

# EVALUACION DE MECANISMOS DE COOPERACION ENTRE GENERACIONES DE DESARROLLADORES EN EL PROCESO DE DESARROLLO DE SOFTWARE LIBRE.

Fernando Castillo Zapata <sup>1</sup>, Jorge Andrick Parra Valencia <sup>2</sup>

<sup>1</sup> Ingeniero de sistemas, UNIVERSIDAD AUTONOMA DEL CARIBE – COLOMBIA, Maestría en software libre Convenio universidad abierta de Cataluña España – Universidad autónoma de Bucaramanga, [ingfcastillo@gmail.com](mailto:ingfcastillo@gmail.com)

<sup>2</sup> JORGE ANDRICK PARRA VALENCIA, Ph.D., M.Sc., SE., Profesor titular programa de Ingeniería de sistemas, Investigador asociado Grupo de investigación en pensamiento sistémico. UNIVERSIDAD AUTONOMA DE BUCARAMANGA-COLOMBIA, Presidente comunidad Colombiana de dinámica de sistemas 2011-2012,2012-2013-2014 [andrick.parra@gmail.com](mailto:andrick.parra@gmail.com)

## RESUMEN

En la relaciones de generaciones de desarrolladores de software libre se presentan relaciones complejas entre los desarrolladores activos de nuevas versiones y los desarrolladores activos de re desarrollo, desde el punto de vista de los intercambios entre generaciones y sus inclinaciones por iniciar o continuar con proyectos de desarrollo de software libre.

Son los intercambios entre generaciones y las deserciones o retiros entre estos, los que demarcan la dinámica entre generaciones que al final afectan la disponibilidad del bien común llamado software libre. La cooperación es fundamental para el crecimiento y el aumento de la calidad del software libre, teniendo en cuenta la dinámica de estas generaciones.

Actualmente no es claro como la cooperación entre generaciones de desarrolladores de software libre afecta el desarrollo de software libre.

En la búsqueda de propuesta de solución, se aplicaron los lineamientos metodológicos de la Dinámica de Sistemas para entender como la cooperación basada en confianza puede afectar la acción colectiva de las generaciones de desarrolladores de software libre. Los resultados obtenidos explican como la complejidad dinámica puede dominarse para gestionar la cooperación entre generaciones de desarrolladores de software de forma efectiva en el desarrollo de software libre. Como conclusión final se obtuvo que altos tiempos de re-desarrollo de software afectan directamente la efectividad de la cooperación basada en confianza.

**Palabras Clave:** Cooperación, software libre, mecanismos de cooperación, generaciones de desarrolladores, dinámica de sistemas.

Recibido: 11 de Agosto de 2014. Aceptado  
*Received: Agust 11, 2014. Accepted:*

# EVALUATION OF COOPERATION MECHANISMS BETWEEN GENERATIONS OF DEVELOPERS IN THE DEVELOPMENT PROCESS OF FREE SOFTWARE.

## ABSTRACT

*In the relations of generations of free software developers complex relationships among active developers of new versions and active developers of re development are presented. from the point of view of exchanges between generations and inclinations for initiating or continuing projects of free software development.*

*Are exchanges between generations and between these desertions or withdrawals, which demarcate the dynamic between generations that ultimately affect the availability of free software called common good. Cooperation is essential for growth and increased quality free software. Currently it is unclear how cooperation between generations of free software developers affects the development of free software. Methodological guidelines for system dynamics to understand how cooperation based on trust can affect the collective action of the generations of free software developers were applied. The results explain how dynamic complexity can be mastered for managing cooperation between generations of software developers effectively in the development of free software. The final conclusion was obtained that high times redevelopment of software affects the effectiveness of cooperation based on trust.*

**Keywords:** *Cooperation, free software, cooperation mechanisms, generations of developers, system dynamics.*

## 1. INTRODUCCION.

La cooperación entre generaciones de desarrolladores de software libre es vital para el aumento de la cantidad y calidad del software libre.

Sin embargo es limitado el conocimiento sobre la cooperación entre generaciones de desarrolladores de software libre. Este artículo estudia el efecto de la promoción de la acción colectiva entre generación de desarrolladores de software en el desarrollo de software libre. A continuación se presenta una referencia al problema de la cooperación desde la perspectiva de los dilemas sociales, luego se detallan algunos aspectos que supone la cooperación entre generaciones de desarrolladores de software libre, y finalmente se indica la brecha que será tratada en este artículo.

### 1.1. DILEMAS SOCIALES

Los dilemas sociales son conflictos que los grupo de individuos enfrentan comúnmente, tomando decisiones entre beneficio individual o colectivo [1]. En dichos dilemas sociales todos los miembros del grupo tendrían mejores beneficios si cooperan.

La disponibilidad del recurso y la realimentación del estado del recurso juegan un papel fundamental en la decisión de cooperar, cuya decisión esta frecuentemente ligada al cuando más que al contribuir o no.

Además, los dilemas sociales contemplan dos aspectos:

- Deserción de individuos (no cooperar).
- Decisión propia y personal de contribuir y la medida de esa contribución

Estos aspectos generan toda una estructura de conflictos:

- Conflictos en el nivel
- Conflictos en la localización
- Conflictos en la distribución

Los conflictos generados por los dilemas sociales pueden afectar el desempeño de los grupos pequeños para manejar los recursos compartidos [2] y la capacidad de sostenibilidad de los recursos de gran escala como la atmosfera [8]. La cooperación es factible para enfrentar dilemas sociales a pequeña escala [9] [10].

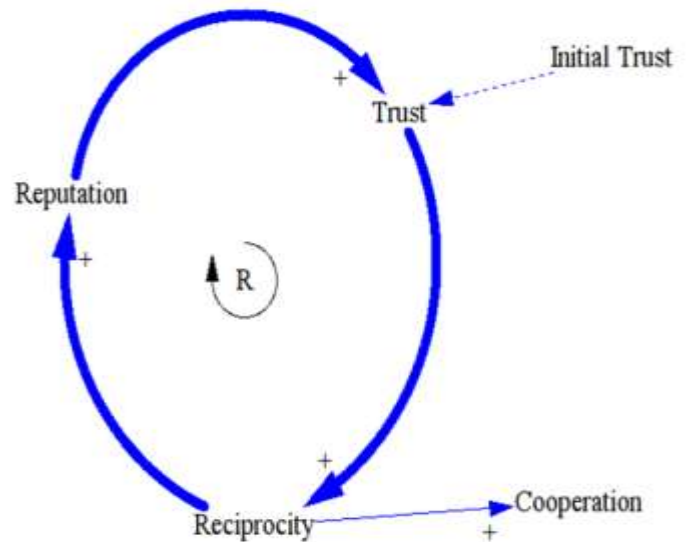


Fig. 1. Versión dinámica genérica del mecanismo de cooperación basado en confianza.

Una versión dinámica del mecanismo basado en confianza promueve reciprocidad. Después, la reputación es afectada por la reciprocidad.

Más reciprocidad produce más reputación e incrementa la cooperación. Finalmente, la reputación mejora la confianza. En términos dinámicos, las condiciones iniciales [3] para la confianza afecta el comportamiento de la cooperación porque estas variables básicas (confianza, reputación y cooperación) están unidas en un ciclo de retroalimentación de refuerzo, que refuerza cualquier condición inicial. Este es el caso con el mecanismo de cooperación basado en confianza que presenta una dependencia a las condiciones iniciales.

### 1.2. PROBLEMAS DE LA COOPERACION BASADA EN CONFIANZA

La dependencia a las condiciones iniciales y los retardos en la información de los resultados de la cooperación, son los problemas identificados para la promoción de la acción colectiva efectiva entre generaciones de desarrolladores de software en el proceso de desarrollo de software libre. No hay una unificación de conceptos a cerca de la posibilidad de aplicar cooperación

basada en confianza en situaciones sociales a gran escala [11].

La anterior circunstancia ha sido descrita por [6] para varias situaciones de cooperación alrededor del cambio climático y crisis de electricidad.

Si las generaciones no disponen de información sobre los resultados de su cooperación, estos podrían dejar de cooperar. Esta situación ha sido estudiada para fenómenos de alta inercia tales como los que tienen que ver con la limitación de gases de alta inercia en la atmosfera tales como el CO<sub>2</sub> [7] y el ozono.

### **1.3. GESTION DE LA COOPERACION**

Este trabajo supone que la cooperación es posible pero su aparición supone una complejidad dinámica. Por lo tanto, se propone estudiar la dinámica que supone la promoción de una cooperación efectiva que supere los problemas estructurales y dinámicos que definen mecanismos de cooperación como el mecanismo basado en confianza. La gestión de la cooperación supone la articulación de reglas, estrategias, normas y mecanismos que permitan mejorar las condiciones para que la dinámica de la cooperación produzca acción colectiva efectiva en la solución de dilemas sociales en grupos. Así, es posible gestionar la cooperación en dilemas sociales a través de mecanismos adicionales combinados con la cooperación basada en la confianza para hacer frente a la dependencia de las condiciones iniciales. Para evaluar esta afirmación, se han desarrollado modelos de simulación para representar el proceso de desarrollo de software integrando el mecanismo de cooperación basado en confianza.

### **1.4. DILEMA DE CONFIANZA**

El dilema de la confianza es un concepto central de la teoría de la acción colectiva. Este dilema explica el proceso de evaluar el riesgo que un individuo asume al confiar en que un actor realizara una acción de cierta manera. En el dilema los individuos evalúan el riesgo o vulnerabilidad generada por la confianza frente a la posibilidad de lograr el comportamiento deseado y los dividendos de la confianza.

### **1.5. GAP**

Luego de la revisión realizada, no se han encontrado trabajos que ofrezcan elementos para una gestión dinámica de la cooperación entre generaciones de desarrolladores de software libre.

Este tipo de investigaciones es pertinente debido que la calidad de la cooperación entre generaciones de desarrolladores de software es clave para aumentar la cantidad y calidad del software libre.

## **2. METODO**

Para el desarrollo de los modelos se aplicaron los lineamientos metodológicos de la Dinámica de Sistemas [4] [5]. Los pasos más importantes son:

- Hipótesis dinámica que explica como el mecanismo de cooperación desarrollado se integra al proceso de desarrollo de software libre entre generaciones de desarrolladores de software.

- Modelo matemático del proceso de desarrollo de software y del funcionamiento del mecanismo de cooperación integrado.

- Simulaciones para evaluar la efectividad del mecanismo de cooperación en la promoción y el sostenimiento de la cooperación.

Se aplico la Dinámica de Sistemas para desarrollar hipótesis dinámicas que representan como la cooperación afecta el desarrollo de software libre y de esta forma evaluar la acción colectiva efectiva del mecanismo de cooperación entre las generaciones de desarrolladores de software libre. Los modelos se desarrollaron usando Vensim 6.1 para Windows.

## **3. RESULTADOS**

Se presentan inicialmente la hipótesis dinámica, el modelo y la simulación correspondiente.

### **3.1. HIPOTESIS DINAMICA**

Se planteo una hipótesis dinámica que expone que a mayor confianza entre generaciones de desarrolladores de software menor tiempo promedio de re-desarrollo y así mismo mayores aplicaciones en re-desarrollo.

En este artículo se realizó una desagregación de la confianza desde el concepto de dilema de confianza. Un dilema de confianza supone el decidir desde la percepción de los costos entre la vulnerabilidad y oportunidad que supone el cooperar con base en la confianza.

Si la confianza no tiene la suficiente calidad en términos de su correspondencia con un comportamiento cooperativo esperado en su respuesta, se genera una vulnerabilidad.

Si la confianza es de la suficiente calidad, se genera una oportunidad de cooperación.

La versión inicial de la hipótesis dinámica:



Fig. 2a. Hipótesis dinámica. Propone la integración del mecanismo de cooperación basado en confianza con la cooperación entre generaciones de desarrolladores de software.

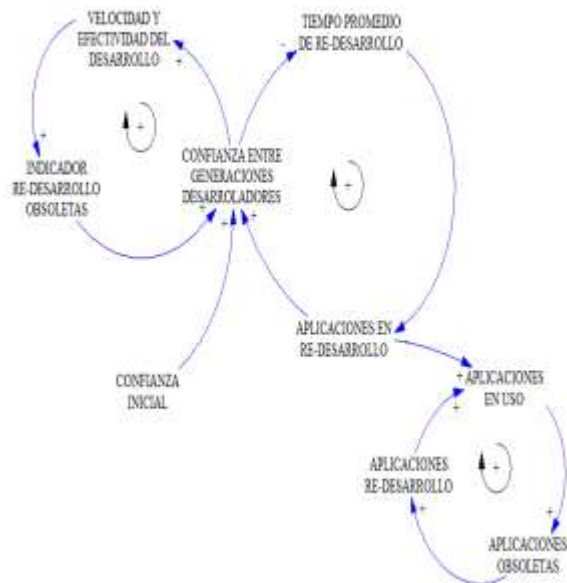


Fig. 2b. Hipótesis dinámica que propone la integración del mecanismo de cooperación

basado en confianza con la cooperación entre generaciones de desarrolladores de software de una manera más ampliada.

### 3.2. MODELO DE SIMULACION

El modelo de simulación presenta como la confianza entre generaciones de desarrolladores de software se representa como un nivel que se alimenta de la percepción retardada de las aplicaciones en re-desarrollo. La confianza degrada mediante una vida media.

La confianza afecta el re-desarrollo que a su vez afecta las aplicaciones en uso. El ciclo se define como del tipo refuerzo.

Finalmente se ilustra como las cantidades de aplicaciones obsoletas se aumentan a medida que aumentan las aplicaciones en uso.

La cantidad de desarrolladores nuevos que se convierten a desarrolladores de re-desarrollos combinado con los desarrolladores de re-desarrollo, teniendo en cuenta la productividad de las dos generaciones aumentan el indicador de re-desarrollo de obsoletas que a su vez aumenta la construcción de la confianza.

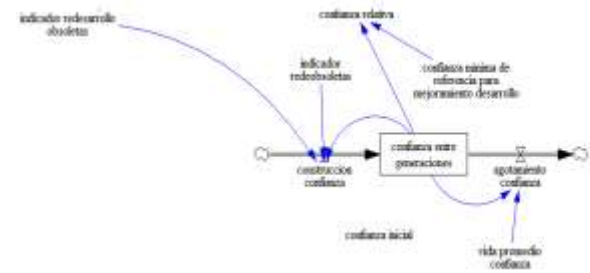


Fig. 3a. Modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Parte 1 de 3.

Ecuación diferencial asociada a la variable de nivel.

$$\frac{d(\text{confianza entre generaciones})}{dt} = \text{construccion confianza}(t) - \text{agotamiento confianza}(t)$$

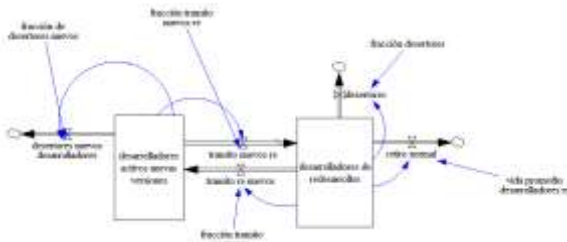


Fig. 3a. Modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Parte 2 de 3 Generaciones de desarrolladores.

Para las variables de nivel, se presentan las siguientes ecuaciones diferenciales:

$$\frac{d(\text{desarrolladores activos nuevas versiones})}{dt}$$

$$= \text{transito de RE a NUEVO } (t) - \text{desertores nuevos desarro } (t) - \text{transito de NUEVO a RE}$$

$$\frac{d(\text{desarrolladores de redesarrollos})}{dt}$$

$$= \text{transito de NUEVO a RE } (t) - \text{desertores } (t) - \text{retiro normal } (t) - \text{transito de RE a nuevos } (t)$$



Fig. 3a. Modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Parte 3 de 3 Aplicaciones.

Ecuación diferencial asociada a la variable de nivel.

$$\frac{d(\text{aplicaciones en desarrollo})}{dt}$$

$$= \text{nuevos proyectos de aplicaciones } (t) - \text{desarrollo } (t)$$

$$\frac{d(\text{aplicaciones en uso})}{dt}$$

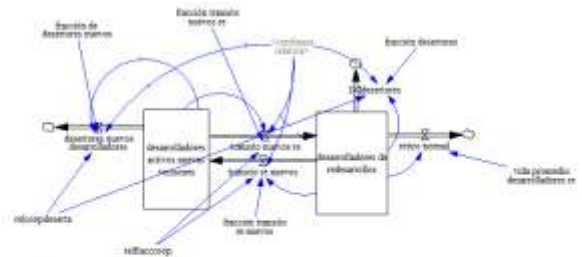
$$= \text{desarrollo } (t) + \text{redesarrollo}(t) - \text{obsolescencia } (t)$$

$$\frac{d(\text{aplicaciones obsoletas})}{dt}$$

$$= \text{obsolescencia } (t) - \text{rescate aplicaciones } (t)$$

$$\frac{d(\text{aplicaciones en redesarrollo})}{dt}$$

$$= \text{rescate aplicaciones } (t) - \text{redesarrollo } (t)$$



1. Confianza relativa
2. rel coop deserta
3. rel facc coop
4. Vida promedio desarrolladores de redesarrollo

Fig. 3b. Nuevo modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Parte 1 de 2.

Ecuación diferencial asociada a la variable de nivel.

$$\frac{d(\text{desarrolladores activos nuevas versiones})}{dt}$$

$$= \text{transito de RE a NUEVO } (t) - \text{desertores nuevos desarro } (t) - \text{transito de NUEVO a RE}$$

$$\frac{d(\text{desarrolladores de redesarrollos})}{dt}$$

$$= \text{transito de NUEVO a RE } (t) - \text{desertores } (t) - \text{retiro normal } (t) - \text{transito de RE a nuevos } (t)$$



Fig. 3b. Nuevo modelo que propone la integración del mecanismo de cooperación basado en confianza con cooperación entre generaciones de desarrolladores de software libre. Parte 2 de 2 Aplicaciones.

Para las variables de nivel, se presentan las siguientes ecuaciones diferenciales:

$$\frac{d(\text{aplicaciones en desarrollo})}{dt} = \text{nuevos proyectos de aplicaciones}(t) - \text{desarrollo}(t)$$

$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia}(t) - \text{rescate aplicaciones}(t)$$

$$\frac{d(\text{aplicaciones en redesarrollo})}{dt} = \text{rescate aplicaciones}(t) - \text{redesarrollo}(t)$$

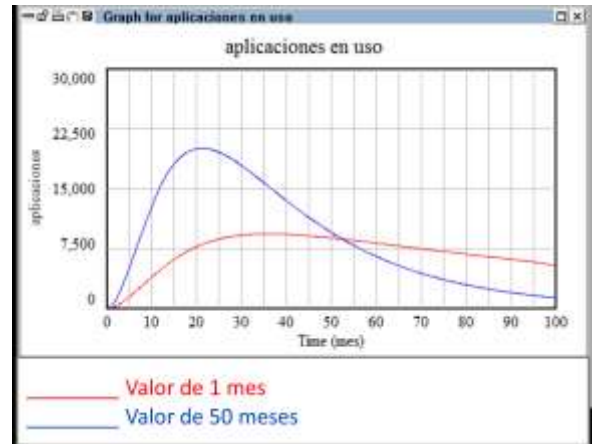
### 3.3. RESULTADO DE LA SIMULACION

Durante el proceso de re-desarrollo de aplicaciones se va aumentando la confianza entre generaciones de desarrolladores de software.

La confianza relativa y teniendo en cuenta las generaciones de desarrolladores de re-desarrollo, la generaciones de desarrolladores de desarrollo que hacen transito a desarrollar re-

desarrollo y la productividad de cada generación de desarrolladores, aumentan las aplicaciones en uso.

Una mayor cantidad aplicaciones en uso genera una mayor cantidad de aplicaciones obsoletas, desde la cual se pueden obtener un alto índice de aplicaciones que entran en etapa de re-desarrollo, de esta manera aumenta la confianza entre generaciones de desarrolladores de software.



Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones en uso

Figura 4a. Resultados para niveles altos y bajo de confianza inicial en referencia a las aplicaciones en uso.

$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$



Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones en re-desarrollo

Figura 4b. Resultados para niveles altos y bajo de confianza inicial en referencia a las aplicaciones en re-desarrollo.

$$\frac{d(\text{aplicaciones en redesarrollo})}{dt} = \text{rescate aplicaciones } (t) - \text{redesarrollo } (t)$$



Conclusión: a mayor confianza inicial mayor cantidad de aplicaciones obsoletas

Fig. 4c. Resultados para niveles altos y bajo de confianza inicial en referencia a las aplicaciones obsoletas.

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia } (t) - \text{rescate aplicaciones } (t)$$

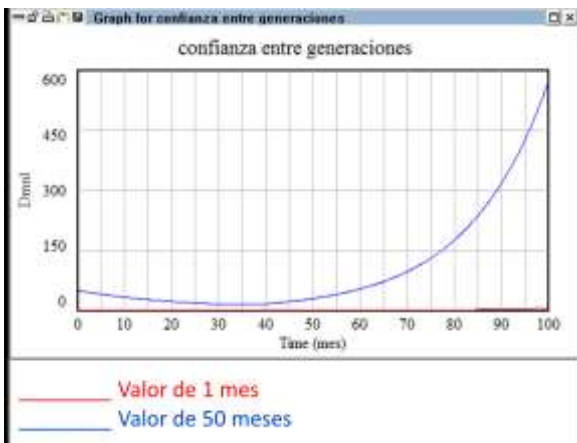


Fig. 4d Resultados para niveles altos y bajos de confianza inicial en referencia a la confianza entre generaciones.

$$\frac{d(\text{confianza entre generaciones})}{dt} = \text{construccion confianza } (t) - \text{agotamiento confianza } (t)$$

### 3.4. ANALISIS DE SENSIBILIDAD

Se realizo un análisis de sensibilidad consistente en 200 simulaciones que variaron de manera uniforme la confianza inicial.

Las siguientes figuras presentan los espacios de probabilidad para el parámetro confianza inicial para diferentes variables descritas.

El análisis de sensibilidad demuestra que la confianza depende de las condiciones iniciales [3] de la confianza de cooperación lo cual afecta la confianza entre generaciones.

A mayor confianza inicial mayor confianza entre generaciones, debido al mejoramiento de la acción colectiva entre generaciones de desarrolladores de software.

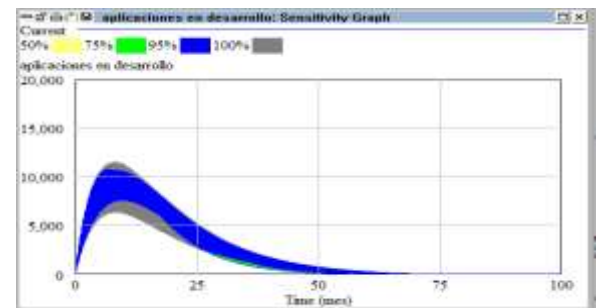


Fig. 5a. Análisis de sensibilidad para las aplicaciones en desarrollo para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

$$\frac{d(\text{desarrolladores de redesarrollos})}{dt} = \text{transito de NUEVO a RE } (t) - \text{desertores } (t) - \text{retiro normal } (t) - \text{transito de RE a nuevos } (t)$$



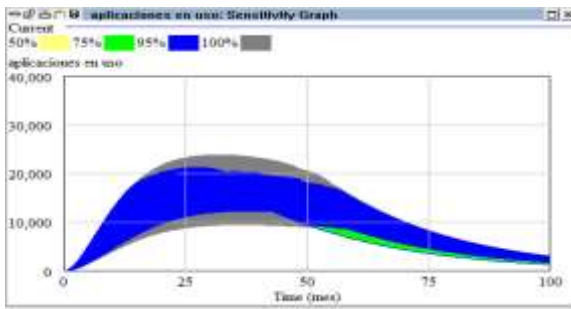


Fig. 5b. Análisis de sensibilidad para las aplicaciones en uso para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

$$\frac{d(\text{aplicaciones en uso})}{dt} = \text{desarrollo}(t) + \text{redesarrollo}(t) - \text{obsolescencia}(t)$$

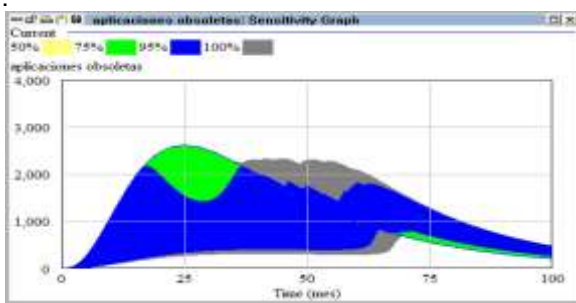


Fig. 5c. Análisis de sensibilidad para las aplicaciones obsoletas para el mecanismo de cooperación basado en confianza en el proceso de cooperación entre generaciones de desarrolladores de software libre.

$$\frac{d(\text{aplicaciones obsoletas})}{dt} = \text{obsolescencia}(t) - \text{rescate aplicaciones}(t)$$

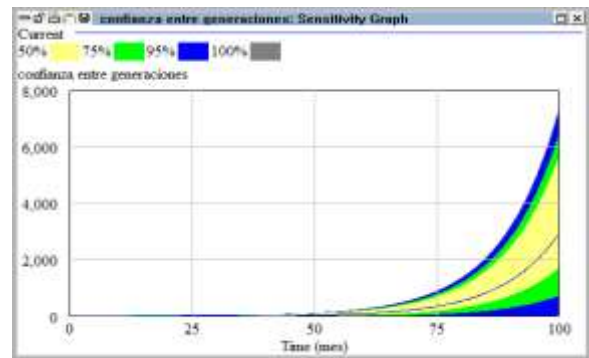


Fig. 5d. Análisis de sensibilidad para la Confianza entre generaciones para el Mecanismo de cooperación basado en confianza en el proceso de cooperación entre Generaciones de desarrolladores de software Libre.

$$\frac{d(\text{confianza entre generaciones})}{dt} = \text{construcción confianza}(t) - \text{agotamiento confianza}(t)$$

#### 4. DISCUSION

En este artículo podemos resaltar la importancia y la relación directa de la cooperación con todo el sistema de desarrollo de software libre y sus relaciones inter generacionales. Cuando aumenta la cooperación aumentan las aplicaciones en desarrollo, aumentas las aplicaciones obsoletas y aumentas las aplicaciones en re desarrollo.

Los análisis previos nos indican la forma en que la cooperación nos permite que desarrollemos mas software y obtengamos mas aplicaciones obsoletas.

El modelo nos muestra como al tener más aplicaciones obsoletas, el uso de estas nos permite incrementar la cantidad de aplicaciones en re desarrollo, lo cual afecta directamente la confianza entre generaciones de desarrolladores.

El anterior resultado fue alcanzado aplicando los lineamientos metodológicos de la dinámica de sistemas, con los cuales se demuestra que la gestión dinámica de la cooperación inter generacional basada en el mecanismos de la confianza puede ser efectiva y sostenible.

Ostrom [2] muestra el mecanismo de cooperación basado en confianza, pero no menciona el efecto sobre la cooperación entre

generaciones de desarrolladores de software libre.

Castillo menciona el problema de la dependencia a las condiciones iniciales [3] del mecanismo de cooperación basado en confianza, pero no evalúa el mejoramiento de la acción efectiva entre generaciones de desarrolladores de software.

Parra y Dyner han reportado el problema en dilemas sociales de gran escala tales como la acumulación de CO<sub>2</sub> y el deterioro del OZONO estratosférico [7], pero no en el desarrollo de software libre. Lo anterior supone que este hallazgo es un aporte al área.

## 5. CONCLUSION

El retardo en los tiempos de re-desarrollo afecta la cantidad de aplicaciones en re-desarrollo lo cual a su vez afecta la efectividad de la cooperación entre generaciones de desarrolladores de software.

Incentivar las aplicaciones en uso, incrementa las aplicaciones obsoletas, desde las cuales obtenemos un incremento de las aplicaciones en re desarrollo, nos afecta de manera positiva la confianza entre generaciones de desarrolladores.

## 6. AGRADECIMIENTOS

Constantemente agradeciéndole a DIOS por la claridad y sabiduría durante la maestría.

A mi familia por el apoyo y la comprensión.  
Especial agradecimiento al Dr. Pascual Matera Lajud por su apoyo constante e incondicional.

## 7. REFERENCIAS

[1] P. Kollock, "Social dilemmas: The anatomy of cooperation," *Annual Review of Sociology*, vol. 24, no. 1, pp. 183–214, 1998.

[2] E. Ostrom, "A behavioral approach to the rational choice theory of collective action," in *Polycentric games and institutions: readings from the Workshop in Political Theory and Policy Analysis*. University of Michigan Press, 2000, p. 472.

[3] D. Castillo and A. Saysel, "Simulation of common pool resource field experiments: a

behavioral model of collective action," *Ecological Economics*, vol. 55, no. 3, pp. 420–436, 2005.

[4] J. Sterman, *Business dynamics: Systems thinking and modeling for a complex world with CD-ROM*. Irwin/McGraw-Hill, 2000.

[5] J. Forrester, *Industrial Dynamics*. MIT press Cambridge, MA, 1961.

[6] J. Parra, "Mecanismo de cooperación en dilemas sociales de recurso agotable de gran escala," Ph.D. dissertation, Universidad Nacional de Colombia. Doctorado en Ingeniería Area Sistemas, 2010.

[7] J. Parra and I. Dyner, "Cooperation mechanism for large-scale social dilemmas involving resource depletion," in *Proceedings International System Dynamics Conference*. System Dynamics Society, 2010, p. 72.

[8] S. Buck, *The global commons: an introduction*. Island Press, 1998.

[9] E. Ostrom and J. Walker, *Trust and reciprocity: Interdisciplinary lessons from experimental research*. Russell Sage Foundation Publications, 2005.

[10] E. Ostrom, T. Dietz, N. Dolsak, P. Stern, S. Stonich et al., *The drama of the commons*. National Research Council, 2002.

[11] M. McGinnis and E. Ostrom, "Will Lessons from Small-Scale Social Dilemmas Scale Up?" *New issues and paradigms in research on social dilemmas*, pp. 189–211, 2008.

## 8. CURRICULUM

JORGE ANDRICK PARRA VALENCIA, es Ph.D., M.Sc., SE., Profesor titular programa de Ingeniería de sistemas, Investigador asociado Grupo de investigación en pensamiento sistémico, presidente comunidad Colombiana de dinámica de sistemas 2011-2012, 2012-2013.

FERNANDO CASTILLO ZAPATA, es ingeniero de sistemas de la Universidad autónoma del Caribe – Colombia estudiante maestría en software libre convenio universidad abierta de Cataluña España – universidad autónoma de Bucaramanga Colombia.