

第二部分 创业计划书

一、创业项目简介

（此项内容为系统填报的信息自动生成，如需修改，请登录申报系统进行更新）

（一）项目背景与意义

代码智能追踪是指利用自动化、智能化的分析技术，生成并维护软件文档与代码之间的关联关系。此技术能够有效地支撑各类软件开发活动，从而帮助软件企业降低运营成本、缩短开发周期、降低开发错误率等。与传统的人工实现代码追踪（如 CMMI 3）相比，代码智能追踪在实现效率与应用前景等方面具有巨大的优势。

（二）目标市场分析

目前，国内的软件开发企业与研究机构不但为国计民生提供了大量基础性、关键性的软件系统支撑，其本身的市场规模也十分可观。仅在 2017 年 1 月到 2 月，全国软件业务的收入就达到了约七千亿人民币，同比增长 12.6%。然而，软件业务收入本身的增速在放缓，而同时期软件从业人员的薪资增长速度却达到了 14.1%，且增速还在缓慢增加。本项目可以为有软件开发任务的企业、研究所、高校、政府机构等提供代码智能追踪工具、定制化解决方案和相关技术服务，从而提升其开发效率，帮助其实现降本增效。

（三）项目研究基础与成熟度

团队先后提出并实现了代码依赖紧密度分析、用户反馈分析、软件托管平台挖掘与分析等多项代码智能追踪关键技术，并结合日常软件开发活动设计并实现了一系列工具原型。在“词汇表失配”等领域内经典难题的解决方面积累了丰富的经验。

二、企业运营及团队情况

(请说明创业企业组织结构，描述申请人及核心团队成员已经取得的代表性业绩和能力。)

在创业初期，公司拟实行扁平化管理，主要职能部门设技术部和运营部，并计划随着公司的发展聘用专门的会计和法务人士。其中技术部负责根据项目和客户需要研发相应产品，同时紧跟技术最前沿，对核心算法不断更新换代；商务部负责客户需求对接，市场推广和人力资源等日常职能管理。这一设置的原因在于：（1）提高决策和执行效率；（2）代码追踪相较于代码测试等传统软件开发工具与服务是一个新兴事物，其相应的产品设计、市场规模、客户认可度等因素需要一个渐进的过程来实现。未来，随着各方面因素的不断成熟，本公司拟增设专门的产品与市场部门，组建一个精干的产品设计与市场运营团队，稳步扩大市场规模。

以下介绍核心团队成员的个人信息。

申请人匡宏宇拟任创业公司总经理，本科与博士学位均在南京大学计算机科学与技术系攻读，2017 年 3 月获得计算机软件与理论专业博士学位。2017 年 8 月入职南京大学软件学院任助理研究员至今。

包括申请人在内的本项目团队在面向 **Internet** 开放环境的软件可信保障技术与软件可追踪技术等相关方面有较深厚的工作积累(参见部分发表论著列表)。近年来，在国家自然科学基金、国家 973 计划、863 计划等项目的支持下，本团队在软件方法学、软件工程、软件可追踪性、程序分析、信息检索、代码库挖掘等技术领域进行了系统的研究，并于华为技术有限公司 2012 实验室装备与测试部展开过为期三个月的实地调研。

在此过程中，申请人在代码依赖捕获、程序依赖图分析、软件可追踪性生成与维护等相关方向取得了一定的初期研究成果，在 *Journal of Software: Evolution and Process (JSEP)* 等期刊以及 *ICSM*（现更名为 *ICSME*）、*SANER*、*NASAC* 等国际、国内会议上发表论文 4 篇。同时，通过国家留学基金委的建设高水平大学博士联合培养项目，与软件可追踪性领域内的知名学者 Alexander Egyed 教授（奥地利林茨大学副校长、林茨大学软件系统工程研究所所长）、Patrick Mäder 副教授（就职于德国伊尔梅瑙工业大学）、黄丽果副教授（就职于美国南方卫理公会大学）等建立了深度合作关系。

教育和工作经历:

- 2003.9-2007.6: 南京大学计算机科学与技术系, 获学士学位
- 2008.9-2017.3: 南京大学计算机科学与技术系, 软件新技术国家重点实验室, 计算机软件与理论专业, 获博士学位, 导师: 吕建院士
- 2017.8-至今: 助理研究员, 南京大学软件学院

荣誉及奖励:

- ACM 中国南京分会优秀博士论文, 2017
- 南京大学优秀研究生, 2009
- 南京大学优秀毕业生, 2007

承担的主要科研项目

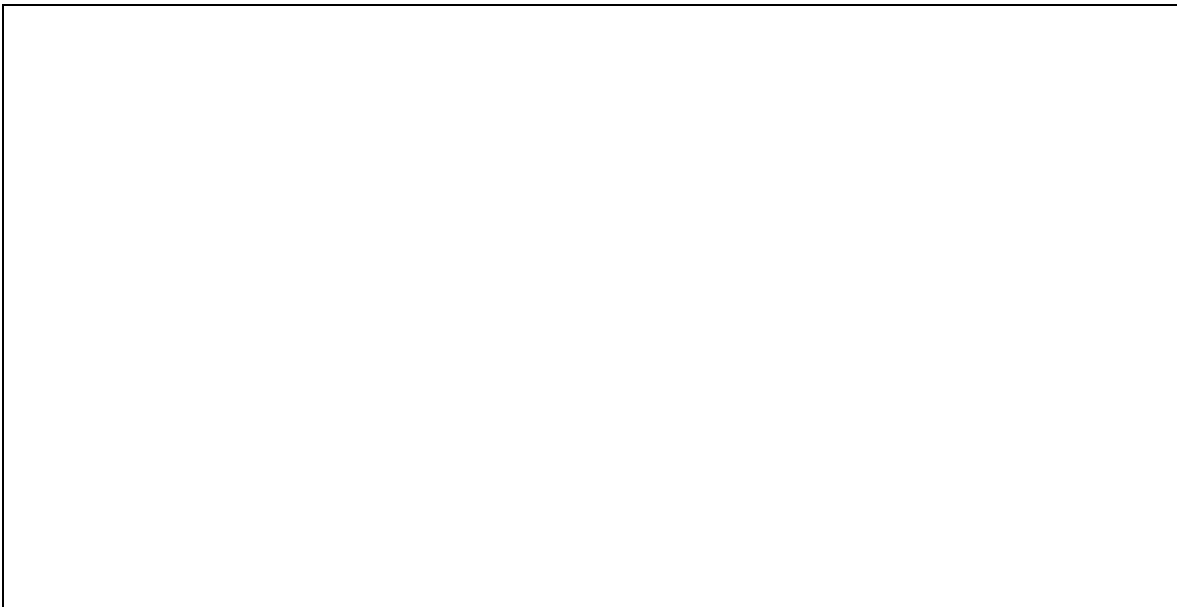
- 某国防 973 项目专题 3-3, 2017 入选, 中电科 28 所信息系统工程重点实验室。
- 基于代码依赖的软件可追踪技术研究, 2017 入选, 计算机软件新技术国家重点实验室创新项目。

代表论著

- **Hongyu Kuang**, Patrick Mäder, Hao Hu, Achraf Ghabi, LiGuo Huang, Jian Lü, and Alexander Egyed, "Do data dependencies in source code complement call dependencies for understanding requirements traceability?", in 28th IEEE International Conference on Software Maintenance (ICSM), 2012, pp.181-190.
- **Hongyu Kuang**, Patrick Mäder, Hao Hu, Achraf Ghabi, LiGuo Huang, Jian Lü, and Alexander Egyed, "Can method data dependencies support the assessment of traceability between requirements and source code?", Journal of software: Evolution and Process (J. Softw. Evol. and Proc.), 2015, Volume 27, Issue 11, pp. 838–866.
- **Hongyu Kuang**, Jia Nie, Hao Hu, Patrick Rempel, Jian Lü, Alexander Egyed, and Patrick Mäder, "Analyzing Closeness of Code Dependencies for Improving IR-Based Traceability Recovery", in 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER), 2017, pp. 68-78.
- **Hongyu Kuang**, Jia Nie, Hao Hu, and Jian Lü, "Improving Automatic Identification of Outdated Requirements by Using Closeness Analysis based on Source Code Changes", in Software Engineering and Methodology for Emerging Domains: the proceedings of the 15th National Software Application Conference (NASAC English Track), 2016, pp. 52-67.

申请专利 (已受理)

- 一种基于代码依赖关系的过时需求自动检测方法, 申请号: 201610154205.4
- 基于用户反馈和代码依赖的软件可追踪生成方法, 申请号: 201810184034.9



三、创新性产品（服务）

(请说明创业项目产品（服务），包括技术（服务）创新内容、创新水平、技术（服务）成熟度、实现成本、以及可靠性、稳定性等性能指标，突出技术（服务）独特之处，可描述创业项目技术涉及的相关自有知识产权情况。)

（一）创新内容与创新水平

本创业项目拟实现四项关键技术创新：1）核心算法创新；2）自动化代码追踪工具创新；3）基于代码托管平台挖掘的代码智能追踪创新；4）代码智能追踪服务外包创新。这些技术创新均为国际领先水平。针对上述四个关键技术创新，我们将介绍本项目研究的主要内容。

1)、代码追踪核心算法创新

图 1 首先给出了需求到代码追踪线索的常用记录形式-需求追踪矩阵，其中横轴代表需求（如“R0”），纵轴代表代码元素（类或函数，如函数“VODClient.init()”），则对应的单元格中有“X”表示二者之间存在基于系统功能的关联关系（Trace），若单元格为空则表示二者之间功能无关（No-trace）。

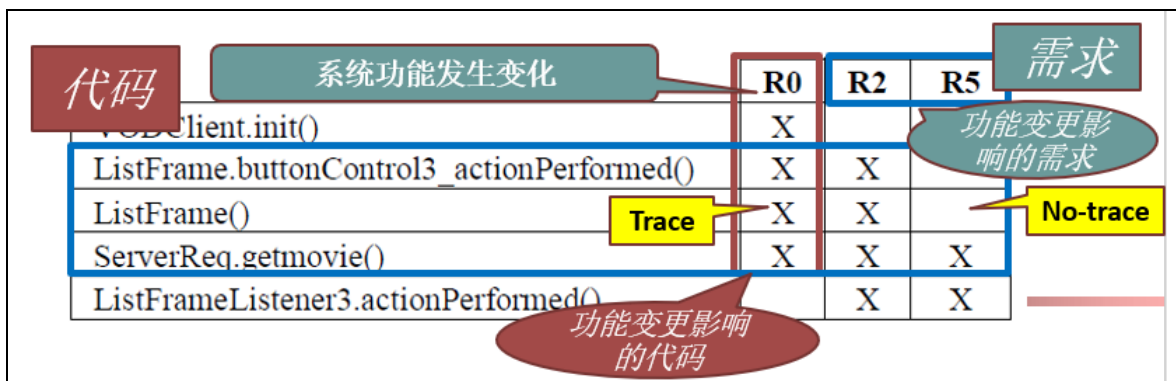


图 1 需求追踪矩阵及其作用示例

当前的软件开发活动以软件维护与演化为主，即不断优化已有系统功能的实现，或逐步地向原系统中添加若干新功能。以图 1 为例，开发项目领导决定优化系统中需求 R0 所代表的系统功能，在需求追踪矩阵的帮助下，系统的开发与维护人员不但能够快速检索到 R0 变化所需要修改的代码，还能够定位到相应的代码变更对于由其它需求所描述的系统功能的影响与冲击，这无疑有助于提升项目开发的效率与正确性。

然而，图 1 中所展示的需求追踪矩阵的生成和维护是困难的。其主要原因是需求和代码处于不同的抽象层次，且具有不同的数据格式。因此，传统的数据与数据库分析技术难以应用于代码追踪场景。为此，研究人员提出通过分析需求和代码在文本上的相似程度来自动化地展开代码追踪，并逐步形成了当前主流的基于信息检索技术的代码追踪方法。然而，所谓的“词汇表失配”问题极大地影响了此类方法的准确性，即不同的软件制品可能通过不同的词汇来表达同一个概念（例：car 与 automobile），或用同一个词汇指代不同的概念（例：schedule 既有“时间表”又有“调度”的含义）。这使得此类方法所分析的软件系统需要有丰富且高质量的文本信息，用以消减“关键词失配”这一问题所带来的不利影响。然而在现实中情况却恰好相反，即软件系统缺乏相应的文档描述，且代码的标识符和注释书写不规范。这使得自动化代码追踪技术的精度不高，难以得到大规模应用。

针对上述问题，本团队经过不懈的努力与探索，逐步发现并掌握了代码文本所特有的结构化特征、代码智能追踪用户验证局部性特征、软件托管平台所隐含的“微追踪线索”等特性，并相应地开发了代码依赖紧密度分析、用户反馈分析、软件托管平台挖掘与分析等多项具有自主知识产权的关键技术，将这些技术与当前主流的基于信息检索技术的代码追踪方法相结合，将能够有效提升自动化、智能化代码追

踪的精度与应用范围。下面以图 2 为例简要说明代码依赖紧密度分析如何帮助提升基于信息检索的代码追踪。图中方块为代码中的类，有箭头的实线和无箭头的虚线分别代表类之间的调用与数据依赖，线段上的数字是 0 到 1 范围内的紧密度值，该值越高则代表两个类交互越紧密，意味着二者大概率实现了相同的需求。代码依赖紧密度值主要是通过代码依赖图的拓扑结构计算得到，这里不再展开讨论。

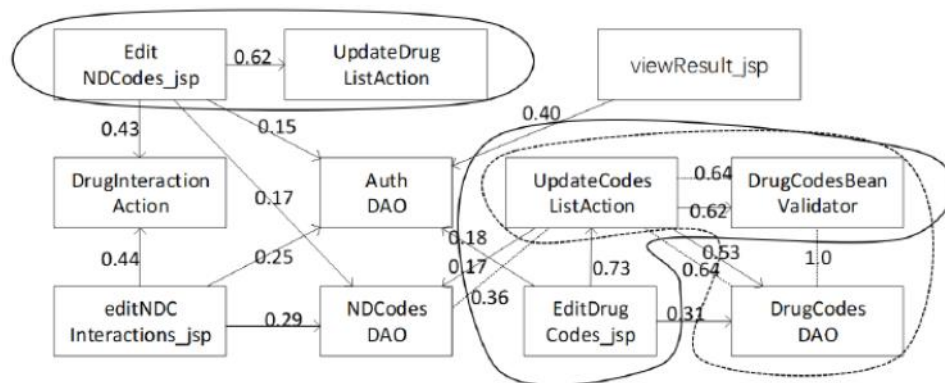


图 2 由代码依赖紧密度得到的紧密交互代码域

对于图 2 中的类，若给定需求“UC 36: Maintain and update national drug code list”，则 UpdateDrugListAction 类因其类名与 UC36 在文本上相似程度高，因而会被认为与 UC36 之间大概率存在追踪线索。而 EditNDCodes_jsp 类因其与 UC36 的文本相似性较弱，会被主流代码追踪方法认为与 UC36 无关。事实上，“ND”正是“national drug”的缩写，因此上述两个类都应该与 UC36 相关。然而，在代码依赖紧密度的辅助下，上述两个类因彼此之间存在紧密的交互关系而被划分到同一个紧密交互代码域中，从而增强后者与 UC36 的文本相似度，进而帮助代码追踪方法判定 EditNDCodes_jsp 也大概率与 UC36 相关。

综上所述，本项目中的算法核心在于充分挖掘代码文本、追踪线索建立流程、基于托管平台开发等一系列软件系统所独有的特征，并与文本相似度分析有机结合，极大地拓展了代码智能追踪方法的分析维度，提升了方法精度，从而有力地支撑了各类代码追踪相关的应用场景（部分成果以论文形式发表于重要国际国内软件工程领域学术会议，并已申请相关专利）。

2) 自动化代码追踪工具创新

一套完整的自动化、智能化代码追踪工具主要包含数据采集与预处理、追踪算

法分析、用户判断算法推荐排序、查询和展示追踪线索集合等四部分。本团队正在搭建并完善自动化程度高、用户体验友善的代码智能追踪工具，包括：（1）高效准确的数据采集与预处理：主要包含相关文本捕获、代码依赖捕获、文本预处理等；（2）准确性高、用户可理解的追踪算法：此类算法是代码智能追踪的核心，需要不断拓展分析维度并有机结合分析结果；（3）用户对追踪算法结果的验证：在实践中，自动化代码追踪的结果还需要用户最终确认以确保其正确性，一个好的追踪工具既要有友善的交互界面以便于用户展开验证，又要充分利用用户的部分验证信息进一步优化未判断内容的精度。（4）追踪线索的使用，一般的开发者要能够轻松地对生成的追踪线索集合展开查询，并为他们提供追踪线索可视化的支持，进一步的，代码智能追踪应该与普通开发者日常使用的开发环境进行深度结合（如通过插件的形式融入 Java 开发平台 Eclipse），为其打造“智能化代码开发平台”，真正实现追踪线索在日常软件开发中的“普适性”。图 3 展示了当前已实现的、结合了文本分析、代码依赖分析以及用户反馈分析的代码智能追踪工具原型。

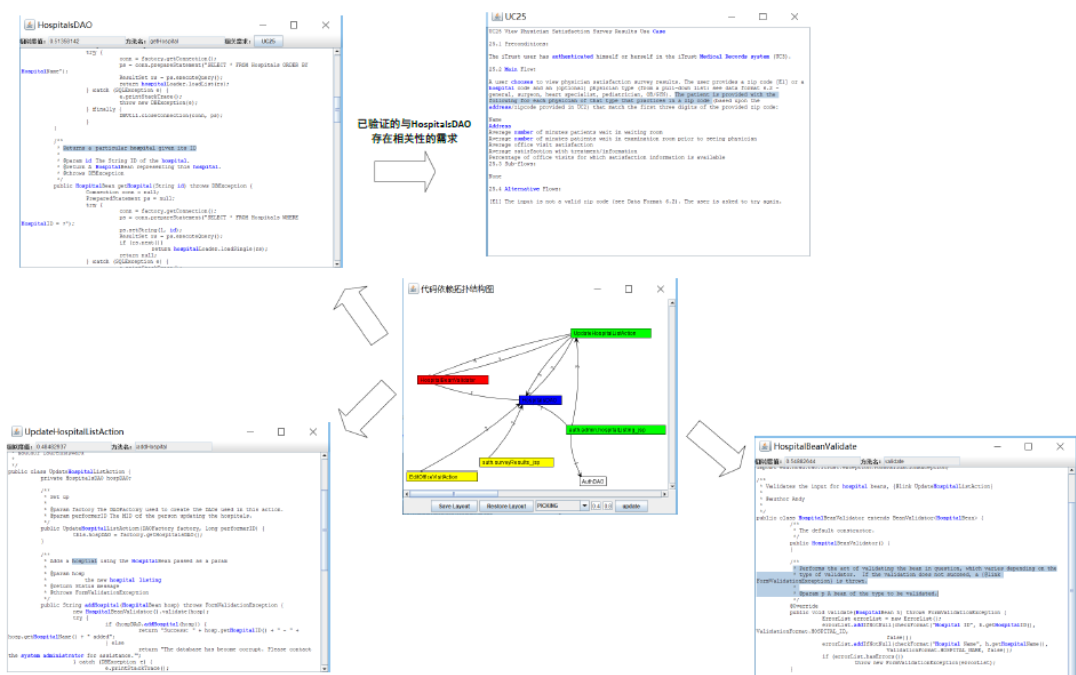


图 3 已实现的代码智能追踪工具原型

3) 基于代码托管平台挖掘的代码智能追踪创新

代码追踪在实践中遇到的一个重大挑战是，随着开源软件思想与支撑技术的不断发展，越来越多的软件系统选择基于以 GitHub 为代表的软件托管平台展开相应的开发工作。此类托管平台虽然极大地提升了协同开发的效率，却不再显式地维护

除代码之外的其它各类软件制品（比如需求），但这些软件系统的开发却依然需要追踪线索，甚至需求文档的支持。幸运地是，GitHub 托管的软件系统依然存在从系统功能提出到代码编程实现的转换过程，只是其粒度较细、抽象层级较低。针对此类软件系统，本团队以无监督的文本聚类算法为基础，辅以代码依赖紧密度分析和其它维度上的分析，实现并不断完善一个综合考虑多维度的聚类算法，将细粒度的与系统功能相关的代码变更记录进行聚合，从而抽取出发人员可理解、可使用的需求到代码追踪线索。并以此技术为基础，针对一些开发者和企业重点关注的基础性、关键性的开源软件，依照其发布的新版本定期生成和维护对应的追踪线索集合。

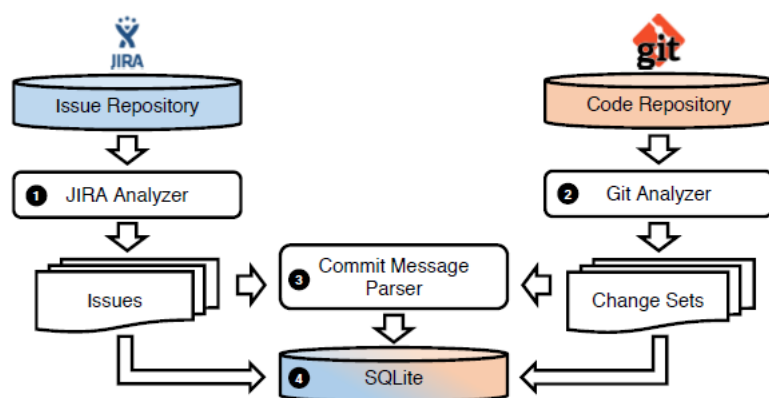


图 4 从代码托管平台 GitHub 上获取微追踪线索

4) 代码智能追踪服务外包

根据本团队的实践观察与亲身经历，大型软件企业因其自身实力较强，且开发的软件系统规模大、复杂性高、开发团队众多，因此一般都会设置一个支撑部门，以日常监督、工具实现、技术支持、员工培训等方式保障其软件开发活动的效率与正确性（如华为公司下属的 2012 实验室）。这些部门往往对于新技术持开放态度，并愿意为之投入，甚至有意识地进行相关领域的技术储备。另一方面，大多数中小型软件企业和新的创业公司缺乏足够的能力和意愿维护这样的一个开发支撑部门。然而，这些企业又迫切需要在软件开发方面实现降本增效，从而提升自身的竞争力，更好的应对愈发残酷的市场竞争。这使得本团队所设想的代码智能追踪服务外包成为可能，且市场前景广阔。本项目深入软件企业开发一线，充分了解不同企业、不同项目的具体需要，规避其中可能存在的涉密、超时等风险，未来将为广大企业和科研院所提供专业化、定制化、端到端的代码智能追踪服务。

（二）技术服务成熟度

本项目团队孵化于软件新技术国家重点实验室、南京大学软件学院、南京大学计算机科学与技术系。课题组多年来在国家自然科学基金、国家 973 计划、863 计划等项目的支持下在软件方法学、软件质量保障、软件工程、数据挖掘等方面多年的工作积累和技术优势为本项目提供了良好的技术保障。而南京大学软件学院完善的本科和硕士培养体系为本项目提供了良好的人力保障。团队负责人致力于软件工程、程序理解、自动化代码追踪等领域的研究，同时具有大量科研项目开发与实践经验，并在华为 2012 实验室展开过实地调研。针对当前主流的自动化代码追踪方法的不足，团队从以代码结构化信息为代表的软件系统特征作为技术突破口，从实验探索到特征抽取，从算法设计到工具实现，解决了多项关键技术问题，并始终关注如何从实验室概念向生产实践应用转化，积累了丰富的经验与技术储备。此外，团队负责人还通过国家留学基金委博士联合培养项目，与奥地利林茨大学副校长 Alexander Egyed 教授、德国伊尔梅瑙工业大学 Patrick Mäder 副教授等多名业内知名学者建立了长期的学术合作交流关系。

下面将展示本团队近期承担的由中电科某研究所邀请协助的国防 973 项目某子专题的相关研究内容与成果。

案例 1：基于代码变更的过时需求识别

客户痛点：客户研发并维护的若干超大型软件系统由自有代码和引入的开源代码组成，即组成了一个混源软件系统。每当开源代码版本更新时，都会对系统的稳定性和正确性带来较大的冲击，用户亟需依据每次较大规模的代码变更（包含自有代码和开源代码）自动化地识别受到影响的项目需求，从而指导开发人员顺利完成整体软件系统的更新。这一案例可以看作是一种特殊的、轻量化的代码追踪应用场景。

解决方案：首先利用启发式规则过滤不会导致系统功能变化的代码变更（如修复 bug、重命名等），再引入代码依赖紧密度分析找到能够清楚描述此次系统功能变化的代码变更元素集合，从集合中抽取关键词，再使用信息检索技术找到最可能受到此次代码变更影响的需求。图 5 中的节点代表函数（红色节点代表变更代码元素），有向边代表函数之间的调用关系，边上的数字是利用图拓扑结构计算出的代码依赖紧密度，红色虚线框内是最能反映系统功能变更的代码元素集合，在这个例子中我们能够抽取 build、log、screen、message 等重要的关键字。事实上，此次代码变更

常的软件开发实践中，需求属于级别非常高的软件文档，不能由普通开发人员轻易修改（一般是维护一个稳定的需求基线），需要对需求更新行为进行管理。

解决方案：考虑到在日常开发中需求等文档一般由项目经理定期维护，因此工具原型将记录并汇总由代码变更所引起的需求更新，并交由项目经理统一处理，如图 7 所示。下一步，工具原型将支持当前主流的代码库管理平台（如 GitHub），与平台上的代码版本管理工具相结合，同时支持更新平台上与系统功能相关的各类文档。

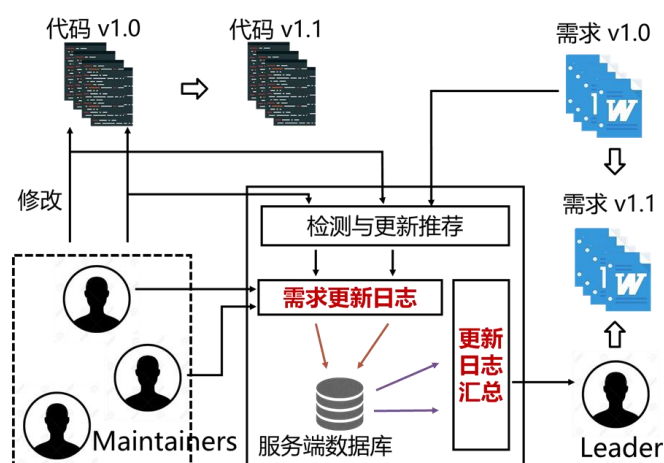


图 7 贴合日常开发场景的需求更新工具原型

四、产品（服务）市场与竞争

（请说明创业项目产品（服务）相关行业情况，市场规模及增长趋势，行业竞争对手等；确定本产品（服务）的目标顾客、目标市场和市场竞争优势，可能的市场地位和市场份额，以及未来 3 年市场销售预测等；分析本项目实施中的风险及应对措施，包括市场风险、管理风险、政策风险等。）

（一）创业项目产品说明

本项目所在的领域是软件可追踪性，是当前软件工程领域的研究热点与难点。将自动化软件可追踪技术用于提升日常软件开发的效率和正确性在国内属于首创。所谓的软件可追踪性是指软件系统所能具有的一种能够将各类软件制品相互关联，并随系统变化对这些关联关系进行维护的特性。具有这一特性的软件系统在开发时的效率和正确性能分别提高 20%和 60%。为此，针对软件企业的日常开发场景我们将提供一系列面向软件系统功能的代码智能追踪解决方案与服务，从而为软件企业的日常开发提供低成本、高精度的追踪线索集合支持，提升软件开发的整体效率，帮助软件企业实现降本增效。代码智能追踪的核心是基于多个分析维度的追踪算法，而对软件日常开发场景的了解与分析也是重要的市场进入壁垒。本团队利用多年的技术积累与实践经验，期望不断深化代码智能追踪算法，建立拥有自主知识产权的代码智能追踪工具、服务与知识体系，能够真正帮助广大软件企业和科研院所提升软件开发的整体效率。

（二）目标客户和市场容量

本项目主要针对三类客户群体：（1）对于拥有开发支撑部门的大型软件企业与研究所，本项目可以为其提供代码智能追踪工具以及相关的咨询服务，进一步强化其软件开发支撑技术以提升其开发效率；（2）对于中小型软件企业，本项目可以采用外包的方式承接其特定软件项目的追踪线索生成与维护工作，从而帮助该软件项目满足特定的软件标准要求（如 CMMI 3 以上标准、部分安全攸关软件强制要求等），有效提升该项目的开发效率与正确性；（3）对于软件开发者个人与相关专业学生，本项目还计划对一些具有基础性、关键性、应用广泛的软件技术框架（如安卓操作系统、深度学习基础平台 tensorflow 等），针对其不断发布的新版本展开定期的追踪线索生成与维护，采用“知识付费”的模式发布追踪粒度不同的追踪分析报告，从而提升领域内整体研究与实践的开发效率和正确性。

虽然目前还没有关于代码追踪的独立的市场调研报告，我们可以参照软件测试领域的市场变化。据有关资料显示，2011 年我国软件测试产业规模约为 1695 亿元人民币，而到 2016 年时规模已经扩大到 4900 亿元人民币，增长约 2.9 倍。与此同时，2018 年 1 月我国软件产业规模已经超过 5 万亿元人民币。考虑到软件企业本身实现降本增效的迫切愿望，则以提升整体开发效率为目标的代码智能追踪的潜在市场规模值得期待。这也与我国持续不断推动新时代软件产业高质量发展、乃至实现“智能制造”的国策战略相吻合。

（三）竞争对手和市场趋势

据本团队成员了解，到目前为止国内尚没有以提升开发效率为业务核心的知名软件服务企业存在，这部分的潜在要求多由各大 IT 公司下属的技术支撑部门实现（也是我们未来展开合作和研讨的对象）。国际上，IBM 公司在著名的软件建模工具 Rational Rose 中整合了代码追踪的人工标记功能，然而由于自动化程度低，人力消耗大，这一工具并未在业内得到广泛应用。相比之下，本项目提出的代码智能追踪具有显著的竞争优势和更为广泛的市场前景。

另一方面，对于大多数中小软件企业而言，为了保障最后交付的软件产品质量，这些企业大多都引入了软件测试工具或外包服务。出于节约总体项目成本的目的，中小企业在软件测试之外再引入新的开发支撑工具或服务的意愿可能较弱。然而，随着软件测试技术的不断发展，虽然测试的覆盖度和自动化程度不断提高，但由于软件测试本身并不分析软件文档与代码背后的语义，在测试发现问题之后一线的开发人员往往是“知其然不知其所以然”，依然需要按照测试报告有限的信息逐步的去寻找可能存在问题的代码片段。因此我们相信，在我们不断优化算法、工具与服务的前提下，代码智能追踪这一全新的开发支撑技术最终将得到软件企业的认可，从而在根本上提升国内软件企业的开发能力，促进我国从软件业大国转向软件业强国，也促使本项目在国内能够得到大力发展。

（四）风险与应对

1、需求离散化风险

代码智能追踪贴近日常开发实践，其应用场景差异化明显，因此项目推进过程中必然要面临定制化，在市场规模一定的前提下，定制化意味着生产成本的极大提升。为此，公司计划以软件基础架构和软件应用领域为关键点，形成体系化的代码

智能追踪工具与服务，尽可能提升已有工具与服务的可复用性，从而大大降低企业生产和扩张的成本，从而快速切入市场，保证企业利润。

2、企业竞争风险

代码智能追踪在领域内算是新兴事物，市场需要一定时间和成功实践的驱动来认可。特别是软件企业已经在软件测试工具与服务方面投入较多的情况下，需要向其展示整体开发效率提升所带来的巨大收益。此外，软件测试与代码追踪本身并不冲突，甚至是相辅相成的关系。未来，打造一体化的软件开发与交付质量保证体系，使得选购此类服务的软件企业能够将全部研发实力投入技术革新和商业模式创新，是本公司所期望达到的一个长远目标。

3、核心技术与人员流失风险

核心技术与核心人员是公司的主要竞争力，因此对于公司核心成员，一方面要通过合同约定，禁止核心技术泄漏；另一方面也考虑适当引入股权激励制度，充分调动个人的主观能动性。此外还要为员工营造开放、宽松、成长的工作环境，并建立健全员工培训机制，完善技术团队的梯队建设，保证技术力量的新老更替和补给。

（五）销售收入预测（金额单位:万元）

	2018 年	2019 年	2020 年	2021 年	2022 年
软件产品收入	10	40	180	300	500
技术服务与咨询	40	110	320	700	1000
销售收入合计	50	150	500	1000	1500

五、产品（服务）的商业模式

(请说明创业项目产品（服务）的开发、生产（服务）策略、营销策略等，在价格、促销、建立销售网络等各方面拟采取的措施及其可操作性和有效性，突出项目产品（服务）获利方式和商业模式的独特性。提出企业未来发展的短期、中期、远期发展规划和目标。)

（一）项目的研发与生产策略

项目的研发和生产主要分为三个部分，即核心算法与工具原型的研发，贴合具体项目开发的代码追踪服务设计，以及针对基础性、关键性开发框架的定期追踪线索生成与维护。这三个部分彼此不同又相互关联，其中核心算法与工具原型的研发是基础，是公司的核心竞争力之所在。而贴近具体项目开发的代码追踪服务一般选择基于最为成熟、性价比最高的追踪算法与工具，这部分的关键在于摸透服务对象的开发团队组织、基本代码构架、以及软件项目背后所隐含的丰富的领域知识（如电商、游戏、安防等软件系统必然有本质性的差异）。这部分看似技术难度不高，实则最具挑战性的工作，从中获取的宝贵经验与领域知识发掘也将是公司重要的隐形资产。而维护基础软件框架（如安卓、tensorflow 等）的追踪线索集合可以有效地拓展公司的知名度，为行业发展谋福利，同时也具有强烈的教学价值，可以在富有经验的团队成员的指导下，依托南京大学软件学院完善的本科和硕士培养体系来完成。

（二）项目的销售策略与盈利模式

本项目盈利方式拟采用“轻产品、重服务”，即向用户提供代码智能化工具、定制化解决方案以及相关的咨询与服务，在产品部分仅针对定制化的工具部分收取一定费用。本项目的产品主要分为以下三类：（1）面向大型软件企业技术支撑部门的技术转让；（2）面向中小型软件企业的代码智能追踪服务；（3）面向领域的基础开发平台追踪线索集合生成与维护。其中第一类产品构成公司的基础收入来源，市场维系主要通过持续的算法和技术更新来实现；第二类产品是公司的重点业务，其关键之处在于能否尽量复用已有的成功样例，同时又能够紧密贴合客户的具体开发实践；第三类产品则计划采用知识付费的模式运营，只覆盖基本功能的、粗粒度的（如功能到代码包的追踪线索）、非最新版本的报告免费，其余的按照生成报告工作量的不同设定不同的价格，允许客户通过付费的形式获取自定义的报告。

（三）项目进度计划

预期的未来五年计划（2018.7-2022.6）如下：

- 第一年完成 1-2 套工具的转让，签订 1-2 个服务外包项目，着手展开基础开发框架的分析工作
- 第二年在推广工具转让的同时也着手核心技术的更新换代，签订 2-4 个服务外包项目，初步发布 1-2 个基础开发框架的追踪线索报告
- 第三年在一切有序的情况下尝试扩大经营规模，签订 4-6 个服务外包项目，充分掌握 2-4 个基础开发框架的追踪线索报告
- 第四年则逐步展开相应的市场推广活动，完成新技术和工具的研发并对外转让，签订 6-8 个服务外包项目，掌握 5 个以上的基础开发框架的追踪线索报告
- 第五年迎来大幅增长期，尝试每一年到两年完成一次技术和工具的更新，签订超过 10 个服务外包项目（在已有项目能够复用的基础上），维护 8 个左右的基础开发框架的追踪线索报告。

六、财务与经济社会效益

(请说明企业财务现状；预测未来 3-5 年的投资、融资计划，以及营业收入、利润、资产回报率等指标。)

1、项目投融资计划

项目投资总额	1116 万元							
项目投资计划	预算支出	金额/万元	2018	2019	2020	2021	2022	备注
	1. 设备材料费	150	30	30	30	30	30	购买服务器、电脑、材料
	2. 差旅费	110	10	20	20	30	30	
	3. 人员费	590	50	110	110	160	160	员工薪酬
	4. 市场运营费	100	10	15	15	30	30	市场推广和运营费用
	5. 其他	70	20	15	15	10	10	办公费用、水电、知识产权费用
	6. 税费	96	1.5	4.5	15	30	45	软件企业税费 3%
	合计	1116	121.5	194.5	205	290	305	
项目融资计划	资金来源	金额/万元	2018	2019	2020	2021	2022	备注
	企业自筹	516	46	70	100	120	180	
	银行贷款	100					100	
	政府资金	50	50					
	其他（融资）	450		150	150	150		融资款
	合计	1116	96	220	250	270	280	

2、项目经济效益分析

项目	2018	2019	2020	2021	2022	合计
营业收入	50	150	500	1000	1500	3200
营业成本	48.5	52.5	175	375	550	1201
税收	1.5	4.5	15	30	45	96
净利润	0	93	310	595	905	1903
销售净利润率	0%	62%	62%	59.5%	60%	59%
投资回收期						4.1

项目未来 5 年预计可实现收入 3200 万元，税收 96 万元，净利润 1903 万元，预计平均销售净利润率 59%，投资回收期约 4.1 年。

2018 年主要为研发试运营阶段，主要以投入为主，预计收入 50 万元。

2019年通过市场培育，争取达到技术与工具转让两项，平均转让费用20万元，预期收入40万元；承接4个服务外包项目，每个项目平均收入25万元，预期收入100万元；基础开发平台追踪线索报告两个，每个报告的知识付费平均5万元，预期收入10万元。

2020年争取达到技术与工具转让三项，平均转让费用33万元，预期收入100万元；承接6个服务外包项目，每个项目平均收入60万元，预期收入360万元；基础开发平台追踪线索报告四个，每个报告的知识付费平均10万元，预期收入40万元。

2021年争取达到技术与工具转让四项，平均转让费用50万元，预期收入200万元；承接8个服务外包项目，每个项目平均收入87.5万元，预期收入700万元；基础开发平台追踪线索报告五个，每个报告的知识付费平均20万元，预期收入100万元。

2022年争取达到技术与工具转让收入维持在300万元；承接10个服务外包项目，每个项目平均收入80万元，预期收入800万元；基础开发平台追踪线索报告八个，每个报告的知识付费平均50万元，预期收入400万元。

3、项目社会效益预测

我们有信心在五年内让国内软件产业接受代码智能追踪产品并成为行业的领导者，将公司的工具、服务与付费知识打造成领域内的知名品牌。同时，我们也会积极关注新技术的发展与演进，继续促进领域内相关研究的发展，并充分利用南京大学软件学院成熟的本科与硕士培养体系，做到产学研的有机结合与共同促进。我们最大的期望在于，通过我们在代码智能追踪这一软件开发关键领域上的不懈努力，能够打造一系列有竞争力的工具与服务，推动以程序理解为核心的智能追踪技术不断发展，培养一批充分掌握代码智能追踪知识与技能的研究人员与工程师，进而从根本上提升整体软件企业的开发效率，使得各种规模的软件企业与科研院所能够更加专注于新技术的使用与业务逻辑的优化，从而增强企业的核心竞争力，带动相关产业的发展，提升整体软件产业的层次，真正促进南京市、江苏省以至于全国的软件产业进一步做大做强，达到真正的国际顶尖水平。