

Dinamica de sistemas aplicada a la industria del software en Colombia

System dynamics applied to industry Software in Colombia

Germán Gonzalo Vargas Sánchez, Jorge Andrick Parra Valencia

Resumen



n este artículo se muestra la construcción de una hipótesis dinámica, aplicada a la industria de desarrollo de software en Colombia, utilizando sistemas dinámicos. Se muestra la construcción de los modelos utilizando software Libre.

Palabras clave: sistemas dinámicos, software, hipótesis, modelos, software Libre.

Abstract



n this article the construction of a dynamic hypothesis, applied to the software development industry in Colombia, using dynamical systems shown. building models using free software is displayed.

Keywords: dynamical systems, software, assumptions, models, Free software.

Recibido / Received: Enero 27 de 2015 Aprobado / Approved: Octubre 22 de 2015

Tipo de artículo / Type of paper: Investigación científica y tecnológica terminada

Afiliación Institucional de los autores / Institutional Affiliation of authors: Universidad El Bosque

Autor para comunicaciones / Author communications: Germán Gonzalo Vargas Sánchez, germanvargas@unbosque.edu.co, gvargas162@unab.edu.co, Jorge Andrick Parra Valencia, e-mail: japparra@unab.edu.co

Los autores declaran que no tienen conflicto de interés.

Introducción

El objetivo básico de la Dinámica de Sistemas es llegar a comprender las causas estructurales que provocan el comportamiento del sistema. Esto implica aumentar el conocimiento sobre el papel de cada elemento del sistema, y ver como diferentes acciones, efectuadas sobre partes del sistema, acentúan o atenúan las tendencias de comportamiento implícitas en el mismo [1] [11] [13].

Como características diferenciadoras de otras metodologías puede decirse que no se pretende predecir detalladamente el comportamiento futuro. El estudio del sistema y el ensayo de diferentes políticas sobre el modelo realizado enriquecerán el conocimiento del mundo real, comprobándose la consistencia de nuestras hipótesis y la efectividad de las distintas políticas. [1] [2] [13]

A través de la dinámica de sistemas es posible la construcción de una hipótesis que permita explicar la relación entre los modelos de desarrollo de software aplicados por la industria de software en Colombia en la productividad empresarial.

Es de vital importancia el desarrollo de esta hipótesis puesto que no existen en la actualidad métricas que permita optimizar el esfuerzo aplicado al utilizar diferentes metodologías para el desarrollo de software en Colombia. [2] [3]

Con respecto a Colombia se observa que las empresas colombianas son reconocidas como “artesanas del software”, debido a que se han concentrado en el desarrollo de software a la medida, es decir adecuando especialmente a las necesidades del cliente. La industria local no ha incurrido en el desarrollo empaquetado, dado que este mercado al ser atendido por multinacionales es demasiado competitivo y existen importantes barreras de entrada. [2] [7]

Utilizando las diferentes metodologías para construir software se han realizado varias investigaciones que tienen como objetivo optimizar el esfuerzo. Algunas de estas se relacionan a continuación:

Métodos Formales:

Investigación que tuvo como objetivo dar a conocer la aplicación de métodos formales en la industria de software y hardware; en la que se tuvo en cuenta los métodos

formales desde el autor Wing Janneth (1990) quien dice que un método formal es una técnica basada en matemáticas, usada para describir sistemas de hardware o software. [4]

Los métodos formales pueden ser aplicados a las Empresas para mejorar sus procedimientos y productos.

Al comparar el método formal con el método convencional se observó que el incremento en el esfuerzo para la realización de un módulo con el método formal, resultó poco significativo comparado con el incremento en la calidad, la cual fue considerable. [4] [5]

Metodologías tradicionales:

Las metodologías tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Letelier. [3].

Metodologías ágiles:

En un estudio realizado por Lapolli se observa que las metodologías ágiles se centran en las personas y no en los procesos; se hace mayor énfasis en la producción del software que en la documentación del mismo. Una característica de las metodologías ágiles es que son de adaptación en lugar de ser predictivas. [8]

Entre las metodologías ágiles se encuentran las metodologías Extreme Programming XP y SCRUM.

Entre las conclusiones del estudio se destacan las siguientes:

- Las metodologías ágiles han sido bien aceptadas por la industria del software y por investigadores en ingeniería de software.

- Los métodos tradicionales se han convertido en pesados e impracticables en ciertas situaciones como por ejemplo el aumento de la cantidad de la burocracia. De la misma manera, modelos de métodos tradicionales y de procesos han sido considerados incapaces para responder a los cambios de las organizaciones, negocios y ambientes técnicos así como no reconocer el talento y la creatividad individual. Desde el cambio de milenio la tendencia está mutando en cierta medida de los métodos tradicionales a los métodos ágiles de desarrollo. Aquellos fueron desarrollados como respuesta y, además, los cambios y los entornos técnicos y de negocio permitieron los usos e invención de métodos de desarrollo ágiles[9].

El estudio realizado producto de la investigación permite la construcción de experimentos que avalan el modelo diseñado y que dan explicación a la hipótesis propuesta.

Los experimentos para la validación del modelo se construyeron utilizando el software Libre System Dynamics versión 1.3

El software systemsdynamics

El proyecto SystemDynamics es organizado por SourceForge.net. El equipo del proyecto lo describe como:

Una aplicación Java gráfica de modelado, visualización y ejecución de modelos de dinámica de sistemas con licencia Gpl V2. Está orientada hacia educación, matemáticas y simulaciones.

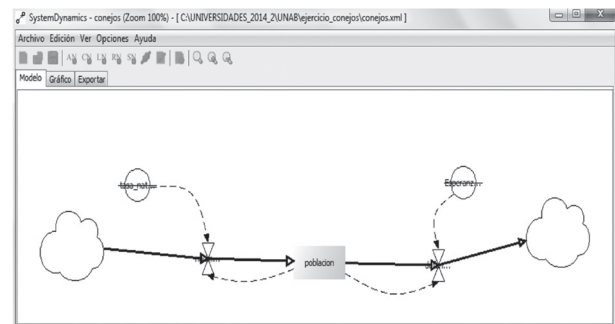
Se decidió utilizar este paquete por el tipo de licencia GPL v2 (totalmente libre y opensource) y por su orientación hacia la educación y simulación matemática, además porque posee una interfase gráfica sencilla, permitiendo la generación de los datos hacia hojas de cálculo así como la generación de gráfica del comportamiento del modelo.[9].

Elementos del software

Diagrama de Forrester:

El Diagrama de Flujos, también denominado Diagrama de Forrester, es el diagrama característico de la Dinámica de

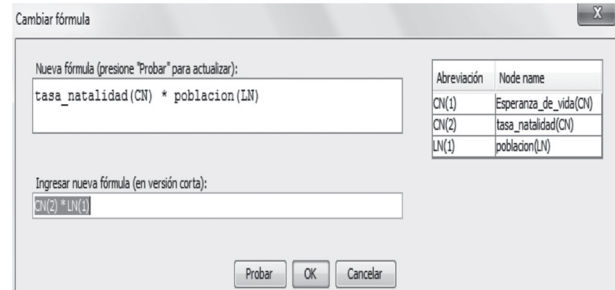
Sistemas. Es una traducción del Diagrama Causal a una terminología que permite la escritura de las ecuaciones en el computador para así poder validar el modelo, observar la evolución temporal de las variables y hacer análisis de sensibilidad.[1][12]



Fuente propia

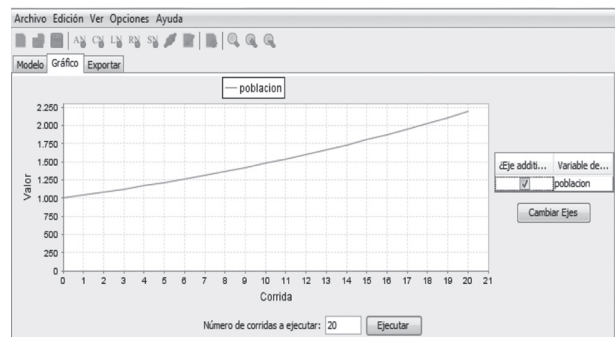
Construcción de fórmulas:

SystemsDynamics permite la construcción de fórmulas que se aplicarán en el modelo de Forrester de la siguiente forma [2]:



Fuente propia.

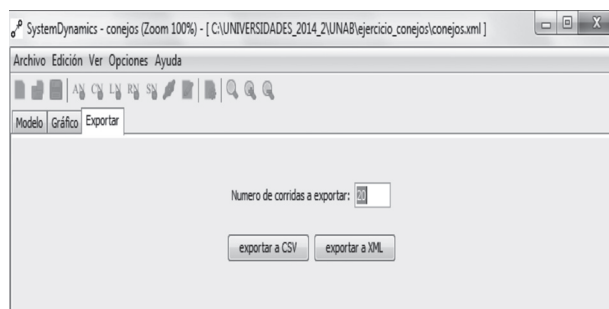
Es posible la creación de la gráfica del modelo utilizando systemsDynamics así:



Fuente propia

Exportación de datos del modelo

El software permite exportar los datos creados por el modelo, con formato cvs ó xml especificándole el número de iteraciones deseado:



Fuente propia.

Metodología

A continuación se enuncian las indicaciones precisas de los pasos a seguir en la creación de un modelo de simulación utilizando la Dinámica de Sistemas.

Es necesario seguir un cierto orden en este proceso.

La primera etapa es construir un diagrama causal donde representamos los elementos que consideramos forman el sistema y las relaciones que existen entre ellos. Es normal que el primer diagrama causal que dibujemos sea incompleto, lo iremos perfeccionando y ampliando si es necesario. La segunda etapa es trasladar este diagrama causal a un diagrama de flujos, que es el formato que pueden manejar los diferentes tipos de software que sirven para realizar este tipo de simulaciones. [1][14][15]

Crear el diagrama causal

- Definir el problema
- Definir las influencias de primer orden
- Definir las influencias de segundo orden
- Definir las influencias de tercer orden
- Definir las relaciones
- Identificar los bucles de realimentación
- Depurar influencias no relevantes
- Idear posibles soluciones al problema

Crear diagramas de flujos

- Caracterizar los elementos
- Escribir las ecuaciones
- Asignar valores a los parámetros
- Crear una primera versión del modelo
- Estabilizar el modelo
- Identificar elementos claves
- Simular [1]

Desarrollo

El problema:

Ausencia de una hipótesis dinámica que relacione los modelos de desarrollo de software con la productividad

Identificación de variables:

Variables de Nivel:

- Tareas (Hace referencia a la cantidad de tareas a realizar)
- Avance (Hace referencia a la cantidad de tareas realizadas)
- Días

Variables de flujo

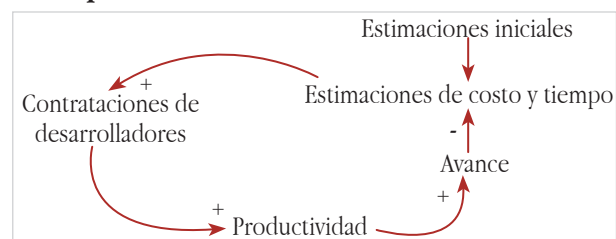
- Productividad
- Cantidad_días
- Variablet

Constantes

- Tasa de Cambio
- Cantidad de Cambios
- Desarrolladores
- Constante de proporción
- Tareas_diarias

A continuación se identifican los bucles del sistema.

Bucle positivo:



#Modelo10		
Días	Avance	Tareas
10	18	32
11	19,8	30,2
12	21,6	28,4
13	23,4	26,6
14	25,2	24,8
15	27	23
16	28,8	21,2
17	30,6	19,4
18	32,4	17,6
19	34,2	15,8
20	36	14
21	37,8	12,2
22	39,6	10,4
23	41,4	8,6
24	43,2	6,8
25	45	5
26	46,8	3,2
27	48,6	1,4
28	50,4	-0,4

Conclusiones

- A través de modelos dinámicos es posible explicar el comportamiento de modelos de construcción de software aplicados en la industria de construcción de software en Colombia.
- Es posible utilizar software Libre para la construcción de modelos dinámicos.
- Es posible diseñar hipótesis dinámicas que representen cómo un modelo de desarrollo afecta la productividad.
- Es posible diseñar el modelo de simulación.
- Se pueden construir experimentos de simulación que permitan evaluar los modelos.

Referencias

- [1] J. García. Teoría y ejercicios prácticos de Dinámica de Sistemas. Juan Martín García, 2003.
- [2] K. Heshusius, “Desafíos y oportunidades de la industria del software en América Latina: Capítulo 5 - Colombia: Desafíos de una industria en formación,” 2009.
- [3] Leterlier, P. Penades, M. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2006
- [4] C. A. Fernandez-y-Fernandez, M. Ambrosio, G. Andrade, G. Cruz, M. José, R. Ortiz, and H. Sánchez, “Métodos formales aplicados en la industria del software,” *Temas Cienc. y Tecnol.*, vol. 15, pp. 3 – 12, 2011..
- [5] E. Montoya Serna, “Métodos formales e Ingeniería de Software,” *Rev. Virtual Univ. Católica ...*, pp. 1–26, 2011.
- [5] Grupo Simon de Investigaciones en modelamiento y simulación. Escuela de Ingeniería Industrial de Santander. Diagrama de Forrester.
- [6] Espinosa, A. Metodologías de desarrollo de software [documento en línea]. Disponible en: « www.slideshare.net/juliopari/4-clase-metodologia-de-desarrollo-de-software » [consulta: 11 de junio de 2012.
- [7] F. S. Marques, “Capítulo 8. América Latina en la industria global de software y servicios: una visión de conjunto,” in *Desafíos y oportunidades de la industria del software en América Latina.*, 2009, pp. 249–307.
- [8] F. Lapolli, C. M. Cruz, C. L. R. Motta, and C. E. T. Oliveira, “Modelo de Desenvolvimento de Objetos de Aprendizagem Baseado em Metodologias Ágeis e Scaffoldings,” *Revista Brasileira de Informática na Educação*, vol. 18, pp. 17–31, 2010
- [9] C. Pardo, J. Hurtado, and C. Collazos, “Agil Software Process Improvement with agile SPI DYNA, pp. 251–263, 2010.
- [10] <http://system-dynamics.sourceforge.net/> Tomada el 12 de Octubre de 2014
- [11] Rodríguez, J. Proportional entropy of the cardiac dynamic systems: Physical and mathematical predictions of the cardiac dynamic for clinical application [Entropía proporcional de los sistemas dinámicos cardíacos: Predicciones físicas y matemáticas de la dinámica cardíaca de aplicación clínica] (2010) *Revista Colombiana de Cardiología*, 17 (3), pp. 115-129. Cited

- 10 times. <http://www.scopus.com/inward/record.url?eid=2-s2.0-78349289542&partnerID=40&md5=8aeda2be79ed0bc3de004d4b3791f3b4>
- [12] Rodriguez-Ulloa, R., Paucar-Caceres, A. Soft System Dynamics Methodology (SSDM): Combining Soft Systems Methodology (SSM) and System Dynamics (SD) (2005) *Systemic Practice and Action Research*, 18 (3), pp. 303-334. Cited 24 times. <http://www.scopus.com/inward/record.url?eid=2-s2.0-22044450172&partnerID=40&md5=e0722549a9c554de191de66c8dbae7fa>
- [13] Zhang, D.-F., Duanmu, J.-S., Lian, W.-X., Liu, D., Zhang, Y. Aviation maintenance safety culture safety management mechanism based on system dynamics simulation (2013) *Advanced Materials Research*, 765-767, pp. 3285-3290. <http://www.scopus.com/inward/record.url?eid=2-s2.0-84885038311&partnerID=40&md5=220e96c52c708a10f9ea8cc62f6f39d1>
- [14] Taborda, J.A., Angulo, F., Olivar, G. Mandelbrot-like bifurcation structures in chaos band scenario of switched converter with delayed-pwm control (2010) *International Journal of Bifurcation and Chaos*, 20 (1), pp. 99-119. Cited 1 time. <http://www.scopus.com/inward/record.url?eid=2-s2.0-77951534090&partnerID=40&md5=df8e9ea505492fab61d2177078218777>
- [15] Sánchez-Román, R.M., Folegatti, M.V., Orellana-González, A.M.G. Water resources situation at piracicaba, capivari and jundiaí watersheds using a dynamic systems model [Situação Dos Recursos Hídricos Nas Bacias Hidrográficas Dos Rios Piracicaba, Capivari E Jundiaí Utilizando Modelo Desenvolvido Em Dinâmica De Sistemas] (2009) *Engenharia Agricola*, 29 (4), pp. 578-590. Cited 7 times. <http://www.scopus.com/inward/record.url?eid=2-s2.0-77955296658&partnerID=40&md5=62d92dbf5ef0b8ae9c6ecb4f63da7268>

Los Autores



German Gonzalo Vargas Sánchez

Ingeniero de Sistemas Universidad Distrital Francisco José de Caldas. Especialista en Ingeniería de software. Especialista en Informática y Ciencias de la Computación. Magister en software Libre, Universidad Autónoma de Bucaramanga Colombia. Estudiante del Doctorado en Pensamiento Complejo en la Multiversidad Mundo Real Edgar Morin México.



Jorge Andrick Parra Valencia

Ph.D., M.Sc., SE. Profesor Titular Programa de Ingeniería de Sistemas UNAB. Líder-CoDirector. Grupo de Investigación en Pensamiento Sistémico. Secretario Comité Institucional de Ética en Investigación CIEI.

Coordinador Académico Programa de Gestión de Sistemas de Información Universidad Autónoma de Bucaramanga Colombia. Presidente Comunidad Colombiana de Dinámica de Sistemas 2011-2012, 2012-2013, 2013-2014. Miembro Capítulo Latinoamericano de la System Dynamics Society.