

Informe final de Investigación:

Propuesta de diseño e implementación de un marco de trabajo para
apoyar la evolución del software

*(An evolutionary visual software analytics framework
for supporting software Maintainability).*

Autor: Wilberth Molina Pérez
Carrera Ingeniería del Software

Marzo 2018

Índice

Índice	2
Resumen ejecutivo	3
Introducción	4
Antecedentes	4
Planteamiento del Problema de investigación	4
Objetivo de la investigación	5
Justificación	5
Marco teórico o referencial	5
Metodología	7
Planificación y revisión de la planificación	8
Diagnóstico	8
Análisis de los hallazgos	8
Resultados	9
Resultados generales	9
Resultados sobre recopilación de datos	11
Resultados sobre Herramientas Generales.....	14
Resultados sobre Herramientas para el proceso de desarrollo del SW.....	16
Resultados sobre la herramientas para apoyar el proceso de desarrollo	18
Resultados sobre herramientas de analítica visual.....	18
Conclusiones	20
Recomendaciones	21
Bibliografía	21

Resumen ejecutivo

El principal objetivo es la comprensión de que los cambios que se realizan a los sistemas de software es un elemento clave para poder continuar con su desarrollo o brindarles mantenimiento. Por lo que se requiere de métodos efectivos que permitan comprender los procesos de cambio y los grandes volúmenes de información que estos producen.

La metodología que se utilizó en el desarrollo de esta investigación se basó en una adaptación del modelo de Investigación Acción [Kemmis 2005].

La planificación inicial llevó a cabo la conceptualización de las metas, objetivos, preguntas de investigación y problema de investigación, los cuales se definen y redefinen de forma constante de acuerdo con la metodología

El análisis de los trabajos de investigación que han sido publicados en los últimos 5 años apoyó la construcción de una encuesta a profesionales que trabajan en la industria del software. Posteriormente, utilizando los resultados obtenidos en dicha encuesta, y con el apoyo de las referencias bibliográficas pertinentes, se realizó una discusión detallada sobre el estado de la investigación en este campo y su impacto en la industria (tomando como punto de partida el uso actual de la visualización y AV para apoyar el proceso de desarrollo y mantenimiento).

Se analizaron los resultados de todo el proceso presentar las principales conclusiones. Con base en los resultados de esta fase, se redefine el plan para el siguiente ciclo del proceso de investigación, que se espera continúe el próximo año.

Luego de que más de 200 profesionales ligados al sector participaran en la encuesta sobre diferentes elementos del proceso de desarrollo de software, herramientas de pruebas y metodologías de desarrollo se puede notar que existe una amplia variedad y profundidad tanto de herramientas como su aplicación en el proceso de desarrollo de software.

Igualmente, se nota que en el campo del análisis de datos, concretamente en la analítica visual no existe una práctica establecida y robusta en su uso, siendo comentado por los participantes en la encuesta como una opción importante a explorar en el delicado proceso de revisar la evolución del software y el proceso de mantenimiento que conlleva.

Introducción

Antecedentes

El desarrollo de este proyecto tiene como antecedente la tesis doctoral del doctor Antonio González Torres, titulada Evolutionary Visual Software Analytics. En dicha tesis se lleva a cabo la definición de un proceso que describe la aplicación de la analítica visual a la evolución de software con el fin de apoyar a los programadores y líderes de proyecto durante las tareas de desarrollo y mantenimiento. Con base en dicho proceso se lleva a cabo la definición e implementación de una arquitectura que es validada mediante un estudio de usabilidad y un caso de uso.

El trabajo desarrollado en la tesis mencionada plantea una buena oportunidad para continuar con el desarrollo de una propuesta más integral para apoyar a los programadores y administradores de proyecto. Por lo que se refierev a esta sección a dicha tesis y al artículo titulado Knowledge discovery in software teams by means of evolutionary visual software analytics, por encontrarse en estos profusamente documentados los antecedentes del área de investigación.

Los procesos de desarrollo y mantenimiento de software producen un gran número de cambios que requieren ser comprendidos por los programadores y los líderes de proyectos para poder realizar cambios adicionales a los sistemas. Lo anterior implica que estos procesos producen grandes volúmenes de datos en la forma de líneas de código, variables, relaciones de acoplamiento, cohesión, herencia e implementación de interfaces por cada revisión (commit) del sistema. El volumen de datos se debe multiplicar por el número de revisiones del sistema, las cuales por lo general se cuentan por miles después de unos pocos meses de evolución de un sistema mediano o grande. Como consecuencia, los datos que se generan durante los procesos de desarrollo y mantenimiento de software cumplen con las propiedades de Big Data, y requiere el uso de enfoques novedosos para transformarlos en conocimiento.

En este contexto, el uso de la Analítica Visual (AV) aplicada a la Evolución de Software (ES) (conocida como Evolutionary Visual Software Analytics) permite el análisis automático de los datos y su representación en elementos visuales que son desplegados mediante vistas enlazadas, las cuales se apoyan en técnicas de interacción persona-computadora y las capacidades cognitivas de los usuarios para obtener conocimiento y facilitar la toma de decisiones para efectuar cambios adicionales a los sistemas y promover su mantenibilidad.

Planteamiento del Problema de investigación

El proceso de desarrollo de software es un proceso complejo que por general involucra a un gran número de programadores que se encuentran distribuidos en diferentes áreas geográficas. Durante este proceso se produce una gran cantidad de información como producto de los cambios que se

realizan a los sistemas, y por el esquema distribuido de los sistemas y la frecuente falta de documentación, los programadores y líderes de proyectos requieren de métodos que les faciliten la comprensión de los cambios que se han efectuado para poder realizar cambios adicionales, con el fin de continuar con el desarrollo del sistema o brindarle mantenimiento.

En este contexto el análisis de la evolución de software resulta de gran utilidad, pero no ofrece información suficiente para llevar a cabo las tareas de comprensión de los cambios al sistema de una manera satisfactoria para apoyar de forma adecuada a los desarrolladores y administradores de proyectos. Por lo tanto, esta investigación ha tenido en cuenta el importante papel que ha tenido Visualización de la Información (VI) en los últimos años al proporcionar conocimiento a partir de grandes y complejos conjuntos de datos mediante el uso de representaciones visuales combinadas con técnicas de interacción.

Debido a lo anterior, los esfuerzos de un sector de la comunidad científica que investiga sobre métodos de ingeniería de software se han centrado en el uso de representaciones visuales con técnicas de interacción para facilitar la comprensión de los sistemas de software y su evolución. Estos esfuerzos de investigación se han centrado en el uso de Visualización de Software (VS) [Diehl 2007] y Visualización de la Evolución de Software (VES) [Voinea 2007]; aunque de forma más reciente algunos esfuerzos de investigación se han centrado en la aplicación de AV a los sistemas de software, pero no se han estudiado de forma amplia su aplicación a la Evolución de software.

Objetivo de la investigación

El objetivo de la investigación es determinar el nivel de aplicación de herramientas para el desarrollo del software, evolución del software y la factibilidad de uso de la analítica visual para la revisión de la evolución del software que lleven a la comprensión de los cambios que se realizan a los sistemas de software para poder continuar con su desarrollo o brindarles mantenimiento.

Justificación

La comprensión de los cambios que se realizan a los sistemas de software es un elemento clave para poder continuar con su desarrollo o brindarles mantenimiento. Por lo que se requiere de métodos efectivos que permitan comprender los procesos de cambio y los grandes volúmenes de información que estos producen.

Marco teórico o referencial

La Analítica Visual (AV) combina las fortalezas de las máquinas con las humanas, como la capacidad de análisis, la intuición, la resolución de problemas y la percepción visual. Por lo tanto, las personas están en el corazón de AV [Dix 2010] e Interacción Persona-Ordenador (IPO) es un componente clave para

apoyar el descubrimiento de conocimientos. Este es un proceso cuyo objetivo es proporcionar conocimiento a partir de la enorme cantidad de datos científicos, forenses, académicos o de las empresas que son almacenados por formatos heterogéneos como bases de datos, archivos HTML, XML, metadatos y código fuente.

Este proceso recopila información de forma iterativa, realiza el pre-procesamiento de datos, efectúa análisis estadístico [Peck 2011], lleva a cabo minería de datos, usa aprendizaje automático [Witten 2005], representación del conocimiento [van Harmelen 2007], interacción del usuario [Sharp 2011], representaciones visuales [Leung 1994a, Johnson 1991, Robertson 1991], la cognición y percepción humana, la exploración y las capacidades humanas para la toma de decisiones [Keim 2006, Llorca 2006].

AV se ha aplicado de forma amplia a problemas tan diversos como la gripe aviar [Proulx 2006], las condiciones pale-oceanográficas [Theron 2006c], el análisis organizacional [Card 2006], eLearning [Gomez-Aguilar 2009, Gomez-Aguilar 2015b], la toma de decisiones [Migut 2011, Savikhin 2008], ingeniería de ontologías [Garcia 2012, Garcia-Penalvo 2012c, Garcia-Penalvo 2014], patrones temporales [Weaver 2006, Ziegler 2010], las redes sociales [Perer 2011], el análisis de la seguridad [Harrison 2011] y los sistemas de software [Reniers 2012, Gonzalez-Torres 2013b]. Por lo tanto, se puede decir que el descubrimiento de conocimiento es una propiedad intrínseca de AV, debido a que está destinada a apoyar a los analistas y tomadores de decisiones en la obtención de conocimiento a partir de grandes conjuntos de datos multivalentes [Thomas 2005].

En consecuencia, AV puede ofrecer soluciones al problema que conlleva brindar apoyo efectivo a programadores y administradores de proyectos durante los procesos de implementación de sistemas, teniendo en cuenta que es un proceso que ofrece un enfoque integral que incluye desde la recuperación de información y análisis hasta la representación visual de los resultados del análisis. Además, ofrece la posibilidad de explorar diferentes niveles de detalle utilizando múltiples representaciones visuales, enlazadas y coordinadas entre sí, mediante el uso de técnicas de interacción [North 2000]. Esto permite facilitar el descubrimiento de relaciones y conocimiento por medio del razonamiento analítico del analista.

Teniendo en cuenta estos factores positivos, se puede decir que una de las propiedades de AV es la capacidad de proporcionar apoyo a la toma de decisiones [Savikhin 2008, Mane 2012] utilizando las capacidades cognitivas de los usuarios, por lo cual su aplicación a la evolución de software ofrece grandes oportunidades para apoyar tanto a los programadores como a los administradores de proyectos. Sin embargo, la aplicación de AV a la evolución del software es nueva y las tareas y necesidades de información de los programadores y administradores de proyectos son complejas [Forsberg 2005, de Oliveira Barros 2004, Munch 2004, Paul 1999].

Esto implica que todavía existe un gran número de desafíos que deben ser superados para apoyar con éxito a los administradores de proyectos en la toma de decisiones. Entre esos retos se encuentran los siguientes:

- Facilitar el análisis visual y la evaluación del proceso de desarrollo.

- Proporcionar métodos para controlar visualmente la evolución de la calidad de los elementos de software (clases, paquetes y módulos), teniendo en cuenta el uso de métricas de calidad del software. Lo anterior con el objetivo de mantener la complejidad y la evolución del sistema bajo control, así como asegurar el control de calidad.
- Proporcionar mecanismos visuales para revisar las medidas de ejecución de tareas, y permitir el análisis de rendimiento y la predicción de los avances.
- Apoyar mediante el uso de métodos visuales la gestión de riesgos, el crecimiento y complejidad del producto de software.
- Mantener informados a los administradores de proyectos sobre los patrones de colaboración entre desarrolladores y sobre los elementos que se han modificado de forma síncrona o asíncrona, así como sobre las consecuencias (en términos de calidad y funcionalidad) de los cambios que se han llevado a cabo.

Mientras que los desafíos que enfrenta AV para apoyar a los programadores en la comprensión de Evolución de Software (ES), de acuerdo con sus necesidades de información [Sillito 2006b] son las siguientes:

- Ofrecer detalles de los elementos de software a los que están efectuando cambios.
- Proporcionar información sobre los componentes de software que son modificados de forma simultánea por otros programadores.
- Permitir que los programadores comprendan las implicaciones de los cambios realizados con base en las relaciones (herencia y implementación de la interfaz) y asociaciones entre los elementos de software (composición, de referencia, y de acoplamiento), así como el efecto de la colaboración entre los objetos.
- Proporcionar detalles sobre la creación de variables así como el acceso y modificación de datos por medio de argumentos a los métodos y variables globales.
- Facilitar que los programadores analicen y comparen para dos o más revisiones el control de flujo, la ejecución y la gestión de excepciones entre revisiones.
- Facilitar la identificación de diferencias entre archivos, elementos de software y tipos en varias revisiones.

Metodología

El proceso desarrollado en esta investigación se planteó siguiendo el modelo

La metodología que se utilizó en el desarrollo de esta investigación se basó en una adaptación del modelo de Investigación Acción [Kemmis 2005]. Las fases de esta metodología que se utilizaron se detallan a continuación:

Planificación y revisión de la planificación

En esta fase se llevó a cabo la conceptualización de las metas, objetivos, preguntas de investigación y problema de investigación, los cuales se definen y redefinen de forma constante de acuerdo con la metodología. Del mismo modo, los conceptos, términos y el proceso de análisis de la Evolución del Software, así como la definición del proceso de Analítica Visual se revisarán en cada iteración de la metodología.

Diagnóstico

En esta fase se realizó el análisis de los trabajos de investigación que han sido publicados y están relacionados con la aplicación de AV a los sistemas de software (y su evolución) en los últimos 5 años. Durante esta etapa, se realizó encuestas a 274 profesionales que trabajan en la industria del software. El objetivo de esta encuesta fue determinar ¿Cuáles herramientas están siendo utilizadas por los programadores y líderes de proyectos durante el desarrollo de productos de software? Posteriormente, utilizando los resultados obtenidos en dicha encuesta, y con el apoyo de las referencias bibliográficas pertinentes, se realizó una discusión detallada sobre el estado de la investigación en este campo y su impacto en la industria (tomando como punto de partida el uso actual de la visualización y AV para apoyar el proceso de desarrollo y mantenimiento). La población considerada en el estudio fue de profesionales que se dedican a las áreas de desarrollo de sistemas, calidad y mantenimiento así como administradores de proyectos de desarrollo. El muestreo se realizó por conveniencia accediendo a los contactos generados en bases de datos de empresas tecnológicas de Costa Rica y profesionales egresados de algunas universidades de Costa Rica.

Análisis de los hallazgos

Esta fase del ciclo de investigación se analizó los resultados de todo el proceso presentando las principales conclusiones. Con base en los resultados obtenidos, se redefine el plan para el siguiente ciclo del proceso de investigación, que se espera continúe el próximo año.

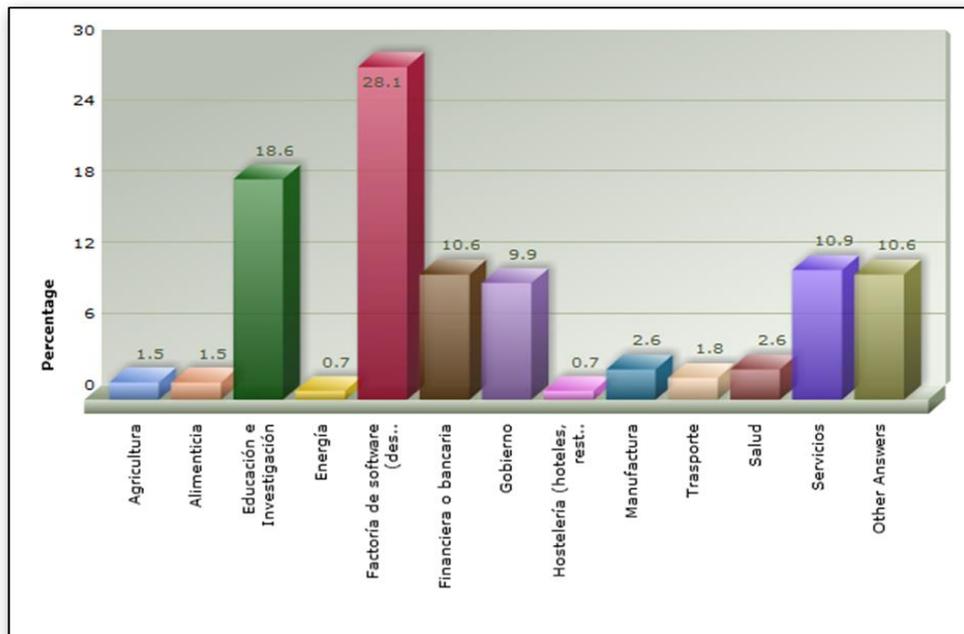
Resultados

Luego de que más de 274 profesionales ligados al sector participaran en la encuesta sobre diferentes elementos del proceso de desarrollo de software, herramientas de pruebas y metodologías de desarrollo se puede notar que existe una amplia variedad y profundidad tanto de herramientas como su aplicación en el proceso de desarrollo de software.

Resultados generales

Existe una amplia gama de sectores donde se aplicó la encuesta, como se aprecia en el gráfico #1 las cinco principales actividades a las que se dedican las empresas de los encuestados son: Desarrollo de software (28.1%), Educación e investigación (18.6%), Servicios (10.9%), Área financiera o bancaria (10.6%), Gobierno (9.9%)

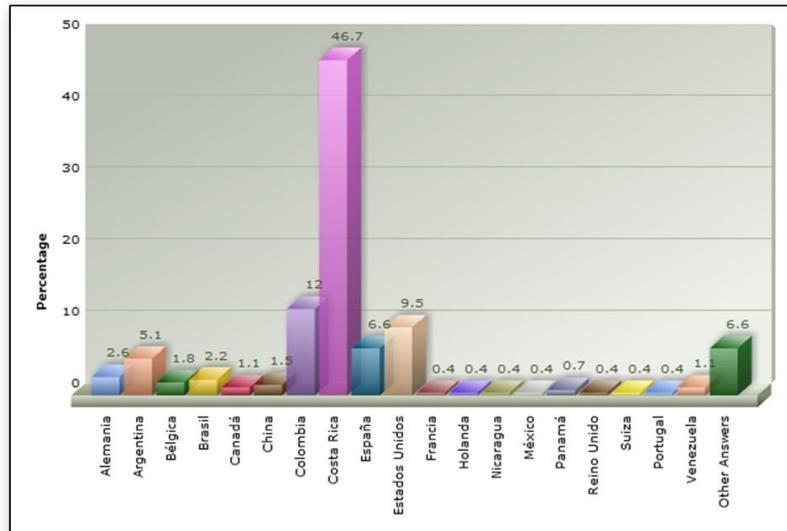
Gráfico 1: Sector al cual pertenece la empresa



Fuente: encuesta aplicada, elaboración del autor

Las empresas mayoritariamente son originales de Costa Rica (46.7%) aunque hay presencia de empresas originarias de Colombia (12%), Estados Unidos (9.5%), España (6,6%) y Argentina (5.1%) entre otros. Mayor detalle se puede observar en el gráfico #2.

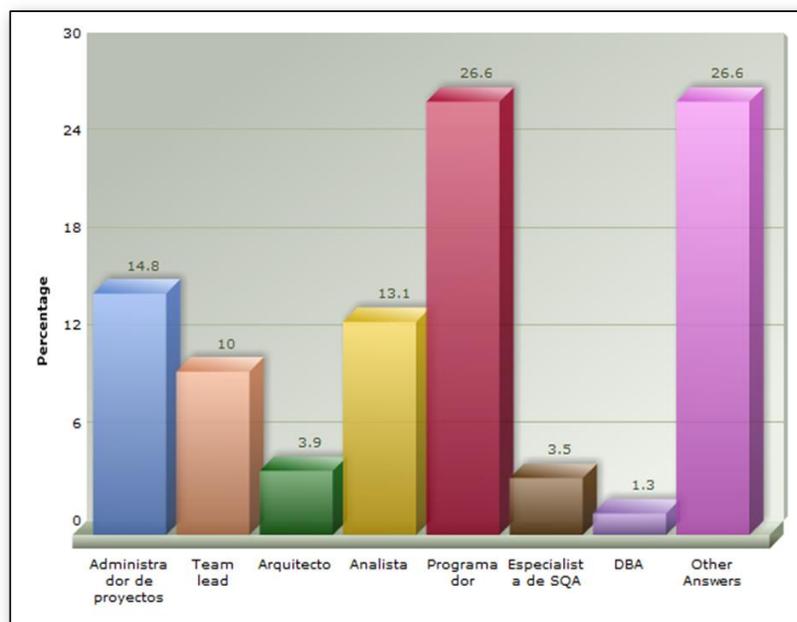
Gráfico 2: ¿Cuál es el país de origen de la empresa?



Fuente: encuesta aplicada, elaboración del autor

Cómo se puede apreciar en el gráfico #3, mayoritariamente los encuestados son programadores (26.6%), administradores de proyectos (14.8%), analistas (13,1%), líderes de proyecto (10%) y arquitectos de software (3,9%). Esta distribución de puestos es consecuente con las premisas de la investigación.

Gráfico 3: Puesto que desempeña en la empresa



Fuente: encuesta aplicada, elaboración del autor

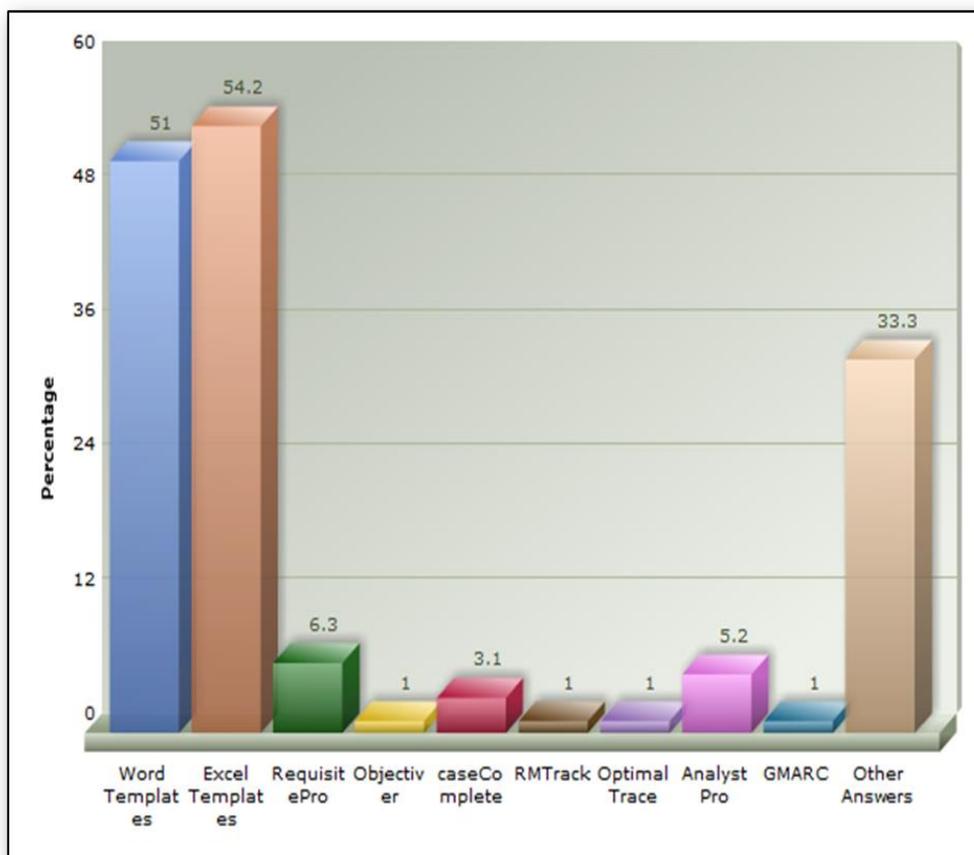
Resultados sobre recopilación de datos

Avanzando en la realización de la encuesta continúan preguntas para conocer si utilizan herramientas que puedan recopilar datos del desarrollo del software. Los datos que arrojan esta sección se detalla a continuación.

Sólo el 44.5% de los encuestados manifiesta que utilizan una herramienta para el levantamiento y análisis de requerimientos.

Como se presenta en la figura #4, la población que utiliza una herramienta, mayoritariamente utiliza plantillas de Word o Excel para esta labor.

Gráfico 4: Indique si utiliza alguna de estas herramientas.

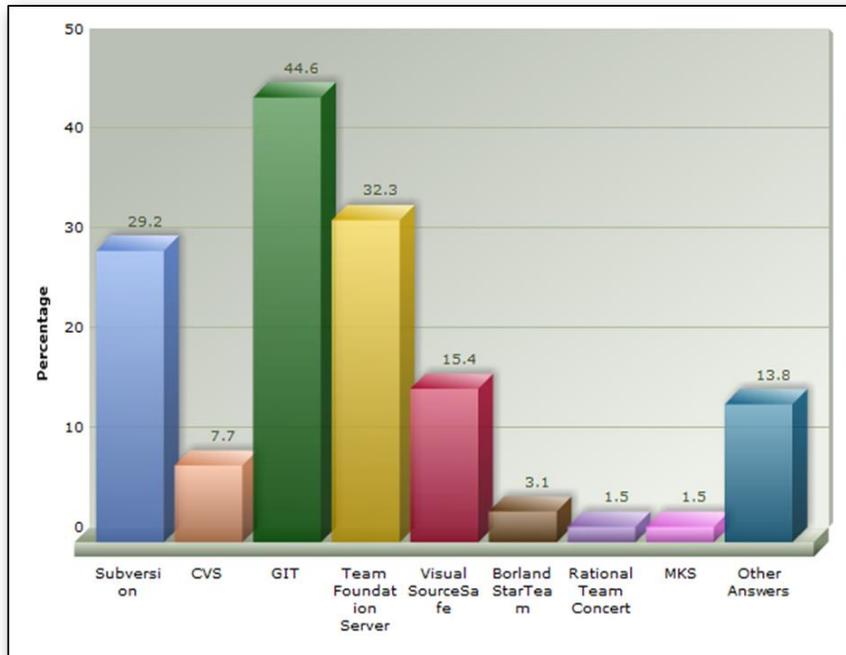


Fuente: encuesta aplicada, elaboración del autor

Menos del 40% de los encuestados indica que utilizan alguna herramienta para el diseño de sistemas. Adicionalmente, menos del 34% indican el uso de una herramienta de control de versiones del software. Y entre las herramientas más utilizadas para el control del software se encuentran GIT

(44.6%), Team Foundation Server (32.3%), Subversión (29.2%), Visual SourceSafe (15,4%), CVS (7.7%), mayores detalles se pueden apreciar en el gráfico #5.

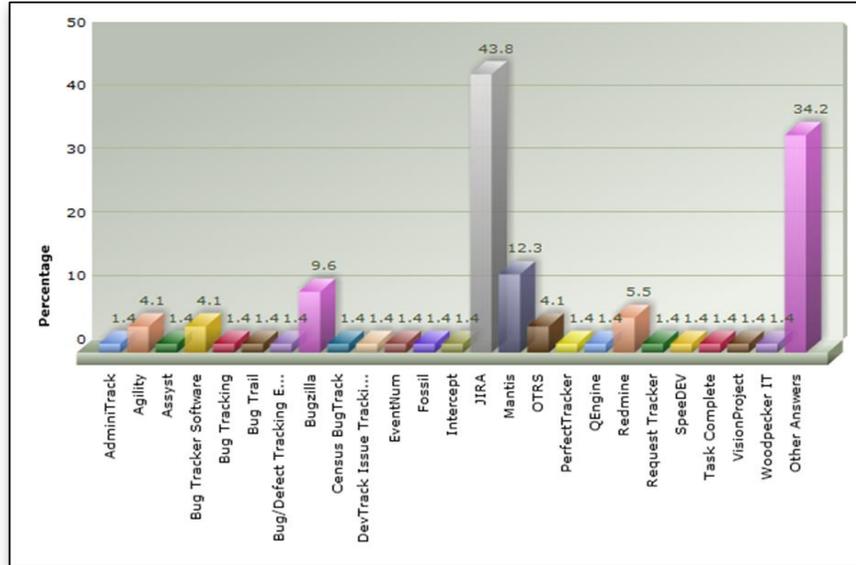
Gráfico 5: Herramienta de control de versiones que utilizan.



Fuente: encuesta aplicada, elaboración del autor

Menos del 40% de los encuestados indica que utiliza herramientas de seguimiento de errores y según se aprecia en el gráfico # 6, la herramienta más utilizada es JIRA en un 43.8% de incidencia le siguen Mantis (12.3%), Bugzilla (9.6%), Redmine (5.5%).

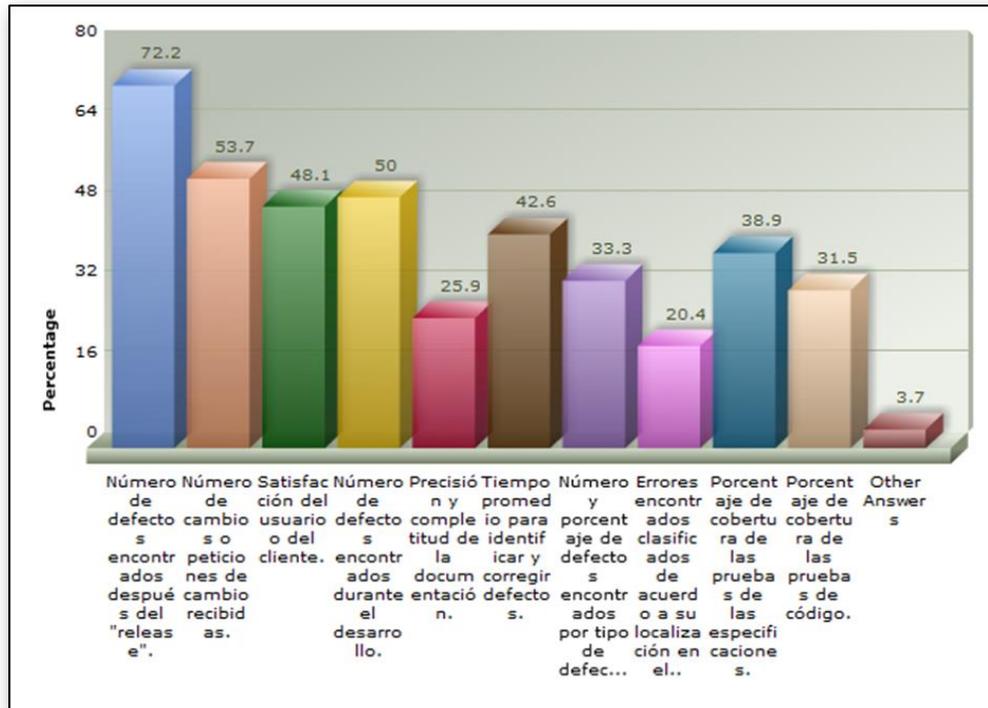
Gráfico 6: ¿Qué herramienta de Bug Tracking usan?



Fuente: encuesta aplicada, elaboración del autor

En el 55,4% de los casos se indica que realizan un seguimiento que vincula los errores con los cambios realizados en los sistemas. Y en el 39,4% de los casos indican que se calculan métricas de software atinentes al aseguramiento de la calidad del software. Siendo Excel la herramienta que en el 40,3% de los casos utilizan para recoger y procesar la información de las métricas. Según se aprecia en el gráfico #7, las principales métricas que calculan son: Número de defectos encontrados después de un “release” (72.2%), Número de peticiones de cambio recibidas (53.7%), Número de defectos encontrados durante el desarrollo (50%), Satisfacción del cliente (48.1%), Tiempo promedio para identificar y corregir errores (42.6%).

Gráfico 7: Enumeración de las métricas que se utilizan



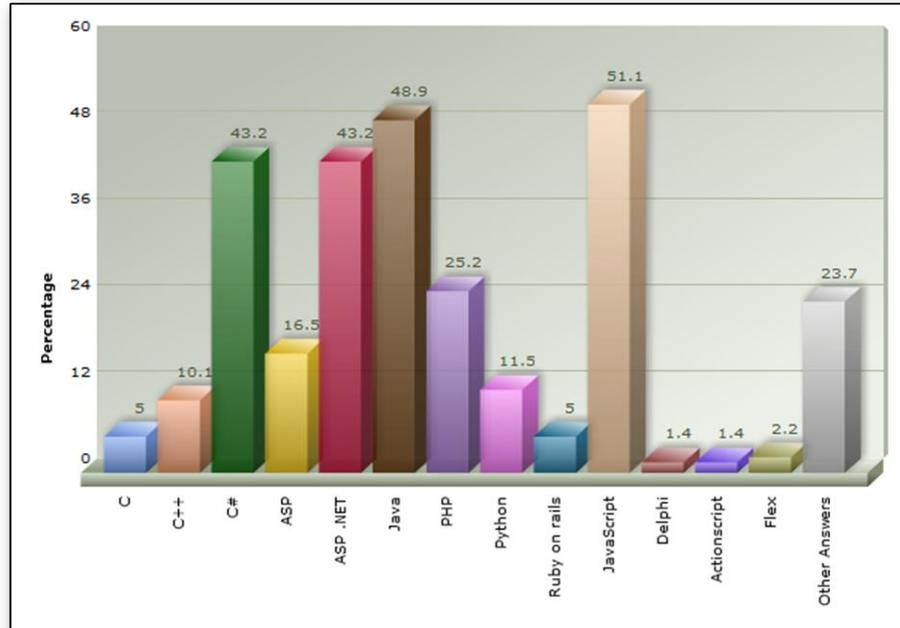
Fuente: encuesta aplicada, elaboración del autor

Resultados sobre Herramientas Generales

Continúa la revisión de las herramientas de carácter general utilizadas en las empresas, con ello se determina los siguientes datos.

Sobre los lenguajes que usualmente utilizan se presenta JavaScript como el más usual con un 51.1%, continúan Java (48.9%), C# y Asp .Net (43.2%), PHP (25.2%), ASP (16.5%). El detalle se puede observar en el gráfico #8.

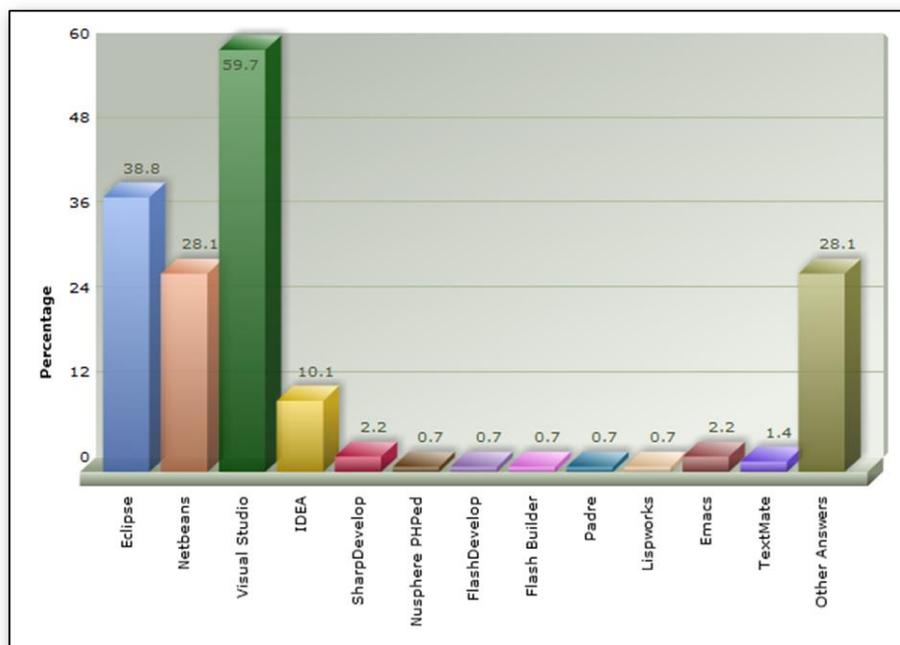
Gráfico 8: Lenguajes de programación que utilizan



Fuente: encuesta aplicada, elaboración del autor

Según se presenta en el gráfico #9, el IDE de desarrollo que normalmente utilizan es Visual Studio con 59.7% siguiendo Eclipse (38.8%), Netbeans (28.1), IDEA (10.1%), SharpDevelop (2.2%).

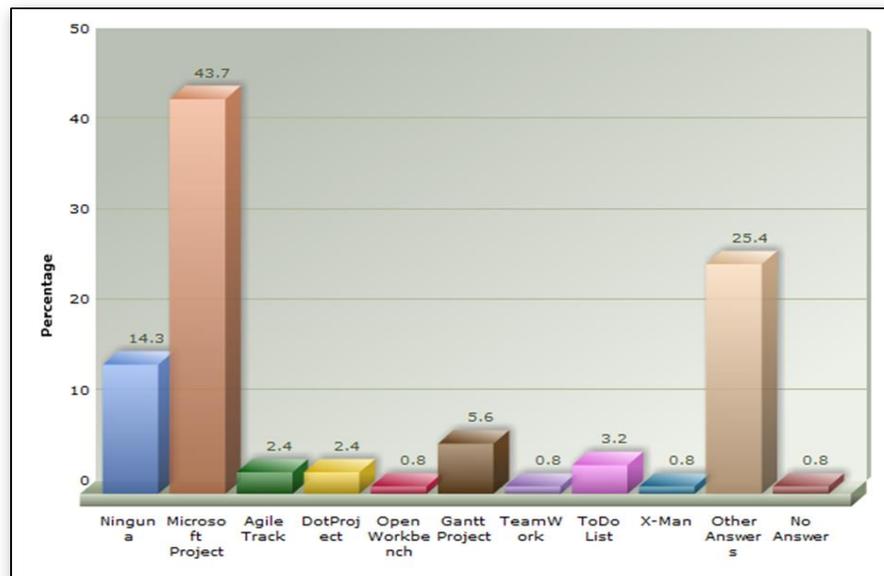
Gráfico 9: ¿Cuál IDE utilizan normalmente?



Fuente: encuesta aplicada, elaboración del autor

Únicamente menos del 13% de los encuestados manifiesta que en la empresa donde labora se utilizan herramientas especializadas para realizar ingeniería inversa. Y el 34.1% indican que utilizan una herramienta de visualización para realizar tareas durante el análisis, diseño, programación o administración de proyectos de software. Según se aprecia en el gráfico #10, mientras que el 43.7% de los encuestados indica que utilizan Microsoft Project como herramienta de seguimiento de proyectos, el 14.3% indica que no utilizan herramientas para el seguimiento de proyectos.

Gráfico 10: ¿Qué herramienta utilizan para la planificación, control y seguimiento de proyectos?



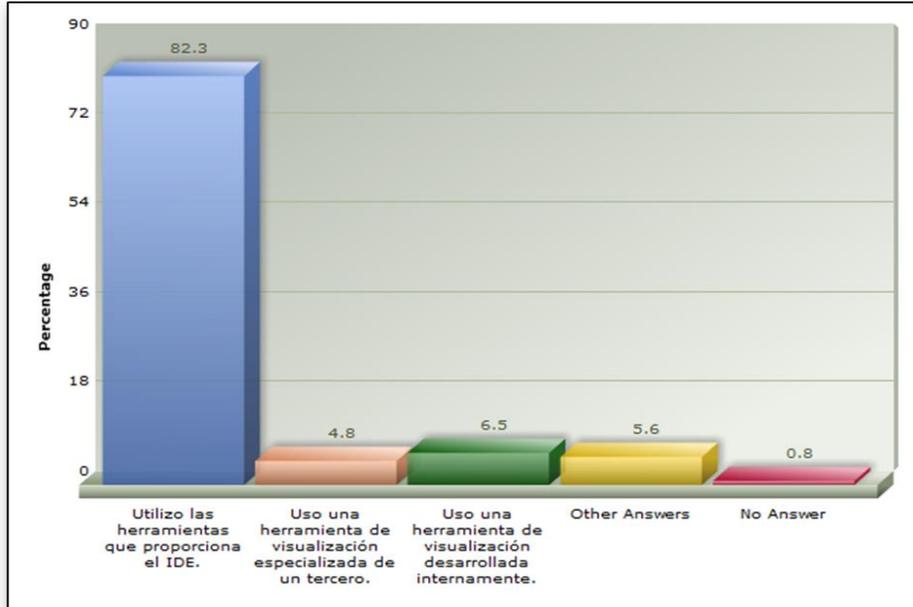
Fuente: encuesta aplicada, elaboración del autor

Resultados sobre Herramientas para el proceso de desarrollo del SW

Como se aprecia en el gráfico #11, mayoritariamente (82.3%) de los encuestados indica que utilizan las herramientas que proporciona el IDE para depurar programas,

El 6.5% indica que utilizan una herramienta interna y el 4.8% que utilizan una herramienta de terceros.

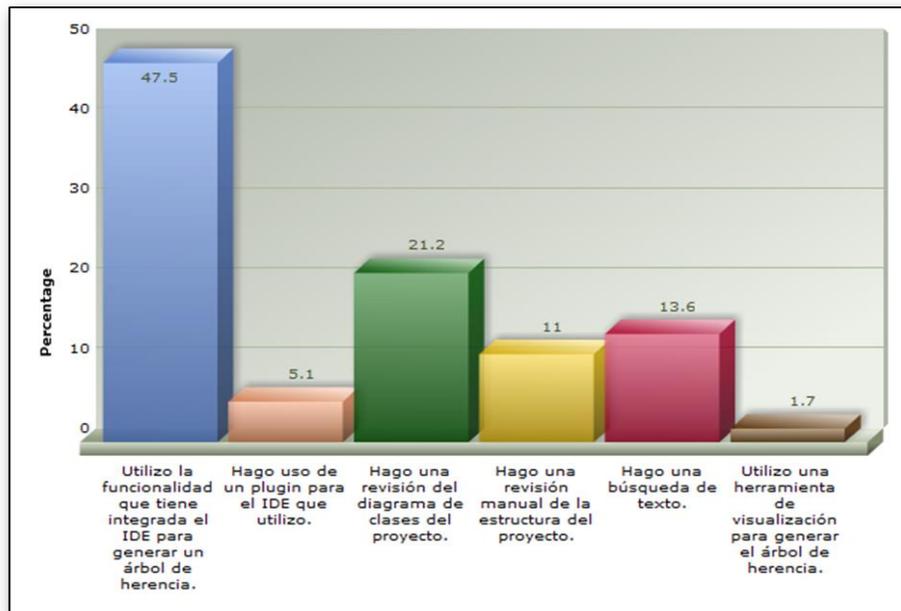
Gráfico 11: Herramienta para la depuración de código



Fuente: encuesta aplicada, elaboración del autor

Según el gráfico #12, el 47% de los participantes en la encuesta indica que utilizan las facilidades que provee el IDE que utilizan, mientras que 45.8% realizan acciones “manuales” para esto.

Gráfico 12: Para determinar temas de herencia, ¿qué utilizan?



Fuente: encuesta aplicada, elaboración del autor

Sólo el 8.5% de los encuestados manifiesta que en la empresa utilizan alguna herramienta especializada para entender la interdependencia entre los elementos del software. Adicional a esto sólo el 20.4% de los encuestados indica que utilizan una herramienta de software especializada para la detección de clones de software. Finalmente el 11.8% indica que utiliza una herramienta de software para determinar las ubicaciones de los elementos de un proyecto trabajado en equipo.

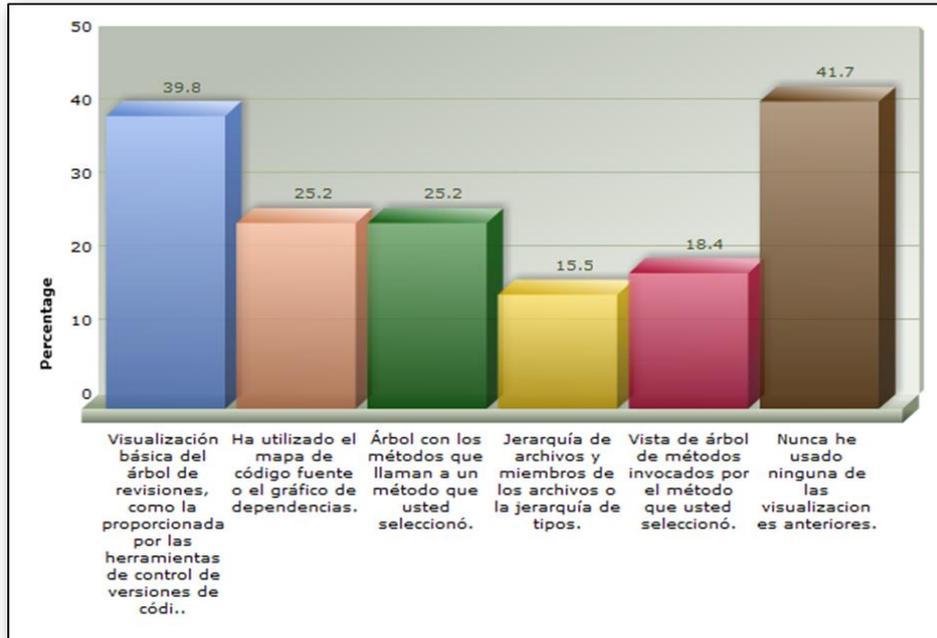
Resultados sobre la herramientas para apoyar el proceso de desarrollo

Sólo el 29.4% de los encuestados indica que en la empresa donde laboran utilizan una herramienta para determinar el aporte de cada miembro del equipo al código del proyecto. Y el 41.5% de los encuestados manifiesta que utilizan una herramienta para determinar la porción del código que está siendo actualizado por diferentes miembros del equipo. El 11.3% de los encuestados manifiesta que en la empresa donde laboran utilizan herramientas para la visualización de métricas de software, finalmente sólo el 16% manifiesta que utilizan una herramienta de visualización para monitorear los cambios de las métricas entre varias revisiones.

Resultados sobre herramientas de analítica visual

Según el gráfico #13, el 41.7% de los participantes nunca ha utilizado una herramienta de visualización, mientras que el 39.8% utilizan herramientas básicas de visualización del IDE.

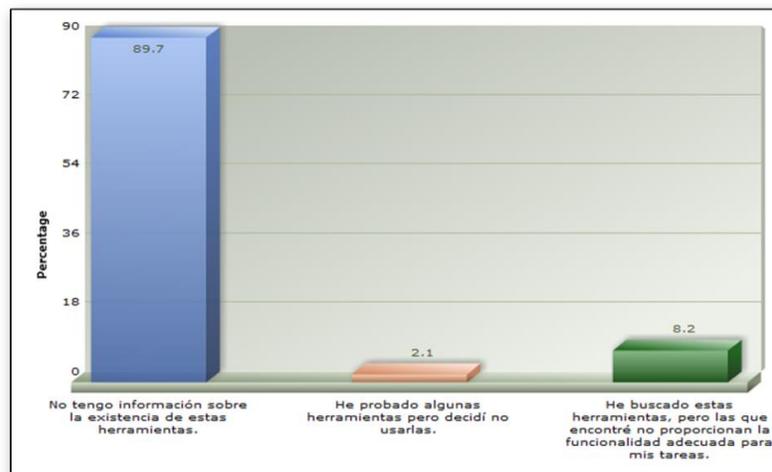
Gráfico 13: Herramientas de visualización incluidas en IDE's



Fuente: encuesta aplicada, elaboración del autor

Sólo es 5.8% utiliza herramientas de analítica visual para soportar procesos de desarrollo y mantenimiento del software. Según el gráfico #14, el 89.7% desconoce de la existencia de este tipo de herramientas, mientras que el 8.2% no encuentra la funcionalidad que requiere y finalmente el 2.1% decide no utilizarlas.

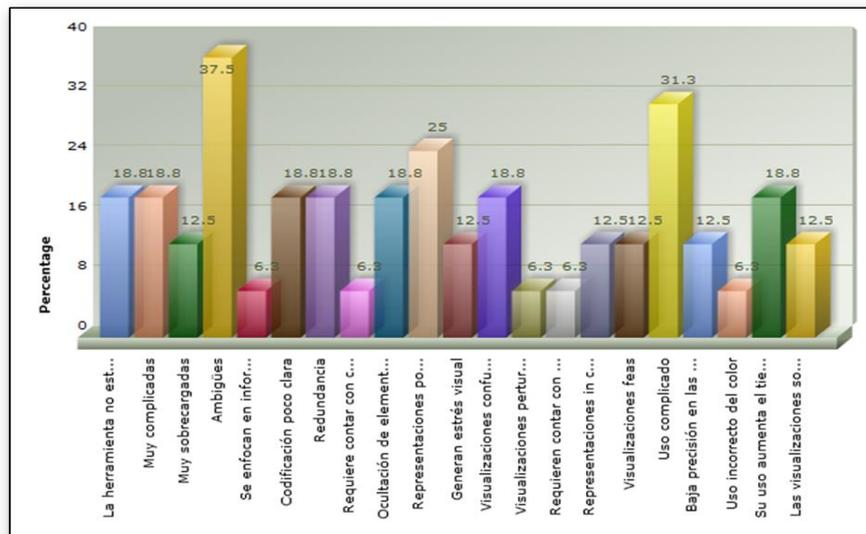
Gráfico 14: Principal razón para no utilizar herramientas de analítica visual?



Fuente: encuesta aplicada, elaboración del autor

Según se aprecia en el gráfico #15, las tres principales desventajas de la herramientas de analítica visual son: son ambiguas (37.5%), su uso es complicado (31.3%), o bien tiene representaciones pobres (25%).

Gráfico 15: Principales desventajas de las herramientas de analítica visual



Fuente: encuesta aplicada, elaboración del autor

Finalmente el 62.5% de los participantes considera que las herramientas de analítica visual reducen el tiempo de desarrollo del software y en el 84.2% de los participantes considera que este tipo de herramientas debe abordarse en los planes de estudio de las carreras de ingeniería del software.

Conclusiones

Se puede notar que hay diferentes niveles de uso de herramientas en el proceso de desarrollo esta variedad ofrece un panorama importante para el desarrollo de análisis visual sobre la información que puede generarse en el proceso.

Como se fundamenta en estudios presentados González (2016) esta amplitud de prácticas es una buena oportunidad para avanzar en el desarrollo de herramientas que faciliten el aprendizaje sobre la evolución del software y por ello los resultados preliminares de esta investigación funcionan como un

base para futuros procesos investigativos que consideren estos resultados y generen a partir de ellos propuestas avanzadas en el área del análisis visual.

El problema planteado igualmente se presenta en la población estudiada, por ello se concluye que en Costa Rica la evolución del software aún no cuenta con herramientas automatizadas, claramente tampoco herramientas de analítica visual que faciliten la comprensión del proceso evolutivo de un software.

Estas limitaciones se suman a una amplia variedad de prácticas que hacen más difícil establecer estándares que se apliquen en el sector de desarrollo de software. Siendo esto una excelente oportunidad para continuar con estudios y propuestas para la creación de un marco de trabajo en el área del análisis visual sobre la evolución del software.

Recomendaciones

El trabajo desarrollado es parte de un proyecto presentado, este proyecto contempló originalmente 2 años de desarrollo del proyecto de investigación, claramente el producto presentado en este informe contempla sólo 8 meses de desarrollo y por tanto es importante buscar la continuación del proyecto original para desarrollar las siguientes etapas.

Los resultados obtenidos ofrecen información que se considera congruentes con lo planteado en el proyecto original por lo su continuación es considerada posible a la luz de los resultados obtenidos.

Bibliografía

[Card 2006] Stuart K. Card, Bongwon Suh, Bryan A. Pendleton and Jeffrey Heer. TimeTree: exploring time changing hierarchies. In IEEE Symposium on Visual Analytics Science and Technology 2006 (VAST 2006), Baltimore; MD; USA. Piscataway NJ, 2006. IEEE Computer Society.

[de Oliveira Barros 2004] Marcio de Oliveira Barros, Claudia Maria Lima Werner and Guilherme Horta Travassos. Supporting risks in software project management. Journal of Systems and Software, vol. 70, no. 1-2, pages 21 – 35, 2004.

[Diehl 2007] Stephan Diehl. Software visualization visualizing the structure, behavior, and evolution of software. Springer Berlin Heidelberg New York, 2007.

[Dix 2010] Alan Dix, Margit Pohl and Geoffrey Ellis. Mastering the information age solving problems with visual analytics, chapter Perception and Cognitive Aspects, pages 109 – 130. Eurographics Association, 2010.

[Forsberg 2005] Kevin Forsberg, Hal Mooz and Howard Cotterman. Visualizing project management: Models and frameworks for mastering complex systems. John Wiley & Sons, 3rd edition, September 2005.

[García 2012] Juan García. Analítica Visual Aplicada a la Ingeniería de Ontologías. PhD thesis, University of Salamanca, 2012.

[Garcia-Penalvo 2012c] Francisco J. García-Penalvo, Ricardo Colomo Palacios, Juan García and Roberto Therón. Towards an ontology modeling tool. A validation in software engineering scenarios. Expert Systems Application, vol. 39, no. 13, pages 11468–11478, 2012.

[Garcia-Penalvo 2014] Francisco J. García-Penalvo, Patricia Ordoñez de Pablos, Juan García and Roberto Therón. Using OWL-VisMod through a decision-making process for reusing OWL ontologies. Behaviour & IT, vol. 33, no. 5, pages 426–442, 2014.

[Gomez-Aguilar 2009] Diego A. Gómez-Aguilar, Roberto Therón and Francisco J. Garcia-Peñalvo. Semantic Spiral Timelines Used as Support for e-Learning. Journal of Universal Computer Science, vol. 15, no. 7, pages 1526–1545, April 2009.

[Gomez-Aguilar 2009] Diego A. Gómez-Aguilar, Roberto Therón and Francisco J. Garcia-Penalvo. Semantic Spiral Timelines Used as Support for e-Learning. Journal of Universal Computer Science, vol. 15, no. 7, pages 1526–1545, April 2009.

[Gómez-Aguilar 2015b] Diego A. Gómez-Aguilar, Ángel Hernandez-García,

[González-Torres 2013b] Antonio González-Torres, Francisco J. García-Peñalvo and Roberto Therón. Human-Computer interaction in evolutionary visual software analytics. Computers in Human Behavior, vol. 29, no. 2, pages 486–495, March 2013.

[Harrison 2011] L. Harrison, Wenwen Dou, Aidong Lu, W. Ribarsky and Xiaoyu Wang. Guiding security analysis through visualization. In 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), pages 317–318, oct. 2011.

[Johnson 1991] Brian Johnson and Ben Shneiderman. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. In VIS '91: Proceedings of the 2nd conference on Visualization '91, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[Keim 2006] Daniel A. Keim, Florian Mansmann, Jorn Schneidewind and Hartmut Ziegler. Challenges in Visual Data Analysis. In IV '06, Proceedings of the Conference on Information Visualization, pages 9–16, Washington, DC, USA, 2006. IEEE Computer Society.

[Kemmis 2005] Stephen Kemmis and Robin McTaggart. The sage handbook of qualitative research (3rd ed.), chapitre Participatory Action Research: Communicative Action and the Public Sphere., pages 559–603. Sage Publications Ltd, 2005.

[Leung 1994a] Ying K. Leung and Mark D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. ACM Transactions in Computer-Human Interaction, vol. 1, no. 2, pages 126–160, June 1994.

[Llora 2006] X. Llora, K. Sastry, F. Alias, D. E. Goldberg and M. I. Welge. Analyzing Active Interactive Genetic Algorithms Using Visual Analytics. In GECCO '06, Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 1417–1418, New York, NY, USA, 2006. ACM.

[Mane 2012] Ketan K. Mane, Chris Bizon, Charles Schmitt, Phillips Owen, Bruce Burchett, Ricardo Pietrobon and Kenneth Gersing. VisualDecisionLinc: A visual analytics approach for comparative effectiveness-based clinical decision support in psychiatry. Journal of Biomedical Informatics, vol. 45, no. 1, pages 101 – 106, 2012.

[Migut 2011] Malgorzata Migut, Jan van Gemert and Marcel Worring. Interactive decision making using dissimilarity to visually represented prototypes. In 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), pages 141–149, oct. 2011.

[Munch 2004] Jurgen Munch and Jens Heidrich. Software project control centers: concepts and approaches. *Journal of Systems and Software*, vol. 70, no. 1, 2, pages 3–19, 2004.

[North 2000] Chris North and Ben Shneiderman. Snap-together visualization: can users construct and operate coordinated visualizations. *International Journal of Human-Computer Studies*, vol. 53, no. 5, pages 715 – 739, 2000.

[Paul 1999] Raymond A. Paul, Toshiyasu L. Kunii, Yoshihisa Shinagawa and Muhammad F. Khan. Software metrics knowledge and databases for project management. *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pages 255 –264, jan/feb 1999.

[Peck 2011] R. Peck, C. Olsen and J.L. Devore. Introduction to statistics and data analysis. Available Titles Aplia Series. Cengage Learning, 2011.

[Perer 2011] A. Perer, I. Guy, E. Uziel, I. Ronen and M. Jacovi. Visual social network analytics for relationship discovery in the enterprise. In 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), pages 71 –79, oct. 2011.

[Proulx 2006] Pascale Proulx, Sumeet Tandon, Adam Bodnar, David Schroh, Robert Harper and WilliamWright. Avian Flu Case Study with nSpace and GeoTime. In 2006 IEEE Symposium on Visual Analytics Science and Technology, pages 27 –34, 31 2006-Nov. 2 2006.

[Reniers 2012] Dennie Reniers, Lucian Voinea, Ozan Ersoy and Alexandru Telea. The Solid* toolset for software visual analytics of program structure and metrics comprehension: From research prototype to product. *Science of Computer Programming*, 2012.

[Robertson 1991] George G. Robertson, Jock D. Mackinlay and Stuart K. Card. Cone Trees: animated 3D visualizations of hierarchical information. In CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 189–194, New York, NY, USA, 1991. ACM Press.

[Savikhin 2008] Anya Savikhin, Ross Maciejewski and David S. Ebert. Applied visual analytics for economic decision-making. In IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08, pages 107–114, oct. 2008.

[Sharp 2011] Helen Sharp, Yvonne Rogers and Jenny Preece. Interaction design: Beyond human-computer interaction. Second edition. John Wiley, UK, 2011.

[Sillito 2006b] Jonathan Sillito, Gail C. Murphy and Kris De Volder. Questions programmers ask during software evolution tasks. In Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering, SIGSOFT '06/FSE-14, pages 23–34, New York, NY, USA, 2006. ACM.

[Theron 2006c] Roberto Theron. Visual Analytics of Paleoceanographic Conditions. In 2006 IEEE Symposium on Visual Analytics Science and Technology, pages 19 –26, 31 2006-Nov. 2 2006.

[Thomas 2005] James J. Thomas and Kristin A. Cook. Illuminating the path: Research and development agenda for visual analytics. IEEE-Press, 2005.

[van Harmelen 2007] Frank van Harmelen, Vladimir Lifschitz and Bruce Porter, editors. Handbook of knowledge representation (foundations of artificial intelligence). Elsevier Science, 2007.

[Voinea 2007] Stefan-Lucian Voinea. Software Evolution Visualization. PhD thesis, Advanced School for Computing and Imaging, Technische Universiteit Eindhoven, 2007.

[Weaver 2006] Chris Weaver, David Fyfe, Anthony Robinson, Deryck Holdsworth and Donna Peuquet. Visual Analysis of Historic Hotel Visitation Patterns. In 2006 IEEE Symposium on Visual Analytics Science and Technology, pages 35 –42, 31 2006-Nov. 2 2006.

[Witten 2005] Ian H. Witten and Eibe Frank. Data mining: Practical machine learning tools and techniques, second edition (Morgan Kaufmann series in data management systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[Ziegler 2010] H. Ziegler, M. Jenny, T. Gruse and D.A. Keim. Visual market sector analysis for financial time series data. In 2010 IEEE Symposium on Visual Analytics Science and Technology (VAST), pages 83 –90, oct. 2010.