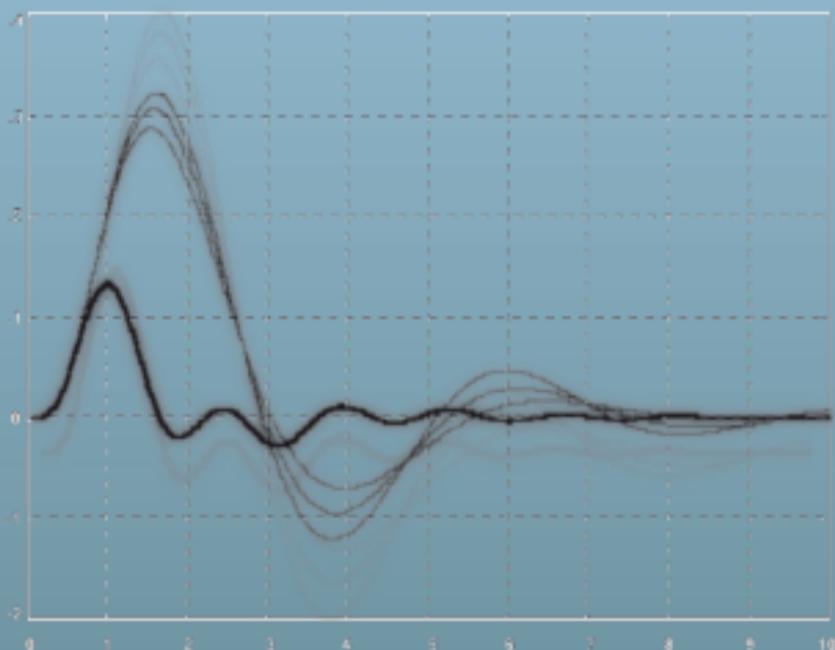


Ingeniería

Revista de la Universidad de Costa Rica
ENERO/DICIEMBRE 2002 • VOLUMEN 12 • Nº 1 y 2



DISEÑO DE UN ESTÁNDAR DE VERIFICACIÓN Y VALIDACIÓN PARA ASEGURAR LA CALIDAD DEL SOFTWARE

Gabriela Salazar Bermúdez
Marcelo Jenkins Coronas

Resumen

Este artículo describe la experiencia obtenida en el proceso de elaboración de un estándar de verificación y validación del software para la Universidad de Costa Rica. El Estándar UCR-61 está basado en los estándares de ingeniería de software de la IEEE. Se explica el contenido del Estándar UCR-61 y la experiencia obtenida en el proceso de elaboración del mismo. Este artículo puede ser de utilidad a organizaciones interesadas en mejorar la calidad del software que producen.

Palabras clave: Ingeniería de software, aseguramiento de la calidad del software, estándares de calidad.

Abstract

This paper describes the experience obtained in the process of elaborating a standard of software verification and validation at the University of Costa Rica. The UCR-61 standard is based on the IEEE software standards. It is explained as well as the experience obtained during the elaboration process. This paper could be useful at organizations interested on improving their software quality.

Keywords: software engineering, securing software quality, quality standards.

1.- INTRODUCCIÓN

Uno de los componentes claves de todo proceso de desarrollo de *software* son las actividades que se llevan a cabo para asegurar la calidad del software que se produce. Este conjunto de actividades y sus productos se conoce como el *Aseguramiento de la Calidad del Software* (ACS). Un componente importante del ACS son las actividades de verificación y validación (V&V) del *software* que se realizan durante las diferentes fases que componen el ciclo de desarrollo de los sistemas. Los objetivos principales del proceso de V&V son dos:

1. Verificar que los productos obtenidos en cada fase del ciclo de vida:
 - Cumplan con los requerimientos de la fase anterior.
 - Satisfagan los estándares, prácticas y convenciones de la fase actual.

- Establezcan las bases apropiadas para iniciar la siguiente fase del ciclo de vida.
2. Validar que el producto final cumpla con los requerimientos del *software* establecidos.

Como parte fundamental del Plan de ACS de la Universidad de Costa Rica (UCR), se elaboró el Estándar UCR-61, Plan de Verificación y Validación del *Software* para la Universidad de Costa Rica, que se describe en detalle en este artículo. Este estándar define un conjunto de tareas y procedimientos que se deben llevar a cabo para verificar y validar tanto los productos de *software* que se desarrollan internamente en la institución como los productos que se adquieren por contratación externa.

Está basado en dos estándares de ingeniería de *software* de la IEEE: Estándar para Planes de Verificación y Validación del *Software* (IEEE Std. 1012-1987) y Estándar para Revisiones y

Auditorías del *Software* (IEEE Std. 1028-1987), así como en los estándares ISO 9001 e ISO 9000-3, esto con el fin de actualizar las metodologías de desarrollo utilizadas actualmente en la UCR de acuerdo con las exigencias internacionales.

El sistema de calidad propuesto en este estándar está elaborado tomando en cuenta las condiciones de las diferentes oficinas de UCR que desarrollan y adquieren *software*. El estándar UCR-61 prescribe una serie de controles y métodos para detectar y corregir las anomalías que puedan presentarse durante el ciclo de vida del software. Controles como los establecidos en ese documento permitirán mejorar el proceso de desarrollo y así lograr un mejor nivel de calidad en los productos finales y consecuentemente una mayor satisfacción del usuario.

El artículo presenta cinco apartados más. El apartado número 2 describe la estructura organizacional de la UCR y su complejidad; el apartado 3 presenta una descripción de los problemas principales en las metodologías de desarrollo que enfrenta actualmente el personal técnico de la organización; en el apartado 4 se indica cómo el estándar UCR-61 es un componente importante del plan de calidad de la UCR; seguidamente, en el apartado 5 se describe en detalle el formato y contenido del Estándar UCR-61; finalmente, en el apartado 6 presentamos las conclusiones del proceso de elaboración del estándar.

2. LA ORGANIZACIÓN

La Universidad de Costa Rica es una institución de educación superior pública con un total de 25.000 estudiantes, 70 unidades académicas, 3.500 profesores y 2.800 funcionarios administrativos. Para apoyar las actividades de la comunidad universitaria esta institución cuenta con una estructura administrativa compleja que requiere un uso extenso de los sistemas de información para la gestión académica.

La UCR es una institución donde coexisten muchos recursos, productos y herramientas, razón por la cual es necesario un proceso de desarrollo,

producción y mantenimiento sistematizada con sus correspondientes controles. Para coadyuvar los procesos administrativos, la UCR dispone de dos tipos de sistemas de información: los institucionales y las aplicaciones.

Los sistemas institucionales se caracterizan por su gran complejidad y tamaño. En su mayoría han sido legados. La documentación, en algunos casos es bastante incompleta o escasa, por esa razón, el personal técnico tiene serios problemas para llevar a cabo las tareas de mantenimiento.

Debido a que actualmente muchos de estos sistemas se están convirtiendo en nuevas plataforma conviene destacar la importancia del Estándar UCR-61 para que la Universidad de Costa Rica pueda enfrentar este nuevo esfuerzo de desarrollo.

El otro tipo de sistemas de información consiste en las llamadas *aplicaciones* o sistemas pequeños. Se utilizan para satisfacer necesidades específicas de las unidades académicas y oficinas administrativas, en alguna medida podrían interactuar con los sistemas *institucionales*.

La mayoría de los *software* que requiere la Universidad de Costa Rica es desarrollado por el personal técnico que labora en las diferentes oficinas administrativas. Cada oficina posee un encargado del desarrollo, producción o mantenimiento de los sistemas computacionales. Sin embargo, en la actualidad debido a la escasez de recursos, la Universidad ha recurrido a la contratación externa.

3. DESCRIPCIÓN DEL PROBLEMA

Previo a la elaboración del estándar UCR-61 se realizó una investigación de las metodologías de desarrollo utilizadas actualmente por el personal informático de la institución. Algunos de los problemas encontrados son los siguientes:

1. El tipo de revisión que se aplica no se adapta a lo solicitado por las normas internacionales. El único tipo de revisión es al final del desarrollo en la fase de pruebas.

2. No existe ningún control que verifique y asegure que los sistemas están siendo documentados adecuadamente, lo que dificulta las fases de producción y mantenimiento.
3. Para proyectos de *software* contratados externamente, la Universidad no cuenta con procedimientos ni estándares que permitan monitorear el proceso de desarrollo ni verificar la calidad de los productos entregados.
4. No existe ningún estándar o procedimiento escrito que guíe esta etapa e indique el contenido mínimo de un plan de proyecto. Este problema dificulta el *monitoreo* del proyecto de *software* durante su desarrollo.

Por otro lado, la Contraloría General de la República, el órgano contralor de más alto rango en el país, en el documento *Manual sobre Normas Técnicas de Control Interno Relativas a los Sistemas de Información Computadorizados [CGR95]*, emitido en noviembre de 1995, exige a todas las entidades y órganos que conforman la hacienda pública, el acatamiento obligatorio de esta normativa.

En esta normativa se indica que para un adecuado control interno del desarrollo de los sistemas computacionales, las instituciones deben disponer de manera indispensable de lo siguiente:

1. Un manual de estándares que incluya los procedimientos y estándares necesarios para guiar las etapas del ciclo de vida de los sistemas computacionales.
2. Documentación completa, adecuada y actualizada que controle el desarrollo y el mantenimiento del *software*.
3. Establecimiento de controles durante las fases del ciclo de vida que garanticen el cumplimiento de los estándares y procedimientos establecidos en el manual de estándares.
4. Aplicación de revisiones y auditorías en los sistemas en desarrollo y en mantenimiento.

En resumen, en la actualidad la Universidad de Costa Rica no cumple con las exigencias de la Contraloría General de la República, no hay un ambiente estable que permita desarrollar y mantener los *softwares* adecuadamente, y el éxito de los proyectos depende del esfuerzo individual y no de un trabajo planificado y controlado.

4. EL PLAN DE CALIDAD

Para mejorar el proceso de desarrollo, se definió un Plan de ACS basado en el estándar de la IEEE (Plan de Aseguramiento de Calidad del *Software* IEEE Std. 730-1989) y el ISO 9001, y se adaptaron una serie de estándares que servirán como complemento al plan para corregir las deficiencias encontradas en las metodologías de desarrollo.

El Plan de ACS contempla aspectos tales como:

- La estructura organizacional del Grupo ACS que se encargará de controlar la calidad en la UCR, incluyendo sus tareas y responsabilidades.
- La documentación necesaria para llevar a cabo esas tareas de ACS.
- Las referencias a los estándares y convenciones que apoyarán el aseguramiento de la calidad en los sistemas de información.
- Las metodologías y herramientas necesarias para llevar a cabo las tareas.

En la Figura 1. se muestra como el Estándar UCR-61 es un componente fundamental del plan de ACS. Como tal ayudará a corregir gran parte de las deficiencias encontradas en las metodologías de desarrollo descritas en la Sección 3. El plan de ACS está basado en cuatro estándares principales.

Los otros cuatro estándares mostrados en la Figura 1. ayudarán a solucionar las áreas problemáticas del desarrollo, particularmente, lo que se refiere a planificación y control de proyectos. Por razones de espacio, en este artículo solamente se explicará el Estándar UCR-61, Plan de Verificación y Validación del *Software*.

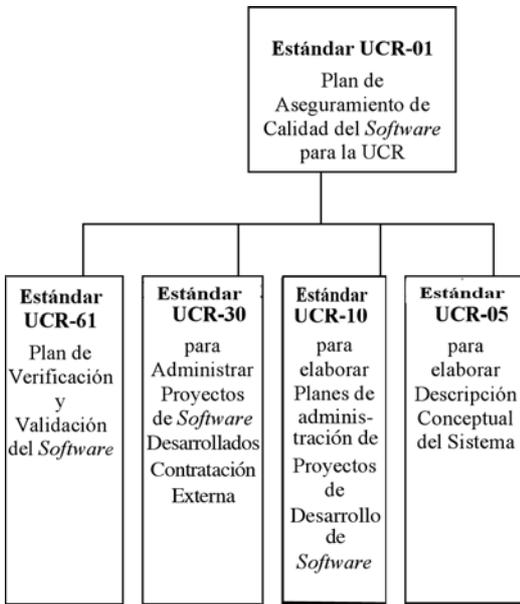


Figura 1. Estructura del Plan de ACS de la UCR

5. ESTRUCTURA DEL ESTÁNDAR UCR-61 DE VERIFICACIÓN Y VALIDACIÓN

La Figura 2. presenta el formato del Estándar UCR-61, Plan de Verificación y Validación del *Software*.

5.1 Propósito

Describe brevemente el propósito y una justificación del estándar.

5.2 Referencias

Enumera todas las citas a las que hace referencias el estándar.

5.3 Definiciones y Abreviaturas

Enumera los términos y abreviaturas más utilizadas por el estándar que requieren una definición.

5.4 Descripción General

El estándar contiene 8 secciones y un anexo que se explican a continuación.

5.4.1 Estructura organizacional

La Figura 3. describe la estructura organizacional necesaria para realizar las actividades de verificación y validación del *software*. Con el fin de aclarar este punto se representa gráficamente la estructura y seguidamente se describen las funciones y responsabilidades de cada uno de los elementos mostrados en la estructura.

El Comité de Desarrollo está conformado por un Director de Proyecto, un equipo técnico de ingenieros de *software*, y un grupo de usuarios. El Grupo de ACS realiza actividades de verificación y validación prescritas en este estándar. Además, la Oficina de Contraloría Universitaria realiza auditorías periódicas para verificar el cumplimiento del estándar.

1. Propósito
2. Referencias
3. Definiciones y Abreviaturas
4. Descripción General
 - 4.1 Estructura Organizacional
 - 4.2 Cronograma
 - 4.3 Recursos
5. Verificación y validación (V&V) durante las fases del ciclo de vida
 - 5.1 V&V del análisis preliminar
 - 5.1.1 Tareas de V&V
 - 5.1.2 Metodologías
 - 5.1.3 Reportes de salida
 - 5.2 V&V de la planificación
 - 5.2.1 Tareas de V&V
 - 5.2.2 Metodologías
 - 5.2.3 Reportes de Salida
 - ...
 - 5.8 V&V del Mantenimiento
 - 5.8.1 Tareas de V&V
 - 5.8.2 Metodologías
 - 5.8.3 Reportes de Salida
 - 5.9 Proceso de Auditorías
6. Metodologías para realizar las tareas de V&V
 - 6.1 Revisión Administrativa
 - 6.2 Revisión Técnica
 - 6.3 Caminatas
 - 6.4 Auditorías
7. Reportes de salida
 - Reporte Técnico (R-1)
 - Reporte Administrativo (R-2)
 - Informe de Auditoría (R-3)
8. Reporte de problema y acciones correctivas
 - Anexo 1 Listas de Verificación

Figura 2. Estructura del Estándar UCR-61 Plan de Verificación y Validación de *Software*

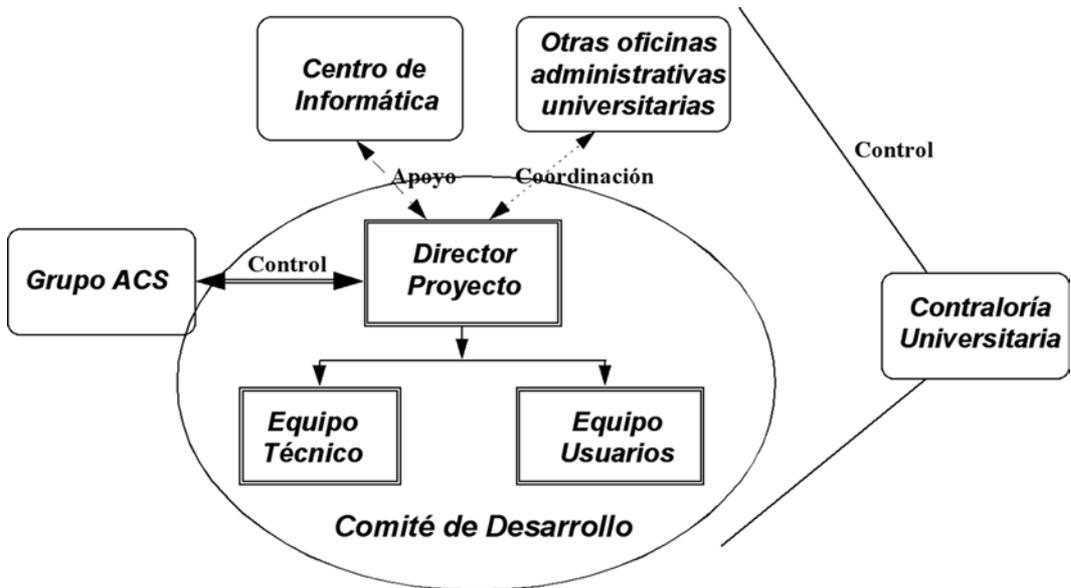


Figura 3. Estructura organizacional encargada de la V&V.

5.2 Cronograma

El Estándar UCR-61 le exige al equipo de desarrollo que incluya en su cronograma el tiempo y los recursos necesarios para llevar a cabo cada una de las tareas de V&V que se prescriben en este estándar.

Por ello, el ciclo de vida se representa como lo muestra la Figura 4. Cada fase del ciclo de vida se representa con una caja dividida en dos triángulos. El superior se refiere a la fase del ciclo de vida y el inferior a las metodologías que se utilizarán para realizar las tareas de V&V correspondientes a la fase. Además, se presentan los productos generados en cada fase; por ejemplo, la fase de análisis produce el documento Especificación de Requerimientos, que a su vez debe pasar por una revisión técnica y una revisión administrativa antes del inicio de la fase de diseño.

De esta manera, cada fase del ciclo de vida está conformada por dos etapas principales, representadas mediante dos triángulos en forma separada

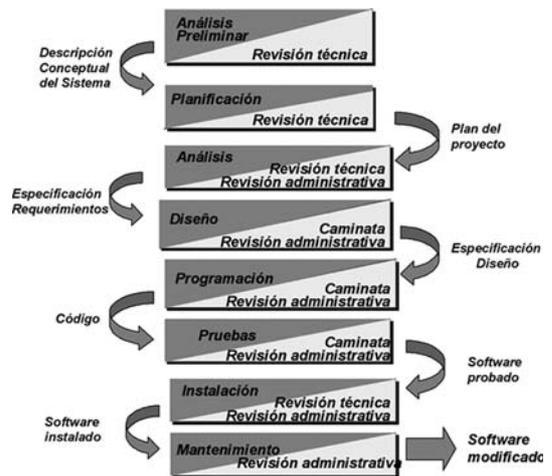


Figura 4. Modelo del ciclo de vida de los sistemas.

como se muestra en la Figura 5. El triángulo superior representa la etapa de desarrollo con sus entradas y salidas y el triángulo inferior las tareas de V&V exigidas por el estándar con sus respectivas salidas.

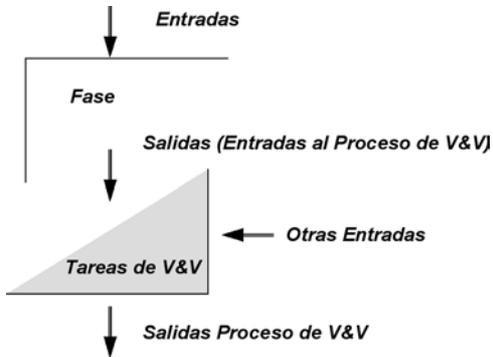


Figura 5. Representación de las fases de desarrollo y las tareas de V&V

Las entradas a las tareas de V&V (del triángulo inferior) corresponde al elemento de *software* al que se le aplica el proceso de V&V, producto de la actividad de desarrollo. Además, se requieren otros documentos de referencia como los estándares de la UCR y las listas de verificación, para evaluar los elementos de *software* de forma correcta. Las salidas del triángulo inferior corresponden a los reportes que se deben obtener en el proceso de V&V.

5.4.3 Recursos

Los recursos necesarios para aplicar el estándar UCR-61 son:

- **Personal que realizará las tareas de V&V:** El plan describe el perfil del personal necesario para realizar cada una de estas tareas.
- **Capacitación del personal en las tareas de V&V:** Todo el personal de desarrollo, aún el nuevo personal que ingrese a laborar en la Universidad de Costa Rica, debe recibir capacitación en las tareas de V&V.
- **Herramientas:** Se debe disponer de herramientas que faciliten las tareas de elaboración de informes y control de documentos. Algunas de ellas son: procesador de palabras, hojas electrónicas y herramientas de planificación y control de proyectos.

5.5 Verificación y Validación durante las fases del ciclo de vida

En esta sección se provee el detalle de las actividades de V&V que se deben llevar a cabo en cada una de las fases del ciclo de vida de los sistemas. Tal y como se muestra en la Figura 4., las fases que se describen en el Estándar son: Análisis Preliminar, Planificación, Análisis, Diseño, Programación, Pruebas, Instalación y Mantenimiento.

Para cada fase se presenta una descripción del proceso de V&V que incluye los siguientes aspectos:

- Diagrama de la fase
- Tareas de V&V que se deben realizar en cada fase
- Metodologías para efectuar esas tareas
- Reportes producto de las actividades de V&V.

Aunque hemos utilizado como modelo el ciclo de vida clásico conocido como “cascada”, es importante indicar que ello no obliga a desarrollar estrictamente sistemas de forma secuencial, que se utilizan por motivos de presentación. Como modelo de desarrollo podría utilizarse cualquier otro ciclo de vida, por ejemplo, el prototipaje o la espiral.

El Estándar UCR-61 prescribe actividades de V&V durante las ocho fases de desarrollo descritas en la figura 4. Sin embargo, por razones de espacio nos referiremos solamente a las actividades de la fase de análisis.

V&V del Análisis

El objetivo principal de esta fase es elaborar el documento de Especificación de Requerimientos (ER). Este documento es clave en etapas posteriores del desarrollo, incluyendo el mantenimiento.

En la Figura 6. se puede observar como para la preparación del ER se requiere de una descripción conceptual del sistema y un plan del proyecto, ambos productos de fases anteriores.

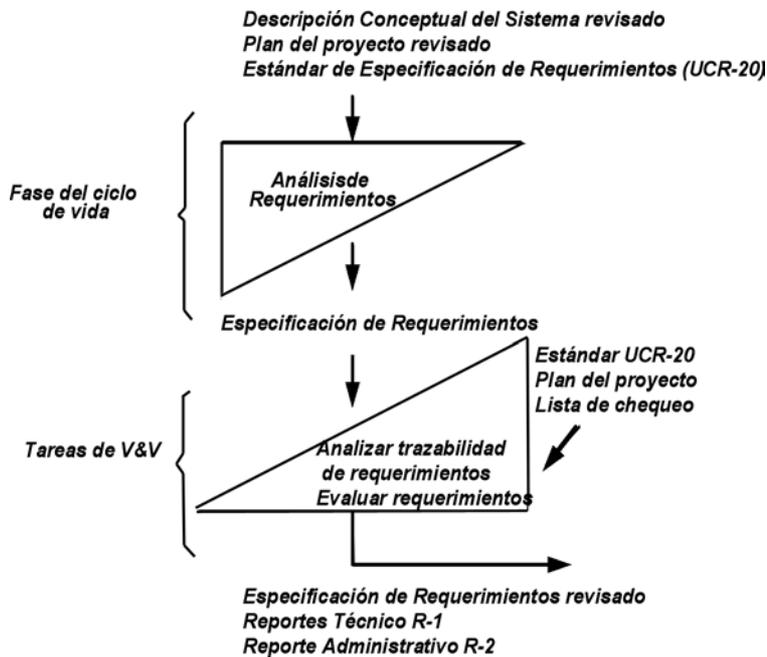


Figura 6. Tareas de V&V de la fase de análisis de requerimientos

La descripción conceptual del sistema sirve como base para realizar un análisis más exhaustivo de los requerimientos definidos en este documento. Por su parte, el plan del proyecto sirve para controlar que el avance de esta fase se realice de acuerdo con lo planificado.

El triángulo inferior de la figura 6 indica las tareas de V&V y los reportes esperados. Se exigen dos tareas de verificación:

1. **Análisis de trazabilidad de los requerimientos.** Es importante que todos los requerimientos descritos en el ER sean revisados para asegurar que todos ellos tengan un origen claro y justificable. Existen dos tipos de referencias recomendadas:

- **Hacia atrás:** Cada requerimiento debe referenciar su origen en anteriores documentaciones. Por ejemplo: el origen de un determinado requerimiento debería poder ubicarse en la descripción conceptual del sistema.

- **Hacia adelante:** Cada requerimiento en el ER tiene un único nombre o número de referencia que se utiliza en cualquier documento derivado del ER.

2. **Evaluación de los requerimientos.** Para cada requerimiento se debe evaluar cada una de las siguientes características:

- **Exactitud:** Evaluar que cada requerimiento tenga una única interpretación para evitar ambigüedades.
- **Completitud:** Verificar que el ER cumpla con el estándar descrito para este fin. Un requerimiento completo debe: 1) tener un objetivo concreto de acuerdo con la funcionalidad, el rendimiento, las restricciones de diseño particulares, la seguridad, la privacidad, la facilidad para interactuar con otros sistemas, etc. 2) Todas las respuestas que dé el sistema deben estar definidas ya sea como valores válidos o inválidos. Cualquier tipo de ayuda como tablas, figuras y diagramas debe estar etiquetada.

- **Certeza:** Verificar que cada requerimiento especificado sea indispensable para satisfacer las necesidades del usuario.
- **Legibilidad:** Verificar la legibilidad de la especificación del requerimiento de manera que esté descrito de una forma sencilla y clara.
- **Posibilidad de probarlo:** Cada requerimiento debe tener la posibilidad de someterse a prueba a través de las documentaciones anteriores u otro medio.

5.5.2 Metodologías

Es necesario aplicar una revisión técnica con el fin de verificar que el producto generado en esta fase satisfaga las especificaciones técnicas y se encuentre listo para comenzar la siguiente fase de desarrollo. Para llevar a cabo esta revisión se requiere corroborar el documento Especificación de Requerimientos contra el Estándar UCR-20 Especificación de Requerimientos (ver figura 6), para verificar que se cumplan los lineamientos establecidos en este estándar.

Posteriormente, se debería llevar a cabo una *Revisión Administrativa* con el objetivo de controlar el estado de esta fase y el avance del proyecto con respecto a lo planificado, para lo cual se requerirá del plan del proyecto.

5.5.3 Reportes de Salida

Para indicar los problemas detectados a través de la tarea de verificación y validación son necesarios los siguientes reportes: Reporte Técnico R-1 y Reporte Administrativo R-2, los cuales se encuentran descritos en la sección 7 de este artículo.

5.6 Metodologías para realizar las tareas de V&V

5.6.1 Revisión Técnica

Se usa para identificar y solucionar asuntos técnicos relativos a la especificación o diseño de los

elementos de software. Los objetivos de una Revisión Técnica son:

1. Verificar que el elemento de software a revisar cumpla las especificaciones técnicas. Un elemento de *software* es cualquier documento, por ejemplo, el plan del proyecto, la especificación de los requerimientos, la especificación del diseño, la documentación de pruebas, la documentación del usuario o cualquier otro documento producido durante el desarrollo.
2. Ofrecer recomendaciones y/o alternativas para la mejora de los elementos del *software*.
3. Informar sobre aspectos técnicos que el elemento del *software* debe cumplir.

Los participantes a estas reuniones, entre tres y cinco personas y al menos un miembro del Grupo ACS, se seleccionan desde la planificación del proyecto. Las reuniones se conforman con el personal técnico del equipo de desarrollo y otras personas afines.

Como mínimo el equipo de revisión debe contar con la siguiente información:

- Enunciado de objetivos de la revisión.
- Especificación del elemento del *software* que se evaluará.
- Elemento del *software* a revisar.
- Planes, estándares o guías que servirán para revisar el elemento del *software*.
- Listas de verificación para asegurar que el elemento del *software* cumpla con los requerimientos mínimos.

La necesidad de esta revisión debe quedar definida en los documentos de planificación del proyecto, las autoridades competentes pueden solicitarla. El elemento del *software* a revisar quedará disponible y cumplirá con los estándares

establecidos por el proyecto en forma y contenido. Toda la información requerida debe estar disponible.

Las reuniones para realizar este tipo de revisión se llevan a cabo cuando así lo establezca un

calendario de fases o cuando se determine adecuada la revisión del elemento del *software* por su disponibilidad.

El procedimiento para llevar a cabo una revisión técnica se describe en el Cuadro 1.

Cuadro 1. Procedimiento para llevar a cabo una revisión técnica.

Paso	Descripción
1. Planificación	Líder verifica los criterios de entrada, establece el calendario, identifica y convoca al equipo de revisores y reparte los materiales.
2. Vistazo (opcional)	Si el líder lo considera necesario, convoca a los participantes a una reunión breve en una fecha y hora específica.
3. Preparación	Cada revisor estudia el material en forma individual.
4. Reunión	<p>Durante la revisión el equipo realiza lo siguiente:</p> <ol style="list-style-type: none"> 1. Revisar el elemento de software contra estándares y guías. 2. Discutir las alternativas o recomendaciones en pro del mejoramiento del elemento de software. 3. Asignar responsabilidades para la solución de problemas encontrados en el elemento de software. Esta responsabilidad puede asignarse al autor o a otros integrantes del proyecto. (Debe hacerse por medio del Director del Proyecto). 4. Documentar los aspectos técnicos, las recomendaciones y al responsable de resolverlos. 5. Cuando los defectos son numerosos o críticos, el líder debe recomendar un proceso de revisión adicional (re-trabajo).
5. Re-trabajo	Autor revisa y corrige el producto atendiendo la lista de recomendaciones.
6. Seguimiento	Líder comprueba la solución de los problemas realizados por el autor y otros responsables, además de determinar si se necesita realizar otra revisión.

La revisión técnica concluye cuando se han resuelto todos los asuntos y defectos detectados en la reunión y se ha generado el reporte técnico R-1.

La salida de la revisión técnica es un reporte técnico R-1 descrito en la Figura 7. Este reporte es un documento auditable. Este documento y los materiales de entrada a la revisión deben mantenerse junto con la información del proyecto y los resultados, además, toda la información tendrá que divulgarse entre los miembros del equipo de desarrollo.

5.7 Reportes de salida

De acuerdo con el formato presentado en la figura 3, este apartado corresponde al punto 7. Este apartado contiene los diferentes formularios

donde se incluirían y especificarían los resultados de las revisiones (descritas en el punto anterior). Se presentan tres formularios: reporte técnico (R-1), reporte administrativo (R-2) e informe de auditoría (R-3).

A manera de ejemplo, la Figura 7. presenta el formulario reporte técnico R-1. Este formulario recoge los resultados de una revisión técnica de algún producto del *software*. Se recogen los datos de los productos que se revisan así como la información correspondiente sobre las personas que participaron en la revisión. Además, se recoge la lista de los defectos encontrados, el lugar, la descripción del defecto, el tipo, la clase y la severidad del mismo. El objetivo es que el reporte R-1 sirva para controlar la corrección de los defectos encontrados durante las revisiones, así como para levantar métricas de calidad y productividad de los procesos de aseguramiento de calidad.

No.	Ubicación	Descripción	Tipo	Clase	Severidad

Tipo defecto: **DC**=Documentación **FA**=Factibilidad **FN**=Funcionalidad **IF**=Interface
 ST=Estándar **RE**=Requerimiento **SN**=Sintaxis **OT**=Otros

Clase: **A**=Ausente **E**=Incorrecto **I**=Incompleto **B**=Ambigüo **X**=Extra

Severidad: **A**=Alta **B**=Baja

8. Acciones requeridas

No.	Acción requerida	Responsable

9. Disposición (Marcar)

Aceptado Condicionado Re-evaluar

Figura 7. Reporte Técnico R-1

6. CONCLUSIONES

El diseño de estándares de ACS para una organización es un proceso complejo que requiere paciencia y especialización en la materia, así como una visión de las necesidades y capacidades particulares de la organización. Los estándares de la IEEE (Std. 1012-1987 y Std. 1028-1987) son guías muy generales que se utilizaron como base para definir nuestros propios estándares. El estándar IEEE Std. 1012-1987 Planes de Verificación y Validación del *Software* no es claro en cuanto a los métodos para realizar las tareas de V&V. Esto nos obligó a hacer uso de otros estándares para seleccionar estas metodologías.

El establecer puntos de control a través de las diferentes fase del ciclo de vida requiere conocer las deficiencias en las metodologías de desarrollo y la complejidad de los proyectos en la organización. Posiblemente, si los proyectos son de poca complejidad, el establecer un punto de control en cada una de las fases puede convertir el estándar de V&V en una herramienta poco práctica y difícil de implementar.

El proceso de elaboración de estándares es un proceso complejo y tedioso que requiere un trabajo de adaptación de estándares internacionales a las necesidades y recursos de la organización. El producto final en la adaptación de estos estándares debe ser un instrumento de apoyo, práctico y fácil de entender, de manera que todo el personal tenga claras sus tareas y responsabilidades y preferiblemente, lo utilice por conveniencia y no por simple obligación.

La efectividad del Estándar UCR-61 fue evaluada por el mismo personal de la organización. Para esta labor se definió un comité de revisión y aprobación conformado por cinco miembros, todos con fuerte liderazgo en la institución, con conocimiento y experiencia en el área informática y capaces de promover políticas de calidad en la organización. Durante el proceso de revisión y aceptación se les distribuyó un borrador del

estándar, ellos lo revisaron y recomendaron correcciones. Estas se discutieron y finalmente, muchas se incorporaron en el documento.

El uso del Estándar UCR-61 Plan de Verificación y Validación del *Software* logrará:

- Establecer controles a través de las diferentes fases del ciclo de vida, lo cual permite asegurar la calidad de los sistemas computacionales.
- Asegurar que la documentación elaborada en cada fase del desarrollo se realice adecuada y oportunamente, lo que facilita, posteriormente, la etapa de mantenimiento.
- Permitir *monitorear* tanto los proyectos desarrollados internamente como los desarrollados a través de contratación externa.
- Ordenar el proceso de desarrollo del *software* al obligar a todo el personal de desarrollo a cumplir sistemáticamente con los lineamientos establecidos en este estándar.
- Al estar sistematizado el proceso de desarrollo, mejorar la comunicación entre el personal técnico, pues se promueve la transferencia de personal de un proyecto a otro o de una oficina a otra y se facilita la capacitación.

Para la UCR es indispensable el Plan de Verificación y Validación del *Software*, también es una disposición de la Contraloría General de la República, debido a que el proceso de desarrollo actual no es aceptable, no hay un ambiente estable para desarrollar y mantener el *software* y el rendimiento depende del esfuerzo individual.

En particular, el Estándar UCR-61 permite cumplir con dos áreas claves del Modelo Capacidad Madurez del SEI [PCCW95]. Las dos áreas claves que se cumplen son: el seguimiento y vigilancia del *Software* y el aseguramiento de calidad. La primera área clave les exige a las empresas que continuamente apliquen revisiones para controlar los logros de los proyectos, así como la definición

de acciones correctivas para solucionar cualquier desvío, mientras que la segunda área clave establece revisiones y auditorías para verificar el cumplimiento de los estándares.

7. BIBLIOGRAFÍA

1. Contraloría General de la República. *Manual sobre normas técnicas de control interno relativas a los sistemas de información computadorizados*. Dirección General de Planificación Interna y Evaluación de Sistemas, noviembre 1995.
2. Ebenau R.G.; Strauss S.H.; *Software Inspection Process*. McGraw Hill, 1994.
3. Humphrey W.; *A discipline for Software Engineering*. Addison-Wesley, 1995.
4. Jenkins. M.; *Adopting Development Standards to Achieve Process Improvement*. Proceedings Sixth International Conference on Software Quality, Montreal, Canada, 1996.
5. Paulk M., Curtis B., Chrissis M.B.; Weber C.V.. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, 1995.
6. Pereira C.; *Guía general para el desarrollo de sistemas de información*. Centro de Informática, Universidad de Costa Rica. Julio 1991.
7. Pereira C. *Guía para la documentación de los sistemas de información*. Centro de Informática, Universidad de Costa Rica. Julio 1991.
8. ANSI/ISO/IEEE Std. 1012-1986, *IEEE Standard for Software Verification and Validation Plans*.
9. ANSI/ISO/IEEE Std. 1028-1988, *IEEE Standard for Software Reviews and Audits*.
10. ISO. ISO 9001:1994 *Quality Systems--Model for Quality Assurance in Design, Development, Production, Installation, and Servicing*. ASQC Press, 1994.
11. ISO. ISO 9000-3 *Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software*. ASQC Press, 1991.
12. Paulk M., Curtis B., Chrissis M.B.; Weber C.V.. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, 1995.

SOBRE LOS AUTORES

M. Sc. Gabriela Salazar Bermúdez

Profesora Adjunta de la Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica.
Tel: 207-5582 Fax: 207-5527
Correo Electrónico: gsalazar@racsa.co.cr

Dr. Marcelo Jenkins Coronas

Profesor Catedrático de la Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica.
Tel: 207-5492 Fax: 207-5527
Correo Electrónico: mjenkins@cariari.ucr.ac.cr

