**摘要**

软件开发的过程中产生了大量的软件开发过程数据，这些软件开发过程数据记录了软件从诞生开始的成长与发展轨迹。这些数据不仅能使项目的管理者与开发者了解到目前项目的具体开发情况，从而做出项目决策，还能帮助人们回答许多软件工程学的问题。因此，对于软件开发过程数据的研究与理解具有重要的意义。而要这些对软件开发过程数据进行理解与使用，则首先要建立起一套基础设施，帮助收集、存储与检索软件开发过程数据。

然而互联网上有众多的软件开发过程数据，具有海量、分布和异构等特点，这对我们利用这些数据进行研究与工作造成了困难。本文尝试应对上述挑战，以北京大学软件工程研究所数据分析小组的数据为基础，尝试提出一种软件开发过程数据的收集，分析与使用方法，并设计与实现一套软件开发过程数据的检索工具。

本文提出的方法与工具有以下几个方面：第一，从数据的处理脚本中提取出脚本的描述信息，包括脚本的处理对象，用途，使用方法等；第二，从软件开发过程数据中提取出数据的描述信息，描述数据的来源，基本信息与存储位置等，；第三，通过数据与脚本的描述信息，建立检索工具。可以让用户方便快速的检索到所需的数据，可以让用户快速的定位到所需的数据，并很方便的对数据进行提取与处理，从而提高软件开发过程数据的使用效率。

关键词：软件开发过程数据，数据描述，数据存储，数据检索

目录

1 引言 3

2 相关技术介绍 6

2.1 软件开发过程数据 6

2.1.1 代码变迁数据 6

2.1.2 缺陷数据 7

2.1.3 邮件数据 7

2.2 MongoDB 7

2.3 pymongo 8

2.4 相关工作 8

3 问题分析与解决方案 9

3.1 问题分析 9

3.2 解决方案 9

4 工具的设计 10

4.1 工具的总体设计 10

4.2 数据提取模块设计 10

4.2.1 日志数据描述 10

4.2.2 缺陷数据描述 11

4.2.3 邮件数据描述 11

4.2.4 脚本描述 11

4.3 数据存储模块设计 11

4.4 数据检索模块设计 11

5 工具的实现 12

5.1 数据提取模块实现 12

5.2 数据存储模块实现 12

5.3 数据检索模块实现 12

6 工具展示 13

7 总结与展望 14

# 引言

软件开发的过程中产生了大量的软件开发过程数据，包括项目的源代码，缺陷报告以及项目开发者们之间的通信邮件等，这些软件开发过程数据记录了软件从诞生开始的成长与发展轨迹。对这些数据进行研究，理解与使用可以帮助人们回答很多软件工程学的问题，例子如下。

开源社区的生存与发展离不开贡献者的参与，如何吸引与维持贡献者就成为开源社区非常关注的一个问题。Zhou和Mockus通过对Mozilla和Gnome两大社区的缺陷数据进行收集、分析与研究，建立模型解释了影响开源社区参与者成为长期贡献者的原因。［Zhou 12］

Mockus针对Apache和Mozilla两个社区的代码变迁数据进行分析与研究，回答了他所提出的六个假设[Mockus 02]，这篇文章也得出了许多有意义的结论，他的这些发现也被大量的软件工程研究者引用与再研究。Mockus还发现，收集与分析整个开源代码库，能够帮助解决许多软件工程的基础问题，包括代码演化等[Mockus 09]。

上述例子都是通过收集、分析与使用软件开发过程数据，总结出项目开发的最佳实践，从而指导软件项目的开发。因此，如果建立起一个统一的，普遍的软件开发过程数据资源库，提供给人们收集、存储与分析这些数据的方法，并提供一套检索软件开发过程数据的工具，就能帮助人们快速的收集，查询与检索软件开发过程数据，并进一步对它们进行研究，理解与使用，从而解决许多软件开发实践问题。

然而，互联网中存在的大量的软件开发过程数据，它们具有海量、分布、异构等特点，这使得我们利用这些数据变得困难，原因如下：首先，分布是指数据分散在互联网的各个角落，这使得寻找和收集这些数据变得困难。其次，异构则指存储软件开发过程数据的系统各异，例如存储源代码的版本控制系统就包括SVN, GIT, Bazaar等，而不同的系统的数据存储格式也不同，这就使得收集、存储和处理这些数据时需要根据不同的系统采用不同的方法。此外，从互联网上直接收集得来的原始数据有许多冗余或是无用的数据，例如从网页上收集而来的缺陷数据，大多是以HTML或是XML的文本进行存储的，这其中就有许多类似于<html>,<head>这样的无用信息，这就使得我们要使用这些数据的时候，许多对它们进行加工后再使用。综上所述，我们要使用软件开发过程数据，需要面对与解决不少问题和挑战。

本文尝试应对上述问题和挑战，以北京大学软件工程研究所数据分析小组的数据为基础（包括从互联网收集的源代码，缺陷数据，邮件数据等软件开发过程数据以及相应的数据处理脚本），尝试提出一种软件开发过程数据的收集，分析与使用方法，并设计与实现一套软件开发过程数据的检索工具。具体来讲，本文提出的方法与工具有以下几个方面：第一，从数据的处理脚本中提取出脚本的描述信息，包括脚本的处理对象，用途，使用方法等，用于建立一套收集，分析与处理软件开发过程数据的方法；第二，从软件开发过程数据中提取出数据的描述信息，描述数据的来源，基本信息与存储位置等，用于描述与检索数据；第三，通过数据与脚本的描述信息，建立检索工具，用于根据用户的检索需求定位和获取所需数据。

本文的新颖之处和难点在于：第一，为纷繁复杂，类型多变的软件开发过程数据建立格式统一，描述简单，易于理解的描述信息。软件开发过程数据的存储系统众多，不同的软件项目使用不同的系统，不同的系统的数据格式不尽相同。本文需要对这些数据进行统一整理，提取出格式统一，描述简单，易于理解的软件开发过程数据的描述信息。第二，为脚本进行分类与建立描述信息。脚本数据与软件开发过程数据不同，不能直接通过编写程序来提取出描述信息，本文需要人为的对处理数据的脚本进行分类，并建立包括脚本输入输出，用途等脚本的描述信息，以帮助建立一套收集，分析与处理软件开发过程数据的方法。第三，从大量的软件开发过程数据中定位到用户要检索的数据。首先选取合适的存储系统对描述信息进行存储，再对上述信息建立一套检索接口，将用户的检索请求分割成可执行的检索语句，利用检索接口进行检索，并将结果合并后返回给用户。

本文的主要贡献在于整理了一套收集、存储与使用软件开发过程数据的方法，设计与实现了一种软件开发过程数据的检索工具。本文的方法可以提供一套获取软件开发过程数据的方法，通过使用这套方法，研究者与项目开发者可以收集，分析与处理互联网上的软件开发过程数据。同时，本文的检索工具可以帮助研究者与开发者快速地获取到想要查找的软件开发过程数据。

本文结构安排如下：第一章是引言；第二章介绍软件开发过程数据的特点以及存储与检索所用到的相关技术；第三章对本文面临的问题与挑战进行分析并提出解决方案；第四章给出工具的设计；第五章介绍工具的具体实现；第六章对工具的效果进行展示；第七章介绍本文的工作进展和对未来工作的展望。

# 相关技术介绍

本文面向包含代码变迁数据，缺陷数据，邮件数据等软件开发过程数据，实现对这些数据的收集，整理，存储与检索。本文需要从大量的日志数据，缺陷数据和邮件数据中抽取数据的描述信息并进行存储，在此基础上建立这些软件开发过程数据的索引，并提供检索这些数据的工具。因此，本文面临着与多种软件开发过程数据的存储系统进行交互，对数据进行处理并建立数据的描述信息，利用描述信息对数据进行检索等问题。针对上面的问题，以下介绍本文面向的软件开发过程数据，数据的存储，数据的检索等方面的相关技术以及本文的相关工作。

## 软件开发过程数据

本文所述的软件开发过程数据就是指在软件开发过程中出现的用于辅助软件开发的数据。本文针对的软件开发过程数据包括三种类型：代码变迁数据，缺陷数据和邮件数据。

### 代码变迁数据

代码是软件项目的重要组成部分，它也是最重要的软件开发过程数据。代码一般被存储于版本控制系统中，以方便管理和协同开发。常见的版本控制系统有SVN, GIT, Bazaar等，这些系统记录了项目中所有的代码历史以及每一次的代码提交信息。本文所面向的代码变迁数据就是指项目开发过程中产生的代码提交日志，标记着项目中的每一次代码提交。这些数据通常包括如下信息：代码提交者，提交时间，本次提交所变更的文件信息以及对本次提交的注释等。数据的格式会由于所使用的版本控制系统的不同而不同，获取它们的方法也不尽相同

代码变迁数据可以让我们了解到项目中的代码是如何变化的，我们可以直接从这些数据中得到许多信息，包括整个项目的提交历史，项目的时间跨度，人员分布。同时，我们还可以通过分析这些数据，得到更细节的信息，包括项目代码是如何演化的等。因此，分析与理解这些信息，能让我们从宏观上和微观上了解项目开发的整个过程。

### 缺陷数据

缺陷数据记录了人们在软件开发过程中发现的软件项目存在的错误，或者对软件项目提出的新需求。

### 邮件数据

在软件项目开发过程中，项目的开发者往往会通过邮件进行交流，邮件数据则记录了这些开发者之间的日常交流活动。

## MongoDB

MongoDB是一种基于文件存储的非关系型数据库，它与传统的关系型数据库相比，具有可移植，可扩展，高性能，存取方便等特点。

MongoDB中的数据的存储模型从上到下分别为：数据库，集合，文档。MongoDB是面向集合进行存储的，并且模式自由。面向集合是指数据被存储在集合中，这里的集合类似于关系型数据库中的表。而模式自由则意味着数据存储在MongoDB中时，不需要定义任何模式，我们可以把任意结构的数据存入其中。存储在集合中的是文档，存储的形式是键-值对，键用于标识一个文档，为字符串类型，而值可以是任意复杂的文件类型。

本文使用MongoDB对数据进行存储，有以下几点原因：

1. MongoDB是一种非关系型数据库，不要求所存储的数据具有特定的结构，存储模式非常灵活。相对而言，关系型数据库则要求数据有确定的结构，并可以支持如表连接等复杂的数据库操作。对于本文所需要存储的数据而言，软件开发过程数据的描述信息之间差异很大，例如需要描述项目邮件数据的时间分布，不同的项目时间跨度不同，有的跨越的时间长，有的跨越的时间短，不方便使用特定的结构进行存储。同时，除了软件开发过程数据本身，本文还要存储处理这些数据的脚本的描述信息。除此之外，本文所存储的数据之间不需要很复杂的表连接操作，所以MongoDB非常适合用于存储本文所需存储的数据的描述信息。
2. MongoDB很好的实现了面向对象的思想，在MongoDB中，每一条记录都是一个文档对象，这使得我们在使用编程时，可以很方便的与数据库进行交互，同时它还具有多种语言配套的API，为我们操作数据库提供了很好的帮助。
3. MongoDB中的数据是使用JSON格式进行存储的。而本文的数据展示部分，其中的图形化界面使用web UI的形式进行展示。在前端与后端之间使用RESTful API进行交互，所传输的数据类型就是JSON，所以使用MongoDB可以为数据存储与传输带来极大便利。

综上所述，本文使用MongoDB作为数据的存储系统。

## pymongo

pymongo是MongoDB的一套python API，封装了MongoDB的基本操作，提供在python中操作MongoDB的一系列方法。

## 相关工作

# 问题分析与解决方案

## 问题分析

本文致力于回答的问题是：如何建立起一个统一的，普遍的软件开发过程数据库，并提供一套存储与检索这些数据的方法与工具。

## 解决方案

# 工具的设计

上一章我们对本文面临的主要问题及其解决方案进行了阐述。本章将介绍工具的设计，以及如何解决之前提出的问题。本章首先介绍工具的总体设计，然后再分别介绍工具各个模块的设计。

## 工具的总体设计

工具按照功能划分，包括三个部分：数据提取，数据存储，数据检索。工具的整体结构如图所示：

* 数据提取模块：

用于提取软件开发过程数据及相关处理脚本的描述信息。

* 数据存储模块：

用于将所提取的描述信息存储到数据库中。

* 数据检索模块：

用于通过所提取的软件开发过程数据的描述信息来对软件开发过程数据进行检索。

## 数据提取模块设计

数据提取模块是用于提取软件开发过程数据及相关处理脚本的描述信息，按照提取的需求可以分为四个小的模块：日志数据描述，缺陷数据描述，邮件数据描述，脚本描述，下文将针对上述几个小的模块详细描述数据提取模块的设计。

### 日志数据描述

### 缺陷数据描述

### 邮件数据描述

### 脚本描述

## 数据存储模块设计

## 数据检索模块设计

# 工具的实现

上一章介绍了工具的总体设计以及各个模块的设计情况，本章将介绍各个模块的实现细节。

## 数据提取模块实现

## 数据存储模块实现

## 数据检索模块实现

# 工具展示

# 总结与展望