencia - conformidad con la eficiencia

Nombre:	Efficiency compliance
Propósito:	How compliant is the efficiency of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance in the specification.
Medición, fórmula:	$X = 1 - A/B$ (X: Ratio of satisfied compliance items relating to efficiency) A = Number of efficiency compliance items specified that have not been implemented during testing

	B = Total number of efficiency compliance items specified.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Product description (User manual or Specification) of compliance and related standards, conventions, style guides or regulations. Testing specification and report
ISO/IEC 12207 SLCP:	Qualification testing Validation
Audiencia:	Supplier User

MANTENIBILIDAD

• CAPACIDAD DE SER ANALIZADO

Tabla 162. Métrica externa de la capacidad de ser analizado - capacidad de auditar

Nombre:	Audit trail capability
Propósito:	Can user identify specified operation which caused failure? Can maintainer easily find specific operation caused failure?
Método de aplicación:	Observe behaviour of user or maintainer who is trying to resolve failures
Medición, fórmula:	$X = A/B$ A = Number of data actually recorded during operation B = Number of data planned to be recorded enough to monitor status of software during operation
Interpretación:	$0 \leq X$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207	Qualification testing

SLCP:	Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 163. Métrica externa de la capacidad de ser analizado - soporte de diagnóstico de la función

Nombre:	Diagnostic function support
Propósito:	How capable are the diagnostic functions in supporting causal analysis? Can user identify the specific operation which caused failure? (User may be able to avoid falling into the same failure occurrence again with alternative operation) Can maintainer easily find cause of failure?
Método de aplicación:	Observe behaviour of user or maintainer who is trying to resolve failures using diagnostic functions.
Medición, fórmula:	$X = A/B$ A = Number of failures which maintainer can diagnose (using the diagnostics function) to understand the cause-effect relationship B = Total number of registered failures
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count $X = \text{count/count}$
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 164. Métrica externa de la capacidad de ser analizado - capacidad de análisis de falla

Nombre:	Failure analysis capability
Propósito:	Can user identify the specific operation which caused failure? Can maintainer easily find cause of failure?
Método de aplicación:	Observe behaviour of user or maintainer who is trying to resolve failures
Medición, fórmula:	$X = 1 - A/B$ A = Number of failures of which causes are still not found B = Total number of registered failures
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer Operator

Tabla 165. Métrica externa de la capacidad de ser analizado - eficacia del análisis de la falla

Nombre:	Failure analysis efficiency
Propósito:	Can user efficiently analyze cause of failure? (User sometimes performs maintenance by setting parameters) Can maintainer easily find cause of failure?
Método de aplicación:	Observe behaviour of user or maintainer who is trying to resolve failures
Medición, fórmula:	$X = \text{Sum}(T)/N$
Interpretación:	$0 \leq X$ The shorter is the better.
Tipo de escala:	Ratio
Tipo de medida:	T = Time
Fuente de medición:	Problem resolution report Operation report

ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 166. Métrica externa de la capacidad de ser analizado – capacidad de monitorear el estado

Nombre:	Status monitoring capability
Propósito:	Can user identify the specific operation which caused failure by getting monitored data during operation? Can maintainer easily find cause of failure by getting monitored data during operation?
Método de aplicación:	Observe behaviour of user or maintainer who is trying to get monitored data recording status of software during operation.
Medición, fórmula:	$X = 1 - A/B$ A = Number of cases which maintainer (or user) failed to get monitor data B = Number of cases which maintainer (or user) attempted to get monitor data recording status of software during operation.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Developer Maintainer Operator

- **FACILIDAD DE CAMBIO**

Tabla 167. Métrica externa de la facilidad de cambio - cambiar la eficacia del ciclo

Nombre:	Change cycle efficiency
Propósito:	Can the user's problem be solved to his satisfaction within an acceptable time scale?
Método de aplicación:	Monitor interaction between user and supplier. Record the time taken from the initial user's request to the resolution of problem.
Medición, fórmula:	Average Time: $T_{av} = \text{Sum}(T_u)/N$ $T_u = T_{rc} - T_{sn}$ T_{sn} = Time at which user finished to send request for maintenance to supplier with problem report. T_{rc} = Time at which user received the revised version release (or status report) N = Number of revised versions
Interpretación:	$0 < T_{av}$ The shorter is the better, except of the number of revised versions was large.
Tipo de escala:	Ratio
Tipo de medida:	T_u = Time T_{rc}, T_{sn} = Time N = Count T_{av} = Time
Fuente de medición:	Problem resolution report Maintenance report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	User Maintainer Operator

Tabla 168. Métrica externa de la facilidad de cambio - cambiar el tiempo que transcurre en la implementación

Nombre:	Change implementation elapse time
Propósito:	Can the maintainer easily change the software to resolve the failure problem?
Método de aplicación:	Observe the behaviour of the user and maintainer while trying to change software.

	Otherwise, investigate problem resolution report or maintenance report.
Medición, fórmula:	Average Time: $T_{av} = \text{Sum}(T_m)/N$ $T_m = T_{out} - T_{in}$ T_{out} = Time at which the causes of failure are removed within changing the software (or status is reported back to user) T_{in} = Time at which the causes of failures are found out N = Number of registered and removed failures
Interpretación:	$0 \leq T_{av}$ The shorter is the better, except of the number of failures was large.
Tipo de escala:	Ratio
Tipo de medida:	T = Time T_{in}, T_{out} = Time T_{av} = Time
Fuente de medición:	Problem resolution report Maintenance report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 169. Métrica externa de la facilidad de cambio - complejidad de la modificación

Nombre:	Modification complexity
Propósito:	Can the user maintainer easily change the software to resolve problem?
Método de aplicación:	Observe behaviour of maintainer who is trying to change the software. Otherwise, investigate problem resolution report or maintenance report and product description.
Medición, fórmula:	$T = \text{Sum}(A/B)/N$ A = Work time spent to change B = Size of software change N = Number of changes
Interpretación:	$0 < T$ The shorter is the better or the required number of changes were excessive
Tipo de escala:	Ratio
Tipo de medida:	A = Time B = Size

	N = Count T = Time
Fuente de medición:	Problem resolution report Maintenance report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 170. Métrica externa de la facilidad de cambio - modificabilidad parametrizada

Nombre:	Parametrised modifiability
Propósito:	Can the user or the maintainer easily change parameter to change software and resolve problems?
Método de aplicación:	Observe behaviour of the user or the maintainer while trying to change the software. Otherwise, investigate problem resolution report or maintenance report.
Medición, fórmula:	$X = A/B$ A = Number of change log data actually recorded B = Number of cases which maintainer attempts to change software by using parameter.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count $X = \text{count}/\text{count}$
Fuente de medición:	Problem resolution report Maintenance report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operador User

Tabla 171. Métrica externa de la facilidad de cambio - capacidad de control de cambio en el software

Nombre:	Software change control capability
Propósito:	Can the user easily identify revised versions? Can the maintainer easily change the software to resolve problems?
Método de aplicación:	Observe the behaviour of user or maintainer while trying to change the software. Otherwise, investigate problem resolution report or maintenance report
Medición, fórmula:	$X = A/B$ A = Number of change log data actually recorded B = Number of change log data planned to be recorded enough to trace software changes.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better or the closer to 0 the fewer changes have taken place.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	User manual or specification Problem resolution report Maintenance report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

- ESTABILIDAD**

Tabla 172. Métrica externa de la estabilidad - cambio en la proporción del éxito

Nombre:	Change success ratio
Propósito:	Can user operate software system without failures after maintenance? Can maintainer easily mitigate failures caused by maintenance side effects?
Método de aplicación:	Observe behaviour of user or maintainer who is operating software system after maintenance. Count failures which user or maintainer encountered during operating

	software before and after maintenance. Otherwise, investigate problem resolution report or maintenance report
Medición, fórmula:	$X = Na/Ta$ $Y = \{(Na/Ta) / (Nb/Tb)\}$ Na = Number of cases which user encounters failures during operation after software was changed. Nb = Number of cases which user encounters failures during operation before software is changed Ta = Operation time during specified observation period after software is changed Tb = Operation time during specified observation period before software is changed.
Interpretación:	$0 \leq X, Y$ The smaller and closer to 0, is the better.
Tipo de escala:	Ratio
Tipo de medida:	Na, Nb = Count Ta, Tb = Time $X = \text{Count/Time}$ $Y = [(\text{Count/Time}) / (\text{Count/Time})]$
Fuente de medición:	Problem resolution report Maintenance report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 173. Métrica externa de la estabilidad - localización del impacto de la modificación (fallo que emerge después del cambio)

Nombre:	Modification impact localization (Emerging failure after change)
Propósito:	Can user operate software system without failures after maintenance? Can maintainer easily mitigate failures caused by maintenance side effects?
Método de aplicación:	Count failures occurrences after change, which are mutually chaining and affected by change.
Medición, fórmula:	$X = A/N$ A = Number of failures emerged after failure is resolved by change during specified period N = Number of resolved failures

Interpretación:	$0 \leq X$ The smaller and closer to 0, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count N = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

- FACILIDAD DE PRUEBA**

Tabla 174. Métrica externa de la facilidad de prueba - disponibilidad de la función de la prueba incorporada

Nombre:	Availability of built-in test function
Propósito:	Can user and maintainer easily perform operational testing without additional test facility preparation?
Método de aplicación:	Observe behaviour of user or maintainer who is testing software system after maintenance.
Medición, fórmula:	$X = A/B$ A = Number of cases in which maintainer can use suitably built-in test function B = Number of cases of test opportunities
Interpretación:	$0 \leq X \leq 1$ The larger and the closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance

Audiencia:	Developer Maintainer Operator
-------------------	-------------------------------------

Tabla 175. Métrica externa de la facilidad de prueba - Realizar varias pruebas de eficiencia

Nombre:	Re-test efficiency
Propósito:	Can user and maintainer easily perform operational testing and determine whether the software is ready for operation or not?
Método de aplicación:	Observe behaviour of user or maintainer who is testing software system after maintenance.
Medición, fórmula:	$X = \text{Sum}(T) / N$ T = Time spent to test to make sure whether reported failure was resolved or not N = Number of resolved failures
Interpretación:	$0 < X$ The smaller is the better.
Tipo de escala:	Ratio
Tipo de medida:	T = Time N = Count X = time/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 176. Métrica externa de la facilidad de prueba - capacidad para recomenzar las pruebas

Nombre:	Test restartability
Propósito:	Can user and maintainer easily perform operational testing with check points after maintenance?
Método de aplicación:	Observe behaviour of user or maintainer who is testing software system after maintenance.
Medición, fórmula:	$X = A/B$

	A = Number of cases in which maintainer can pause and restart executing test run at desired points to check step by step B = Number of cases of pause of executing test run
Interpretación:	$0 \leq X \leq 1$ The larger and the closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

- **CONFORMIDAD DE FACILIDAD DE MANTENIMIENTO**

Tabla 177. Métrica externa de la conformidad con la facilidad de mantenimiento - conformidad con la la capacidad de mantenimiento

Nombre:	Maintainability compliance
Propósito:	How compliant is the maintainability of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.
Medición, fórmula:	$X = 1 - A/B$ A = Number of maintainability compliance items specified that have not been implemented during testing B = Total number of maintainability compliance items specified.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de	Product description (User manual or Specification) of compliance and related

medición:	standards, conventions, style guides or regulations. Testing specification and report
ISO/IEC 12207 SLCP:	Qualification testing Validation
Audiencia:	Supplier User

PORTABILIDAD

• ADAPTABILIDAD

Tabla 178. Métrica externa de la adaptabilidad - adaptabilidad de las estructuras de datos

Nombre:	Adaptability of data structures
Propósito:	Can user or maintainer easily adapt software to data sets in new environment?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to adapt software to operation environment.
Medición, fórmula:	$X = A/B$ A = The number of data which are operable and but are not observed due to incomplete operations caused by adaptation limitations B = The number of data which are expected to be operable in the environment to which the software is adapted.
Interpretación:	$0 \leq X \leq 1$ The larger and closer to 1, the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 179. Métrica externa de la adaptabilidad - adaptabilidad del ambiente de hardware

Nombre:	Hardware environmental adaptability
Propósito:	Can user or maintainer easily adapt software to environment? Is software system capable enough to adapt itself to operation environment?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to adapt software to operation environment.
Medición, fórmula:	$X = 1 - A/B$ A = Number of operational functions of which tasks were not completed or not enough resulted to meet adequate levels during combined operating testing with environmental hardware B = Total number of functions which were tested.
Interpretación:	$0 \leq X \leq 1$ The larger is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 180. Métrica externa de la adaptabilidad - adaptabilidad al ambiente organizacional

Nombre:	Organizational environment adaptability
Propósito:	Can user or maintainer easily adapt software to environment? Is software system capable enough to adapt itself to operation environment?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to adapt software to operation environment.
Medición, fórmula:	$X = 1 - A/B$ A = Number of operated functions in which the tasks were not completed or not enough resulted to meet adequate levels during operational testing with user's business environment. B = Total number of functions which were tested.
Interpretación:	$0 \leq X \leq 1$

	The larger is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 181. Métrica externa de la adaptabilidad - amigabilidad del usuario

Nombre:	Porting user friendliness
Propósito:	Can user or maintainer easily adapt software to environment?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to adapt software to operational environment.
Medición, fórmula:	T = Sum of user operating time spent to complete adaptation of the software to user's environment, when user attempt to install or change setup.
Interpretación:	$0 < T$ The shorter is the better.
Tipo de escala:	Ratio
Tipo de medida:	T= time
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 182. Métrica externa de la adaptabilidad - adaptabilidad del ambiente de software del sistema

Nombre:	System software environmental adaptability
Propósito:	Can user or maintainer easily adapt software to environment?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to adapt software to operation environment.
Medición, fórmula:	$X = 1 - A/B$ A = Number of operational functions of which tasks were not completed or were not enough resulted to meet adequate level during combined operating testing with operating system software or concurrent application software. B = Total number of functions which were tested.
Interpretación:	$0 \leq X \leq 1$ The larger is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

• FACILIDAD DE INSTALACION

Tabla 183. Métrica externa de la facilidad de instalación - facilidad de instalación

Nombre:	Ease of installation
Propósito:	Can user or maintainer easily install software to operation environment?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to install software to operation environment.
Medición, fórmula:	$X = A/B$ A = Number of cases which a user succeeded to in changing the install operation for his/her convenience B = Total number of cases which a user attempted to change the install operation for his/her convenience.

Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	A = count B = count X = count/count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 184. Métrica externa de la facilidad de instalación - facilidad de reintentar el setup

Nombre:	Ease of Setup Re-try
Propósito:	Can user or maintainer easily retry setup installation of software?
Método de aplicación:	Observe user's or maintainer's behaviour when user is trying to retry setup installation of software.
Medición, fórmula:	$X = 1 - A/B$ A = Number of cases in which user fails in retrying setup during setup operation. B = Total number of cases in which user attempt to retry setup during setup operation
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

- **COEXISTENCIA**

Tabla 185. Métrica externa de coexistencia - coexistencia disponible

Nombre:	Available co-existence
Propósito:	How often user encounters any constraints or unexpected failures when operating concurrently with other software?
Método de aplicación:	Use evaluated software concurrently with other software which user often uses.
Medición, fórmula:	$X = A/T$ A = Number of any constraints or unexpected failures which user encounter during operating concurrently with other software. T = Time duration of concurrently operating other software.
Interpretación:	$0 \leq X$ The closer to 0, is the better
Tipo de escala:	Ratio
Tipo de medida:	A = Count B = Time $X = \text{count/time}$
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer SQA Operator

- **REEMPLAZABILIDAD**

Tabla 186. Métrica externa de reemplazabilidad - uso continuo de datos

Nombre:	Continued use of data
Propósito:	Can user or maintainer easily continue to use the same data after replacing this software to previous one? Is software system migration going on successfully?
Método de aplicación:	Observe user's or maintainer's behaviour when user is replacing software to previous one.
Medición, fórmula:	$X = A/B$

	A = Number of data which are used in other software to be replaced and are confirmed that they are able to be continuously used B = Number of data which are used in other software to be replaced and planned to be continuously reusable.
Interpretación:	$0 \leq X \leq 1$ The larger is the better
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Problem resolution report Operation report
ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 187. Métrica externa de reemplazabilidad - inclusividad de la función

Nombre:	Function inclusiveness
Propósito:	Can user or maintainer easily continue to use similar functions after replacing this software to previous one? Is software system migration going on successfully?
Método de aplicación:	Observe user's or maintainer's behaviour when user is replacing software to previous one.
Medición, fórmula:	$X = A/B$ A = Number of functions which produce similar results as previously produced and where changes have not been required. B = Number of tested functions which are similar to functions provided by another software to be replaced.
Interpretación:	$0 \leq X \leq 1$ The larger is the better
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Problem resolution report Operation report

ISO/IEC 12207 SLCP:	Qualification testing Operation Maintenance
Audiencia:	Developer Maintainer Operator

Tabla 188. Métrica externa de reemplazabilidad - consistencia funcional al soporte del usuario

Nombre:	User support functional consistency
Propósito:	How consistent are the new components with existing user interface?
Método de aplicación:	Observe the behaviour of the user and ask the opinion.
Medición, fórmula:	$X = 1 - A1/A2$ A1 = Number of new functions which user found unacceptably inconsistent with the user's expectation A2 = Number of new functions
Interpretación:	$0 \leq X$ Larger is better.
Tipo de escala:	Absolute
Tipo de medida:	A1 = count A2 = count $X = \text{count}/\text{count}$
Fuente de medición:	Test report Operation report
ISO/IEC 12207 SLCP:	Integration Qualification testing Operation Quality Assurance
Audiencia:	User User interface designer Maintainer Developer Tester SQA

- **CONFORMIDAD DE LA PORTABILIDAD**

Tabla 189. Métrica externa de la conformidad con la portabilidad - conformidad con la portabilidad

Nombre:	Portability compliance
Propósito:	How compliant is the portability of the product to applicable regulations, standards and conventions?
Método de aplicación:	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.
Medición, fórmula:	$X = 1 - A/B$ A = Number of portability compliance items specified that have not been implemented during testing B = Total number of portability compliance items specified.
Interpretación:	$0 \leq X \leq 1$ The closer to 1, is the better.
Tipo de escala:	Absolute
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Product description (User manual or Specification) of compliance and related standards, conventions or regulations. Testing specification and report
ISO/IEC 12207 SLCP:	Qualification testing Validation
Audiencia:	Supplier User

ANEXO D

Explicación Detallada de los tipos de escala de las métricas y de los tipos de medidas

Tipos de escala de las métricas

Una de las siguientes tipos de medición de escala de las métricas debería ser identificada para cada medida, cuando un usuario de métricas tiene el resultado de una medición y utiliza la medida para el cálculo y la comparación. El promedio, la proporción o la diferencia de valores pueden no tener un significado para algunas medidas. Los tipos de escala de métricas son: escala nominal, escala ordinal, escala de intervalo, escala proporción, y escala absoluta.

a) Escala Nominal

Esta clasificación incluye, por ejemplo, tipos de falla de software (de datos, de control y otros).

Ejemplos: Número de identificación de la línea de transporte de una ciudad, número de mensajes de error de identificación de un compilador.

b) Escala Ordinal

Esto incluye pedidos, por ejemplo, fallos del software por gravedad (insignificante, marginal, crítico, catastrófico).

Ejemplos: resultados del examen del colegio (excelente, bueno, aceptable, no aceptable)

c) Escala de intervalo

Esto incluye escalas de calificación ordenadas donde la diferencia entre dos medidas tiene un significado empírico. Sin embargo, la proporción de dos medidas en una escala de intervalo no tiene el mismo significado empírico.

Ejemplo: Temperatura (Celsius, Fahrenheit, Kelvin) diferencia entre el tiempo de computación real y el tiempo previsto.

d) Escala promedio

Este incluye escalas de calificación ordenadas, donde la diferencia entre dos medidas tiene el mismo significado empírico. Un promedio y una proporción tienen significados respectivamente y ellos dan significado a los valores.

Ejemplos: longitud, peso, tiempo, tamaño, contador

e) Escala absoluta

Cualquier declaración relativa a las medidas es significativa. Por ejemplo el resultado de dividir una medida de tipo de escala de proporción por otra medida de tipo de escala donde la unidad de medición es la misma, es absoluta. Una medición de tipo de escala absoluta es de hecho, una sin ningún tipo de unidad.

Ejemplo: número de líneas de código con comentarios dividido por el total de líneas de código.

Tipos de medida

General

Con el fin de diseñar un procedimiento para la recolección de datos, interpretación de significados, y la normalización de las medidas para la comparación, un usuario de métricas debería identificar y tomar en cuenta el tipo de medida de medición empleada por una métrica.

1. Tipo de tamaño de medida

General

Una medida de este tipo representa un tamaño particular del software de acuerdo a lo que se afirma para ser medido dentro de su definición.

Nota: el software puede tener muchas representaciones de tamaño (como cualquier entidad puede ser medida en más de una dimensión – masa, volumen, área de la superficie, etc.)

La normalización de otras medidas con una medida de tamaño puede dar valores comparables en términos de unidades de tamaño. Las medidas de tamaño descritas a continuación pueden ser usadas para la medición de la calidad del software.

- **Tipo de tamaño funcional**

El tamaño funcional es un ejemplo de un tipo de tamaño (una dimensión) que el software puede tener. Cualquier instancia del software puede tener más de un tamaño funcional dependiendo de, por ejemplo:

- a) El propósito para medir el tamaño del software (no influye en el ámbito de aplicación del software incluido en la medición);
- b) El método particular usado (esto cambiará las unidades y la escala)

- **Tipo de tamaño del programa**

En esta cláusula, el término “programación” representa las expresiones de cuando son ejecutados resultados en acciones, y el término “lenguaje” representa el tipo de expresión usada.

- ❖ **Tamaño del programa fuente**

El lenguaje de programación debería ser explicada y debería ser proveída como las declaraciones no ejecutables, así como las líneas comentadas, son tratadas.

- ❖ **Tamaño de las palabras del programa**

La medición puede ser computada de la siguiente manera usando las medidas de Halstead:

Vocabulario del programa = $n_1 + n_2$; Longitud del programa observado = $N_1 + N_2$, donde:

n_1 : es el número de palabras de distintos operadores las cuales son preparadas y reservadas por el lenguaje del programa en el código fuente de un programa.

n2: es el número de palabras de distintos operandos las cuales son definidos por el programador en el código fuente de un programa.

N1: es el número de ocurrencias de distintos operadores en el código fuente de un programa.

N2: es el número de ocurrencias de distintos operandos en el código fuente de un programa.

❖ **Número de módulos**

La medición es contar el número de objetos independientemente ejecutables tales como los módulos de un programa.

- **Tipo de medida de los recursos utilizados**

Este tipo identifica los recursos utilizados por la operación del software siendo evaluado. Los ejemplos son:

- a) *Cantidad de memoria*, por ejemplo, cantidad de disco o memoria ocupada temporalmente o permanentemente durante la ejecución del software.
- b) *Carga de la entrada y la salida*, por ejemplo, cantidad de tráfico en la comunicación de los datos (*Significativo de las herramientas de copia de seguridad en una red*)
- c) *Carga de la CPU*, por ejemplo, porcentaje de CPU ocupada del conjunto de instrucciones por segundo (este tipo de medida es significativa para medir el uso de la CPU y la eficiencia del proceso de distribución en múltiples hilos de software corriendo sobre sistemas concurrentes o paralelos).

d) *Archivos y datos registrados*, por ejemplo, longitud en bytes de archivos o registros

e) *Documentos*, por ejemplo, número de páginas del documento.

- **Tipo de paso de procedimiento operativo especificado**

Este tipo identifica pasos estáticos de procedimientos los cuales son descritos en una especificación de diseño de la interfaz o en un manual de usuario.

El valor medido puede variar dependiendo de que tipos de descripción sean usados para la medición, tales como un diagrama o un texto en representación de los procedimientos operativos de usuario.

2. Tipo de medida de tiempo

General

El usuario de métricas del tipo de medida de tiempo debería registrar los periodos de tiempo, cuántos sitios fueron examinados y cuántos usuarios participaron en las mediciones.

Hay muchas formas en las cuales el tiempo puede ser medido como una unidad, como se muestra en los siguientes ejemplos.

a) Unidad de tiempo real

Este es un tiempo físico: es decir, segundo, minuto u hora. Esta unidad es usualmente usada para describir tareas de tiempo de procesamiento de un software de tiempo real.

b) Unidad de tiempo de la maquinaria del computador

Este es la tiempo del reloj del procesador del computador: es decir, segundo, minuto, u hora del tiempo de la CPU.

c) Unidad del tiempo oficial previsto

Este incluye horas de trabajo, días de calendario, meses o años.

d) Unidad de tiempo del componente.

Cuando hay múltiples sitios, el tiempo del componente identifica a cada sitio y esto es una acumulación de tiempo individual de cada sitio. Esta unidad es usualmente usada para describir la fiabilidad del componente, por ejemplo, la tasa de fallos del componente.

e) Unidad de tiempo del sistema

Cuando hay múltiples sitios, el tiempo del sistema no identifica a cada sitio pero identifica todos los sitios que corren, en su conjunto en un solo sistema. Esta unidad es usualmente usada para describir la fiabilidad del sistema, por ejemplo, la tasa de fallos del sistema.

- **Tipo de tiempo de operación del sistema**

El tipo de tiempo de operación del sistema provee unas bases para medir la disponibilidad del software. Esto es principalmente usado para la evaluación de la fiabilidad. Debe identificarse si el software está en funcionamiento discontinuo o funcionamiento continuo. Si el software opera discontinuamente, Debe tener la

seguridad de que la medición del tiempo esta hecha sobre los periodos donde el software esta activo (esto es obviamente extendido a operaciones continuas).

a) Tiempo transcurrido

Cuando el uso del software es constante, por ejemplo en sistemas de operación para el mismo período de tiempo cada semana.

b) Tiempo de potencia de la máquina

Para tiempo real, el software del sistema operativo o embebido que se encuentra en pleno uso todo el tiempo, el sistema está funcionando.

c) Tiempo en que la máquina está normalizada

Como en el “tiempo de potencia de la máquina”, pero combinando datos de varias máquinas de diferentes “potencias de máquina” y aplicando un factor de corrección.

- **Tipo de tiempo de ejecución**

El tipo de tiempo de ejecución es el tiempo en el cuál es necesario ejecutar el software para completar una tarea específica. La ejecución bajo condiciones específicas, particularmente una condición sobrecargada, debería ser examinada. El tipo de tiempo de ejecución es principalmente usado para la evaluación de la eficiencia.

- **Tipo de tiempo del usuario**

El tipo de tiempo del usuario es medido bajo periodos de tiempo gastado por usuarios en la finalización de las tareas mediante el uso de las operaciones del software. Algunos ejemplos son:

a) Tiempo de la sesión

Medido entre el comienzo y el fin de una sesión. Son útiles, por ejemplo, para señalar el comportamiento de los usuarios de un sistema bancario.

b) Tiempo de tarea

Tiempo gastado por un usuario para realizar una tarea mediante el uso de las operaciones del software en cada intento. El comienzo y el fin de los puntos de medición deberían ser bien definidos.

c) Tiempo del usuario

Tiempo gastado por un usuario usando el software desde el tiempo de comienzo hasta un punto en el tiempo. (Aproximadamente, esto es cuantas horas o días el usuario usa el software desde el comienzo).

- **Tipo de esfuerzo**

Tipo de esfuerzo es el tiempo productivo asociado con una tarea específica de un proyecto.

a) Esfuerzo individual

Esto es el tiempo productivo el cual es necesitado por una persona quien es desarrollador, mantenedor u operador para trabajar y completar una tarea específica.

b) Esfuerzo en la tarea

Esfuerzo en la tarea es un valor acumulado de todas las personas del proyecto: desarrolladores, mantenedores, operadores, usuarios u otros quienes trabajaron para completar la tarea específica.

- **Tipo de intervalos de tiempo de los acontecimientos**

Este tipo de medida es el tiempo entre un evento y el siguiente durante un periodo de observación. La frecuencia de un periodo de tiempo de observación puede ser usada en lugar de esta medida. Esto es típicamente usado para describir el tiempo entre fallas que ocurren sucesivamente.

3. Tipo de medida del contador

- **Tipo de número de fallos detectados**

Esta medición cuenta los fallos detectados durante la revisión, las pruebas, las correcciones, la operación o el mantenimiento.

- **Tipo de número de complejidad del programa estructural**

Esta medición cuenta la complejidad del programa estructural. Un Ejemplo es el número de rutas distintas.

- **Tipo de número de inconsistencias detectadas**

Esta medida cuenta las inconsistencias detectadas las cuales son preparadas para la investigación.

- **Tipo del número de cambios**

Este tipo identifica los temas de la configuración del software los cuales se detectó que se ha cambiado. Un ejemplo es el número de líneas cambiadas en el código fuente.

- **Tipo del número de fallos detectados**

La medición cuenta el número de fallos detectados durante el producto desarrollado, pruebas, operación o mantenimiento.

- **Tipo del número de intentos (ensayos)**

Esta medida cuenta el número de intentos a corregir el defecto o la falla. Por ejemplo, durante revisiones, pruebas y mantenimiento.

- **Tipo de puntaje**

Este tipo identifica el puntaje o resultado de un cálculo aritmético. El puntaje puede incluir la cuenta o el cálculo de los todos los pesos comprobados en las listas de chequeo. Ejemplos: puntajes de las listas de chequeo, puntaje de los cuestionarios.

ANEXO E

TÉRMINOS Y DEFINICIONES

Adquisidor/Adquirente

Organización que adquiere o procura un sistema, producto de software o servicio de software por parte de un proveedor.

Atributo

Propiedad inherente o característica de una entidad que puede ser distinguida cuantitativamente o cualitativamente por el ser humano o medio automatizado.

Atributo para medir la calidad

Atributo que relaciona al producto de software en sí mismo, al uso del producto de software o a su proceso de desarrollo.

Calidad

Todas las características de una entidad que tiene la habilidad de satisfacer necesidades implícitas y específicas.

Calidad del software

La capacidad del producto de software para satisfacer estados y necesidades implicadas cuando es usado bajo condiciones específicas.

Calidad en uso (medida)

El grado para el cual un producto usado por usuarios específicos resuelve sus necesidades para alcanzar metas específicas con efectividad, productividad, seguridad y satisfacción en contextos específicos de uso.

Calidad en uso del software

Capacidad del producto de software para permitir a usuarios específicos alcanzar metas específicas con efectividad, productividad, seguridad y satisfacción en contextos específicos de uso.

Calidad externa del software

La capacidad de un producto de software que le permite al comportamiento de un sistema satisfacer estados y necesidades implicadas cuando el sistema se usa bajo condiciones específicas.

Calidad interna del software

Capacidad de un conjunto de atributos estáticos de un producto de software para satisfacer las necesidades indicadas e implicadas cuando el sistema es usado bajo condiciones específicas.

Características de la calidad del software

Categoría de los atributos de la calidad del software que se soporta sobre la calidad del software.

Desarrollador

Individuo u organización que realiza el desarrollo de actividades (incluyendo análisis de requisitos, diseño, pruebas de aceptación) durante el proceso del ciclo de vida del software.

Evaluador

Individuo u organización que realiza una evaluación.

Evaluación de un producto de software

Operación técnica que consiste en producir un gravamen de una o más características de un producto de software, de acuerdo con un procedimiento específico.

Falla

Paso, proceso o definición de dato incorrecto en un programa de computador.

IEC (International Electrotechnical Commission)

La Comisión Electrotécnica Internacional (CEI o IEC, por sus siglas del idioma inglés International Electrotechnical Commission) es una organización de normalización en los campos eléctrico, electrónico y tecnologías relacionadas. Numerosas normas se desarrollan conjuntamente con la ISO (normas ISO/IEC).

ISO (International Standard Organization)

La Organización Internacional de Normalización (ISO) es una federación mundial de organismos nacionales de normalización.

La misión de ISO es estimular en todo el mundo el desarrollo de la normalización y las actividades relacionadas de manera de facilitar el intercambio de productos y servicios y alcanzar acuerdos en los campos intelectual, científico, técnico y económico. Su trabajo involucra todos los campos, excepto los de normas de ingeniería eléctrica o electrónica, que están bajo el ámbito de IEC.

ISO/IEC 9126

Estándar que define un modelo de calidad de producto software. El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y métricas de calidad de uso.

ISO/IEC 14598

Estándar que proporciona la dirección y los requisitos para el proceso de la evaluación de un software.

El estándar está dividido en seis partes: visión general, planificación y gestión, procedimiento para desarrolladores, procedimiento para compradores, procedimiento para evaluadores, documentación de los modelos de evaluación.

ISO/IEC 25000:2005 - SQuaRE

SQuaRE (Software Quality Requirements and Evaluation) es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598 (Evaluación del software).

SQuaRE incluye un estándar de requerimientos de calidad. Está compuesto por 14 documentos agrupados en 5 tópicos: (1) Administración de la calidad – 2500n, (2) Modelo de calidad – 2501n, (3) Medidas de calidad – 2502n, (4) Requerimientos de calidad – 2503n y (5) Evaluación de la calidad – 2504n.

Mantenedor

Individuo u organización que realiza actividades de mantenimiento.

Medición

Conjunto de operaciones que tienen el objeto de determinar un valor de una medida.

Medida

Variable para la cual un valor es asignado como el resultado de una medida.

Medida externa

Medida indirecta de un producto, derivada de las medidas que se hacen al comportamiento del sistema del cual hace parte. El sistema incluye hardware, software (tanto software para el consumidor como software “no común”) y usuarios.

Medida interna

Medida del producto mismo, directa o indirecta. El número de líneas de código, medidas complejas y número de fallas encontradas durante todo el proceso, son todas medidas internas hechas al producto mismo.

Método de evaluación

Procedimiento que describe acciones a ser realizadas por el evaluador para obtener resultados para la medida especificada, aplicada para los componentes del producto específico o sobre el producto en su totalidad.

Métrica

Método de medición definido y escala de medición. Las métricas pueden ser internas o externas, directas o indirectas. Las métricas incluyen métodos para categorizar datos cualitativos.

Modelo de Calidad

Define un conjunto de características, y de relaciones entre ellos, las cuales proveen un marco para requisitos de la especificación de la calidad y evaluación de la calidad.

Necesidades implicadas

Necesidades que pueden no haber sido declaradas, pero que son necesidades actuales.

Operador

Individuo u organización que opera el sistema.

Proceso

Sistema de actividades, las cuales usan recursos para transformar entradas en salidas.

Producto de software

Conjunto de programas de computadora, procedimientos, y documentación y datos posiblemente asociados.

Requisitos

Expresión de una necesidad percibida de algo que se logró o realizó.

Sistema

Compuesto integrado que consiste de uno o más de los procesos, hardware, software, facilidades y gente, que proveen una capacidad de satisfacer las necesidades u objetivos estipulados.

Software específico

Producto de software desarrollado para una aplicación específica desde una especificación de requisitos de usuario.

Software genérico

Son sistemas aislados producidos por una organización de desarrollo y que se venden al mercado abierto a cualquier cliente que le sea posible comprarlos. Algunas veces estos se denominan software empaquetados.

Proveedor

Individuo u organización que entra en contacto con el adquisidor para suministrarle un sistema, producto de software o servicio de software bajo los términos de un contrato.

Uso del contexto

Usuarios, tareas, equipos (hardware, software y materiales), y los ambientes físicos y sociales en los cuales un producto es usado.

Usuario

Individuo u organización que usa el sistema para realizar una función específica.

Usuario final

Persona individual quien se beneficia en última instancia de los resultados del sistema.

ANEXO F

MANUAL DE USUARIO

**PLANTILLA PARA EVALUAR LA CALIDAD DE UN PRODUCTO
SOFTWARE**

Manual del Usuario

Plantilla para evaluar la Calidad de un Producto Software

“Según el modelo de calidad de un producto software planteado en la norma SQuaRE y utilizando las métricas externas e internas de la norma ISO/IEC 9126”

Versión 1.0

Sebastián Piedrahita Mesa
Octubre de 2007

INTRODUCCIÓN

La herramienta usada en este desarrollo fue una plantilla creada en Microsoft Excel, lo cual permite una fácil manipulación por parte de los usuarios finales.

La plantilla es un desarrollo sencillo creado con el fin de poder realizar la evaluación de un producto de software genérico y/o específicos, ya que permite medir las características que contiene un desarrollo de software cualquiera.

Cuando el desarrollo de software es genérico, la plantilla puede utilizarse para establecer el nivel de calidad que se pretende alcanzar con ese producto. Igualmente si el desarrollo es un producto hecho a la medida (específico), es posible establecer los niveles de calidad que se pretenden tener cada vez que se van creando las etapas del desarrollo.

En ella, se definen cada una de las características y sub-características a ser evaluadas, se presenta un resumen detallado de los resultados obtenidos, se expresan esos resultados de manera gráfica y se incluye la descripción de cada una de las métricas disponibles, con el fin de facilitar el uso de la plantilla.

Las características y sub-características utilizadas en esta plantilla, son las que presenta la norma ISO/IEC CD 25010 (SQuaRE) en su modelo de calidad.

Estas características son: funcionalidad, seguridad, interoperabilidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

Nota: En esta plantilla no se usaron todas las métricas propuestas en la norma ISO/IEC 9126:2003 pues algunas de ellas son muy difíciles de calcular y no se considero necesario su inclusión. El total de métricas que se usaron fueron 103.

OBJETIVO GENERAL

Explicar a los nuevos usuarios de la plantilla, cómo utilizar la herramienta y hacer una descripción de las hojas donde se encuentra la información.

INGRESO DE DATOS

El ingreso de los datos por parte del usuario debe hacerse en las celdas que están pintadas únicamente con color gris claro. Una vez que el usuario introduzca un valor dentro de una celda, éste se establece de color negro. Los datos que vienen de una hoja previa o que se van actualizando, se presentan en color rojo.

Para que el usuario recuerde el rango de valores a ingresar, se muestra en cada celda una nota como la que se expone a continuación:

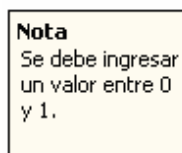


Figura15 – Nota de información para el ingreso de datos

Las hojas en las que se hace necesario el ingreso de datos por parte del usuario y el orden correspondiente, se muestran a continuación:

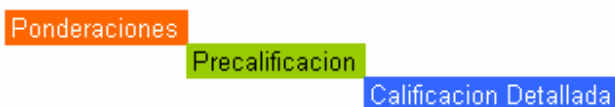


Figura16 – Orden de las hojas que requieren ingreso de datos por parte del usuario

Restricciones

- ❖ Los valores que se ingresan en cada una de las hojas deben estar en un rango entre 0 (mínimo) y 1 (máximo). Si se ingresa un número negativo o mayor a 1, el programa sacará un mensaje de error, lo que indica que el valor ingresado es incorrecto.

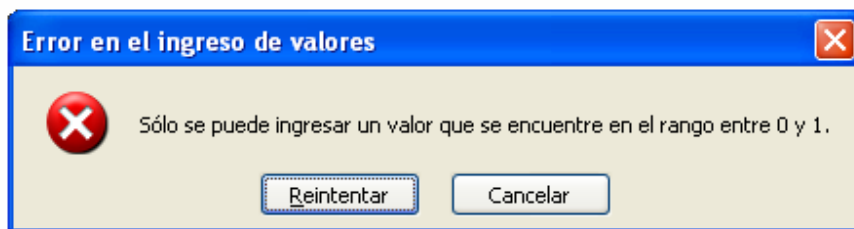


Figura17 – Cuadro de error cuando se ingresa mal un dato

- ❖ La suma de las ponderaciones siempre debe sumar 1. Si esto no se cumple, se muestra un mensaje de error en el programa donde se indica lo ocurrido. Además, si no se corrige este error, no se podrá tener una evaluación correcta del producto de software.

Error !!! **Error !!!**

Figura18 – Mensaje de error cuando no se cumple la suma de las ponderaciones

- ❖ Sólo se pueden ingresar valores en las celdas que están coloreadas de gris claro. Al intentar modificar alguna otra celda, inmediatamente el programa lo restringe, puesto que las celdas están protegidas; esto con el fin de que no se modifique información que no sea necesaria.

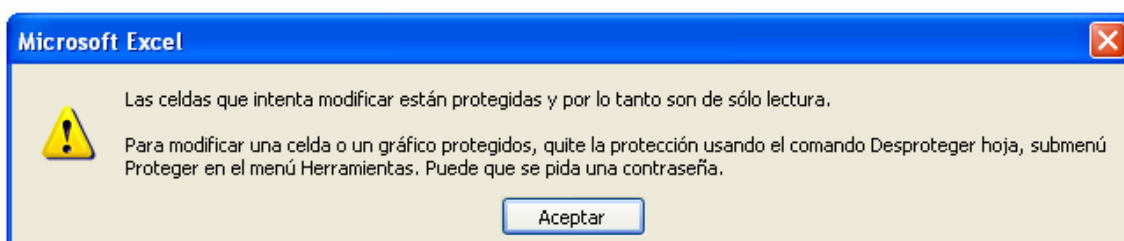


Figura19 – Cuadro de advertencia cuando se intenta acceder a una celda diferente a las del ingreso de datos disponibles

Botones

El siguiente botón proporciona ayuda a los usuarios sobre el manejo de cada una de las hojas del programa. En cada hoja existe un botón de ayuda en donde se explica el ingreso de los datos y las columnas en donde se debe realizar dicho ingreso.

Un botón rectangular con un fondo naranja y el texto "AYUDA (?)" en color blanco.

Figura 20 – Botón de ayuda

A continuación se presenta un ejemplo del cuadro que despliega el botón de ayuda una vez se de clic sobre él.

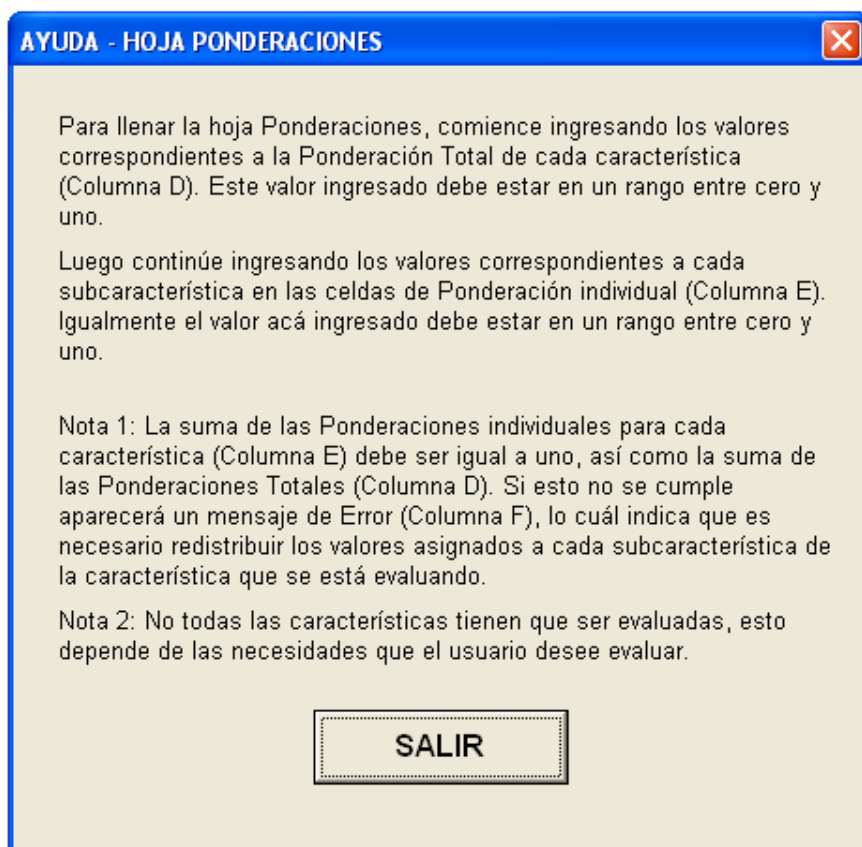


Figura 21– Ejemplo de cuadro de ayuda

Descripción de colores



El color amarillo claro en cada una de las hojas de la plantilla, hace referencia a los títulos que enmarcan cada columna.



El color azul pálido en cada una de las hojas de la plantilla, hace referencia a las características que serán evaluadas en el producto software.



El color verde claro en cada una de las hojas de la plantilla, hace referencia a las sub-características que serán evaluadas en el producto software.



El color canela en cada una de las hojas de la plantilla, hace referencia a las sub-características que serán evaluadas en el producto software.



El color gris claro en cada una de las hojas de la plantilla, hace referencia a las celdas en donde el usuario puede ingresar valores correspondientes a la ponderación y/o calificación de cada aspecto a evaluar del producto software.



El color turquesa en cada una de las hojas de la plantilla, hace referencia al nombre de las métricas externas que pueden ser evaluadas por el usuario.



El color azul cielo en cada una de las hojas de la plantilla, hace referencia al nombre de las métricas internas que pueden ser evaluadas por el usuario.



El color rojo en cada una de las hojas de la plantilla, es usado para destacar los valores que vienen de una hoja previa o que se están actualizando. Estos valores no se pueden modificar.

 El color negro en cada una de las hojas de la plantilla, es usado para destacar los valores que el usuario ingresa. Estos valores se pueden modificar cuantas veces sea necesario.

Definiciones

Las siguientes definiciones están listadas de acuerdo al orden de aparición en las plantillas.

Ponderación Total: es el peso (o valor) que se le atribuye a cada una de las características a evaluar. Esto se evalúa de acuerdo a la importancia que tiene la característica para el usuario. Este valor es el que el usuario espera que la característica evaluada del producto software alcance después de la evaluación. Su rango varía en una escala de cero a uno.

Ponderación individual: es el peso (o valor) que se le atribuye a cada una de las sub-características a evaluar. Este valor es el que el usuario espera que la sub-característica evaluada del producto software alcance después de la evaluación. Su rango varía en una escala de cero a uno.

La suma de las ponderaciones individuales para cada característica debe ser igual a uno.

En el siguiente cuadro, se muestra un ejemplo de lo mencionado anteriormente.

<i>Ponderación Total:</i> 0,30		<i>Ponderación individual</i>
Adecuación	La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario.	0,20
Exactitud	La capacidad del producto de software para proveer los resultados o efectos acordados con un grado necesario de precisión.	0,50
Conformidad con la funcionalidad	La capacidad del producto de software de adherirse a los estándares, convenciones o regulaciones legales y prescripciones similares referentes a la funcionalidad.	0,30
<i>Suma de las Ponderaciones</i>		1,00

Figura 22 – Ejemplo de ingreso de datos para las ponderaciones

$$\text{Suma de las Ponderaciones} = \sum(\text{Ponderaciones individuales por característica})$$

Calificación: en la hoja “Precalificación” es el valor que el usuario le asigna a cada una de las sub-características que desea evaluar del producto software sin haber hecho la evaluación correspondiente con las métricas.

En la hoja “Calificación Detallada” es el valor que la sub-característica evaluada del producto software alcanza después de la evaluación con las métricas y su rango varía en una escala de cero a uno.

Calificación ponderada: es el valor que el programa retorna de cada una de las sub-características. Se calcula así:

$$\text{Calificación ponderada} = (\text{Ponderación individual} * \text{calificación})$$

Calificación Total: es la suma de cada una de las calificaciones pertenecientes a una misma característica. Este valor es retornado por el programa.

$$\text{Calificación Total} = \sum(\text{calificaciones por característica})$$

Calificación Ponderada Total: es la multiplicación de la *Ponderación Total* con la *Calificación Total*, por característica. Este valor es retornado por el programa.

$$\text{Calificación Ponderada Total} = (\text{Ponderación Total} * \text{Calificación Total})$$

DESCRIPCIÓN DE LAS HOJAS

Cada una de las hojas de Excel, mostradas en el siguiente gráfico contiene información correspondiente para realizar la evaluación de un producto software. Para realizar una correcta evaluación, se debe diligenciar cada una de las hojas en el orden en que se muestran, es decir, es necesario primero ingresar todos los datos requeridos en la hoja “Ponderaciones”, para luego pasar a la hoja “Precalificación” y así sucesivamente.



Figura 23 – Nombres de las hojas de la plantilla

La primera hoja con la que el usuario se encuentra una vez abra el archivo de Excel correspondiente a la Plantilla, es la hoja “Ponderaciones”. En esta hoja se hace una breve descripción de cada una de las características y sub-características a ser evaluadas por el usuario sobre un producto software.

Acá se comienza con el ingreso de valores correspondiente a la *Ponderación Total* y a la *Ponderación individual*. Dichos valores son copiados automáticamente a las hojas “Precalificación” y “Calificación Detallada”.

En la Figura 24 se detalla ésta hoja.

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	Definición	
Funcionalidad	Ponderación Total:	0.00	Ponderación individual
La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.	Adecuación	La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario.	
	Exactitud	La capacidad del producto de software para proveer los resultados o efectos acordados con un grado necesario de precisión.	
	Conformidad con la funcionalidad	La capacidad del producto de software de adherirse a los estándares, convenciones o regulaciones legales y prescripciones similares relevantes a la funcionalidad.	
	Suma de las Ponderaciones		0.00
Seguridad	Ponderación Total:	0.00	Ponderación individual
La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o sistemas autorizados no se les niegue el acceso a ellos.	Resistente al Acceso	La capacidad del software para proteger de accesos ilegales y no autorizados.	
	Resistente a la copia	La capacidad del producto de software para proteger de copias ilegales.	
	Facilidad de cifrado	La capacidad del producto de software para proteger de buscadores ilegales por encriptación.	
	Resistente a la falsificación	La capacidad del producto de software para no permitir el análisis de la estructura interna y de datos almacenados.	
	Robustez	La capacidad del producto de software para recuperarse de entradas y situaciones anómalas.	
	Conformidad con la seguridad	La capacidad del producto de software para adherirse a estándares, convenciones o regulaciones en lo relacionado con seguridad.	
Suma de las Ponderaciones			0.00
Interoperabilidad	Ponderación Total:	0.00	Ponderación individual
La capacidad del producto de software de interactuar con uno o más sistemas específicos.	Compatibilidad de la OSI	La capacidad del producto de software para interactuar con uno o más sistemas especificados en cada nivel de la capa de la OSI (Open Systems Interconnection).	
	Compatibilidad del Software	La capacidad del producto de software para ser cooperativamente operable con uno o más productos de software.	
	Compatibilidad de los datos	La capacidad del producto de software para intercambiar los datos con uno o más sistemas específicos.	
	Interoperabilidad	La capacidad del producto software para registrar los registros operacionales.	

Figura 24 – Hoja ponderaciones

La hoja siguiente es la de “Precalificación”, allí también se detallan cada una de las características y subcaracterísticas, y además se debe ingresar la calificación correspondiente a cada subcaracterística. La calificación de cada característica es calculada de manera automática por el programa con los datos de la hoja “Ponderaciones” ingresados previamente y con los datos ingresados en la columna *Calificación*.

De esta hoja no pasa ningún valor a la hoja “Calificación Detallada”, puesto que los datos que hay en esta hoja no son calculados con las métricas como en la hoja “Calificación Detallada”. La idea de diligenciar la hoja “Precalificación” es de hacer un estimativo de lo que el usuario considera de cada característica y subcaracterística, antes de realizar la evaluación con las respectivas métricas.

CARACTERÍSTICAS	Ponderación Individual	Calificación Individual	Calificación Ponderada	SUBCARACTERÍSTICAS	DEFINICIÓN	Ponderación Individual	Calificación Individual	Calificación Ponderada
Funcionalidad	0,00	0,00	0,00					
La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software de software bajo condiciones específicas.				Adecuación	La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas a efectuar especificadas por el usuario.	0,00		0,00
				Exactitud	La capacidad del producto de software para proveer los resultados o efectos esperados con un grado aceptable de precisión.	0,00		0,00
				Conformidad con la funcionalidad	La capacidad del producto de software de adherirse a los estándares, convenciones o regulaciones legales y percepciones similares referentes a la funcionalidad.	0,00		0,00
				Suma de las Ponderaciones		0,00		0,00
Seguridad	0,00	0,00	0,00					
La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o sistemas autorizados no se les niegue el acceso a ellos.				Resistente al Acceso	La capacidad del software para proteger de accesos ilegales y no autorizados.	0,00		0,00
				Resistente a la copia	La capacidad del producto de software para proteger de copias ilegales.	0,00		0,00
				Facilidad de cifrado	La capacidad del producto de software para proteger de bases de datos ilegales por recuperación.	0,00		0,00
				Resistente a la falsificación	La capacidad del producto de software para no permitir el análisis de la estructura interna y de datos almacenados.	0,00		0,00
				Robustez	La capacidad del producto de software para recuperarse de entradas y salidas anómalas.	0,00		0,00
				Conformidad con la seguridad	La capacidad del producto de software para adherirse a estándares, convenciones o regulaciones de la relación con seguridad.	0,00		0,00
				Suma de las Ponderaciones		0,00		0,00
Interoperabilidad	0,00	0,00	0,00					
La capacidad del producto de software de interactuar con uno o más sistemas especificados.				Compatibilidad de la OBI	La capacidad del producto de software para interactuar con uno o más sistemas especificados en cada nivel de la capa de la OSI (Open Systems Interconnection).	0,00		0,00
				Compatibilidad del Software	La capacidad del producto de software para ser cooperativamente operable con uno o más productos de software.	0,00		0,00
				Compatibilidad de los datos	La capacidad del producto de software para intercambiar los datos con uno o más sistemas especificados.	0,00		0,00
				Trazabilidad	La capacidad del producto software para registrar los registros operacionales para seguir la causa de la interacción no deseada.	0,00		0,00

Figura 25 – Hoja precalificación

La siguiente hoja es “Calificación Detallada”, allí se presenta información sobre cada una de las métricas que contiene cada sub-característica. Cada métrica, presenta el propósito y la fórmula para ser calculada. Si se desea mayor información al respecto, se puede hacer clic sobre la métrica y esta lo conduce directamente a la hoja Métricas donde se puede detallar toda la información correspondiente a la métrica seleccionada.

También hay un campo que se denomina Justificación Evaluación/Calificación en el cuál se puede agregar algún comentario sobre la métrica que se está evaluando. Sirve para llevar un registro de la evaluación que se está realizando.

En esta hoja se deben ingresar datos en los campos *Ponderación individual* que se encuentra en el encabezado ATRIBUTOS y en *Calificación* que se encuentra en el encabezado MÉTRICAS.

Las figuras 26 y 27 ilustran los campos donde se debe ingresar la información.

CARACTERÍSTICAS			SUBCARACTERÍSTICAS			ATRIBUTOS			Métricas Internas	Métricas Externas
Calificación Total	Ponderación Individual	Calificación ponderada	Calificación Total	Ponderación Individual	Calificación ponderada	Calificación Total	Ponderación Individual	Calificación ponderada		
FUNCIONALIDAD	0,00	0,00	ADECUACIÓN	0,00	0,00	Suficiencia Funcional	0,00	0,00	Que tan adecuadas son las funciones verificadas? $X = 1 - A/B$ A = número de funciones en las cuales los problemas son detectados en la evaluación B = número de funciones verificadas	
						Integridad de la implementación funcional	0,00	0,00	Qué tan completa está la implementación funcional? $X = 1 - A/B$ A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos	
						Alcance de la implementación funcional	0,00	0,00	Qué tan correcta es la implementación funcional? $X = 1 - A/B$ A = número de funciones faltantes o implementadas incorrectamente B = número de funciones descritas en la especificación de requisitos	
						Estabilidad (volatilidad) de la especificación funcional	0,00	0,00	Qué tan estable es la especificación funcional durante el desarrollo del ciclo de vida? $X = 1 - A/B$ A = número de funciones cambiadas durante las fases de desarrollo del ciclo de vida B = número de funciones descritas en la especificación de requisitos	
						Suficiencia Funcional	0,00	0,00	Que tan adecuadas son las funciones verificadas? $X = 1 - A/B$ A = número de funciones en las cuales los problemas son detectados en la evaluación B = número de funciones verificadas	
						Integridad de la implementación funcional	0,00	0,00	Qué tan completa es la implementación de acuerdo a la especificación de requisitos? $X = 1 - A/B$ A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos	
Ponderaciones / Precalificación / Calificación Detallada / Resumen Cal Det / Gráficos Cal Det / Métricas										

Figura 26 – Hoja calificación detallada

ATRIBUTOS			MÉTRICAS						
	Calificación	Ponderación Individual	Calificación ponderada		Justificación Evaluación / Calificación		Calificación	Ponderación Individual	Calificación ponderada
Suficiencia Funcional	0,00		0,00	Que tan adecuadas son las funciones verificadas? X = 1 - A/B A = numero de funciones en las cuales los problemas son detectados en la evaluación B = número de funciones verificadas				1,00	0,00
Integridad de la implementación funcional	0,00		0,00	Qué tan completa está la implementación funcional? X = 1 - A/B A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos				1,00	0,00
Alcance de la implementación funcional	0,00		0,00	Qué tan correcta es la implementación funcional? X = 1 - A/B A = numero de funciones faltantes o implementadas incorrectamente B = número de funciones descritas en la especificación de requisitos				1,00	0,00
Estabilidad (volatilidad) de la especificación funcional	0,00		0,00	Qué tan estable es la especificación funcional durante el desarrollo del ciclo de vida? X = 1 - A/B A = numero de funciones cambiadas durante las fases de desarrollo del ciclo de vida. B = número de funciones descritas en la especificación de requisitos				1,00	0,00
Suficiencia Funcional	0,00		0,00	Que tan adecuadas son las funciones verificadas? X = 1 - A/B A = numero de funciones en las cuales los problemas son detectados en la evaluación B = número de funciones verificadas.				1,00	0,00
Integridad de la implementación funcional	0,00		0,00	Qué tan completa es la implementación de acuerdo a la especificación de requisitos? X = 1 - A/B A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos				1,00	0,00
Ponderaciones / Precalificación / Calificación Detallada / Resumen Cal Det / Gráficos Cal Det / Métricas									

Figura 27 – Hoja calificación detallada (continuación)

La hoja Resumen Calificación Detallada contiene un cuadro donde se resumen los resultados de los valores calculados anteriormente en la hoja “Calificación Detallada” para cada característica y subcaracterística. Allí se detallan los valores para la *Calificación Total*, *Ponderación Total* y *Calificación Ponderada Total*. Ver Figura 28.

También contiene información gráfica de cada característica y sub-característica respectivas, haciendo un comparativo entre la *Calificación Ponderada Total* y la *Calificación Total*. Ver Figura 29.

A partir de esta hoja ya no hay que ingresar mas valores, pues las hojas que siguen son de información para el usuario de acuerdo a los valores ingresados previamente.

CARACTERÍSTICAS	Calificación Total	Ponderaciones Total	Calificación Ponderada Total	SUBCARACTERÍSTICAS	Calificación Total	Ponderaciones Total	Calificación Ponderada Total
Funcionalidad	0	0	0	Adecuación	0	0	0
				Exactitud	0	0	0
				Conformidad con la funcionalidad	0	0	0
Seguridad	0	0	0	Resistente al Acceso	0	0	0
				Resistente a la copia	0	0	0
				Facilidad de cifrado	0	0	0
				Resistente a la falsificación	0	0	0
				Robustez	0	0	0
				Conformidad con la seguridad	0	0	0
Interoperabilidad	0	0	0	Compatibilidad de la OSI	0	0	0
				Compatibilidad del Software	0	0	0
				Compatibilidad de los datos	0	0	0
				Trazabilidad	0	0	0
				Conformidad con la interoperabilidad	0	0	0
Fiabilidad	0	0	0	Madurez	0	0	0
				Tolerancia a fallos	0	0	0
				Recuperabilidad	0	0	0
				Conformidad con la fiabilidad	0	0	0
Usabilidad	0	0	0	Apropiabilidad	0	0	0
				Facilidad de Aprendizaje	0	0	0

Figura 28 – Hoja resumen calificación detallada

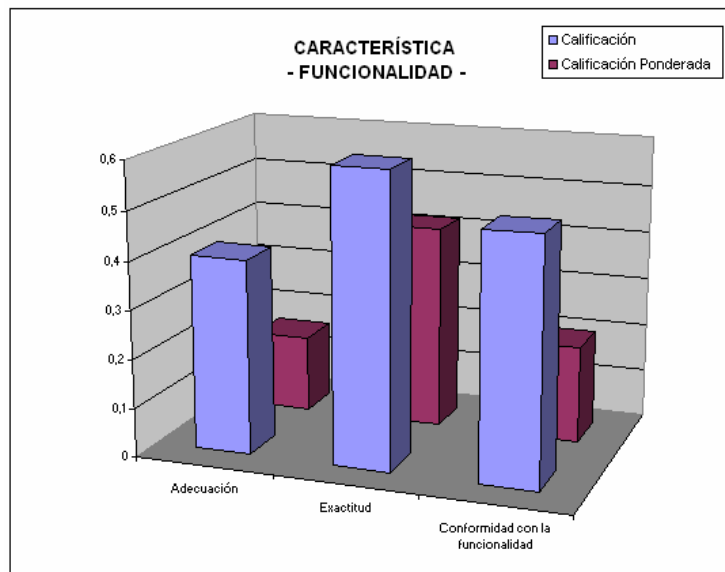


Figura 29 – Muestra de la gráfica de una característica - Hoja resumen calificación detallada

La hoja Gráficos Calificación Detallada muestra la relación de cada uno de los valores obtenidos de las características, tanto numéricamente como gráficamente. Esos valores de cada característica son: *Calificación Total*, *Ponderación Total*, *Calificación Ponderada Total*. Ver Figura 30.

CARACTERÍSTICAS	Calificación	Ponderación	Calificación Ponderada
Funcionalidad	0	0	0
Seguridad	0	0	0
Interoperabilidad	0	0	0
Fiabilidad	0	0	0
Usabilidad	0	0	0
Eficiencia	0	0	0
Mantenibilidad	0	0	0
Portabilidad	0	0	0

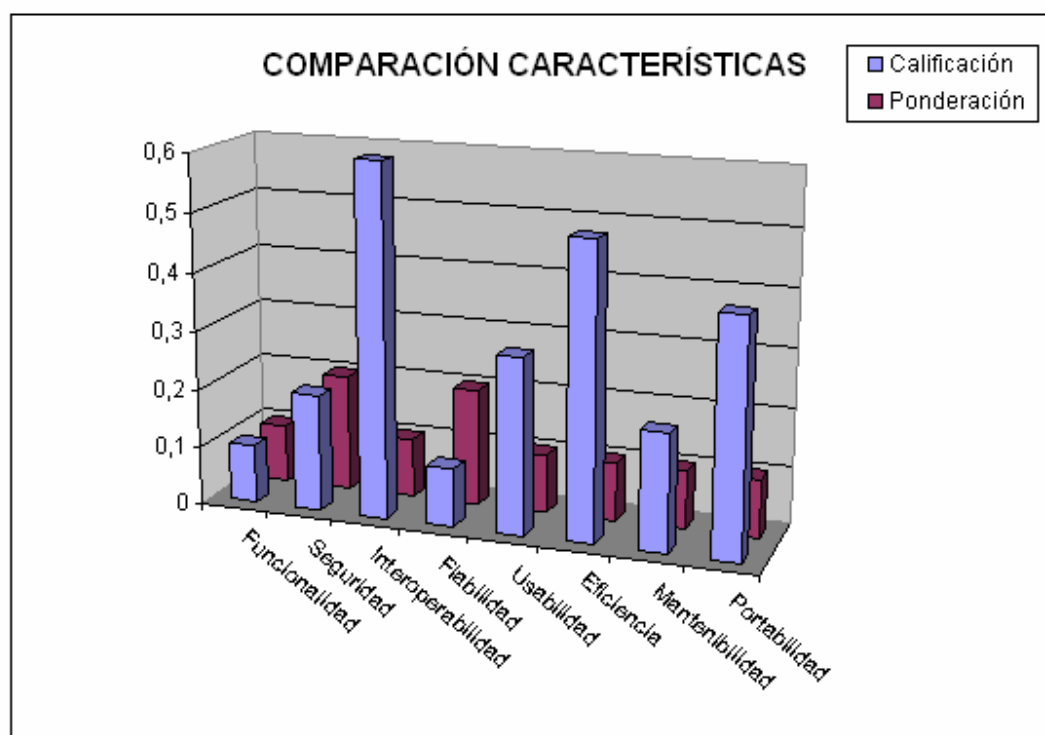


Figura 30 – Hoja gráficos calificación detallada

Finalmente, la hoja Métricas contiene la información detallada de cada una de las métricas que se encuentra en la hoja Calificación Detallada, con el fin de servir de apoyo al usuario en el momento en que necesite mayor información al respecto sobre alguna métrica.

MÉTRICAS

FUNCIONALIDAD

ADECUACIÓN

Métrica externa de Adecuación - Alcance de la implementación funcional

Nombre:	Functional implementation coverage
Propósito:	How correct is the functional implementation?
Método de aplicación:	Count the number of incorrectly implemented or missing functions and compare with the number of functions described in the requirement specifications.
Medición, fórmula:	$X = 1 - A/B$ A = Number of incorrectly implemented or missing functions detected B = Number of functions described in requirement specifications
Interpretación:	$0 \leq X \leq 1$ The closer to 1, the more correct
Tipo de escala:	Absolute
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Req. Spec Design Source code Review report
ISO/IEC 12207	Verification

Precalificación / Calificación Detallada / Resumen Cal Det / Gráficos Cal Det / **Métricas** /

Figura 31 – Hoja métricas