



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO, CHILE



MODELO DE CALIDAD PARA EVALUAR CONTINUIDAD DE DESARROLLO

Memoria presentada como requerimiento
para optar al título profesional de
INGENIERO CIVIL EN INFORMÁTICA

por

Pablo Alejandro Acuña Rozas

Comisión Evaluadora:

Dr. Hernán Astudillo

Dr. Raúl Monge

DICIEMBRE 2013

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO, CHILE

TÍTULO DE LA MEMORIA:

MODELO DE CALIDAD PARA EVALUAR CONTINUIDAD DE DESARROLLO

AUTOR:

PABLO ALEJANDRO ACUÑA ROZAS

Memoria presentada como requerimiento para optar al título profesional de **Ingeniero Civil en Informática** de la Universidad Técnica Federico Santa María.

Profesor Guía

Dr. Hernán Astudillo

Profesor Correferente

Dr. Raúl Monge

Diciembre 2013.
Valparaíso, Chile.

Índice general

Index of Figures	IV
0.1. Introducción	1
0.2. Estado del Arte	2
0.2.1. Estudios acerca de mantenibilidad	2
Bibliography	6

Índice de figuras

0.1. Introducción

0.2. Estado del Arte

0.2.1. Estudios acerca de mantenibilidad

Ya en la década de los años noventa, se consideraba que la mantenibilidad de software era un tema fundamental y que podía generar un impacto notorio en el costo de mantenimiento de un producto de software. En [4] se definen ciertas métricas para realizar mediciones de mantenibilidad y se presenta un índice de mantenibilidad que permite unificar estas métricas en una sola.

Uno de los estudios conocidos de esta época se puede encontrar en [?]. En este trabajo altamente citado se presentan guías para automatizar el análisis de mantenibilidad con el fin de apoyar la toma de decisiones relacionadas con el software en cuestión. Se evaluaron 5 métodos utilizando datos reales y luego se menciona como estos métodos pueden ser utilizados en un ambiente industrial. Los modelos aplicados fueron: modelos jerárquicos multidimensionales, modelos de regresión polinomial, medidas de complejidad basadas en entropía, análisis de componentes principales y análisis de factores.

Otro trabajo importante llevado a cabo en esta década es presentado en [10]. Se presentan algunas definiciones relativas a mantenibilidad de software así como la importancia que tiene este atributo en cualquier sistema. Además se presenta un método básico para evaluar mantenibilidad utilizando dos indicadores principales: la medida de mantenibilidad (MM) y el índice de mantenibilidad (MI). Este último es utilizado hasta hoy en día para tener una referencia rápida del comportamiento estático de un sistema.

Ya para los siguientes años, el tema de mantenibilidad de software es considerado de alta importancia y se puede encontrar diversos estudios que resumen y presentan el estado del arte del área. A continuación se nombran los más relevantes para esta memoria.

En [3] se presentan los principales factores que afectan a la mantenibilidad de software. Estos factores son extraídos de diferentes autores los cuales proponen diferentes modelos de mantenibilidad de software. En el desarrollo del reporte se presentan factores tales como analizabilidad, modificabilidad, estabilidad, capacidad

de pruebas, modularidad, descriptividad, consistencia, simplicidad, etc. Estos son tomados de modelos conocidos tales como el modelo de calidad de la ISO 9126, el modelo de calidad de Boehm, modelo de McCall, el modelo Fuzzy, modelo ME-MOOD, etc.

En [8] se describen guías para estudiar la mantenibilidad de software utilizando métricas basadas en orientación a objetos. Para este estudio se generó un catálogo de investigaciones importantes relevantes utilizando ciertas heurísticas. Se encontraron 606 métricas de las cuales 570 eran métricas orientadas a objetos y 71 métricas orientadas a aspectos. Otro resultado interesante que surgió del estudio fueron los tópicos encontrados al buscar investigaciones relacionadas con mantenibilidad. Estos fueron limitaciones de arquitectura de software, herencia, cohesión, acoplamiento, complejidad y tamaño. De estos conceptos se encontró que cohesión y acoplamiento eran los tópicos mayormente investigados en la literatura con respecto a mantenibilidad.

Otro estudio relacionado con mantenibilidad enfocada en orientación a objetos se puede encontrar en [5]. En este trabajo se revisan estudios de métricas de mantenibilidad de software orientado a objetos correspondientes a la década pasada (2003-2012). Del estudio se concluyó que los métodos basados en análisis de regresión (RA) fueron los mayormente empleados para evaluación durante la década. Otros modelos de calidad existentes tales como ISO 9126 o el modelo de McCall son escasamente utilizados en el desarrollo de modelos de mantenibilidad. La mayoría de los estudios obtenidos en la investigación hacen uso de las métricas orientadas a objetos existentes sin una revisión y adaptación crítica antes de ser utilizadas para desarrollar modelos.

Otro estudio de métricas existentes para software orientado a objetos se puede encontrar en [9]. En este trabajo se discuten un rango de opciones para categorizar métricas. Lo que se busca es solucionar el problema de la falta de información útil acerca de métricas para mantenibilidad de software orientado a objetos que apoye la toma de decisiones acerca de cuales de estas métricas debiesen ser adoptadas en estudios académicos o incluso en actividades del día a día en la industria del software. Este trabajo es una continuación de [8], en el cual se encontraron 570 métricas

para software OO. El objetivo es categorizar este conjunto de métricas para facilitar la implementación de un modelo para mantenibilidad. Se generaron un total de 15 categorías que van desde métricas de evaluación hasta métricas de herramientas de ayuda.

En [1] se presentan los resultados luego de realizar una revisión de la literatura para identificar métricas contemporáneas asociadas con mantenibilidad de software y propuestas para tecnologías orientadas a aspectos y orientadas a características. En este trabajo se puede encontrar la lista de métricas y propiedades medibles para programación orientada a aspectos y orientada a características. Con estas métricas se elaboró un catálogo unificado de métricas aplicables a ambas tecnologías y además se presenta sus principales referencias.

Otro tipo de tecnología relevante es la orientada a servicios. En [11] se analizan factores que afectan en la mantenibilidad de software y presenta un método de evaluación para mantenibilidad de software orientado a servicios. Se divide la mantenibilidad de software en analizabilidad, cambiabilidad, estabilidad y capacidad de prueba como índice de evaluación. Para cada uno de estos atributos de calidad se presentan métricas que pueden ayudar a mejorar el diseño de software y seleccionar mejores estrategias de mantenimiento.

Otra tópico importante corresponde a los modelos de predicción de mantenibilidad. Estos modelos son herramientas matemáticas que de acuerdo a ciertas métricas estudiadas en el software, pueden entregar predicciones del comportamiento futuro del software en lo que respecta a su mantenimiento. En [6] se presenta una revisión sistemática de predicción de mantenibilidad de software y de métricas utilizando distintas pregunta para conducir su investigación. Algunas de estas preguntas fueron: ¿Qué medidas han sido utilizadas para medir la precisión de las predicciones de mantenibilidad de aplicaciones de software?, ¿Qué factores y métricas han sido investigados como predictores de mantenibilidad para aplicaciones de software?. Este estudio está más focalizado en estudiar predictores de mantenibilidad y al igual que la mayoría de este tipo de investigaciones, se realizaron consultas adecuadas a un repositorio de investigaciones para analizar los resultados que se repiten con mayor

frecuencia y que pudiesen responder a las preguntas planteadas. Uno de los resultados de la revisión es que no existe un modelo de predicción obvio para mantenibilidad. Se encontraron distintos tipos de modelos para predecir mantenibilidad, la mayoría basados en algoritmos de regresión y de validación cruzada. También se observó que los predictores más utilizados fueron aquellos basados en tamaño, complejidad y acoplamiento.

Una aplicación más directa de modelo de calidad establecido se puede encontrar en [2]. En este trabajo se entrega una descripción del método utilizado por el *Software Improvement Group* para analizar calidad de código enfocándose en mantenibilidad. Este método utiliza un modelo de medición basado en la conocida ISO/IEC 9126, específicamente en la definición de mantenibilidad y métricas de código fuente. Utilizando las métricas adaptadas para este modelo se genera un repositorio que luego es utilizado como *benchmark* en el cual se acumulan distintos resultados de evaluaciones los que cuales permiten ir calibrando el modelo de medición.

Se han realizado numerosos estudios en el campo de sistemas OO. El tema de mantenibilidad también ha sido crucial en estos estudios. En [7] se realiza una revisión de diversos modelos de mantenibilidad encontrados en la literatura. En este estudio sólo se han considerado aquellos trabajos en los cuales se utiliza OO. Para el análisis se tomaron investigaciones de diversas fuentes. En el trabajo se detallan las principales técnicas encontradas en los artículos.

Bibliografía

- [1] R. Abilio, P. Teles, H. Costa, and E. Figueiredo. A systematic review of contemporary metrics for software maintainability. In *Software Components Architectures and Reuse (SBCARS), 2012 Sixth Brazilian Symposium on*, pages 130–139, 2012.
- [2] Robert Baggen, José Pedro Correia, Katrin Schill, and Joost Visser. Standardized code quality benchmarking for improving software maintainability. *Software Quality Control*, 20(2):287–307, June 2012.
- [3] B. Kumar. A survey of key factors affecting software maintainability. In *Computing Sciences (ICCS), 2012 International Conference on*, pages 261–266, 2012.
- [4] P. Oman and J. Hagemester. Metrics for assessing a software system’s maintainability. In *Software Maintenance, 1992. Proceedings., Conference on*, pages 337–344, 1992.
- [5] B.A. Orenyi, S. Basri, and Low Tan Jung. Object-oriented software maintainability measurement in the past decade. In *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on*, pages 257–262, 2012.
- [6] Mehwish Riaz, Emilia Mendes, and Ewan Tempero. A systematic review of software maintainability prediction and metrics. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM ’09*, pages 367–377, Washington, DC, USA, 2009. IEEE Computer Society.

- [7] Prof. (Dr.) Ajay Rana Sanjay Kumar Dubey, Amit Sharma. Analysis of maintainability models for object oriented system. *International Journal On Computer Science And Engineering*, 3(12):3837–3844, 2011.
- [8] J. Saraiva. A roadmap for software maintainability measurement. In *Software Engineering (ICSE), 2013 35th International Conference on*, pages 1453–1455, 2013.
- [9] J. Saraiva, S. Soares, and F. Castor. Towards a catalog of object-oriented software maintainability metrics. In *Emerging Trends in Software Metrics (WETSoM), 2013 4th International Workshop on*, pages 84–87, 2013.
- [10] Richard West. Improving the maintainability of software. *Journal of Software: Evolution and Process*, 8, 1996.
- [11] Ma Zhe and Ben Kerong. Research on maintainability evaluation of service-oriented software. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 4, pages 510–512, 2010.