

Tiempo medio entre fallas: investigación de confiabilidad de proyectos ágiles y estudio comparativo de modelos

Gabriel R. Pena y Néstor R. Barraza

Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina
gpena@untref.edu.ar

Resumen En este trabajo abordamos el problema de medir la confiabilidad de un sistema de software mediante el estudio del tiempo medio entre fallas, parámetro ya ampliamente observado en la literatura. A partir de varios conjuntos de datos reales, encuadrados bajo el paradigma de desarrollo ágil, investigamos las características generales del tiempo medio entre fallas en este tipo de proyectos. Se presenta un análisis comparativo del comportamiento de varios modelos de confiabilidad, Musa-Okumoto, Goel-Okumoto, Delayed S-shaped, Logístico y un modelo basado en contagio propuesto por los autores. Los resultados evidencian la necesidad de un análisis multietapa donde se hace necesario ajustar la primera etapa de fallas creciente y la predicción del momento donde comienza el crecimiento de la confiabilidad.

Keywords: Tiempo medio entre fallas · Confiabilidad · Software · Ágiles.

1. Introducción

Con el auge del software en la década de los 70 surgió la necesidad de un marco teórico para el control de calidad. Una medida importante de la calidad de un sistema es su *confiabilidad*, métrica importante y ya entonces ampliamente estudiada, definida como "la probabilidad de que un sistema se comporte adecuadamente durante el tiempo esperado bajo las condiciones encontradas" [1]. Si bien todos los estudios previos en ingeniería de confiabilidad se habían enfocado en sistemas físicos y electrónicos (ver por ejemplo [7]), parecía natural investigar su posible extensión al campo del software, lo que fue hecho exhaustivamente en la década de 1980 por autores como Musa [13] y Yamada [21]. Desde entonces, la ingeniería de software ha evolucionado de forma constante, cambiando no solamente las tecnologías utilizadas sino también las modalidades de desarrollo. Con el advenimiento de las metodologías ágiles en la última década, la utilización del tradicional modelo en cascada (*waterfall*) ha quedado restringida a ciertos ámbitos particulares. Resulta necesario, pues, lograr una adecuada actualización de los modelos de confiabilidad (diseñados originalmente bajo la suposición del desarrollo en cascada) a las nuevas formas de trabajo en

la ingeniería de software. El presente trabajo tiene como objetivo complementar las publicaciones previas de los autores (ver [14,3,4,5]) con un análisis orientado específicamente al **tiempo medio entre fallas** (*MTBF*) en el marco de desarrollos ágiles. Dicha métrica ya ha sido ampliamente estudiada en los modelos tradicionales basados en procesos no homogéneos de Poisson (ver por ejemplo [20,12]). Nos proponemos aplicar esos mismos modelos a productos de software modernos, construidos bajo el paradigma ágil, y realizar un estudio comparativo para discutir su aplicabilidad; asimismo, introducimos en el estudio un modelo moderno, original de los autores de este artículo, y comparamos su eficiencia con la de aquellos. Este trabajo se desarrolla de la siguiente manera: el marco teórico se presenta en la sección 2, los experimentos sobre casos reales se presentan en la sección 3, finalmente, las conclusiones se desarrollan en la sección 4.

2. Marco teórico

2.1. Procesos de nacimientos

Los modelos de confiabilidad que aquí trataremos están basados en procesos de Markov de tiempo continuo (ver [9,17,19]). En estos modelos $P_n(t)$, la probabilidad de que hayan ocurrido exactamente n fallas en el intervalo $(0, t)$, satisface el sistema de ecuaciones diferenciales estocásticas (1), para todo $n \in \mathbb{N}_0$,

$$\begin{cases} P'_n(t) = -\lambda_n(t)P_n(t) + \lambda_{n-1}(t)P_{n-1}(t) & n > 0, \\ P'_0(t) = -\lambda_0(t)P_0(t), \end{cases} \quad (1)$$

La Ecuación (1) se llama *ecuación de nacimientos puros* [9]. La cantidad de fallas n se denomina estado del proceso, y t se mide en unidades de tiempo. La función $\lambda_n(t)$, usualmente llamada tasa de fallas o función de intensidad [1] es el parámetro identificador que determina los diferentes modelos basados en esta clase de procesos, como veremos en la Sección 2.2 para algunos casos particulares notables. Aunque no siempre es posible (y en general no es sencillo) obtener la solución exacta de 1, en la práctica suele bastar con conocer la función de valor medio del proceso estocástico, que llamamos $\mu(t)$. El tratamiento práctico habitual (ver por ejemplo [5]) consiste en ajustar los datos reales a la curva de valor medio de algún modelo. En la Figura 1 se muestran, a modo de ejemplo, las curvas de valor medio de varios modelos ajustados a los datos del clásico proyecto NTDS (ver [10]).

2.2. Procesos no homogéneos de Poisson

Un caso particular importante de la Ecuación (1) se presenta cuando $\lambda_n(t) = \lambda(t)$, es decir, la tasa de fallas es independiente del tiempo transcurrido. En dicho caso la solución tiene una forma explícita dada según (2),

$$P_n(t) = \frac{[\mu(t) - \mu(t_0)]^n}{n!} e^{-[\mu(t) - \mu(t_0)]}, \quad (2)$$

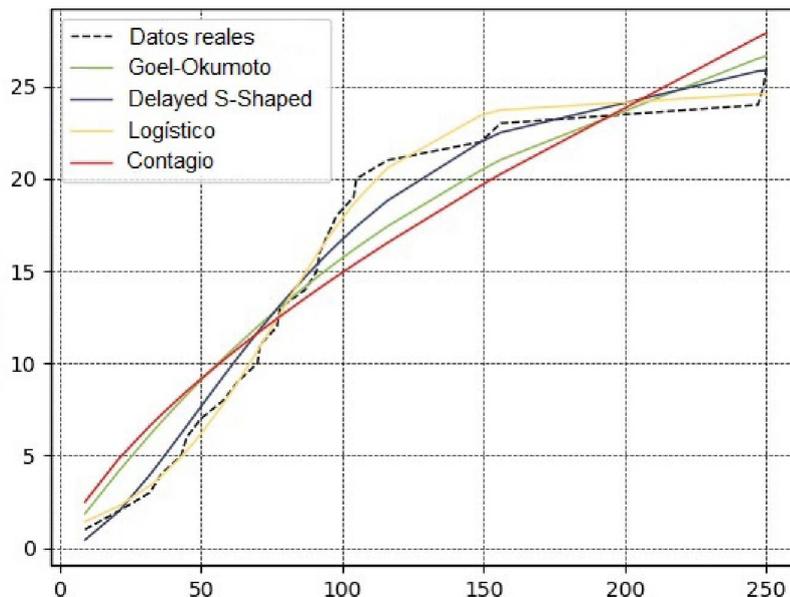


Fig. 1. Ajuste de varios modelos de confiabilidad a los datos del proyecto NTDS. En el eje horizontal, el tiempo en días. En el eje vertical, la cantidad acumulada de fallas detectadas.

donde $\mu(t) - \mu(t_0) = \int_{t_0}^t \lambda(u) du$. En este caso particular, la integral indefinida $\int \lambda(u) du$ coincide con la función de valor medio del proceso, razón por la que la llamamos $\mu(t)$. Esta clase de procesos han sido ampliamente estudiados en la literatura (ver [13,21,8,10]), puesto que casi todos los modelos tradicionales de confiabilidad de software pertenecen a ella, incluyendo los de Musa-Okumoto, Goel-Okumoto, Delayed S-Shaped y Logístico (Tabla 1), que consideramos en este trabajo. Por la relación antes mencionada entre la tasa de fallas y la función de valor medio, es indistinto utilizar una u otra para identificar a cada modelo (únicamente tratándose de procesos no homogéneos de Poisson). Notar que para el caso particular $\lambda(t) = \lambda$, el modelo se reduce al proceso de Poisson convencional.

2.3. Tiempo medio entre fallas

Para todos los procesos listados en la Tabla 1, y en general para todos los que satisfagan la Ecuación (2), es posible obtener una expresión para la función

Tabla 1. Modelos de confiabilidad tradicionales basados en procesos no homogéneos de Poisson

Modelo	$\mu(t)$	Parámetros
Musa-Okumoto	$a \ln(1 + bt)$	$a > 0, \quad 0 < b < 1$
Goel-Okumoto	$a(1 - e^{-bt})$	$a > 0, \quad 0 < b < 1$
Delayed S-Shaped	$a(1 - (1 + bt)e^{-bt})$	$a > 0, \quad 0 < b < 1$
Logístico	$\frac{a}{1 + e^{-b(t-\sigma)}}$	$a > 0, \quad 0 < b < 1, \quad c > 0$

densidad de probabilidad (*PDF*) del tiempo hasta la k -ésima falla (3), ver por ejemplo [12] y [16].

$$f(t_k | t_k < \infty) = \lambda(t_k) \frac{[\mu(t_k) - \mu(t_0)]^{k-1}}{\gamma(L, k)} e^{-(\mu(t_k) - \mu(t_0))}, \quad (3)$$

donde $\gamma(L, k)$ es la función gamma incompleta (4)

$$\gamma(L, k) = \int_0^L z^{k-1} e^{-z} dz, \quad (4)$$

y $L = \lim_{t \rightarrow +\infty} \mu(t) - \mu(t_0)$.

El tiempo medio hasta la k -ésima falla (*MTTF*) se define como la esperanza matemática de la variable aleatoria t_k (5)

$$E[t_k] = \frac{\int_0^{+\infty} t_k \lambda(t_k) [\mu(t_k) - \mu(t_0)]^{k-1} e^{-(\mu(t_k) - \mu(t_0))} dt_k}{\gamma(L, k)}. \quad (5)$$

El tiempo entre fallas s_k se define como la diferencia entre dos tiempos de falla sucesivos (conviniendo en definir $s_1 = t_1$); en consecuencia, el **tiempo medio entre fallas** (*MTBF*) es (6)

$$E[s_k] = E[t_k] - E[t_{k-1}]. \quad (6)$$

Es importante notar que para calcular los tiempos medios *MTTF*, y por consiguiente los *MTBF* es necesario resolver una integral que depende de las funciones $\mu(t)$ y de $\lambda(t)$, distintas para cada modelo, lo que implica que previamente hay que efectuar la estimación de los parámetros del modelo, lo que puede hacerse [5] ajustando los datos a $\mu(t)$ por mínimos cuadrados, o bien por el método de máxima verosimilitud utilizando las Ecuaciones (3) o (2) según sea el formato de los datos.

Para procesos más generales, como los procesos de nacimiento puros [9], la Ecuación (3) deja de ser válida y se requiere un tratamiento distinto, como el que proponemos en la Sección 2.4 y el Apéndice A.

2.4. El proceso de contagio

En trabajos previos (ver [3,4]) los autores de éste trabajo hemos propuesto un modelo propio basado en el conocido proceso de contagio de Polya [2].

Introducimos aquí una versión corregida del mismo. Nuestro modelo toma los siguientes supuestos:

1. El código es probado todo el tiempo y cada falla detectada es corregida de inmediato.
2. Nuevo código es agregado al sistema constantemente. Esto introduce nuevas fallas a una tasa proporcional al número de fallas detectadas (ley de contagio).
3. Debido al proceso de *testing* y corrección, la cantidad de fallas decrece proporcionalmente al tiempo transcurrido.

Tales hipótesis conducen a la expresión de la Ecuación (7) para la función de intensidad:

$$\lambda_n(t) = a \frac{1 + bn}{1 + at}. \quad (7)$$

Debido a la dependencia de la intensidad sobre la cantidad de fallas n , dicha función corresponde a un proceso markoviano con incrementos dependientes, y por consiguiente no son válidas las Ecuaciones (2)-(6), y debe estudiarse como caso particular partiendo únicamente de la ecuación de nacimientos puros (Ecuación (1)). En [6] obtenemos una solución exacta para la función de probabilidad $P_n(t)$ de este modelo mediante el procedimiento descrito en [18]. Para hallar la función de valor medio el método es más sencillo: multiplicando la Ecuación (1) por n y sumando se deduce la ecuación diferencial dada por la Ecuación (8),

$$\mu'(t) = \sum_{n=1}^{+\infty} \lambda_{n-1}(t)P_{n-1}(t), \quad (8)$$

que para $\lambda_n(t)$ dada como en la Ecuación (7) más la condición de 0 fallas en $t = t_0$ tiene como solución la Ecuación (9)

$$\mu(t) = \frac{1}{b} [(1 + at)^b - 1] \quad (9)$$

para $b \neq 0$. En el caso particular $b = 0$, el modelo se reduce al de Musa-Okumoto (Tabla 1) con parámetros $a_{MO} = 1$ y $b_{MO} = a$. La función de valor medio de la Ecuación (9) tiene la particularidad de que su forma cambia dependiendo del valor del parámetro b . Para $0 < b < 1$, $\mu(t)$ es una curva de concavidad negativa, como se muestra en la Figura 2a, lo que permite modelar un crecimiento de la confiabilidad. En cambio, si es $b > 1$, la curva de valor medio tiene concavidad positiva, como se muestra en la Figura 2b, modelando así una tasa de fallas creciente.

Para calcular el *MTBF*, primero derivamos una expresión para la *PDF* del tiempo entre dos fallas sucesivas (Ecuación (10)),

$$f(t|t_{n-1}, t_n < \infty) = \frac{\lambda_{n-1}(t) \cdot \exp\left(-\int_{t_{n-1}}^{t_n} \lambda_{n-1}(u)du\right)}{1 - \exp\left(-\int_{t_{n-1}}^{\infty} \lambda_{n-1}(u)du\right)}, \quad (10)$$

6 G. R. Pena y N. R. Barraza

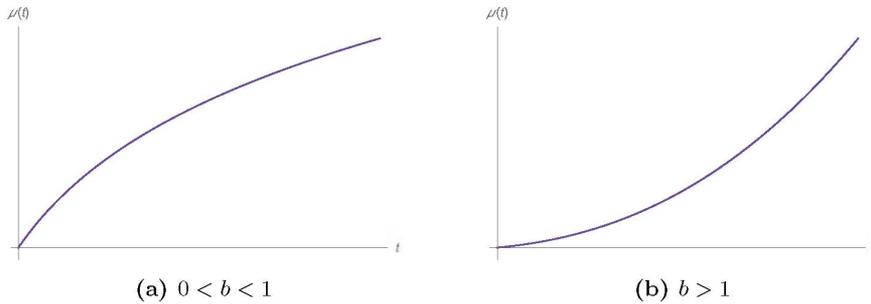


Fig. 2. Distintas formas que puede adoptar $\mu(t)$ en función del parámetro b .

cuya esperanza siendo $\lambda_n(t)$ la de la Ecuación (7) es precisamente el tiempo medio entre las fallas $n - 1$ y n , Ecuación (11):

$$MTBF_n(t_{n-1}) = \frac{1}{a} \frac{1 + at_{n-1}}{(1 + at_{n-1})^b - 1}. \quad (11)$$

Una demostración completa se incluye en el Apéndice A.

3. Experimentos

Los experimentos que a continuación listamos fueron realizados empleando tres *datasets* reales provenientes de proyectos llevados a cabo por desarrolladores de una *software factory* bajo metodologías ágiles, presentados en el formato *número de falla - tiempo de falla*. Para los tres (identificados como Proyectos Ágiles N° 1, 2 y 3) hemos calculado los tiempos medios entre fallas predichos por los modelos Delayed S-Shaped y Logístico, así como para el modelo de contagio que presentamos en la Sección 2.4. A excepción del modelo de contagio, la estimación de los parámetros para todos los modelos se hizo por el método de máxima verosimilitud. Para aquél, se utilizó el método de los mínimos cuadrados.

3.1. Proyecto Ágil N°1

Éste proyecto ya ha sido estudiado por los autores en [5]; consultar la referencia para los detalles sobre el mismo. La Figura 3 muestra las curvas interpoladas de tiempo medio entre fallas para cada modelo, contra los tiempos entre fallas reales registrados. Todos los tiempos están medidos en días. Es notable que los tres modelos predigan un *MTBF* bastante superior a los tiempos reales observados (la mayoría de los cuáles, como se aprecia en la Figura 3, caen en la franja de entre 0 y 10 días). Los tres coinciden en predecir tiempos de entre 10 y 20 días para todas las fallas de la número 5 en adelante, radicando sus diferencias en la etapa más temprana y en la forma de las curvas. El modelo de contagio y el Delayed S-Shaped se comportan muy parecido, siendo casi la única diferencia

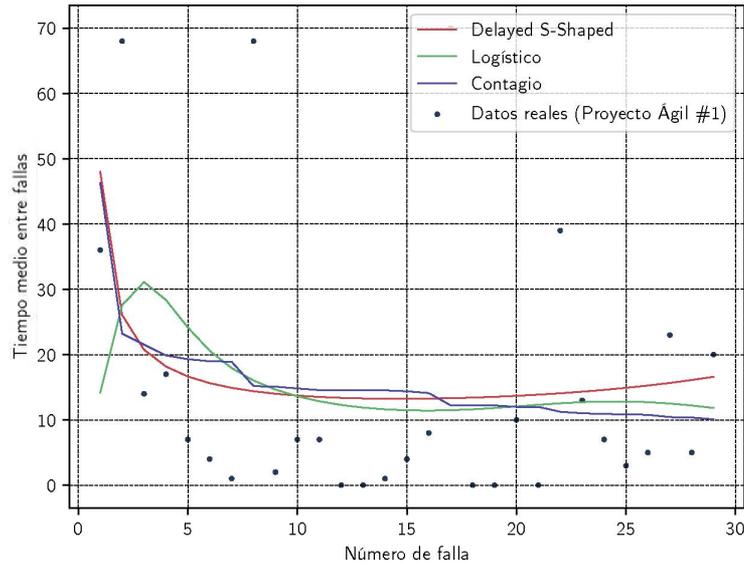


Fig. 3. MTBFs vs tiempos reales entre fallas, Proyecto Ágil N°1.

que la curva de nuestro modelo es monótona decreciente, es decir, predice una tasa de fallas totalmente creciente, mientras que la de aquél muestra un mínimo alrededor de la falla 15 (curva roja) y luego vuelve a crecer, reflejando una etapa de crecimiento de la confiabilidad. La curva del modelo Logístico tiene un comportamiento muy distinto en la etapa inicial, comenzando con tiempos muy bajos y creciendo abruptamente hasta un pico alrededor del día 3, luego del cuál decrece suavemente. El pico inicial es característico del modelo Logístico, y es debido a que éste es el único modelo que considera una cantidad de fallas inicial detectada distinta de cero. La curva tiene, además, dos máximos y un mínimo, sugiriendo varias etapas bien diferenciadas en donde la confiabilidad crece y decrece. Los datos reales presentan, además, varios valores *outlier* muy lejanos a lo que todos los modelos predicen.

3.2. Proyecto Ágil N°2

Los datos del presente fueron provistos por la empresa Grupo Esfera S.A., pero por razones de confidencialidad no nos es posible indicar los detalles del proyecto. La Figura 4 muestra las curvas interpoladas de tiempo medio entre fallas para cada modelo, contra los tiempos entre fallas reales registrados. Todos los tiempos están medidos en días. En contraste a lo ocurrido con el Proyecto

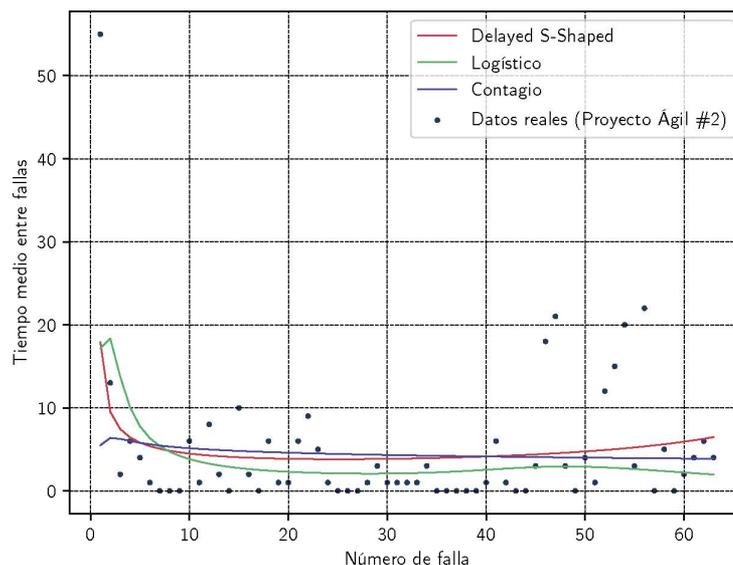


Fig. 4. *MTBF*s vs tiempos reales entre fallas, Proyecto Ágil N°2.

Ágil N°1 (Sección 3.1), aquí las tres curvas predicen tiempos muy cercanos a los datos reales, que se localizan en alta densidad en la franja de entre 0 y 10 días. La comparación aquí se dificulta por lo cercanas en el tiempo que son las fallas (observar que hay muchos puntos localizados en la misma línea horizontal, indicando que todas ellas ocurrieron el mismo día), pero aún es posible observar que la curva del modelo de contagio (curva azul) predice un *MTBF* monótonamente decreciente (indicando una tasa de fallas creciente) pero casi constante, mientras que la curva del modelo Delayed S-Shaped presenta un mínimo alrededor del día 25, creciendo luego. El modelo Logístico presenta un pico bien destacado en su etapa inicial, luego de lo cual tiene varias etapas sucesivas de crecimiento y decrecimiento. Ninguno de los valores *outlier* presentes en los datos es tomado en consideración por los modelos.

3.3. Proyecto Ágil N°3

Los datos del presente fueron provistos por la empresa Grupo Esfera S.A., pero por razones de confidencialidad no es posible indicar los detalles del proyecto. La Figura 5 muestra las curvas interpoladas de tiempo medio entre fallas para cada modelo, contra los tiempos entre fallas reales registrados. Todos los tiempos están medidos en días. Se observa aquí, de repetida cuenta, todo lo

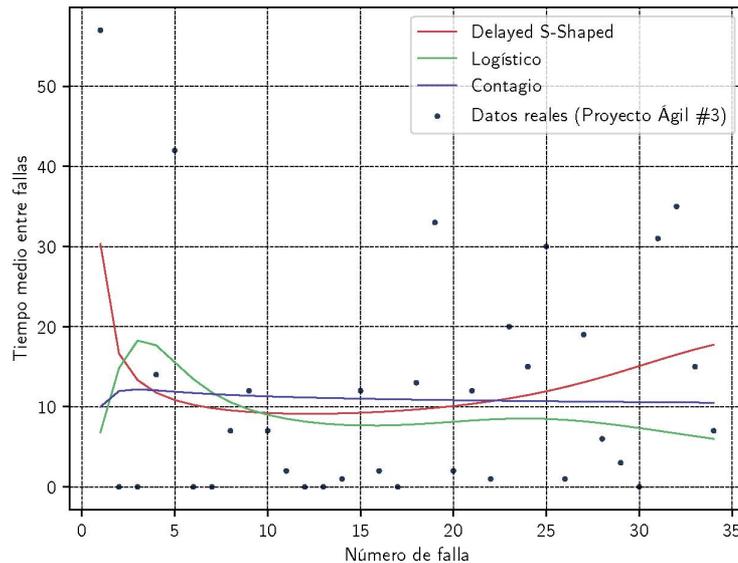


Fig. 5. *MTBFs* vs tiempos reales entre fallas, Proyecto Ágil N°3.

indicado acerca del Proyecto Ágil N°2 (Sección 3.2). Resta destacar, únicamente, que en éste caso se observa un crecimiento de la confiabilidad mucho más marcado en la etapa final, apreciable por la reducción de la densidad de puntos en la franja de entre 0 y 10 días, característica que únicamente el modelo Delayed S-Shaped predice correctamente.

3.4. Discusión

Los proyectos evaluados muestran que en desarrollos ágiles los valores del *MTBF* siguen una curva del tipo "baño de calor" (*bathhtub*) [11], característico de los estudios de confiabilidad en hardware, con una etapa de tasa de fallas creciente al principio. Hemos comprobado que ni el modelo de Musa-Okumoto ni el de Goel-Okumoto se pueden ajustar a ninguno de los tres *dataset*, algo razonable ya que ambos poseen curvas de valor medio de concavidad negativa y los algoritmos de cuadrados mínimos no convergen. Ésto constituye una nueva confirmación empírica de nuestra hipótesis de que en los proyectos ágiles la confiabilidad no siempre crece, sino que de hecho, decrece en las etapas tempranas. El modelo Logístico exhibe algunas peculiaridades que lo diferencian de los demás. En los tres experimentos se observa que la curva del *MTBF* alcanza un máximo bien marcado en los primeros días, para luego decrecer suavemen-

te. También hay una cierta desviación respecto de la forma típica de la curva de baño de calor, pues las observadas resultan tener varios máximos y mínimos poco marcados en lugar de solo un mínimo. No obstante, la diferencia se da estrictamente en las formas, pues los valores numéricos se mantienen siempre en el mismo rango que los demás modelos y no se alejan de los datos reales. Resulta razonable que un punto de inflexión en $\mu(t)$, si existe, tenga como reflejo un mínimo en la curva del MTBF. En [15] se efectúa un estudio más general en cuyos experimentos ya se puede observar este fenómeno. El comportamiento ideal y más similar al esperado lo tiene el modelo Delayed S-Shaped, que presenta una etapa de tasa de fallas creciente (MTBF decreciente) seguida de un mínimo y luego una etapa de crecimiento de la confiabilidad (MTBF creciente). El MTBF del modelo de contagio no presenta mínimos en ninguno de los tres casos, como se esperaba, debido a que su curva de valor medio en ningún caso tiene punto de inflexión (Figura 2).

4. Conclusiones

En este trabajo se ha realizado un análisis del tiempo medio entre fallas de proyectos de software actuales desarrollados bajo el paradigma ágil, comparando el ajuste de cinco modelos de Confiabilidad: Musa-Okumoto, Goel-Okumoto, Delayed S-Shaped, Logístico y de contagio. Se ha hallado evidencia de que los modelos de crecimiento de confiabilidad puros no resultan útiles para modelar los proyectos donde se espera una tasa de fallas creciente, debido a la concavidad negativa invariable de las curvas resultantes. Los modelos más apropiados para ajustar la etapa inicial resultan ser el Delayed S-Shaped y el Logístico, aunque fallan en ajustar los datos en un rango más amplio. El modelo de contagio demuestra ser muy útil en el ajuste de la primera etapa de tasa de fallas creciente pero requiere de otro modelo para ajustar la etapa de crecimiento de la confiabilidad. Como es de esperar, no existe un modelo que ajuste muy bien todo el rango de datos, lo cual expresa la necesidad de efectuar un análisis multietapa, en el cual es posible aplicar el modelo de contagio en la etapa temprana junto con otro modelo que admita una tasa de fallas decreciente para la etapa final (incluyéndose la necesidad de predecir el comienzo de ésta). En este último caso, también aparece como una posibilidad el modelo de contagio, ya que como se explicó, permite modelar casos tanto de concavidades positivas como negativas. Actualmente los autores están trabajando en este tipo de análisis y los resultados serán desarrollados en próximas publicaciones.

A. Cálculo del tiempo medio entre fallas para el proceso de contagio

De la ecuación de nacimientos puros, Ecuación (1), se deduce la conocida fórmula del tiempo de espera exponencial, Ecuación (12):

$$P(T_n \geq t | t_{n-1}) = \exp\left(-\int_{t_{n-1}}^t \lambda_{n-1}(u) du\right). \quad (12)$$

Por definición, la Ecuación (12) es el complemento de la función distribución del tiempo (finito) hasta la n -ésima falla, conocido el tiempo de la anterior. Esto permite escribir la Ecuación (13):

$$F(t, t < \infty | t_{n-1}) = 1 - \exp\left(-\int_{t_{n-1}}^t \lambda_{n-1}(u) du\right). \quad (13)$$

Por otra parte, la probabilidad de que el tiempo hasta la n -ésima falla sea infinito también se deduce del tiempo de espera exponencial, de manera que su complemento es:

$$P(t_n < \infty | t_{n-1}) = 1 - P(t_n = \infty | t_{n-1}) = 1 - \exp\left(-\int_{t_{n-1}}^{\infty} \lambda_{n-1}(u) du\right). \quad (14)$$

Dividiendo la Ecuación (13) por la Ecuación (14) y diferenciando resulta la Ecuación (10). A partir de esta última se toma la expectación condicional de t para obtener la Ecuación (15)

$$MTBF_n(n-1, t_{n-1}) = \frac{1}{a} \frac{1 + at_{n-1}}{b(n-1)}, \quad (15)$$

y sustituyendo $n-1$ por el número medio de fallas se obtiene la Ecuación (11). Notar que aquella expresión está condicionada a que haya ocurrido exactamente el número medio de fallas, $\mu(t_{n-1})$, en el instante t_{n-1} , que es en la práctica el único $MTBF$ que se puede calcular para todo t a partir del modelo ajustado únicamente.

Referencias

1. Barlow, R., Proschan, F.: Mathematical Theory of Reliability. Society for Industrial and Applied Mathematics, Philadelphia, PA 19104-2688 (1996), <http://epubs.siam.org/doi/abs/10.1137/1.9781611971194>
2. Barraza, N.R.: Software reliability modeled on contagion. In: 2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 49–50 (Oct 2016). <https://doi.org/10.1109/ISSREW.2016.22>
3. Barraza, N.R.: Mining bugzilla datasets with new increasing failure rate software reliability models. In: XLIII Latin American Computing Conference, CLEI 2017, Cordoba, Argentina, September 4-8 (2017), <http://www.clei2017-46jaiio.sadio.org.ar/sites/default/files/Mem/SLISW/slisw-04.pdf>

4. Barraza, N.R.: Software reliability modeling for dynamic development environments. In: Anand, A., Ram, M. (eds.) *Recent Advancements in Software Reliability Assurance*, chap. 3, pp. 29–37. *Advances in Mathematics and Engineering*, CRC Press (2019), <https://books.google.com.ar/books?id=5p6RDwAAQBAJ>
5. Barraza, N.R., Pena, G.: Modelos de confiabilidad de software con tasa de fallas creciente aplicados a proyectos Ágiles. In: VII Congreso Nacional de Ingeniería Informática - Sistemas de Información (11 2019)
6. Barraza, N.R., Pena, G., Moreno, V.: A non-homogeneous markov early epidemic growth dynamics model. application to the sars-cov-2 pandemic. To be published in *Chaos, Solitons & Fractals* **140** (2020)
7. Bazovsky, I.: *Reliability Theory and Practice*. Dover Civil and Mechanical Engineering, Dover Publications (2013), https://books.google.com.ar/books?id=Zf_DAgAAQBAJ
8. Chi, J., Honda, K., Washizaki, H., Fukazawa, Y., Munakata, K., Morita, S., Uehara, T., Yamamoto, R.: Defect analysis and prediction by applying the multistage software reliability growth model. In: 2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP). pp. 7–11 (March 2017). <https://doi.org/10.1109/IWESEP.2017.16>
9. Feller, W.: *Introducción a la teoría de probabilidades y sus aplicaciones*. Limusa (1989)
10. Hossain, S.A., Dahiya, R.C.: Estimating the parameters of a non-homogeneous poisson-process model for software reliability. *IEEE Transactions on Reliability* **42**(4), 604–612 (Dec 1993). <https://doi.org/10.1109/24.273589>
11. Klutke, G.A., Kiessler, P.C., Wortman, M.A.: A critical look at the bathtub curve. *IEEE Transactions on Reliability* **52**(1), 125–129 (2003)
12. Koshimae, H., Tanaka, H., Osaki, S.: Some remarks on MTBF's for non-homogeneous Poisson processes. *IEICE Transactions on Fundamentals* **E77-A**, 144–149 (01 1994)
13. Musa, J.D., Iannino, A., Okumoto, K.: *Software Reliability - Measurement, Prediction, Application*. McGraw-Hill series in software engineering and technology, McGraw-Hill (1987)
14. Pena, G., Cozzi, G., Barraza, N.R.: Aplicación de los procesos de nacimientos puros en confiabilidad de software. In: ASSE 2016, 17^o Simposio Argentino de Ingeniería de Software. pp. 76–86 (2016)
15. Pena, G., Barraza, N.R.: Increasing failure rate software reliability models for agile projects: A comparative study. In: Anand, A. (ed.) *System Performance Modeling*, chap. 4. To be published by De Gryuter (2020)
16. Pham, H.: *System Software Reliability*. Springer Publishing Company, Incorporated, 1st edn. (2010)
17. Ross, S.: *Introduction to Probability Models*. Elsevier, 11 edn. (2014)
18. Sendova, K., Minkova, L.: Introducing the non-homogeneous compound-birth process. *Stochastics* pp. 1–19 (09 2019). <https://doi.org/10.1080/17442508.2019.1666132>
19. Stroock, D.W.: *An Introduction to Markov Processes*, Graduate Texts in Mathematics, vol. 230. Springer-Verlag Berlin Heidelberg, 1st edn. (2005)
20. Yakovlev, G., Rundle, J., Shcherbakov, R., Turcotte, D.: Inter-arrival time distribution for the non-homogeneous poisson process (08 2005)
21. Yamada, S., Ohba, M., Osaki, S.: S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability* **R-32**(5), 475–484 (Dec 1983). <https://doi.org/10.1109/TR.1983.5221735>