



# Diseño de artefactos y su almacenamiento para el ciclo de vida de desarrollo de *software* de Banco Falabella en Colombia

## Design of Devices and their Storage for the Falabella Bank Software Development Lifecycle in Colombia

Claudia Milena Arteaga<sup>1</sup> Fernando Augusto Barrera<sup>2</sup> Jarold Javier Chaparro<sup>3</sup>

### ARTÍCULO DE REFLEXIÓN

Fecha de recepción:

02-06-2014

Fecha de aceptación:

03-06-2017

ISSN: 2344-8288

Vol. 5 No. 1

Enero - junio 2017

Bogotá-Colombia

**Para citar este artículo:** Arteaga, C. M.; Barrera, F. A.; Chaparro, J. J. (2017). Diseño de artefactos y su almacenamiento para el ciclo de vida de desarrollo de software de banco Falabella en Colombia. *TIA*, 5(1), pp. 121-129.

### Resumen

El presente artículo señala los factores relevantes en la propuesta de diseño de artefactos en el ciclo de desarrollo de software de Banco Falabella en Colombia. Se nombran los estándares utilizados para encontrar un diseño de artefactos con la estructura necesaria para el proceso de desarrollo de software; se describen las herramientas usadas para recopilar información: observación directa y encuesta. Con la primera se incursionó en el ambiente en que los individuos desarrollan normalmente actividades relacionadas con el objeto de estudio y con la segunda se recopiló información para establecer los artefactos que los usuarios participantes en el proceso de desarrollo de software consideran deben implementarse. Con el análisis de la información recopilada, se continuó con la propuesta de los diseños a aplicar para satisfacer la necesidad de contar con artefactos estandarizados que faciliten el proceso y que luego sean incluidos en un repositorio que permita la ubicación de la información para ser usada en procesos posteriores.

**Palabras clave:** artefactos, caso de uso, ciclo de vida de desarrollo de software, estándar, requerimiento.

<sup>1</sup> Ingeniera de Sistemas e Ingeniera Telemática, Universidad Icesi. Staff Consultant, Oracle. Correo electrónico: claudiamilena11@hotmail.com

<sup>2</sup> Administrador de Empresas. Analista Funcional, ITS Solutions. Correo electrónico: fbarrerapi@gmail.com

<sup>3</sup> Ingeniero de Sistemas. Ingeniero de Desarrollo Senior, Banco Falabella. Correo electrónico: jaroldjavier@gmail.com

## Abstract

This article points out relevant factors of devices design in the software development cycle of Falabella Bank in Colombia. We name standards used to find devices design with structure necessary for software development process. We describe tools used to collect information: direct observation and survey. With the first we enter into the environment in which individuals normally develop activities related to the object of study and with the second we collect information to establish devices that users—who are participants in the software development process—consider should be implemented. With the analysis of information collected, we continue with the proposal of designs to be applied to satisfy the need to have standardized devices that facilitate the process and then be included in a repository that allows to locate information to be used in later processes.

**Keywords:** devices, use case, software development lifecycle, standard, requirement.

## INTRODUCCIÓN

Es de resaltar que el resultado de este proyecto fue la entrega de una propuesta de diseño de artefactos para el apoyo al ciclo de vida de desarrollo de *software* y su correspondiente almacenamiento, sin embargo, la aplicación e implementación de los mismos queda en manos del Banco Falabella en Colombia. Teniendo en cuenta este contexto, el objeto del trabajo consistió en identificar formalmente las falencias del área de tecnología del banco, debido a la falta de estandarización en la documentación del ciclo de vida de desarrollo de *software*.

Se diseñaron artefactos para los puntos más críticos del ciclo de desarrollo de *software*, tomando como base las recomendaciones de los estándares IEEE relacionados. Los formatos obtenidos con este proceso garantizarán que se tenga un único punto de entrada de la información, esto para asegurar que no se presenten inconsistencias por información desactualizada.

En la primera parte del documento se presentan el planteamiento del problema detectado al interior del área de tecnología del Banco Falabella, los argumentos que justifican la realización de este proyecto, así como los objetivos, alcances y limitaciones del mismo. Posteriormente, se presentan los resultados del análisis realizado para detectar y clasificar las falencias del área; finalmente, se relaciona

la propuesta para la solución del problema detectado.

## PRELIMINARES

En esta sección se presenta el tema de investigación, en el cual se plantea la importancia de contar con un proceso documental que soporte el ciclo de desarrollo de *software* durante su ciclo de vida. Este es precisamente el tema que le da origen al proyecto, para el cual se considera ideal el escenario que nos brinda el área de tecnología del Banco Falabella en Colombia, tema que se desarrolla con el apoyo de los estándares IEEE.

Una vez establecido el tema de investigación, se continúa con el planteamiento del problema, el cual se origina a partir las falencias detectadas en el Banco Falabella, las cuales se detallan a continuación:

- No existe una metodología que indique formalmente los artefactos que deben generarse durante el ciclo de vida de desarrollo. Realmente no existe el concepto “artefacto” al interior del área, en su lugar, en cada proyecto se generan documentos a discreción del equipo de trabajo con la forma y contenido que dicho equipo crea necesarios.
- Una pieza de información puede estar repetida en diferentes documentos, con la imposibilidad

de saber a ciencia cierta cuál es la versión más reciente de dicha información.

- Se dejan de documentar puntos importantes de los diferentes desarrollos debido a la informalidad de la documentación.
- No existe una biblioteca que permita almacenar los documentos generados durante el ciclo de vida de desarrollo de un proyecto de *software*, razón por la cual hay mucha información extraviada y en muchas ocasiones es necesario realizar levantamiento de información desde cero.

De las falencias mencionadas anteriormente se plantean las siguientes preguntas de investigación:

- ¿Qué artefactos deberían generarse para apoyar la gestión del proceso de desarrollo de *software*?
- ¿Por qué hay información duplicada en los documentos que se manejan actualmente en el proceso de desarrollo?
- ¿Qué información debería incluir cada artefacto del proceso?
- ¿Qué esquema de almacenamiento se debería definir para los artefactos que se generen durante el ciclo de vida de desarrollo de *software*?

Es así que después de este proceso se obtiene el planteamiento de los objetivos generales y específicos, los cuales se presentan a continuación.

## Objetivo general

Diseñar artefactos del ciclo de vida de desarrollo de *software* y su respectivo almacenamiento para el área de tecnología del Banco Falabella en Colombia, con el fin de minimizar las falencias que se derivan de la falta de estandarización en la documentación que maneja el área.

## Objetivos específicos

- Identificar las falencias derivadas de la falta de estandarización en la documentación del ciclo de vida de desarrollo en el área de tecnología del Banco Falabella.
- Diseñar artefactos para las fases del ciclo de vida de desarrollo de *software* que lo requieran, de acuerdo a las falencias identificadas, tomando como base las buenas prácticas documentadas en los estándares IEEE para ingeniería de *software*.
- Evitar la redundancia de información en los artefactos que se diseñen, de manera que la información esté en un solo punto.
- Generar un esquema propuesto de almacenamiento de los artefactos que apoyarán el proceso dentro del banco.

Una vez planteados los objetivos de ejecución de este trabajo de investigación, se identificaron los motivos que lo justifican.

Es así que se planteó una justificación que se soporta básicamente en los estándares ISO/IEC/IEEE 12207, 15289, 25010, 26511, 29119 y 29148 en sus versiones más recientes.

Es importante contar con un proceso documental estructurado, pues:

**... la documentación es un componente esencial para el éxito de cualquier proyecto de software, la producción de documentación implica tiempo, esfuerzo y dinero. Es responsabilidad de la gerencia asegurar el uso de recursos que reconozcan la importancia que tiene la documentación en la calidad y el éxito de un proyecto de software [1]**

Del análisis de este concepto y de las falencias detectadas en Banco Falabella, se da como resultado esta propuesta que buscó crear artefactos para los puntos críticos del proceso de desarrollo de *software*, lo anterior beneficiará al banco al trabajar con buenas prácticas enunciadas en normas internacionales.

Además de plantear el diseño de los artefactos, como CMMI lo indica también se realizó una propuesta que facilite el almacenamiento de “los activos de proceso que incluyen documentación relativa a los procesos, tal como políticas, procesos definidos, listas de comprobación, documentos de lecciones aprendidas, plantillas, estándares, procedimientos, planes y materiales de formación” [2].

Al finalizar esta parte del proceso de investigación se planteó el alcance y las limitaciones con las que se desarrolla, en la cual expresamente se resalta que el presente trabajo buscó satisfacer una necesidad del área de tecnología del Banco Falabella en Colombia. Se resalta que no se abarcaran todas las etapas del desarrollo de *software* sino las más críticas, las cuales se detectaron con la realización del proceso de diagnóstico del área de tecnología.

La premisa fue que los diseños de los artefactos generados debían contener como mínimo lo recomendado por las normas IEEE, además de puntos específicos que estén relacionados con la operación del área.

## ANÁLISIS

En el siguiente apartado se presentan los resultados del análisis realizado para detectar y clasificar las falencias del área.

Con el fin de conocer el estado del proceso actual, se optó por el uso de herramientas como la observación directa con la que se busca ver las actividades que realizan las personas involucradas en el lugar donde normalmente desarrollan sus actividades; adicionalmente, se desarrolló una encuesta para la recopilación de datos que permitiera elaborar un diagnóstico el

cual ayude a identificar las necesidades del área respecto a los artefactos que se necesitan para apoyar el proceso de desarrollo de *software*.

De acuerdo al proceso planteado, el resultado de la observación directa arrojó los siguientes comentarios:

- Se acudió a la jefatura del área de desarrollo, se indagó sobre el procedimiento y artefactos generados durante todo el ciclo de vida del desarrollo de *software*.
- La jefatura, quien tiene el rol líder de proyectos, recomendó observar los artefactos generados para el requerimiento “Manejo de saldos negativos y compras internacionales”, ubicados en la herramienta SRT (Sistema Requerimientos de Tecnología). Desde allí se tomaron dichos artefactos y se analizaron desde la creación del requerimiento hasta la puesta en producción.
- También se revisó en la herramienta SRT la documentación de otros requerimientos ya instalados en producción, con lo anterior se corroboró que todos los artefactos son generados con características similares respecto a los del requerimiento en mención.

Por otro lado, se realizó una encuesta para recopilar información que permitiera identificar las principales falencias al interior de la gerencia, así como identificar los puntos críticos respecto a la necesidad del diseño de artefactos, los resultados de la aplicación de la encuesta se observan en la Tabla 1.

Una vez conocidos los resultados de la observación directa y de la encuesta, se procedió a identificar los estándares que se podrían utilizar para la elaboración de la alternativa de solución para el área de tecnología, a continuación se describen los puntos más importantes de los que se utilizaron en el presente trabajo.

**Tabla 1.** Resultados de la encuesta

Pregunta	TOTAL RESPUESTAS							
	A	B	C	D	E	F	G	H
1	26	4	NA	NA	NA	NA	NA	NA
2	5	25	NA	NA	NA	NA	NA	NA
3	5	25	NA	NA	NA	NA	NA	NA
4	5	25	NA	NA	NA	NA	NA	NA
5	5	13	12	NA	NA	NA	NA	NA
6	6	24	NA	NA	NA	NA	NA	NA
7	8	22	NA	NA	NA	NA	NA	NA
8	0	2	5	13	8	2	NA	NA
9	7	3	5	5	10	NA	NA	NA
10	24	6	NA	NA	NA	NA	NA	NA
11	9	21	NA	NA	NA	NA	NA	NA
12	9	14	4	3	NA	NA	NA	NA
13	15	2	13	0	NA	NA	NA	NA
14	4	2	4	0	0	9	11	11
15	0	0	0	0	0	10	20	20
16	8	22	NA	NA	NA	NA	NA	NA
17	10	0	13	7	NA	NA	NA	NA
18	11	11	8	NA	NA	NA	NA	NA
19	9	21	NA	NA	NA	NA	NA	NA
20	24	6	NA	NA	NA	NA	NA	NA
21	8	23	4	16	6	4	4	3
22	0	18	4	4	2	2	0	NA
23	26	4	NA	NA	NA	NA	NA	NA

Fuente: elaboración propia

### ISO/IEC/IEEE 25010 Software engineering -Software Product Quality Requirements and Evaluation (SQuRE)- Software and Quality in Use Models

Este estandar establece criterios para:

[la] especificación de requisitos de calidad de productos de software, sus métricas y su evaluación, e incluye un modelo de calidad compuesto por características y subcaracterísticas como las siguientes:

Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad, dentro de las cuales se evalúan una serie de subcaracterísticas, que definen la calidad del software estudiado [1]

### ISO/IEC/IEEE 26511:2012 Systems and software engineering -Requirements for Managers of User Documentation

En este estándar se resalta lo siguiente:

- Requisitos de gestión en el inicio de un proyecto, incluyendo el establecimiento de procedimientos y especificaciones, el establecimiento de la infraestructura y la construcción de un equipo, con ejemplos de los roles necesarios en un equipo de documentación para el usuario
- Mediciones y estimaciones necesarias para el control de gestión

- La aplicación de control de gestión para el trabajo de documentación del usuario
- El uso de los procesos de apoyo, tales como la gestión del cambio, la programación y el control de costes, gestión de recursos, gestión de calidad y mejora de procesos [1]

## ISO/IEC/IEEE 29119-1 Software and Systems Engineering -Software Testing- Part 1 Concepts and Definitions

El objetivo de este estándar es:

facilitar la comprensión y el uso de todos los otros estándares de la serie 29119. También introduce el vocabulario en el que todas las normas de la serie 29119 son construidas y proporciona ejemplos de la aplicación de cada concepto en la práctica. Proporciona definiciones, una descripción de los conceptos de pruebas de software y la forma de aplicar los procesos, documentos y técnicas definidas en la serie 29119 [3].

## IEEE 29119-3 Software and Systems Engineering -Software Testing- Part 3 Test Documentation

El objetivo de este estándar es:

definir plantillas para la documentación de prueba que cubre todo el ciclo de vida de las pruebas de software. Cada plantilla se puede adaptar para satisfacer las necesidades únicas de cada organización, para apoyar la aplicación de la norma dentro de cualquier etapa del ciclo de vida de desarrollo de software. Todas las plantillas se alinean con el proceso definido en la norma ISO/IEC/IEEE 29119-2 y pueden producirse mediante la aplicación de los procesos que están definidos en dicha norma [4].

## ISO/IEC/IEEE 29148:2011 Systems and Software Engineering -Life Cycle Processes- Requirements Engineering

Esta estándar proporciona un tratamiento unificado de los procesos y productos que

intervienen en el establecimiento de los requisitos de ingeniería en todo el ciclo de vida del *software*.

## PROPUESTA

La siguiente y última etapa en el desarrollo del proyecto fue la propuesta para reducir los puntos débiles que tiene el área de tecnología en su proceso de desarrollo de *software*. Como se evidenció en la encuesta, varios colaboradores manifestaron que el proceso actual respecto a los documentos que soportan el proceso de desarrollo de *software* no facilita la entrega al usuario de la aplicación que requiere; adicionalmente, se demostró el interés de los funcionarios de contar con documentos formales que faciliten la generación de los diferentes entregables como los son casos de uso, casos de prueba, requerimientos y guía de instalación.

A continuación, se relacionan los artefactos diseñados según la recopilación de la información anteriormente descrita, acompañados de la fundamentación teórica que los sustenta; igualmente, se encuentra la propuesta para almacenar los formatos que se generen durante el ciclo de desarrollo de *software*.

## Requerimientos

Inicialmente se realizó la revisión del tema de requerimientos, donde según la norma IEEE 15289 de 2011 se describe de manera general el contenido de un documento de requerimientos; sin embargo, es el estándar IEEE 29148 de 2011 el que propone tres tipos de documentos para realizar la especificación de requerimientos. A continuación se describe brevemente cada uno de ellos.

- Especificación de requerimientos de stakeholders:

es el formato en el que se documentan los requerimientos de alto nivel desde el punto de vista de los stakeholders. Es decir es el que me indica quienes están relacionados con el proyecto, su nivel de injerencia y cómo pueden afectarme las decisiones que cada usuario tome desde su posición [5].

- Especificación de requerimientos del sistema: “es él que comunica la especificación de los requerimientos de los stakeholders al equipo técnico que construirá el sistema, por lo tanto, este documento debe ser entendible tanto para los stakeholders como para el equipo técnico del proyecto” [5].
- Especificación de requerimientos de software: “en este documento queda la especificación de un producto de software, programa o conjuntos de programas que realizarán ciertas funciones en un entorno específico” [5].

Es de anotar que debido a las condiciones del proceso actual en el cual no se cuenta con un documento estándar hace poco viable pasar a tener tres, por lo cual este proceso desarrolló un solo documento que combinara la propuesta que realiza el estándar, incluyendo ítems según el proceso definido en el área de tecnología del Banco Falabella.

El documento desarrollado se denominó “TI-IDProyecto-Requerimientos.doc”.

## Casos de uso

El siguiente formato que se sugirió formalizar es el de casos de uso, debido a que se considera que el proceso de ingeniería de requerimientos de *software* es uno de los procesos clave en el desarrollo de *software*, donde el objetivo es capturar, entender y analizar los requerimientos del cliente o usuario; los casos de uso se utilizan para especificar requerimientos funcionales, además de facilitar los requerimientos de *software* desde el punto de vista de los actores. Teniendo en cuenta lo mencionado anteriormente, se puede concluir que los casos de uso complementan la especificación de requerimientos, hacen parte de

ella y son el punto medio entre las necesidades de los usuarios y lo que deben tener en cuenta los desarrolladores para construir el resultado esperado del sistema. De ahí que en el documento de especificaciones se haya incluido el diagrama de casos de uso del sistema a construir como una mirada general de lo que va a realizar el sistema y, por otro lado, los casos de uso quedaron como documentos independientes que complementan la interpretación de los requerimientos.

El documento desarrollado se denominó “TI-CU-IDProyecto-NombreCU-001.doc”.

## Casos de prueba

Otro de los formatos que se trabajó dentro de la investigación fue el de casos de prueba, el cual es necesario para obtener información respecto de las características de calidad del *software*.

**Al mismo tiempo, aunque se tiene la percepción de que no es posible crear un software perfecto, también es clara la necesidad de probar el software antes de que se libere a los usuarios con el fin de reducir riesgos de tener errores en producción y generar impactos negativos en el negocio [3].**

El formato que se propuso implementar pretende ayudar a medir la calidad respecto a ocho características: “pertinencia funcional, eficiencia en el desempeño, compatibilidad, usabilidad, confiabilidad, seguridad, facilidad de mantenimiento y portabilidad” [1]

Según el concepto mencionado, el Banco Falabella requería contar con un formato estándar que le permitiera realizar la documentación de sus procesos de prueba, “ese artefacto debe contar como mínimo con lo siguiente:

- Identificador único
- Objetivo
- Prioridad
- Trazabilidad
- Precondiciones
- Entradas
- Resultados esperados

Resultados obtenidos y resultado de la prueba (exitosa o no exitosa)” [4]

El documento resultado de este proceso es “TI-CP-IDProyecto-NombreCU-001.xls”.

## Guía de instalación

El último de los formatos propuestos fue la guía de instalación, cuyo objetivo es servir de apoyo para colocar un producto de *software* en un ambiente objetivo que puede ser de desarrollo, pruebas o producción.

“Es de vital importancia que el plan de instalación sea formalmente documentado, ya que el encargado de dicha actividad debe instalar el producto de software de acuerdo a lo que se haya definido en el respectivo plan de instalación” [6].

El plan de instalación debe incluir prerequisites de *software* y *hardware*, y las instrucciones detalladas de instalación [7]; adicionalmente, debe incluir la estrategia (pasos detallados) para devolver el sistema a su último estado estable en caso que la instalación no sea exitosa, por lo tanto, es indispensable que se incluya un punto para la realización de un *backup* antes de iniciar la instalación del desarrollo [6].

El resultado de este proceso de investigación es el formato “TI-IDProyecto-Guía\_Instalación.doc”.

## Esquema de almacenamiento

Con fin de garantizar la centralización de la información y su posterior reutilización y también para evitar la pérdida de conocimiento por factores como rotación de personal se propuso un esquema de almacenamiento con la herramienta de versionamiento “Tortoise SVN”.

A continuación se presenta algunas ventajas que ofrece esta herramienta:

- Trazabilidad absoluta del control de cambios; se puede conocer la fecha, el cambio realizado y el usuario que realizó la modificación del artefacto.
- La herramienta cuenta con una opción que muestra en forma gráfica todas las versiones generadas para un documento, de forma que es más sencillo detectar inconsistencias.
- Permite comparar dos versiones de un documento e identifica las diferencias entre las mismas, resaltando con colores dichas diferencias.

Con el uso de esta herramienta se propuso el siguiente esquema de almacenamiento:

- Creación de una carpeta para cada proyecto
- En la carpeta de cada proyecto se debe crear una subcarpeta para cada etapa del ciclo de vida del desarrollo de *software*, de tal forma que cada artefacto se guarde en la carpeta del ciclo que le corresponde.

Además, se creó una carpeta para almacenar los artefactos que fueron resultado de este proyecto, con el objetivo de mantenerlos a disposición de los integrantes del área de tecnología, para que sean usados en los proyectos que se trabajen.

Algunas ventajas del esquema propuesto son las siguientes:

- Coordinación del trabajo de un equipo de personas evitando que los cambios se traslapen.
- En caso que se hagan modificaciones a los artefactos diseñados, se mantiene el historial de los cambios realizados a los diseños que quedaron definidos como la línea base para el área de tecnología.
- En conclusión, con el esquema propuesto se mantiene la autenticidad e integridad de los artefactos, la disponibilidad de los mismos en la aplicación y control de versiones, todos ellos factores claves para fortalecer la calidad del software en cada etapa de desarrollo.

## CONCLUSIONES

A continuación se presentan las principales conclusiones que se obtuvieron luego de la realización del trabajo de grado en mención :

En primer lugar, cabe resaltar que los estándares IEEE para la ingeniería de *software* ofrecen un conjunto de recomendaciones para diseñar los diferentes artefactos de apoyo al ciclo de vida de desarrollo. Dichos estándares fueron tomados en consideración para diseñar los artefactos que, de acuerdo a los resultados de la encuesta aplicada, debían diseñarse con prioridad por ser de mayor impacto en el proceso de desarrollo.

El área de tecnología del Banco Falabella tiene definidos una serie de procesos para el ciclo de vida del desarrollo de *software*; sin embargo, los resultados obtenidos mediante el proceso de observación directa y la aplicación de la encuesta, permitieron detectar que no se tenían artefactos formalmente establecidos al interior del área y que se carecía de un esquema de almacenamiento para la documentación de los proyectos. Estas falencias han comprometido la calidad y los resultados de los diferentes proyectos que maneja el área.

Se involucró a la jefatura del área de desarrollo de *software* en la evaluación de los diseños definitivos de los artefactos, de forma que los artefactos entregados incluyen las recomendaciones de la IEEE y también son pertinentes con las necesidades del área de tecnología del banco.

Se definió un esquema de almacenamiento para que la documentación se guarde de forma centralizada y pueda ser reutilizada al interior del

área. El repositorio de información se creó con una estructura de carpetas para cada proyecto, de manera que sea más sencillo ubicar los artefactos generados durante el proyecto de desarrollo. La implementación de dicho repositorio se hizo utilizando la herramienta Tortoise SVN, la cual también ofrece la posibilidad de hacer seguimiento a todos los cambios realizados a los documentos en sus diferentes versiones.

## REFERENCIAS

- [1] ISO/IEC and IEEE 26511. (2012). *ISO/IEC/IEEE 26511:2012 Systems and software engineering - Requirements for managers of user documentation*, First., vol. 2012. Geneva, Switzerland: IEEE.
- [2] SEI Software Engineering Institute. (2010). *CMMI para Desarrollo, Versión 1.3*. Massachusetts, USA: SEI.
- [3] ISO/IEC and IEEE 29119-1. (2013). *ISO/IEC/IEEE 29119-1 Software and systems engineering - Software testing - Part 1 Concepts and definitions*.
- [4] ISO/IEC and IEEE 29119-3. (2013). *IEEE 29119-3 Software and systems engineering - Software testing - Part 3 Test documentation*, vol. 2013.
- [5] ISO/IEC and IEEE 29148. (2011). *ISO/IEC/IEEE 29148:2011 Systems and software engineering - Life cycle processes - Requirements engineering*.
- [6] ISO/IEC and IEEE 12207. (2008). *ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes*, Second., vol. 8. Piscataway, USA: IEEE.
- [7] ISO/IEC and IEEE 15289. (2011). *ISO/IEC/IEEE 15289:2011 Systems and software engineering - Content of life cycle information products (documentation)*, First., vol. 2011. Geneva, Switzerland: IEEE.