表04 外文翻译

# T.R. Gopalakrishnan Nair,R. Selvarani. Defect proneness estimation and feedback approach for software design quality improvement[J]. Information and Software Technology . 2011 (3)

# Abstract

Context: Modern software engineering demands professionals and researchers to proactively and collectively work towards exploring and experimenting viable and valuable mechanisms in order to extract all kinds of degenerative bugs, security holes, and possible deviations at the initial stage. Having understood the real need here, we have introduced a novel methodology for the estimation of defect proneness of class structures in object oriented (OO) software systems at design stage.

Objective: The objective of this work is to develop an estimation model that provides significant assessment of defect proneness of object oriented software packages at design phase of SDLC. This frame work enhances the efficiency of SDLC through design quality improvement.

Method: This involves a data driven methodology which is based on the empirical study of the relation- ship existing between design parameters and defect proneness. In the first phase, a mapping of the relationship between the design metrics and normal occurrence pattern of defects are carried out. This is represented as a set of non linear multifunctional regression equations which reflects the influence of individual design metrics on defect proneness. The defect proneness estimation model is then generated by weighted linear combination of these multifunctional regression equations. The weighted coefficients are evaluated through GQM (Goal Question Metric) paradigm.

Results: The model evaluation and validation is carried out with a selected set of cases which is found to be promising. The current study is successfully dealt with three projects and it opens up the opportunity to extend this to a wide range of projects across industries.

Conclusion: The defect proneness estimation at design stage facilitates an effective feedback to the design architect and enabling him to identify and reduce the number of defects in the modules appropriately. This results in a considerable improvement in software design leading to cost effective products.

# Introduction

Intelligent systems are the pressing need for advancement in every facet of work, and software engineering is no exception. The software generation can be made autonomic by incorporating cognitive features in every phase of development life cycle to achieve cost effective defect free software product. Thus development of software becomes an expertise of the machines, and human interaction is necessary only in approving the designs and finished products. One way to effectively manage this interaction is to invent methods for machines to evaluate the quality of its work and report back to the designer for improvement and/or approval. This perhaps would be a primary achievement in realizing closed loop design system.

The design phase of an object oriented software system is critical, as it dictates the quality levels that could be achieved. Several interesting points elucidating the impact of design metrics on external quality parameters have been proposed in the past. Han [28] worked on quantifying the impact of change on design and implementation of documents. However, his approach does not address invocation of interdependencies and identifiers, and their impact on design.

The purpose of this work is to develop an estimation model that provides significant assessment of defect-proneness of object oriented software packages at the design phase of SDLC. This mod- el is synthesized based on the intricate relationship between the design properties represented by CK metrics and the defect-prone- ness of the software. The estimation is appropriate and useful in providing effective guidelines to the design architect in fine-tuning the design for zero defect level before implementation.

The structure of this paper is as follows; Section 2 describes the background of the study. Section 3 introduces the internal design properties of the object oriented paradigm in terms of CK (Chidamber and Kemerer) metrics. Section 4 explains the frame work for the estimation of defect proneness of the software at the design stage. Section 5 elaborately discusses the modeling of influence of design metrics on defect proneness through empirical studies. Section 6 presents an analysis of the design properties and the estimation model for influence of design metrics. Section 7 demonstrates the formation of Software Defect Proneness Estimation (SDPE) model through judicious use of design properties identified through CK metrics. Section 8 illustrates the model evaluation and validation results and Section 9 summarizes the contribution of this paper and the scope for future work.

# 2. Background

Defect proneness of a class can be defined as the possibility of occurrence of fault in the class’’. In the software industry, the defect density of a software system is usually measured only after the implementation of the project through code. Basili and Warmer [6,50] emphasize that an early warning about the probable defect levels in the system will benefit the team in software development process and promote better defect management strategies. The need for research for a better understanding of the determinants of software quality and other project outcomes such as productivity, reusability, and maintainability is emphasized by many researchers [7,12,19]. Grady [24] effectively acknowledged the detection and removal of defects prior to the deployment, owing to its potential role in influencing the customer satisfaction.

Akiyama [2], Kitchenham et al. [36], Veevers and Marshall [49] and Warmer and Kleppe [50] have proposed different approaches like improvement in design clarity, effective use of process and product metrics, realization of stability and maturity in the development process, training the software development teams on defect management, and promotion of practices such as peer reviews and causal defect analysis respectively to reduce defects in software. Janes [3] and Krishnan et al. [37] put forth the necessity of detecting flaws in the design stage, which will permanently settle many of the issues that can lead to the creation of major defects in software. Agresti and Evanco [1] proposed a model for projecting defects from designs of ADA structures based on product and process characteristics. The model was built on the basis of multivariate regression analysis conducted with empirical data. Context coupling emerged as a consistently significant variable in the model.

Kazman et al. [32] presented a theoretical approach on the impact of changes on a system’s architecture but no specific models were suggested for impact estimation. Kiran et al. [33] and Kung et al. [38] suggested an impact model with regression testing, based on three links viz., inheritance, association, and aggregation. Li and Offutt [39] proposed an algorithm for investigating the effects of encapsulation, inheritance, and polymorphism on maintainability at class level. A model for identifying high risk components based on the influence of coupling metrics on design features is formulated in [43].

# 摘要

背景：现代软件工程需要专业人员和研究人员积极主动地共同努力，探索和尝试可行的和有价值的机制，以便在初始阶段提取各种退化的错误，安全漏洞和可能的偏差。 在了解了真正的需求之后，我们在设计阶段引入了一种新颖的方法来评估面向对象（OO）软件系统中类结构的缺陷倾向性。

目标：这项工作的目标是开发一个评估模型，在SDLC的设计阶段提供对面向对象软件包的缺陷倾向性的重要评估。 该框架工作通过提高设计质量来提高SDLC的效率。

方法：这涉及数据驱动的方法论，该方法基于对设计参数与缺陷倾向性之间存在关系的实证研究。 在第一阶段，进行设计度量与缺陷正常出现模式之间关系的映射。 这被表示为一组非线性多功能回归方程，其反映了个体设计度量对缺陷倾向性的影响。 然后通过这些多功能回归方程的加权线性组合产生缺陷倾向性估计模型。 加权系数通过GQM（目标问题度量）范式进行评估。

结果：模型评估和验证通过选定的一组案例进行，这些案例被认为是有前途的。 目前的研究已经成功地处理了三个项目，并且有机会将其扩展到各行各业的众多项目中。

结论：设计阶段的缺陷倾向性估计有助于向设计架构师提供有效的反馈，并使他能够适当地识别和减少模块中的缺陷数量。 这导致软件设计的显着改进，从而导致成本效益的产品。

# 简介

智能系统迫切需要在各个工作方面取得进步，软件工程也不例外。 软件生成可以通过在开发生命周期的每个阶段结合认知功能来实现自主，从而实现具有成本效益的无缺陷软件产品。 因此软件的开发成为机器的专业知识，只有在批准设计和成品时才需要人工交互。 有效管理这种互动的一种方式是发明机器的方法来评估其工作质量，并向设计人员报告以改进和/或批准。 这也许是实现闭环设计系统的主要成就。

面向对象软件系统的设计阶段至关重要，因为它决定了可以达到的质量水平。 过去已经提出了几个有趣的观点来阐明设计度量对外部质量参数的影响。 Han [28]致力于量化变化对文档设计和实现的影响。 然而，他的方法并未涉及相互依存关系和标识符的调用及其对设计的影响。

这项工作的目的是开发一个评估模型，在SDLC的设计阶段提供对面向对象软件包的缺陷倾向性的重要评估。 该模型是基于CK度量表示的设计属性与软件缺陷倾向性之间的复杂关系而合成的。 该估算对于为设计架构师提供有效的指导方针，在实施之前对零缺陷级别的设计进行微调方面是适当和有用的。

本文的结构如下： 第2节介绍研究的背景。 第3节介绍了CK（Chidamber和Kemerer）度量方面面向对象范式的内部设计特性。 第4部分解释了在设计阶段评估软件缺陷倾向性的框架工作。 第5节通过实证研究精心讨论了设计度量对缺陷倾向性影响的建模。 第6节介绍设计属性分析和设计指标影响评估模型。 第7节通过明智地使用通过CK度量指标识别的设计属性来演示软件缺陷倾向性估计（SDPE）模型的形成。 第8节说明了模型评估和验证结果，第9节总结了本文的贡献和未来工作的范围。

# 背景

类的缺陷倾向性可以定义为“在课堂上发生错误的可能性”。 在软件行业中，软件系统的缺陷密度通常只有在通过代码实施项目后才能测量。 Basili和Warmer [6,50]强调，关于系统中可能的缺陷级别的早期预警将有利于团队在软件开发过程中，并促进更好的缺陷管理策略。 许多研究人员都强调，研究需要更好地理解软件质量和其他项目成果的决定因素，如生产力，可重用性和可维护性[7,12,19]。 Grady [24]有效地承认在部署之前检测和清除缺陷，因为它在影响客户满意度方面可能发挥作用。

Akiyama [2]，Kitchenham等人。 [36]，Veevers和Marshall [49]以及Warmer和Kleppe [50]提出了不同的方法，如改进设计清晰度，有效使用流程和产品度量标准，实现开发过程的稳定性和成熟度，培训软件开发团队 关于缺陷管理以及分别推广同行评审和因果缺陷分析等做法，以减少软件缺陷。 Janes [3]和Krishnan等人。 [37]提出了在设计阶段检测缺陷的必要性，这将永久解决许多可能导致软件缺陷产生的问题。 Agresti和Evanco [1]提出了一种基于产品和工艺特性的ADA结构设计中预测缺陷的模型。 该模型建立在用经验数据进行的多元回归分析的基础上。 背景耦合是模型中始终存在的显着变量。

Kazman等人 [32]提出了一种关于系统结构变化影响的理论方法，但没有提出具体的模型用于影响估计。 基兰等人。 [33]和Kung等人。 [38]基于三个环节，即继承，关联和聚合，提出了一个带回归测试的影响模型。 Li和Offutt [39]提出了一种算法来研究封装，遗传和多态对班级可维护性的影响。 [43]中提出了一种基于耦合度量对设计特征影响的高风险成分识别模型。