Resumen: Una investigación sobre la distribución del esfuerzo entre las fases de desarrollo: Un modelo progresivo de estimación de costos de software de cuatro etapas

Autores: Papatheocharous, E., Bibi, S., Stamelos, I., Andreou, A. S.

Redactado por: Zegarra Puma, Franco Sebastián

La estimación de costos en el desarrollo de software suele hacerse en etapas tempranas, con información incompleta, lo que compromete su precisión. Este estudio propone un modelo progresivo de cuatro etapas para actualizar las estimaciones de esfuerzo conforme avanza el ciclo de vida del desarrollo (SDLC), utilizando el dataset ISBSG R10 y redes neuronales artificiales (RNA) como herramienta principal de predicción. Este enfoque busca reducir la incertidumbre presente en modelos tradicionales y mejorar la planificación de proyectos.

Se plantean cuatro preguntas clave a responder durante el transcurso del paper:

- ¿Cómo se distribuye el esfuerzo total entre las fases del SDLC?
- ¿Existe correlación entre los esfuerzos de diferentes fases?
- ¿Actualizar las estimaciones mejora su precisión?
- ¿Qué tan bien se puede predecir el esfuerzo de la siguiente fase?

Los métodos utilizados para la estimación de costos más comúnes se agrupan en los métodos tradicionales, los cuales en este paper se dividen en: juicio experto, modelos paramétricos, históricos e híbridos (Jørgensen & Shepperd, 2007). Sin embargo, se recalca que estos tienden a ser optimistas o poco ajustables en el tiempo. El estudio incorpora una visión iterativa que mejora con retroalimentación continua, en línea con enfoques modernos y adaptativos. También se reconoce la existencia de distribuciones empíricas (como la regla 40-20-40 que consiste en asignar estos porcentajes a los procesos de análisis y diseño, programación y mantenimiento y testeo respectivamente) y análisis basado en datos, visto con mayor detenimiento en el paper de Moløkken & Jørgensen (2003) en el que se menciona su potencial en la práctica industrial y que guarda cierto parecido con lo visto en este modelo por su aplicación del análisis empírico.

Para el modelo propuesto, se definen cuatro puntos de estimación: antes de la planificación, después de la planificación, post-especificación y post-diseño. Se usan RNA para capturar relaciones no lineales entre variables del proyecto y el esfuerzo. La arquitectura empleada es un perceptrón multicapa de propagación hacia adelante, con validación cruzada quíntuple, hiperparámetros ajustados y la función sigmoide tangencial como activación. Según Popescu et al. (2009), este tipo de red es eficaz para modelar relaciones complejas en contextos con datos ruidosos, como la estimación de esfuerzo.

Se utilizó el dataset ISBSG R10. Los datos se transformaron en variables binarias. Se empleó validación cruzada quíntuple y una función sigmoide tangencial para las predicciones. La red se entrenó hasta que se cumpliera alguna condición: 1000 épocas, error mínimo ($<10^{-4}$), o gradiente bajo ($<10^{-6}$).

Se aplicaron las siguientes métricas para la evaluación de la precisión de estimación promedio de los modelos derivados:

- MRE (Mean Relative Error): usado para medir el error entre el esfuerzo actual y el esfuerzo predicho.
- CC (Coeficiente de correlación): mide la capacidad de las muestras predichas para seguir la evolución ascendente o descendente de la serie original a medida que

- evoluciona en la secuencia de predicción de la muestra. Donde un valor de 1 representa un resultado perfecto.
- Pred(I) y RE(n): se usan para revisar la cantidad de predicciones con buen rendimiento en cuanto al total.

Los resultados de las pruebas, en cuanto a las preguntas realizadas demostraron que:

- La fase de construcción concentra aproximadamente el 37% del esfuerzo, lo que demuestra que los ingenieros de software se centran principalmente en estas actividades. También se concluye que los resultados obtenidos difieren de los resultados en la literatura. Las fases de planificación, diseño y construcción presentan fuerte correlación con el esfuerzo total. Luego de varias pruebas, se concluye que la fase de construcción acumula la mayor cantidad de esfuerzo y los gerentes deben enfocarse más en ofrecer un valor con la menor desviación posible.
- Alta correlación entre fases consecutivas (ej. planificación-especificación, diseño-construcción).
- Las estimaciones se vuelven más precisas al incorporar datos de fases anteriores. La mejor precisión se da en la estimación post-diseño.
- Más difícil que estimar el total, pero los modelos post-especificación y post-diseño son los más eficaces.

El modelo reduce la incertidumbre significativamente comparado con el cono de incertidumbre de Boehm, reduciendo el error inicial hasta en un tercio, supera las limitaciones de enfoques estáticos al permitir reevaluaciones del esfuerzo en puntos críticos del ciclo de vida. Se destaca que las fases de planificación, diseño y construcción tienen una fuerte influencia en el esfuerzo total. Se encontró también que las características del proyecto (lenguaje, plataforma, tamaño del equipo) impactan en la distribución del esfuerzo. La mejor estimación del esfuerzo total se alcanza después del diseño. Finalmente, se propone que futuras investigaciones incluyan feedback profesional y validación en contextos reales para mejorar la adopción del modelo.

En conclusión, opino que el paper revisado presenta un modelo novedoso que aprovecha al máximo las opciones que se tenían para el contexto del paper (RNA, Perceptrón multicapa de propagación hacia adelante), sin embargo, no permitieron mostrar con mucha precisión los resultados obtenidos del modelo o alguna superioridad en cuanto a los demás modelos mostrados. Sin embargo, considero que con las opciones que se cuentan en la actualidad (Transformers, entre otros), sería posible mejorar significativamente el desempeño del modelo y obtener resultados más sólidos y comparables.

Referencias

- Papatheocharous, E., Bibi, S., Stamelos, I., & Andreou, A. S. (2017). An investigation of effort distribution among development phases: A four-stage progressive software cost estimation model. *Journal of Software: Evolution and Process*, 29(10), e1881.
- Boehm, B. W. (1984). Software engineering economics. *IEEE transactions on Software Engineering*, (1), 4-21.
- Jorgensen, M., & Shepperd, M. (2006). A systematic review of software development cost estimation studies. *IEEE Transactions on software engineering*, 33(1), 33-53.
- Moløkken-Østvold, K., & Jørgensen, M. (2003). A review of surveys on software effort estimation.
- Popescu, M. C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009).
 Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, 8(7), 579-588.