

### Introducción a Software Analytics: Visión General desde la Ingeniería de Software

Manuel Maldonado Mendoza, Armando Botero Vila

Departamento de Informática

Universidad Carlos III de Madrid

Madrid - España

{thekingmanuel, armando.botero}@gmail.com

**Abstract:** *Software Analytics (SA) Software analytics is to enable software practitioners to perform data exploration and analysis in order to obtain insightful and actionable information for data-driven tasks around software and services [1], knowledge can pull to need. This paper describes the overall structure with SA, Visual Analytics, Visual SA, Evolutionary SA and the importance of SA in Software Engineering.*

**Resumen:** *Software Analytics (SA) da soporte a las necesidades de los profesionales de software, para tener una visión específica y detallada de los puntos que son relevantes y procesables a partir de una colección de datos, en las tareas de software y servicios [1], para poder extraer el conocimiento necesario. En este artículo se describe la estructura general de SA, Visual Analytics, Visual SA, Evolutionary SA y la importancia de SA en la Ingeniería de Software.*

**Keywords:** *Software Analytics, Visual Analytics, Visual Software Analytics, Evolutionary Software Analytics, Ingeniería de Software.*

## 1. Introducción

En la actualidad, la ingeniería de software se enfrenta a la administración de grandes volúmenes de datos para realizar las distintas actividades que comprenden los proyectos de software, además, esta información se encuentra estructurada en fuentes de datos heterogéneas [2]. El gran tamaño de las aplicaciones y complejidad de sus componentes presenta grandes retos a los profesionales del software, dificultando la ejecución de sus tareas con la calidad necesaria y en el tiempo estimado. Una evidencia de este hecho, son las enormes desviaciones presupuestarias que se originan en los defectos del software [3], aspecto que deja ver que los artefactos con los que se cuenta en la ingeniería de software no son comparables con los desafíos de los desarrollos de los productos software requeridos por el mercado.

Como resultado de lo anterior, surge la necesidad de aplicar herramientas automatizadas que optimicen las actividades que deben realizarse en una implementación de un proyecto de software. Como respuesta a esta necesidad aparece SA, herramienta que se rige por los siguiente pilares: Calidad, Productividad y la Experiencia [4-5].

El presente artículo busca realizar una introducción a la temática, buscando las ventajas y desventajas que este enfoque puede presentar a los proyectos de software. El documento está estructurado de la siguiente manera: sección 2.Visual Analytics, sección 3.Software Analytics, sección 4.Impacto de Software Analytics, sección 5.Visual Software Analytics, 6.Evolutionary Visual Software Analytics, sección 7.Software Analytics aplicada a Ingeniería de Software, sección 8.Frameworks Analytics, sección 9.Ventajas y Desventajas de SA y por último, sección 10.Conclusiones.

## 2. Visual Analytics

Visual Analytics es utilizado en distintos dominios, tales como, monitores de tráfico en la red, procesos bancarios, seguridad de la información, ingeniería de software o ciencia forenses [6]. *Visual Analytics* se ha convertido en una herramienta significativa, principalmente por la facilidad de optimizar tareas tediosas y complicadas necesarias en los sistemas a gran escala. *Visual Analytics* realiza un combinación balanceada entre minería de datos y técnicas de visualización de información, buscando ayudar a los usuarios a razonar y abstraer información relevante de grandes volúmenes de datos [7-8].

De una manera más formal *Visual Analytics* se define en [7] como: “la ciencia del razonamiento analítico facilitado por interfaces visuales interactivas”. La cual tiene como ingrediente principal una sinergia entre minería de datos y técnicas de visualización para analizar características específicas en un conjunto de datos que ayuden a la toma de decisiones. *Visual Analytics* difiere con la minería de datos pura, ya que incorpora un componente de razonamiento a los datos que muestra, aspecto que no es posible en una consulta simple a los datos. Además, el proceso de *Visual Analytics* va más allá de la simple visualización de datos, ya que otorga un grado de interpretación de los datos, el cual, puede cambiar según el contexto, consultas o variación de los bloques de datos [6].

En este contexto, la aplicación de *Visual Analytics* permite a los usuarios realizar análisis exhaustivos para obtener información práctica y detallada de los datos estudiados. Por información detallada se entiende como información que otorga una comprensión significativa y útil o genera algún tipo de conocimiento con respecto al objetivo de la tarea a realizar. Además, información práctica es la que concede herramientas representativas a los usuarios para encontrar soluciones óptimas a los problemas analizados [5].

Un factor esencial para *Visual Analytics* es la interacción que existe entre los usuarios con las herramientas de visualización, ya que es en ésta en donde se generan las ventajas competitivas de las herramientas que siguen este enfoque. Por esta razón, en [2] sintetizan a Visual Analytics como un proceso que combina las ventajas que proporcionan los sistemas computacionales con cualidades de los seres humanos, como la capacidad de análisis, la intuición, la resolución de problemas y la percepción visual.

Visual Analytics tiene distintas implementaciones y procesos dependiendo del dominio o área en donde sea aplicado, buscando reforzar algunos puntos específicos con el objetivo de obtener mejores resultados en el análisis de contextos específicos. Un proceso general que presentan las herramientas de Visual Analytics es: en primer lugar se realiza una extracción de datos en fuentes de datos heterogéneas. Segundo, realizar un tratamiento inicial a los datos extraídos con distintas técnicas, por ejemplo, minería de datos para generar una base de conocimiento. Tercero, analizar y transformar los datos de la base de conocimiento, esto para facilitar la aplicación de técnicas de visualización. Cuarto, aplicar técnicas de visualización, teniendo en cuenta el contexto de los usuarios que utilizaran la herramienta, un ejemplo sería grafos de dependencias. Por último, el usuario interactúa con la herramienta, genera análisis y razonamiento a partir de la visualización de los datos, con la posibilidad de solicitar a la herramienta detallar en algunos aspectos alguna información que le sea interesante [2]. Este proceso se puede observar de una manera gráfica en la Ilustración 1.

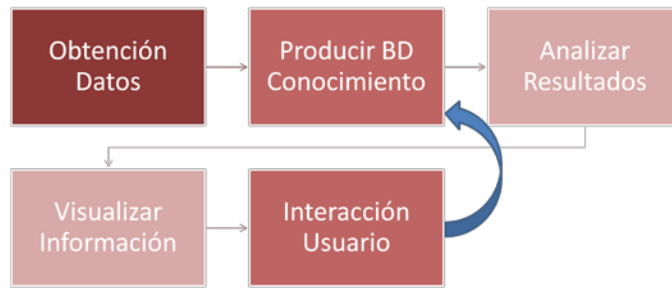


Ilustración 1: Proceso Visual Analytics

### 3. Software Analytics

SA da soporte a los profesionales de software (Ingenieros de soporte, ingenieros de usabilidad, testers, diseñadores, gestores del programa, desarrolladores, administrador del personal) para la exploración y análisis de datos a partir de la minería de datos y, de esta manera, obtener la información relevante y procesable para las tareas del *data-driven* en el software o servicio [5].

En la Ilustración 2 se muestra gráficamente que es SA.



Ilustración 2: Software Analytics

En SA se identifican dos tipos de información [1]:

- Relevante.
- Procesable.

La información relevante es aquella que es útil para entender y comprender las necesidades de los profesionales de software a la hora de transmitir el conocimiento y tener una visión específica del negocio.

La información procesable es la que indica que procesos se tienen que mejorar y de esta forma proporcionar una solución concreta para el negocio (mejor que las soluciones existentes en su caso) [1] a fin de completar la tarea objetivo de los procesos.

Por lo general el ciclo de vida de los proyectos de SA está dividido en 4 fases [1] Las fases que conforman al ciclo de vida son:

- Definición de tareas.
- Preparación de datos.
- Desarrollo de tecnologías analíticas.
- Despliegue y recuperación de información.

Definición de tareas: Es la definición de las tareas objetivo que van a ser asistidas por SA [1]

Preparación de datos: Es la colección de datos que se van a analizar [1].

Desarrollo de tecnologías analíticas: Es desarrollar la estructura del problema, algoritmos, sistemas de análisis y exploración de datos, entendimiento del problema, para obtener una visión final de los datos [1].

Despliegue y recuperación de la información: Generalmente son 2 tipos de escenarios, el de los investigadores y profesionales. El primer escenario es el de ¿Cómo los investigadores han obtenido información relevante?, esto es partiendo de algunos datos recopilados y se requiere la revisión y verificación de los expertos del dominio o profesionales de software [1]. Y el segundo escenario es lo que necesitan los expertos de dominio, para utilizar las herramientas analíticas que se hayan desarrollado, y de esta manera obtener ideas por ellos mismos [1].

Cuando se aplican las tecnologías analíticas en la práctica de SA, se deben observar las siguientes etapas [1]:

- Incorporar una amplia gama de conocimientos y experiencias de los dominios. Por ejemplo (gestión, visualización de la información y procesamiento de datos a gran escala).
- Investigar como los profesionales o expertos lo llevan a la práctica, produciendo información procesable. Y de esta manera dar un soporte efectivo, basado en la información obtenida.

En la Ilustración 3 se muestra el ciclo de vida de SA.



Ilustración 3: Ciclo de vida de SA

#### 4. Impacto Software Analytics

Se realizaron unas búsquedas de los artículos científicos sobre la temática a partir del año 2009 y hasta el 2012, con el motor de scholar.google, y se obtuvieron los siguientes valores que se muestran en la tabla 1 y en la ilustración 4.

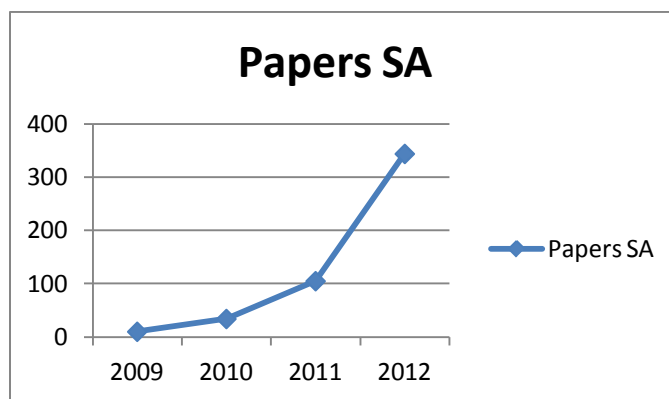


Ilustración 4: Papers SA por año

Como se puede ver en la Ilustración 4, el tema de SA va aumentando por año significativamente, es importante mencionar que se filtraron las patentes y las citas de los artículos para evitar la redundancia en el.

Año	Número de artículos
2009	10
2010	34
2011	105
2012	344

Tabla 1: Impacto de SA teniendo en cuenta las publicaciones sobre la materia

## 5. Visual Software Analytics

Visual SA es una vertiente de Visual Analytics, la cual se enfoca en estudiar los problemáticas presentadas en ámbitos de la Ingeniería de Software. El proceso que la componen es similar al de Visual Analytics, pero la forma en la que se ejecutan cada una de las tareas está dirigida a ofrecer soluciones optimas y aplicables a la ingeriría de software [2].

En este sentido, el principal objetivo de Visual Software Analytics es otorgar los beneficios de las herramientas de Visual Analytics, pero dirigido específicamente a solucionar problemas y optimizar tareas referentes a la disciplina de ingeniería de software.

Los componentes del proceso de Visual SA están diseñados teniendo en cuenta las el contexto de la ingeniería de software, conformado el siguiente proceso: primero la extracción de datos se realiza de repositorios de código fuente, documentos UML, archivos binarios, logs, correos electrónicos. Segundo, la generación de la base de conocimiento se realiza a partir de aplicación de métricas utilizadas en la ingeniería de software, análisis de dependencias en código fuente, minería de datos. Tercero, generación de correlación con los datos de la base de conocimiento, elaboración de diagramas, tales como, grafos o UML, análisis de metadatos. Cuarto, dependiendo el contexto, problemática o tareas que se desee ejecutar se elige la técnica de visualización más adecuada. Por último, el usuario interactúa con la herramienta dependiendo del rol y la tarea que este tenga. La ilustración a continuación describe de forma gráfica el proceso [2].

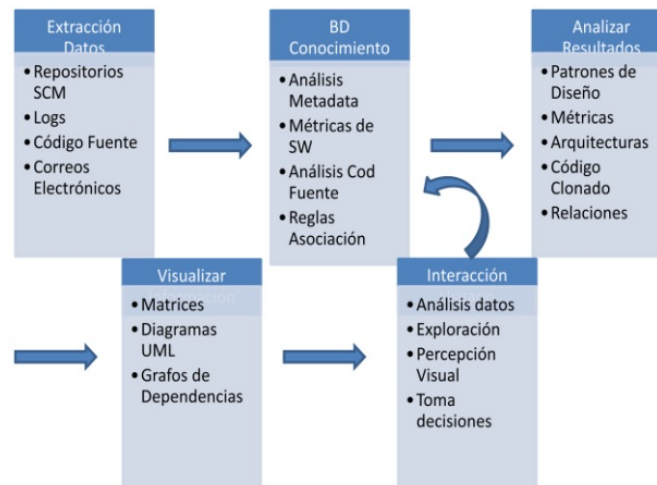


Ilustración 4: Proceso Visual Software Analytics

## 6. Evolutionary Visual SA

Evolutionary visual SA es el proceso que da el soporte al mantenimiento de software, con la partición activa de los usuarios. A través del entendimiento y la comprensión de la evolución del software por los recursos de visual analytics y de la interacción humano computadora. Se trata de un área de aplicación relativamente reciente de visual analytics para la evolución del software [2].

Para realizar la minería en los repositorios de software con procesos evolutivos se requiere el uso en conjunto de técnicas que ofrezcan el soporte de extracción y análisis, para que busquen y descubran patrones, relaciones, métricas de software de calidad [9].

En la Ilustración 5 se muestra la representación del concepto Evolutionary Visual SA.

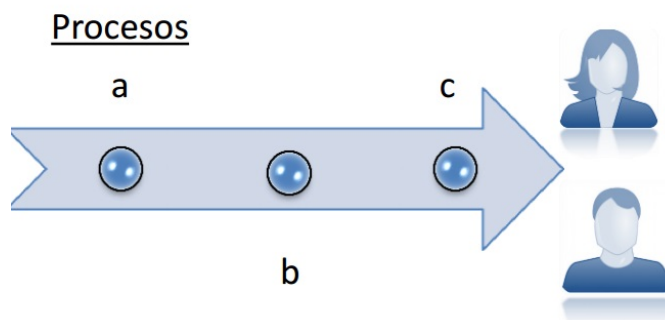


Ilustración 5: Evolutionary Visual Software Analytics

## 7. SA en la Ingeniería del Software

En cada una de las distintas actividades y tareas que componen el ciclo de vida del software, es necesario administrar grandes volúmenes de datos, aspecto que incrementa aún más en la actualidad, ya que las aplicaciones que se construyen tienen dimensiones cada vez mayores y con complejidades más altas, características que implican un aumento de las fuentes de datos.

En este sentido, la ingeniería del software ha implementado distintas estrategias y métodos, que ayudan a construir aplicaciones software más precisas, con la calidad requerida, en el tiempo previsto y con el presupuesto estimado. Cada uno de estos métodos conlleva a la ejecución de tareas que implican administración, transformación y análisis de grandes volúmenes de datos, para así, por ejemplo, poder tomar decisiones, mitigar riesgos o analizar estados del proyecto.

Como consecuencia a lo anterior, surge la necesidad de contar con herramientas que automatizen tareas, buscando optimizar tiempos y facilitando la toma de decisiones. Una respuesta a dicha necesidad son las herramientas de SA, que están diseñadas para analizar información, transformar datos y efectuar un razonamiento sistemático que apoye la toma de decisiones. SA es especialmente útil en la búsqueda de respuestas, tales como, “¿Qué pasó?”, “¿Cómo fue que sucedieron las cosas y por qué sucedieron?” [10], preguntas que surgen frecuentemente en los proyectos de ingeniería de software, y que dependiendo del caso y el contexto la complejidad de encontrar dichas respuestas varía.

SA aplicada a la ingeniería de software busca principalmente, optimizar actividades, automatizar tareas y otorgar herramientas que faciliten la toma de decisiones, todo con el propósito de aumentar la productividad y calidad de los proyectos de software [10].

El proceso de SA se puede observar desde la perspectiva de aplicar técnicas de ingeniería de software a la disciplina de ingeniería de software, es decir, el cliente del proyecto de SA es la misma ingeniería de software.

Además, los datos con los que trabaja provienen de experiencias reales de la ingeniería de software, se deben solucionar problemas reales utilizando los datos y una de las principales métricas de éxito de SA es su influencia e impacto en la práctica de la ingeniería de software [11].

En la actualidad se cuenta con distintas herramientas que soportan actividades de varias etapas de un proyecto de ingeniería de software, estas se diseñan pensando en la tarea a realizar, el contexto en la que se realiza y los roles que tendrán los usuarios en el proyecto. Las herramientas más comunes van enfocadas hacia la ayuda de evaluación de métricas, ayuda al mantenimiento del software y herramientas que analizan código fuente. Esto debido a que estas tareas requieren de una administración de grandes volúmenes de datos y las operaciones para transformar y encontrar relaciones no son triviales [6].

## 8. Frameworks Analytics

Existen herramientas de SA como Solid SX o Software Statistics Service, las cuales se pueden integrar al proyecto de Software de manera sencilla y relativamente rápida, sus frameworks son compatibles con algunos lenguajes de programación como C# o Java. En la tabla 2 se representan las herramientas.

Nombre del Software	Lenguajes de programación compatibles
Solid SX [12]	.NET, Java, Visual C++
Software Statistics Service [13]	NET, C++, Java, Android, Delphi, Microsoft Silverlight, WPF, Windows Phone 7, Mac OS, iOS

Tabla 2: Framework SA

## 9. Ventajas y desventajas

En la tabla 2 se muestran las ventajas y desventajas que se han identificado al utilizar SA.

Ventajas	Desventajas
Proporciona información relevante para la mejora del Software.[5]	La dificultad de interpretar los datos es una barrera importante para el uso de tecnologías analytics [14].
Las herramientas analytics han sido adoptadas exitosamente en algunos productos de Microsoft[5]	La visualización de la información es compleja y requiere de experiencia del usuario.
Cuando se utilizan adecuadamente las tecnologías analytics a lo largo del tiempo pueden ahorrar inspecciones costosas en código fuente [5]	Integrar las tecnologías analytics a proyectos, sin un control de datos bien definido o estructurado.
Son una buena combinación el análisis y la visualización de la información [15].	Los costos de la calidad y colección de datos [16].
Da soporte a la mejora de calidad de los sistemas de software [5]	La gran inversión que se requiere realizar en la compra de frameworks con este enfoque.
Ayuda a la toma de decisiones de los próximos proyectos [10].	
Se puede identificar donde mejorar los procesos de Software [5]	
Existen frameworks que se pueden integrar a lenguajes de programación como por ejemplo a Java o C#.	

Tabla 3: Ventajas y Desventajas



## 10. Conclusiones

La ingeniería de software necesita de herramientas que le ayuden a optimizar y soportar sus tareas, ya que los artefactos con los que se cuentan hoy en día no están cumpliendo a cabalidad las expectativas necesarias para obtener proyectos de calidad, ejecutados en el tiempo estimado y no gastando más recursos del presupuesto asignado.

SA, busca dar respuesta a estas necesidades, otorgando herramientas que automaticen las tareas tediosas y complicadas por el gran volumen de datos que manejan, además, quiere otorgar mecanismos que faciliten el análisis y la toma de decisiones; aspectos de vital importancia en el desarrollo de un proyecto.

Sin embargo, las herramientas de SA no han llegado a satisfacer en la totalidad los requerimientos de la ingeniería de software, principalmente porque sus desventajas son bastante notorias, tales como, la complejidad para los usuarios de entender los diagramas visuales que presentan las herramientas, el mantenimiento a nivel de datos que se debe tener para estas herramientas, y la poca usabilidad.

SA da soporte a los profesionales de software para mejorar en los próximos proyectos, puesto que tienen como base el análisis de los procesos o áreas anteriores y de esta manera puedan identificar los procesos a mejorar, a nivel de código fuente existen estudios realizados por el grupo de Asia de Microsoft [17] en el que SA da soporte a identificar duplicidad de código y así re-factorizar ciertas instancias de código. Existen frameworks que pueden incorporarse a los proyectos, el requerimiento es: que se tengan gestionadas las colecciones de datos.

SA aún se encuentra en etapa exploratoria, y el camino sigue abierto a nuevas investigaciones y al descubrimiento de mejoras que intervengan en las desventajas de estas herramientas. Aunque SA no tiene la llave para solucionar todos los problemas de la ingeniería de software si es un enfoque valido y aplicable en el futuro.

## Referencias

- [1] D. Zhang, Y. Dang, S. Han, and T. Xie, "Teaching and Training for Software Analytics," *2012 IEEE 25th Conference on Software Engineering Education and Training*, no. Iwsc 2011, pp. 92–92, Apr. 2012.
- [2] A. González-Torres, F. J. García-Peñalvo, and R. Therón, "Human–computer interaction in evolutionary visual software analytics," *Computers in Human Behavior*, vol. 29, pp. 486–495, Feb. 2013.
- [3] G. Tasse, "The economic impacts of inadequate infrastructure for software testing," *National Institute of Standards and Technology, RTI Project*, no. 7007.011, 2002.
- [4] S. Han, Y. Dang, S. Ge, D. Zhang, and T. Xie, "Performance debugging in the large via mining millions of stack traces," *2012 34th International Conference on Software Engineering (ICSE)*, pp. 145–155, Jun. 2012.
- [5] D. Zhang, Y. Dang, J. G. Lou, S. Han, H. Zhang, and T. Xie, "Software analytics as a learning case in practice: Approaches and experiences," in *Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering*, 2011, pp. 55–58.
- [6] A. Telea and L. Voinea, "Visual software analytics for the build optimization of large-scale software systems," *Computational Statistics*, vol. 26, no. 4, pp. 635–654, Mar. 2011.
- [7] P. C. Wong and J. Thomas, "Visual Analytics," *IEEE Computer Graphics and Applications*, 24 (5): 20–21, vol. 24, no. PNNL-SA-41935, 2004.
- [8] K. A. Cook and J. J. Thomas, *Illuminating the path: the research and development agenda for visual analytics*. IEEE-Press (2005), 2005.

- [9] M. D'Ambros, H. Gall, M. Lanza, and M. Pinzger, "Analyzing software repositories to understand software evolution," *Software Evolution*, pp. 37–67, 2008.
- [10] R. P. L. Buse and T. Zimmermann, "Information needs for software development analytics," in *Software Engineering (ICSE), 2012 34th International Conference on*, 2012, pp. 987–996.
- [11] D. Zhang, "MSR 2012 Keynote Software Analytics in Practice – Approaches and Experiences," no. Promise 2011, p. 4673, 2012.
- [12] SolidSourceIT, "SolidSX - Software Explorer," 2012. [Online]. Available: <http://www.solidsourceit.com/download/SolidSX-download.html>.
- [13] M. I. Always., "Software Statistics Service." [Online]. Available: <http://www.software-statistics-service.com/>.
- [14] R. P. L. Buse and T. Zimmermann, "Information needs for software development analytics," *2012 34th International Conference on Software Engineering (ICSE)*, pp. 987–996, Jun. 2012.
- [15] M. Van Den Brand, S. Roubtsov, and A. Serebrenik, "SQuAVisiT: A Flexible Tool for Visual Software Analytics," *2009 13th European Conference on Software Maintenance and Reengineering*, pp. 331–332, 2009.
- [16] T. Menzies and T. Zimmermann, "Goldfish bowl panel: Software development analytics," in *2012 34th International Conference on Software Engineering (ICSE)*, 2012, pp. 1032–1033.
- [17] D. Zhang, "Software Analytics in Practice," 2012.