

Elección de una Metodología de Desarrollo a partir de las Ventajas de una Metodología Ágil y un Modelo Robusto como CMMI-DEV 1.3

Choosing a Development Methodology from the Advantages of an Agile Methodology and a Robust Model like CMMI-DEV 1.3

*Patty Pedroza Barrios**

RESUMEN

La competitividad de las organizaciones de desarrollo de software exigen el conocimiento de las diferentes metodologías existentes en el mercado, para así adoptar la que conlleve al éxito en la culminación del proyecto teniendo en cuenta las condiciones de calidad propuestas para un producto de software. Por esta razón, este artículo hace un recorrido por los avances en la gestión de proyectos de desarrollo de software, utilizando una metodología ágil y los que han aplicado el modelo CMMI-DEV, tanto a nivel nacional como internacional.

Palabras clave: SCRUM, CMMI-Dev, Software, Método ágil, Desarrollo.

ABSTRACT

Competitiveness of software development organizations requires knowledge of the various existing methods on the market in order to adopt the one that lead to successful completion of the project taking into account the conditions of quality proposals for a software product. For this reason, this article talks about the advances in the management of software development projects using an agile methodology and those that have applied the CMMI-DEV model, both nationally and internationally.

Key words: SCRUM, CMMI-Dev, Software, Agile methodology, Development.

** Docente del programa de Ingeniería de Sistemas de la Universidad Libre Seccional Barranquilla.
ppedroza@unilibrebaq.edu.co*

I. INTRODUCCIÓN

La tecnología avanza de manera muy rápida y herramientas y plataformas que fueron adecuadas para un proyecto pueden ya no serlo para otro con requerimientos similares, por el surgimiento de herramientas nuevas o mejores, que beneficiarían en costo o tiempo a las empresas desarrolladoras. De igual forma, parece suceder con las metodologías de desarrollo de un producto de software, desde el punto de vista de directivas para su aplicación. Hasta hace unas décadas, el proceso de desarrollo de productos de software, hacía énfasis en el control de los procesos mediante una rigurosa definición de roles, actividades, recursos y artefactos, incluyendo los modelos de diseño, debidamente detallados y documentados. Esta manera “tradicional” para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño, en lo que respecta al tiempo y los recursos asignados, donde por lo general se exige un alto grado de cumplimiento y valga la redundancia, rigurosidad de las actividades definidas en el proceso. Algunas veces el equipo desarrollador se enfrenta a un entorno del sistema muy cambiante, en donde los plazos de entrega son cortos, por ende los tiempos de desarrollo también se recortan, pero se quiere mantener y proveer de productos de una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo sacrifican las buenas prácticas de la ingeniería de software, asumiendo el riesgo que ello puede conllevar. Bajo estas condiciones, surgen las metodologías ágiles como una posible respuesta para llenar ese vacío metodológico. Las empresas desarrolladoras de software, reconocen cada vez más la necesidad de la adopción de herramientas de ingeniería de software, procesos, metodologías y arquitecturas, pero reconocerlo no ha sido suficiente.

El informe CHAOS, que elabora el Standish Group, es el informe más famoso sobre el éxito y fracaso de los proyectos en el sector de las tecnologías de la información (TI), suele realizarse cada dos años. Según este, CHAOS Report 2013 [1], la cifra de proyectos exitosos ha aumentado solo del 35 al 39 %, desde el anterior informe. Mostrando como proyectos exitosos aquellos que fueron entregados a tiempo, dentro del presupuesto y con las características y funciones necesarias. La gran pregunta es entonces ¿En qué se está fallando?, se trabajan tres factores que han sido incidentes en la justificación del fracaso de los proyectos. Un primer factor es que algunas organizaciones tratan de realizar un buen trabajo de desarrollo, sin contar con procesos bien definidos; esto se ha convertido en uno de los principales inconvenientes, debido a que el proceso de desarrollo debe ser contemplado como un proceso de negocio que tiene que ser gestionado, ser eficiente, evolutivo, rápido y altamente adaptable a las condiciones organizacionales. Como un segundo factor, muchas de estas organizaciones de desarrollo de software aún no han implementado un modelo o metodología que gestione no solo el proceso de desarrollo de software sino a toda la organización, esto se debe a factores como: el desconocimiento del tema, temor a invertir en la implementación de una metodología de gestión del proceso de desarrollo de software al no ver el retorno de la inversión a corto plazo o simplemente no han notado la importancia de valorizar sus proyectos de desarrollo en función de convertirse estos en aporte a los logros de las metas del negocio y como tercer factor, la utilidad y eficacia de algunas prácticas, técnicas, metodolo-

gías y modelos varían entre unas organizaciones y otras, o unos proyectos u otros [2]. A nivel general, los factores planteados anteriormente, se han visto involucrados en los fracasos de proyectos de software, pero también se debe dejar claro que no son el motivo principal, y para algunas organizaciones de desarrollo, pueden ser útiles estas prácticas, mientras que en otras influyen de manera negativa en el logro de sus objetivos. Así, Turner y Jain [3] afirman que las organizaciones que usan métodos ágiles corren el riesgo de hacer demasiado énfasis en el conocimiento tácito y poco en la comunicación formal dentro de un equipo de desarrollo de software, esto depende, en muchos casos, de la experiencia de las personas que conforman el equipo, así como de su capacidad de comunicación de la información del proyecto entre los diferentes miembros. Sin embargo, aunque las metodologías ágiles enfatizan en las comunicaciones informales continuas y a la reacción en los cambios, con la finalidad de ser más adaptativos, poseen comunicación formal como: código fuente, casos de pruebas, y una mínima cantidad de documentación, pero en la mayoría de los casos esta es deficiente, y es en este punto en donde la aplicación de buenas prácticas como las de CMMI-DEV aportan al manejo del proceso de desarrollo en los ámbitos de anteproyecto, proyecto y liberación del producto y transforman las debilidades de la informalidad de metodologías ágiles como SCRUM en fortalezas y nuevas oportunidades.

II. METODOLOGÍAS DE GESTIÓN DESARROLLOS ÁGILES

Las metodologías de desarrollo ágil, a pesar de haber evolucionado a mediados de la década de 1990, como parte de una reacción contra los métodos muy estructurados y estrictos, es en la actualidad en donde su uso se ha proliferado. En el año 2001, se reunieron críticos de los modelos de mejora del desarrollo de software basados en procesos, convocados por Kent Beck, para tratar sobre técnicas y procesos para desarrollar software. En la reunión se acuñó el término Métodos Ágiles para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales a las que consideraban excesivamente pesadas y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. Estos críticos se agruparon formando lo que se conoce hoy como Alianza Ágil, que promueve el desarrollo ágil de aplicaciones. Es de esta forma como actualmente cada vez más empresas de desarrollo de software le apuestan al uso de estas metodologías para el desarrollo de sus productos y ha sido así como su popularidad ha ido en aumento.

Los métodos ágiles fueron oficialmente iniciados con el Manifiesto Agile en el 2001, según el cual se valora a:

- Los **individuos y su interacción**, por encima de los procesos y las herramientas.
- El **software que funciona**, por encima de la documentación exhaustiva.
- La **colaboración con el cliente**, por encima de la negociación contractual.
- La **respuesta al cambio**, por encima del seguimiento de un plan.

“Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.”

Algunos métodos ágiles realizan mayor énfasis en la descripción de prácticas y técnicas como Extreme Programming-XP (2003) (2004), otros en la gestión de las actividades como SCRUM (2001) y otros en la Integración Continua (2007), que es una práctica propuesta por XP.

III. CONTEXTO ACTUAL

A. Uso de la metodología en Colombia

En la XI Jornada de Gerencia de TI se presentó por parte de la Asociación Colombiana de Ingenieros de Sistemas-ACIS, las tendencias actuales en cuanto al desarrollo de proyectos colombianos relacionados con tecnología informática. Este estudio fue realizado con una participación de 248 proyectos.

Cómo se puede observar en la Figura 1 a nivel nacional los proyectos de desarrollo de software tanto en el año 2011 como en el año 2012 son los que en mayor porcentaje se llevan a cabo, con una diferencia altamente visible frente a los demás proyectos de tecnología de informática. Cabe aclarar que en la gráfica comparativa 2010-2011, ambos estaban alrededor del 40 % y en 2011-2012, hay una disminución de 5 % en el último año y se muestra el crecimiento de los proyectos de implementación de servicios.

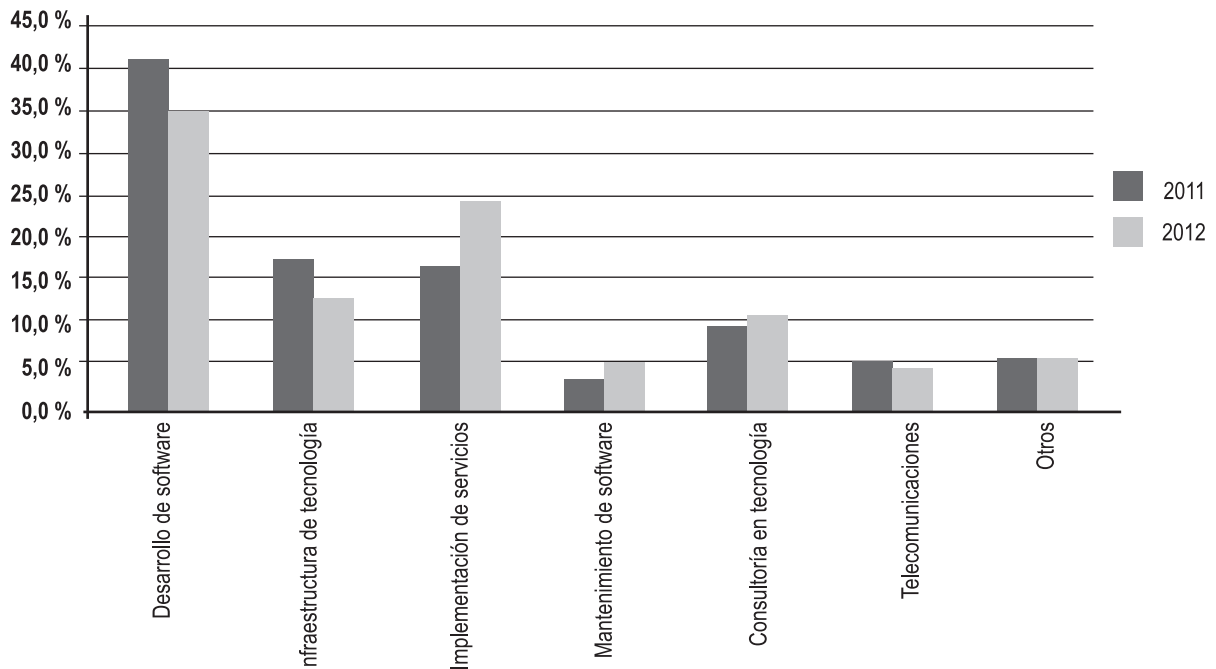


Figura 1. Tipo de proyectos de tecnología de información en Colombia [4]

En cuanto al sector donde se desempeñan estos proyectos, se puede visualizar en la siguiente figura que el sector que mayor auge tiene en los proyectos es el sector privado, con 71,01 %, un poco más del doble que el porcentaje de proyectos realizados en el sector público con tan solo 28,99 %. Esta tendencia se ha mantenido, con respecto a 2011, con una diferencia del 4,49 %.

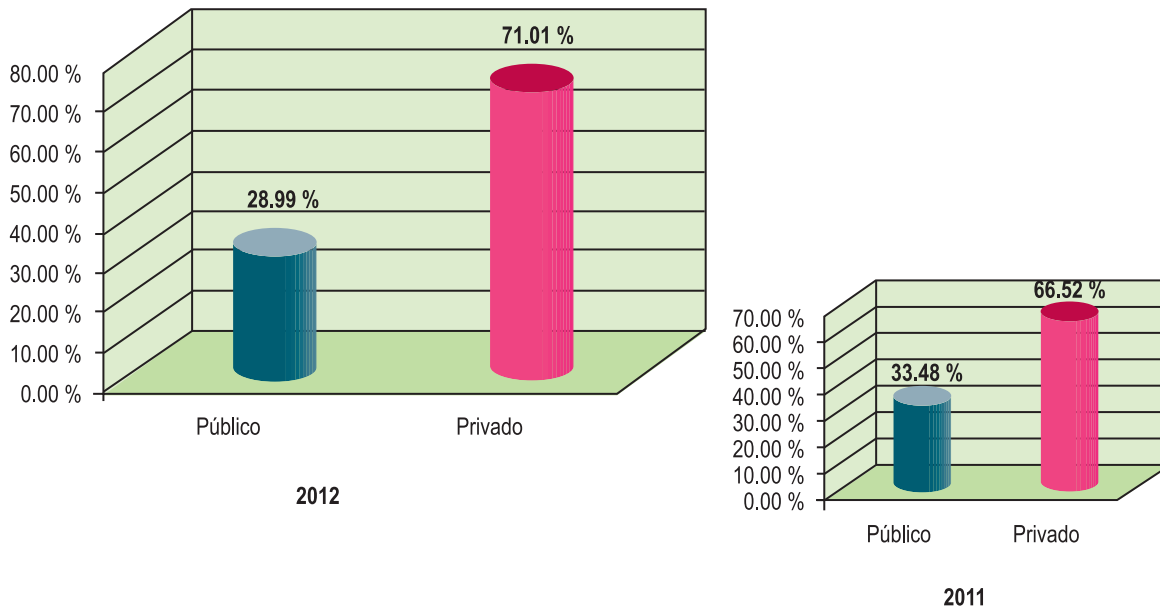


Figura 2. Distribución de sectores 2012 vs 2011 [4]

Para analizar el uso de herramientas y técnicas se muestra la siguiente figura donde se resume el porcentaje de utilización de herramientas y técnica para los proyectos en Colombia.

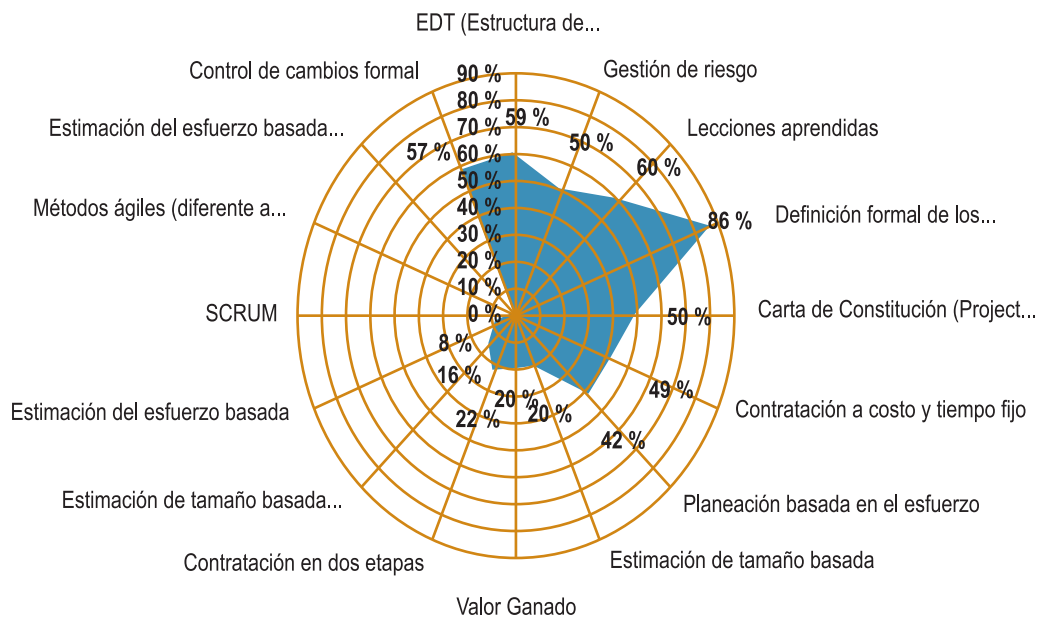


Figura 3. Uso de herramientas y técnicas [4]

B. Uso de CMMI en Colombia

Al 22 de julio de 2010 existían 3.060 certificaciones activas proporcionadas por el SEI [5]. Los países que encabezan el listado con el mayor número de certificaciones a nivel mundial, son algunas de las economías más grandes del planeta, por Latinoamérica se destacan Brasil y México en el top 10 de este listado. A nivel nacional, en Colombia, se han realizado esfuerzos para mejorar la competitividad del sector informático, entre los cuales se destacan: la colaboración con modelos de calidad como CMMI, la capacitación comercial y administrativa, entre otros [6]. Según el informe del SEI de octubre de 2013, para Colombia fueron evaluadas en CMMI 36 empresas, en su mayoría aplicadas para nivel de madurez 2 y 3 entre ellas 7 empresas en nivel 2, 19 nivel 3.

Algunas de las empresas colombianas valoradas oficialmente con CMMI a la fecha de octubre de 2013:

Tabla 1. Empresas colombianas valoradas oficialmente con CMMI [7]

Empresa	Nivel
Sistemas Integrales Ltda.	2
Informática & Tecnología S.A.	2
Heinsohn Business Technology S.A.	5
PSL S.A.	5
Unisys de Colombia S.A.	3
INDRA SOFTWARE LABS, S.L.	3
SoftManagement S.A.	3
Everis Colombia	3
PersonalSoft S.A.S.	5
IT Matica Software Factory SAS	2
Quipux S.A.S.	3
Alsus IT Group	2
Flag Soluciones S.A.S.	2
Asesores de Sistemas Especializados en Software	5
HDS Colombia S.A.S.	3
Carvajal Tecnología y Servicios	3
IG WebServices S.A.S.	5
Sistemas y Asesorías de Colombia S.A.	3

En octubre de 2007 el Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología “Francisco José de Caldas”-Colciencias, presentó la convocatoria 432-2007 “Apoyo al Fortalecimiento de la Capacidad Nacional de Software”. Inicialmente la Unión Temporal Red Colombiana de Calidad de Software denominada UT_ RCCS, conformada por la Universidad Industrial de Santander-UIS, la Universidad EAFIT, Procesix Colombia Ltda. y la Asociación de Parques Tecnológicos Parquesoft Colombia presentaron una propuesta de proyecto pertinente en respuesta a la convocatoria, la cual fue seleccionada para su financiación.

El proyecto se denominó desarrollo del programa Apoyo al Fortalecimiento de la Capacidad Nacional de Software por parte de la UT_RCCS incluyó varios objetivos de trabajo asociados a ampliar el conocimiento y aplicación del Modelo CMMI-DEV 1.2 en Colombia. Partiendo de la meta de impacto esperada para la implementación del Modelo CMMI-DEV en PYMES de software colombianas, las instituciones miembros de la UT_RCCS, generaron de forma integrada una metodología del proceso de asesoría y acompañamiento que permitiera lograr resultados acertados en la implementación del Modelo CMMI-DEV en un entorno PYME [8]. Se definió asesoría y acompañamiento de las PYMES de software para la implementación del Modelo CMMI-DEV v 1.2 y se ejecutó bajo un ciclo de vida de proyecto creado especialmente para atender los requerimientos del programa.

A manera de resultados el proyecto logró un despliegue amplio del Modelo CMMI-DEV y los conceptos de mejora de procesos en todas las regiones del país y en múltiples tamaños de Pymes como son las micro, pequeñas micro, pequeñas y medianas empresas, consiguiendo que un 83 % de las 55 pymes culminara un proceso de mejoramiento de principio a fin, y se perfilaran un 46 % de estos para realizar una evaluación oficial [9]. Las áreas de proceso del Modelo CMMI-DEV preferidas por las PYMES porque impactaron positivamente el desarrollo de sus productos y la organización interna de las empresas fueron: Planificación de Procesos (PP), Seguimiento y control de proyectos (PMC), Gestión de Requisitos (REQM) y Gestión de Configuración (CM), las cuales obtuvieron las mayores frecuencias como áreas de proceso seleccionadas para la planeación de mejora y mostraron los más altos promedios de adherencia a nivel nacional durante el diagnóstico final [10].

Algunas de las dificultades más sobresalientes durante el proceso de implementación del Modelo CMMI-DEV en las PYMES fueron: la interpretación de las prácticas del modelo, la identificación y definición de indicadores gerenciales y de proceso, y la dedicación de mayor esfuerzo a la definición de procesos que a su institucionalización [11].

IV. CMMI VS METODOLOGÍAS ÁGILES

Como se cita en *CMMI or Agile: Why Not Embrace Both!* [12], pueden ser muchas las razones de por qué se perciben los métodos ágiles y CMMI como elementos opuestos. Quizás porque las primeras implantaciones de CMMI fueron en organizaciones muy grandes, y las primeras implantaciones ágiles, en empresas pequeñas, pequeños equipos, con requisitos volátiles. Sin embargo, diferentes estudios demuestran que pueden ser compatibles y trabajar juntos mejorando drásticamente el rendimiento y el éxito del proyecto.

Según lo expuesto en este informe del SEI: existen dos razones principales por las que se considera que son incompatibles las metodologías ágiles y CMMI:

1. Los primeros en usar CMMI son los desarrolladores de gran escala, *riskaverse*, sistemas de misión

crítica, generalmente con altos niveles de supervisión de gestión de proyectos, mientras que los primeros en adoptar métodos ágiles en general, se centran en un solo equipo de proyectos de desarrollo u organizaciones pequeñas.

2. La información inexacta acerca de CMMI Ágil y el mal uso de ambos dio lugar a malas interpretaciones. Estas percepciones negativas que la posición de CMMI Ágil en desacuerdo entre sí surgió en gran parte de los siguientes factores:
 - Falta de información veraz y escasez de información precisa acerca de CMMI en la comunidad de métodos ágiles y viceversa.
 - Dificultades para el uso de terminología de la terminología y métodos ágiles.
 - Uso indebido de CMMI Ágil.

A lo largo del mundo han surgido nuevas situaciones, como por ejemplo qué compañías certificadas en CMMI necesitan introducir métodos de desarrollo ágiles para adaptarse a los cambios en el mercado, sin que ello conlleve a desechar totalmente las prácticas CMMI renunciando inclusive a su certificación CMMI que tan gran y exitosa inversión les ha llevado hasta el momento. También podría ocurrir el caso de organizaciones que tienen ya implantados procesos de desarrollo ágiles de software bien consolidados, cuyos clientes les exigen cierto nivel de conformidad a CMMI [13]. Existen casos de estudios sobre la correspondencia entre CMMI y métodos ágiles, que permiten a pequeñas y medianas empresas implantar procesos de desarrollo de software conformes a CMMI a través de métodos ágiles.

CMMI-DEV en su versión 1.3 ha sido ampliamente adoptado en todo el mundo para la disciplina de ingeniería de alta calidad. Esta versión tiene la capacidad de incluir modernos enfoques, así, como el uso de los métodos ágiles centrados en la arquitectura de desarrollo.

Considerando la metodología ágil más popular, SCRUM, y el modelo de madurez más extendido, CMMI, veamos cómo pueden complementarse. Para ello, es necesario conocer tanto las actividades, roles y productos de SCRUM [14] como la estructura y áreas de proceso de CMMI [15].

Las principales mejoras mutuas que puede traer el uso conjunto de SCRUM y CMMI son:

Mejoras que CMMI aporta a SCRUM

1. CMMI ayuda a determinar de una manera más formal las mejoras que se pueden introducir en los procesos, por ejemplo mediante las métricas o las auditorías internas.
2. La planificación formal ayuda a capturar y dar seguimiento a las decisiones de gestión del proyecto, especialmente cuando los proyectos y la organización crecen y la presión aumenta.
3. Ayuda a involucrar de manera común al resto de la organización y a los actores externos, tanto en el seguimiento de los proyectos como en el aprendizaje y difusión de las mejoras.

4. Define más claramente los roles a nivel de equipo y fuera de los equipos, hecho que facilita la asignación clara de las responsabilidades.
5. Facilita que se determine la formación que no puede adquirirse autónomamente por los miembros de los equipos, especialmente importante en proyectos grandes.
6. Normaliza la realización de ciertas tareas, de manera que puede reaprovecharse mejor el conocimiento, se es más eficiente y se evitan problemas de calidad.
7. El desarrollo más formal de requisitos de cliente ayuda a estimar y planificar mejor el proyecto que un simple *roadmap* de producto.

V. CONCLUSIONES

En la actualidad gran parte de los proyectos no concluyen satisfactoriamente, uno de los motivos que se visualiza a través de los estudios presentados es que no se realizaron unas adecuadas prácticas o un uso correcto de metodologías. Parte del fracaso en el desarrollo de los proyectos que no concluyen se debe a factores como tiempos eternos de desarrollo, bajos niveles de calidad, aumento de gasto y resistencia al cambio en la adopción de alguna metodología lo que conduce finalmente a clientes no satisfechos.

Según las estadísticas presentadas por el Instituto de Ingeniería de Software-SEI se muestran que el uso del Modelo de Madurez y Capacidad Integrado (*Capability Maturity Model Integration-CMMI*) ha sido adoptado en grandes compañías a nivel mundial con resultados satisfactorios para dar lugar a mejoras en la calidad de los procesos y los productos, cumplimiento de los presupuestos, y satisfacción de los clientes. Sin embargo, no se puede desconocer que las estrategias de mejora de Procesos Software basadas en CMMI-DEV requieren de procesos de desarrollo de software con gran inversión en términos de coste y tiempo que muchas pequeñas y medianas compañías (PYMES) no pueden asumir. Aún pudiendo permitírsele, una gran organización necesita de un largo camino para llegar a una madurez en los procesos.

En cuanto a los métodos ágiles según los estudios presentados se puede afirmar que a nivel mundial un 60 % de los proyectos ejecutados dentro de la empresa o el departamento de desarrollo son ágiles, pero no se puede desconocer que a pesar del repunte en el apoyo a la gestión ágil, resultados de la encuesta sugieren que el mayor obstáculo para una mayor adopción de métodos ágiles no parece ser el conocimiento de la metodología, pero sí la cultura interna de la compañía ya que en su mayoría presentan resistencia para su utilización.

Es posible combinar SCRUM y CMMI, un método ágil y uno robusto, si se conocen tanto las actividades, roles y productos de SCRUM como la estructura y áreas de proceso de CMMI para complementarlos y que pueda ser aplicable en organizaciones de diferentes tamaños.

VI. REFERENCIAS

- [1] The Standish Group International, Incorporated. The CHAOS manifesto 2013, think big, act small. Introduction, 1.
- [2] T. A.-B. Chow, "A survey study of critical success factors in agile software projects", *J. syst. Softw*, n°. 6, vol. 8, pp. 961-971, 2009.
- [3] R. Turner & A. Jain, Agile meets CMMI: culture clash or common cause? en extreme programming and agile methods - xp/agile universe, 2002.
- [4] ACIS - Memorias XI Jornada de Gerencia de Proyectos de TI - La X Encuesta Nacional de Gerencia de Proyectos por Alberto Cueto Gil. Disponible en: <http://www.acis.org.co/index.php?id=2026>
- [5] CMMI Institute for Carnegie Mellon. Published Appraisal Results. Available in: <https://sas.cmmiinstitute.com/pars/pars.aspx>
- [6] P. Abrahamsson, O. Salo, J. Ronkainen, & J. Warsta, Agile Software Development Methods: Review and Analysis. VTT Publications 478, 2002.
- [7] CMMI Institute for Carnegie Mellon. Published Appraisal Results Filter Country: Colombia. Available in: <https://sas.cmmiinstitute.com/pars/pars.aspx>
- [8] K. Beck, Extreme Programming Explained: Embrace Change. Addison-Wesley, 2002.
- [9] K. Beck, Test Driven Development: By Example. Addison-Wesley, 2002.
- [10] K. Schwaber, The Enterprise and SCRUM. Washington: Microsoft Press, 2007.
- [11] C. Larman and B. Vodde, Scaling Agile and Lean Development. Addison-Wesley Professional, 2008.
- [12] Hillel Glazer (Entinex, Inc.), Jeff Dalton (Broadsword Solutions Corporation), David Anderson (David J. Anderson & Associates, Inc.), Michael D. Konrad, Sandra Shrum. SEI 2008. CMMI or Agile: Why Not Embrace Both!
- [13] P. MacMahon, Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement. Addison-Wesley Professional, 2010.
- [14] La guía de Scrum. Ken Schwaber, Jeff Sutherland. Julio 2013. [en línea]. Disponible en: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-ES.pdf>
- [15] CMMI V1.3. www.cmmiinstitute.com