



中国科学院软件研究所

杰出青年人才发展专项计划年度进展报告书

(2012 年度)

姓名	杨叶	课题名称	经验软件工程方法研究
资助金额	160 万	课题起止时间	2009. 9-2013. 9
资助类别		所在部门	基础软件中心
研究工作主要进展和阶段性成果（含论文、研究生培养等）			
本项目 2012 年度的工作围绕以软件成本估算和缺陷预测为代表的经验软件工程研究领域的预测模型和方法，对典型预测模型的性能优化问题、缺失数据处理问题等进行了深入研究，并结合基于开源社区和基于众包等典型软件开发场景进行了分析和预测模型及方法研究。主要工作成果包括以下几个部分： ● 缺失数据处理技术 ● 开源社区演化网络分析研究 ● 开源社区缺陷解决人推荐研究 ● 软件定制化的可变性分析的决策支持 ● 众包环境下的软件开发任务定价 1. 缺失数据处理技术 缺失数据中实际上存在两种缺失机制：结构性缺失和非结构性缺失。前者指在数据收集的过程中，软件项目本来就不具备某种或某类特征；后者指软件项目存在数据收集过程中给定的项目特征，但实际数据测量不准确或者没有被收集上来。我们以 ISBSG 和 CSBSG 为研究对象，分析了软件工作量数据中的缺失机制。首先，我们假定数据中的缺失服从某种缺失机制，然后利用在此种机制下的缺失数据处理方法对缺失数据进行修正并进行软件项目工作量的预测。当缺失数据被假定为结构性缺失条件下，最大间距分类（Max-Margin Classification）被用来处理缺失的工作量数据并更根据历史数据预测软件工作量。 图 1 显示了以上缺失数据处理方法在 ISBSG 和 CSBSG 数据集上的软件工作量预测性能对比。实验结果表明，无论对于 ISBSG 数据集还是 CSBSG 数据集，建立在非结构性数据缺失假设基础上的缺失数据处理方法比建立在结构性数据缺失假设基础上的软件工作量预测			

性能要好。而且，在非结构性数据缺失处理方法内，在 ISBSG 上，MINI 方法要比 CMI 方法要好，而在 CSBSG 上，其结论恰好相反。我们将其解释为 CSBSG 数据集上不同工作量类别的数据集之间的方差较小，而 ISBSG 上的数据则方差较大。

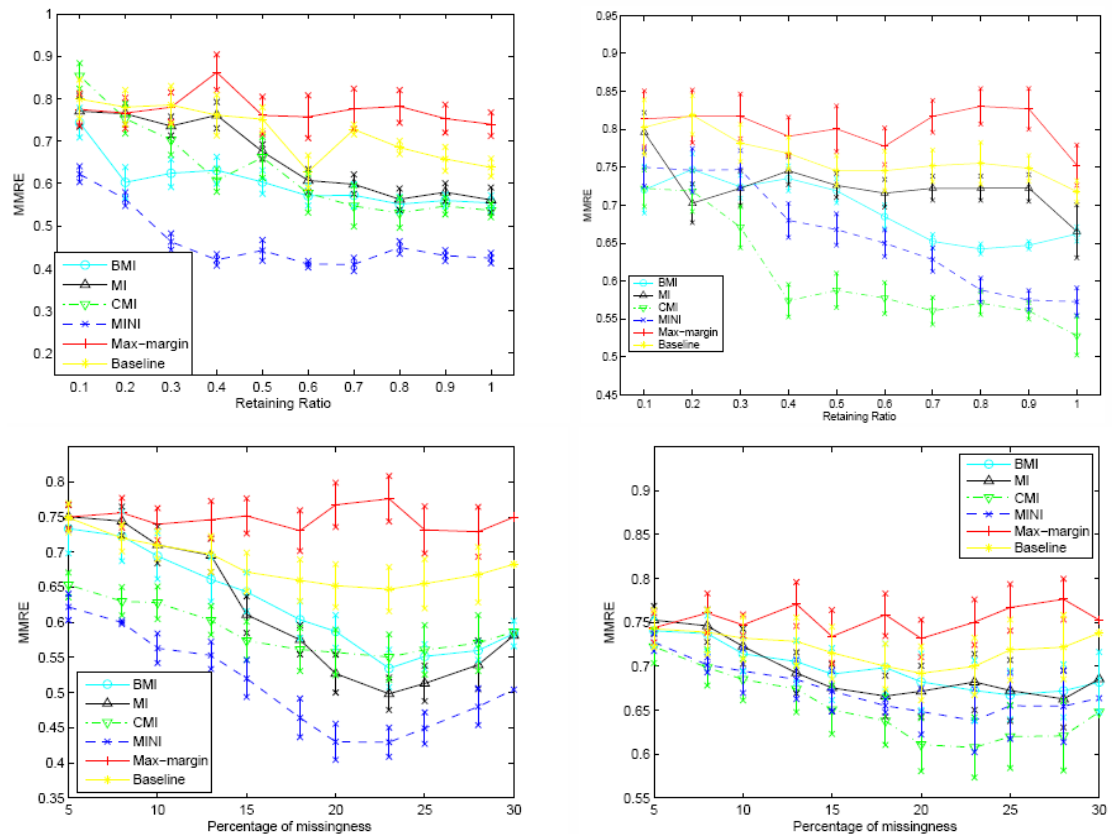


图 1 结构性和非结构性缺失数据处理方法在不同的特征子集规模（上）和不同缺失水平（下）条件下在 ISBSG 数据集（左）和 CSBSG 数据集（右）上进行工作量类别预测的性能对比。

2. 开源社区演化网络分析研究

本研究以一个真实的开源项目社区 ArgoUML 社区 (<http://argouml.tigris.org/>) 为研究案例，研究了开源社区中核心开发人员识别、开发人员在不同软件模块级别的交流分布和开发人员随开源软件版本演化而发生的开发人员变动趋势。我们收集了从 2000-07-10 到 2010-03-06 期间 11 个 ArgoUML 版本的开发人员邮件列表交流和源代码提交记录。在此期间 980 名独立开发人员总共讨论了 6087 个话题，其中的 34 名具有代码提交权限的开发人员总共提交了 15921 个代码变更记录。表 1 列出了 ArgoUML 项目 11 个版本的历史邮件列表和代码提交信息。

利用邮件列表中开发人员之间的交流记录，我们基于 Clique Network 形式建立了在不同

版本期间的开发人员邮件交流网络(包括有权重网和无权重网)。图 2 示例性的显示了从 0.10 版本向 0.12 版本演化过程中，依据开发人员的邮件交流记录构建的无权重的社会网络。依据这个网络，利用节点度 (Degree)、介数 (Betweenness)、PageRank 和 HITS Rank 节点中心性度量指标，我们进行了项目开发核心开发人员识别研究。

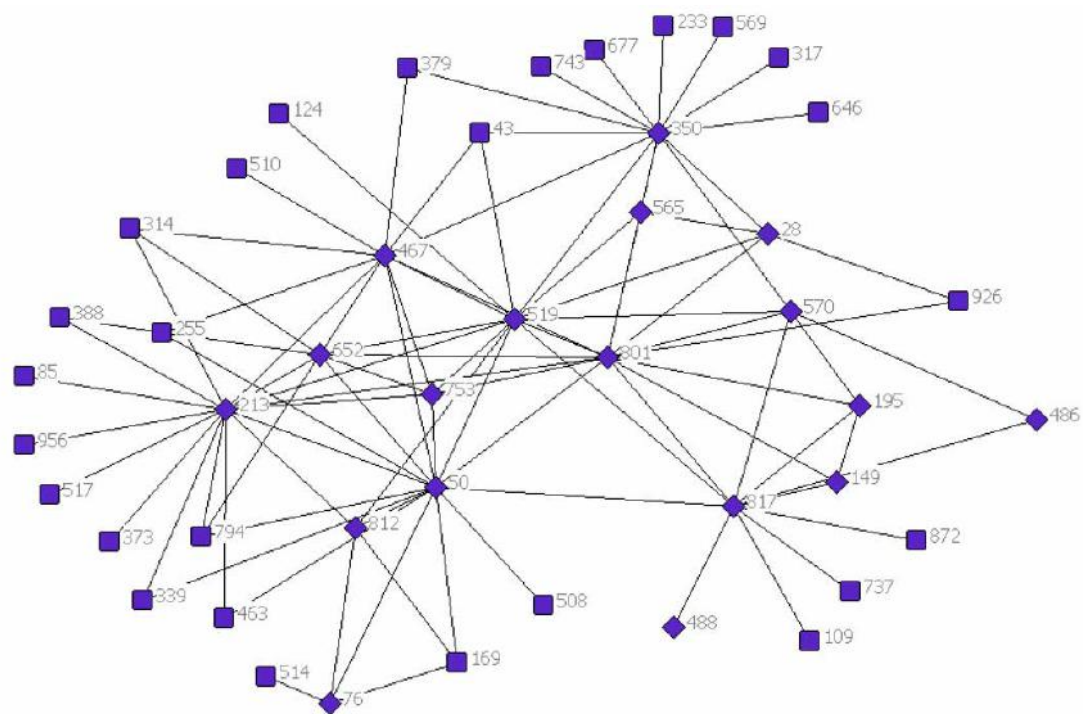


图 2. 从 0.10 版本向 0.12 版本演化过程中的开发人员邮件交流网络。

表 1: ArgoUML11 个版本上的不同类型开发人员的演化情况。

version	# of n.c.	# of e.c.
0.12	71/4/2	432/12/1
0.14	89/3/2	82/1/0
0.16	52/5/2	83/0/0
0.18	36/0/2	60/8/0
0.20	30/4/1	42/3/2
0.22	32/1/0	32/2/3
0.24	22/2/2	30/3/0
0.26	82/2/2	18/5/1
0.28	16/1/1	83/3/1
0.30	24/2/0	20/2/0

表 2: ArgoUML 项目核心开发人员的重复比例。

Release #	08 – 10	10 – 12	12 – 14	14 – 16	16 – 18	18 – 20	20 – 22	22 – 24	24 – 26	26 – 28
10 – 12	0.67									
12 – 14	0.67	1.00								
14 – 16	0.83	1.00	1.00							
16 – 18	0.67	0.83	0.88	0.90						
18 – 20	0.50	0.67	0.75	0.70	0.78					
20 – 22	0.33	0.33	0.38	0.40	0.44	0.64				
22 – 24	0.50	0.50	0.50	0.50	0.56	0.75	1.00			
24 – 26	0.50	0.33	0.38	0.50	0.56	0.63	1.00	0.86		
26 – 28	0.33	0.17	0.38	0.50	0.56	0.63	0.80	0.72	0.88	
28 – 30	0.33	0.17	0.38	0.50	0.56	0.63	0.80	0.72	0.88	1.00

结果表明,各种节点中心性度量指标在核心开发人员识别上的准确率和召回率的差距不大。各个历史版本中的大约 60%的核心开发人员能被社会网络分析指标准备识别出来。具体而言,节点度在有权网络上的识别效果最好,达到了 64%以上。同一模块内的开发人员之间的交流频繁,而不同模块内的开发人员之间的交流则较少。并且,如果一个开发人员参与的开发工作越多,那么他(她)与其他开发人员之间的交流也越多。邮件列表中的一般开发人员之间的在相邻版本上的加入和离开项目的数量在 100 的级别上,而在拥有源代码提交权限的开发人员中,他们离开和加入项目的数量一般在 10 个一下。

3. 开源社区缺陷解决人推荐研究

我们从 Bugzilla 缺陷跟踪系统中提取出 Eclipse JDT 项目和 Mozilla Firefox 项目的缺陷数据,包括缺陷报告的 ID 号、标题、描述以及参与其讨论和修复的开发人员列表。对于 Eclipse JDT 项目,选取 2001/10/10 至 2009/01/31 期间的缺陷报告数据做为训练集,2009/02/01 至 2010/06/16 期间的缺陷报告数据做为测试集;对于 Mozilla Firefox 项目,选取 2008/09/01 至 2009/07/31 之间的缺陷报告数据做为训练集,2009/08/01 至 2009/08/31 期间的缺陷报告数据做为测试集。

实验结果中可以看出,对于 Eclipse JDT 项目而言,本研究实现的方法可以在推荐 5 个开发人员的情况下,使推荐召回率达到 82%;对于 Mozilla 项目而言,本研究实现的方法可以在推荐 7 个开发人员的情况下使召回率达到 50%。另外,召回率会随着参数 θ 的变化而变化,这也说明本研究之前的观点是正确的,即开发人员在不同主题上的兴趣度确实会影响开发人员参与缺陷修复的行为。

4. 软件定制化的可变性分析的决策支持

商业化软件通常通过定制化来满足客户的需求。在 2013 年对 ERP 系统的调研中,只有 23%的机构通过少量的定制满足客户需求,而 34%的机构存在大量的定制。随着需求

的多样化和和不断增长的软件复杂性，定制化已经成为软件开发中一个棘手的问题。定制化不仅会导致软件开发和维护成本的增加，并且对于定制化的功能，还需要为集成到下一个软件版本提供额外的测试和验证。虽然将事先将频繁变更的需求提前部署在软件中，允许客户自定义系统，从而避免一定程度上的定制化。但是当系统提供的功能不能满足客户的要求时，定制化是无法避免的。为了减少定制化带来的风险，一种有效地方法是在客户需求的基础上对现有的系统进行改进和优化，生成一个新的主干版本。在新的演化版本上再进行定制化开发后他交付给相应的客户。这样做的优势在于将原本需要定制化的需求优化后集成到主干版本中，不仅提高了复用，而且减少了定制化带来的额外开销。图 3 显示了这样的一个定制化过程。在这个过程，需要区分哪些客户需求将集成到下一个主干版本上，哪些是需要定制化？

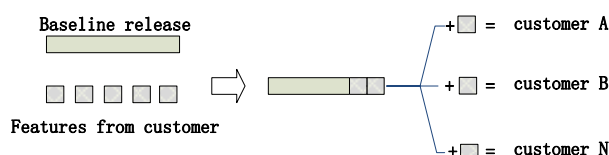


图 3 软件定制化的过程

领域分析技术通常用于分析在产品线中哪些需要时所有产品共有的（commonality），哪些是可变的（variability）。该技术同样可以应用于解决上述的问题。客户中共有的需求可以直接集成到下一个主干版本中。而对于频繁出现可变性需求，可以通过优化后集成到作为下一个主干版本。其他的可变性需求在通过定制化实现。但是现有的领域分析方法大多是通过 NLP 技术或者聚类发现需求之间的共性和可变性，只识别哪些需求是相似的哪些需求是不同的，并没有结合客户的需求对现有的系统进行分析和优化。

因此，为了弥补该缺陷，我们提出了一种领域分析和优化的方法 FOAP（Feature Oriented Analysis and Optimization）。该方法以用户的变更请求描述为输入，包含三个部分：特征抽取，可变性发现，可变性优化。首先从用户的变更获取变更的类型以及需求的描述；而后通过对比用户需求和主版本需求识别系统的可变性；在识别的可变性基础上，从需求语义和业务两个角度对可变性进行优化，通过发掘更多的可集成主干的需求减少定制化。该方法的输出为系统的特征模型以及新的主干版本上的需求和定制的需求。为了验证方法的有效性，我们通过一个软件过程管理软件 Qone 进行了案例研究，结果表明 FOAP 方法可以有效辅助项目决策进行领域分析和优化，减少了定制化需求。

5. 众包环境下的软件开发任务定价

目前已经有许多公司和组织通过众包的形式来进行软件开发,在这种新型开发环境下存在着如何对开发任务进行合理定价的问题。在众包开发过程中不合理的任务定价往往会造成资金利用的低效或过低的用户参与度,进而影响项目预算及产品交付。本文研究如何利用经验历史数据来对基于众包的软件开发任务进行定价。我们的研究动机还来自于观测到的两个反例,在这两个例子中传统软件成本中的规律在新环境下不再适用(帕金森法则与 COCOMO 模型)。虽然之前的研究覆盖了众包的各个方面,但很少研究侧重于基于众包的软件开发及相关模型。据我们所知这是第一篇研究如何在某一众包平台使用经验数据对软件开发任务进行定价的文章。

我们使用数据驱动的方法基于世界上最大的软件开发众包平台 TopCoder 进行了相关研究,爬取并分析了 2,895 个软件设计阶段和 3,015 个软件开发阶段的任务数据。通过分析这些任务数据的特征我们提出了 4 类共 16 个潜在影响价格的成本因子,这 4 个类别包括:1. 开发类型:例如是否是升级项目,使用的语言等;2. 输入的复杂度,涉及前一设计阶段产出设计的复杂程度:例如所涉及到的技术的种类数量,依赖的组件数量,需求规格文档的页数,组件规格文档的页数,设计时序图的个数,设计阶段所耗费的工作量,UL 设计文件的大小,以及估计的项目规模等;3. 输入的质量,上一阶段输出的质量:例如设计阶段产出的得分,获胜者在平台的级别,注册用户数和提交数等;4. 前一阶段决策:例如前一阶段的价格信息等。根据这 16 个价格因子,我们进一步使用多种基于统计和机器学习方法建立了 9 个经验定价模型(见图 4)。

在收集到的 5,910 个任务中,980 个任务对应 490 个项目(每个项目包含一个设计阶段任务和一个开发阶段任务)包含 16 个因子的完整信息,我们在这 490 的项目上使用留一法交叉验证对 9 个经验定价模型和 3 个基准模型进行了验证(实验结果见图 5)。同时我们还通过回归分析检查了这些因子的显著性。

我们的实验结果说明给定软件设计阶段的产出,软件开发任务的价格具有较高的可预测性(Pred(30)指标值大于 0.8)。通过提出的 16 个价格因子建立的 9 个经验定价模型的效果全部要优于 3 个基准模型的效果。通过建立的经验定价模型,还可以抽取一些简洁易用的经验法则来指导 TopCoder 众包软件开发平台的参与者更好的理解与制定任务价格。

基准模型					
Naïve 模型		Random Guessing 模型		Basic COCOMO 模型	
经验模型(使用16个成本因子来建立)					
1个多元线性回归	3个基于决策树的模型	2个基于案例的推理模型	1神经网络模型	1个支持向量机模型	1个多项Logistic回归模型
LReg	C5.0, CART, QUEST	KNN_1, KNN_k	NNet	SVMR	Logistic

图 4. 实验中用到的 3 个基准模型和 9 个经验定价模型

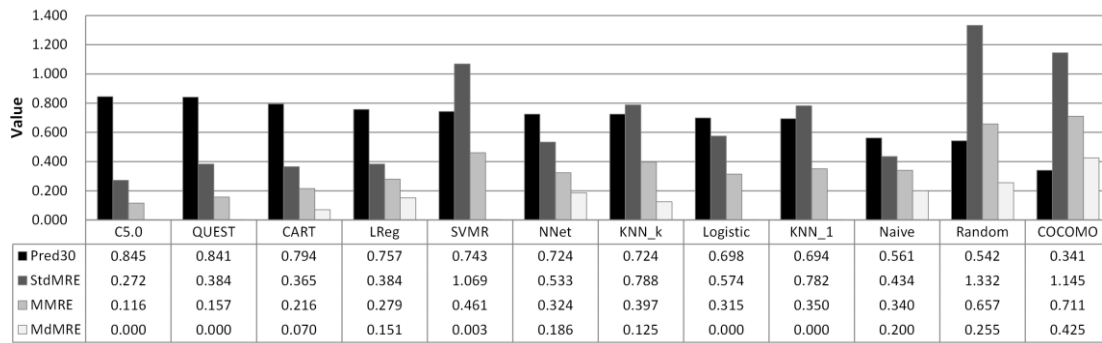


图 5. 每种价格模型的实验结果，横轴所示方法按照 Pred(30)指标排序

下一年度工作计划，包括国内外合作与交流计划

2013 年度的主要研究计划围绕以下几方面的内容展开：

1. 基于迁移学习的软件数据漂移问题研究
2. 软件定制化的可变性分析的决策支持
3. 众包环境下的软件开发任务定价
4. 众包软件过程研究

国际合作与交流计划如下：

1. 2012.11-2013.4 月，派出一名博士生到美国 West Virginia University 的 Tim Menzies 教授团队交流学习，开展迁移学习方面的合作研究；
2. 2013 年 3 月，访问南加州大学 Barry Boehm 教授团队，开展众包软件过程和定价的合作研究。
3. 2013 年 7-9 月，访问美国北卡州立的 Laurie Williams 和谢涛教授团队，开展经验软件工程的相关合作研究。

年度经费使用情况概要
2012 年度经费拨付 33.97 万元，全部用于支付科研人员的劳务费用。
存在的问题、建议及其他需要说明的情况
无

备注：1.杰出青年人才入选者应实事求是地填写此报告，禁止弄虚作假；
2.此报告作为专项计划跟踪、管理的主要依据，每年报送人力资源处；
3. 此报告将在所网站对外发布。