###### **3月9日**

**项目流程:**

分析- +设计- 实现- 执行- 维护

woniuboss项目

JavaEE开发的应用.

1.首先配置Java的环境

。JRE Java的运行时环境

Java应用运行的最基本环境要求

。JDKI

Java开发工具包，当要开发Java的应用程序的时候需要安装

包含了JRE

在企业里面在服务器配置环境的时候-般会在服务器部署JDK， 而不是JRE

1. 所以要配置关系型数据库ROBMS,部署mysql,版本>5.1
2. 应用服务器tomcat,提供一个web服务

**项目部署:**

1.首先登录进入mysql数据库

2.创建一个名为woniuboss2.5的数据库

3. source导入sq|文件或者退出数据库 ，通过数据库还原的指令来导入sql

4.修改一下数据库默认用户root的默认密码为root

5.将war包复制到tomcat的webapps目录下

6.重启mysqI服务

7.重启tomcat服务

8.我们就可以通过浏览器访问woniuboss的项目了。

Ip addr查不到ip地址的原因:

Centos6: vi /etc/sysconfig/network-scripts/ifcfg-eth0 将ONboot的no改成yes

Centos7: vi /etc/sysconfig/network-scripts/ifcfg-ens33 将ONboot的no改成yes

配置jdk和tomcat环境变量命令: . vi ~/.bash\_profile

更新环境变量命令: source ~/.bash\_profile

内容:

JAVA\_ HOME=/opt/jdk1.8.0 231

PATH=$PATH: $H0ME/bin: $JAVA\_ HOME/bin

CL ASSPATH=$JAVA HOME/lib

CATALINA HOME=/ opt/apache- tomcat-9.0.27

CATALINA BASE=$CATAL INA HOME

关闭防火墙:service iptables stop

查看防火墙关闭状态：service iptables status

永久防火墙关闭：chkconfig iptables off

Linux安装mysql命令:yum install mysql-community-server

安装不完整,再次运行命令会自动补全安装包

登录mysql:mysql -uroot -p123456

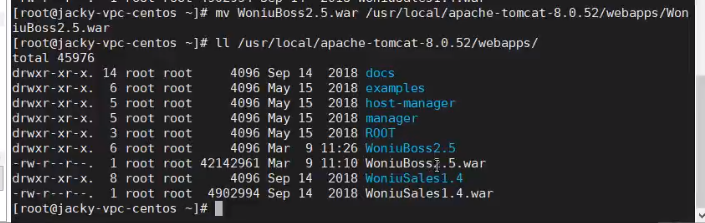
建库: create database `woniuboss2.5`;

退出:exit

查询tomcat位置命令: find / -name catalina.sh

重启mysql:service mysqld restart

移动文件:mv WoniuBoss2.5.war /usr/....



导入mysql数据文件 先将sql文件拖到root目录下,

source /root/woniuboss2.5.sql

mysql -uroot -p woniuboss2.5 < woniuboss2.5.sql

重启数据库:service mysqld restart 重启tomcat:进去bin目录先关 再开

# mysql 赋给用户权限 grant all privileges on命令

grant all privileges on `woniuboss2.5`.\* to 'root'@'%' identified by '123456';

项目配置出现问题如何查错

1. 首先检查本机是否能够ping通你的虚拟机

2.能够正常ping通的情况，需要检查tomcat管理页面是否能够打开

3.能够打开的基础上，再去检查tomcat安装目录下logs目录，查看其中localhost时间.log这样的文件中的错误信息.

###### 3月10日

VersionControl-版本控制；

ChangeControl-变更控制；

ProcessSupport-过程支持。

o配置管理管理的对象是软件

。配置管理是对软件整个生命周期过程中的任何时间节点在软件配置上进行标识，对这些被标识过的配置项的变更进行管理控制，从而保证软件产品的完整性和可回溯性的过程。I

常用的软件配置管理工具主要分为三个级别：

l Rational ClearCase，Perforce，CA CCC/Havest l Merant PVCS l Microsoft VSS，CVS

常用的开源免费的软件配置管理工具有：SVN、[GIT](https://baike.baidu.com/item/GIT" \t "https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E9%85%8D%E7%BD%AE%E7%AE%A1%E7%90%86/_blank)、CVS

CMMI共有5个级别，代表软件团队能力成熟度的5个等级，数字越大，成熟度越高，高成熟度等级表示有比较强的软件综合开发能力。

CMMI一级，执行级。

CMMI二级，管理级。

CMMl三级，明确级。

CMMI四级，量化级。

CMMI五级，优化级。

配置项

合同类文档

建议书、用户意向书、用户需求、工作任务书、合同

计划类文档

包括各类项目相关计划，比如项目过程手册、项目计划，配置管理计划等

工程类文档

包括需求规格文档、测试计划(含测试用例)、设计文档、需求跟踪矩阵等

程序代码

所有开发的源代码，包括各类支持数据，二进制文件

第三方程序代码

由供应商提供的源代码，并接受供应商的维护

工具

支持软件开发、建立、维护的工具管理，比如语言开发工具，编译工具，测试工具，配

置管理工具等

用户文档

包括用户手册，安装指南等

运行环境

包含系统运行环境的相关内容，比如系统运行平台，环境设置要求等

常用术语

➢版本和版本标识

◆版本是表示一个配置项具有 - -组定义的功能的一-种标识。

◆

随着功能的增加，修改或删除，配置项的版本随之演变。

◆

版本以版本号进行标识。

号码

发布类型

典型描述

XX

主版本号

增加了一个大的特性-可能导致与原先版本不兼容

YY

次版本号

增加了一个小的特性-保持与原先版本兼容

可以包含一些更改，并且包含自上一-次版本发布以后所有的补丁。-一般地，这个版本要

zz

维护版本号

根据保证和/或支持/维护协议主动提交给所有的用户

包含对客户或测试组发现和报告的一个或多个问题的解决。此版本只发布给报告问题的

PP

补丁版本号

客户或测试组。解决一次问题发布一个补丁版本给报告问题的客户或测试组。补丁也可

能只是一种优化

构建常用工具

➢代码相关

SVN、git

➢文档相关

wiki

➢构建相关

**ant**、 **maven**、

➢

Bug相关

jira、禅道、 Bugilla

配置控管：导入变更控管流程。该流程通常由变更控制委员会来运行，其主要的职责是核准或拒绝有悖任何基准的所有变更请求。

配置状态报告：记录和呈报与开发过程状态相关的所有必要信息。

配置审核：确保这些配置包含所有预期内容，且备有完整的规定文件（包括要求、结构规范和用户手册）。

建构管理：管理用于建构的流程和工具。

流程管理：确保遵循企业组织的开发流程。

环境管理：管理承载系统的软硬件。

团队合作：促进流程中团队彼此间的交互。

缺陷追踪：确保可溯及每个缺陷的源头。

在软件配置管理计划的制定过程中，它的主要流程应该是这样的：

CCB根据项目的开发计划确定各个里程碑和开发策略；

CMO根据CCB的规划，制定详细的配置管理计划，交CCB审核；

CCB通过配置管理计划后交项目经理批准，发布实施。

###### 3月11日

软件配置管理的组织

**项目经理（Project Manager，PM）：**

**项目经理是整个软件研发活动的负责人，他根据软件配置控制委员会的建议批准配置管理的各项活动并控制它们的进程。其具体职责为以下几项：**

**制定和修改项目的组织结构和配置管理策略；批准、发布配置管理计划；决定项目起始基线和开发里程碑；接受并审阅配置控制委员会的报告。**

**配置控制委员会（Configuration Control Board，CCB）：**

**负责指导和控制配置管理的各项具体活动的进行，为项目经理的决策提供建议。其具体职责为以下几项：**

**定制开发子系统；定制访问控制；制定常用策略；建立、更改基线的设置，审核变更申请；根据配置管理员的报告决定相应的对策。**

**配置管理员（Configuration Management Officer，CMO）：**

**根据配置管理计划执行各项管理任务，定期向CCB提交报告，并列席CCB的例会。其具体职责为以下几项：**

**软件配置管理工具的日常管理与维护；提交配置管理计划；各配置项的管理与维护；执行版本控制和变更控制方案；完成配置审计并提交报告；对开发人员进行相关的培训；识别软件开发过程中存在的问题并拟就解决方案。**

**系统集成员（System Integration Officer，SIO）：**

**系统集成员负责生成和管理项目的内部和外部发布版本，其具体职责为以下几项：**

**集成修改；构建系统；完成对版本的日常维护；建立外部发布版本。**

**开发人员（Developer，DEV）：**

**开发人员的职责就是根据组织内确定的软件配置管理计划和相关规定，按照软件配置管理工具的使用模型来完成开发任务。**

**配置管理过程**

**➢配置项标识**

**➢配置管理计划**

**➢配置库的建立与维护**

**➢基线变更管理**

**➢配置状态发布**

**➢配置审计与验证**

**Git:分布式版本管理工具**

**软件配置管理，贯穿于整个软件生命周期，它为软件研发提供了一套管理办法和活动原则。软件配置管理无论是对于软件企业管理人员还是研发人员都有着重要的意义。软件配置管理可以提炼为三个方面的内容：**

**VersionControl-版本控制；**

**ChangeControl-变更控制；**

**ProcessSupport-过程支持。**

**Git与SVN的区别:**

1.GIT是分布式的，SVN不是

2.Git直接记录快照，而而SVN是按文件

3.GIT分支和SVN的分支不同

4.GIT没有一个全局的版本号，而SVN有

5.GIT的内容完整性要优于SVN

6.SVN运行需要服务器支持,Git不需要

工作模式的区别:

无论是 svn 还是 git 的工作流，都是在本地解决冲突再提交，而不是在提交时解决冲突的。所以：

svn 的模式是：  
1。写代码。  
3。从服务器拉回服务器的当前版本库，并解决服务器版本库与本地代码的冲突。  
5。将本地代码提交到服务器。

分布式版本管理的模式是：  
1。写代码。  
2。提交到本地版本库。  
3。从服务器拉回服务器的当前版本库，并解决服务器版本库与本地代码的冲突。  
4。将远程库与本地代码合并结果提交到本地版本库。  
5。将本地版本库推到服务器。

所以，分布式版本管理仅仅是增加了本地库这个概念，其余的概念与集中管理并无区别。——但是 svn 在与服务器同步之前无法提交代码，因而本地修改更容易出问题。



###### 3月12日

Git的基本概念分区:

。工作区workspace(日常操作的文件数据)

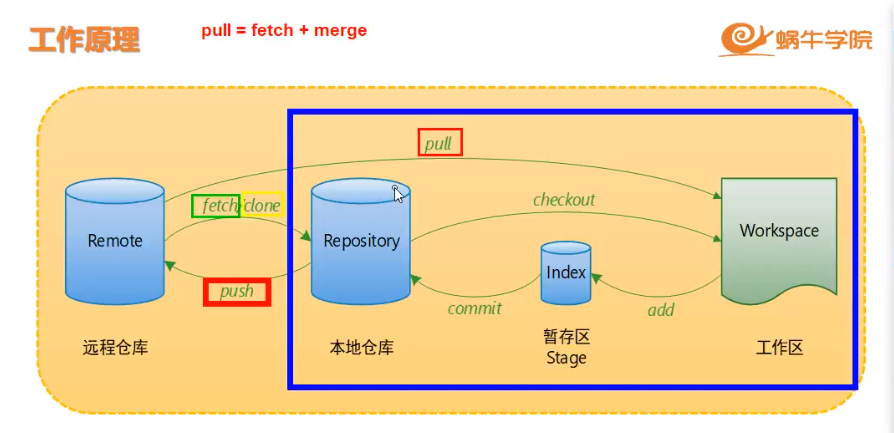
。暂存区stage(用于提交版本仓库的暂存管理.主要提供后悔机制)

。版本仓库repository(用于管理提交的文件数据，这里保存的是文件数据的所有版本。)

■本地仓库local repository

■远程仓库remote repository(不在本地机器上，主要用于分享)

Git的工作原理:



Git常见的图形化工具:

### **Source Tree, TortoiseGit, GitHub for Desktop**

**Git基本配置:**

# 配置用户名 git config --global user.name "username" //（ "username"是自己的账户名，） # 配置邮箱 git config --global user.email ["username@email.com"](mailto:\"username@email.com\")

用 git config --global --list 命令查看配置是否OK

git使用的基本流程

1.首先对windows硬 盘中要管理的文件夹进行一个初始化的动作(init初始化)

2.需要利用添加的动作把要管理的文件数据添加到暂存区中(add添加)

3.经过对暂存区的检查没有问题后就可以利用提交的动作将文件保存到本地仓库中了(commit提交)

4.可以利用推送的方式将本地仓库的数据分享到远程仓库(push推送)

基本用法

创建仓库

◆git init [repository path]初始化仓库

◆git init --bare [repository path]初始化裸仓库

克隆仓库

◆git clone [url]

url格式:户名@服务器地址:仓库路径

例: git clone git@host:path

●

建立仓库关联

◆git remote add [remote name] [ur]url格式:用户名@服务器地址:仓库路径

例: git remote add origin git@host:path

1 git init [path] #如果指定了path那就是对指定的path进行初始化,如果没有指定，那么就是对当前目录进行初始

2 git add # -A表示当前工作区全部内容添加到暂存区;。 表示当前目录全部;添加指定文件就是直接写文件名，可以有多个，但是中间需要用空格隔开

3 git comnit -m“这是当前版本的说明信息”# -m必须有这个参数，在参数后添加待提交版本的信息文字。

4 git status #查询当前仓库的状态

5 git 1og 查看历史记录

6 git reflog 查看操作步骤,也就是查看命令操作历史

7 git diff#用于进行文件比较或者版本比较。如果后面跟的是文件名，代表把工作去中的文件和当前版本中的对应文件进行比较。如果后面跟的是提交的名字，代表指定的版本间的比较。

8 #关于版本可以使用head head\*的形式表示，其中head代表当前最新版本，而head代表当前版本的前一个版本，如果是更多的^那么含义以此类推.

9 #关于版本还有另外-种表示，就是通过git reflog看到 的那种。head@ {0 )表示当前最新版本，head@{1)表示当前版本前一个，数字的变化代表之前第几个版本。

10 git reset 默认操作,修改版本仓库,只影响版本仓库和暂存区,对工作区不影响

11 # --soft: 修改版本仓库，保留暂存区，保留工作区。

12 # --mixdt: 修改版本仓库，修改暂存区，保留工作区。

13 # --hard:修改版本仓库，修改暂存区(清空暂存区)，修改工作区。

14 git checkout [commit] [file] #可以从本地仓库提取指定提交或者指定文件

配置一个linux系统环境下的git服务器，当作远程仓库来使用。

安装git

yum -y install git

添加配置linux用户， 主要出于安全考虑

groupadd git

useradd git -g git

passwd git #需要指定这个linux用户的登录密码，注意-定要记住。 因为未来要用。

对系统使用git用户名登录做一个安全处理， 不让正常登录

###### 3月13日

创建仓库

1 mkdir -p /home/git/XXX.git

git init --bare /home/git/XXX.git #注意用于分享的仓库必须使用bare参数初始化成- -个裸仓库

3 chown -R git:git /home/git/XXX.git

5.每次push或者pu或者clone统统都要输入密码， 很烦人。有没有可能偷点懒。

1.偷懒技巧

在客户端生成证书

1 ssh-keygen -t rsa -C” XXX@ABC . com"

生成证书后将会得到id\_ rsa和id\_ rsapub这一 对证书，其中id\_ rsa.pub是公钥，这个将用于git远程服务器

配置，而id. ,rsa是私钥，自己要保存好。

在服务器端添加公钥

mkdir -p /home/git/ . ssh

touch /home /git/ . ssh/ authorized. keys

vi /home/git/ . ssh/authorized. keys #注意将公钥的全部文本复制进去即可

chown -R git:git /home/git/ . ssh

chmod 700 /home/git/ .ssh

6 chmod 600 /home/git/ . ssh/ authorized. keys

如此处理后证书就配置好了，下面在客户端再次利用git访问服务器就不用输入密码了。

I

1 git checkout #有两种类型的操作

2 # 操作文件或者提交

3 #操作分支

Git分支操作

◆列出本地分支

◆git branch

◆查看远程分支

◆git branch -r

◆查看全部分支

◆git branch -a

###### 3月14日

Robot Framework

简介

1.基本特点

2工作原理

安装配置

1. robot framework配置环境需要配置哪些?

。安装robot framework (RF) 框架本身

Robot --version

，对于python 2和python 3都能够很好的支持。

。安装一个官方的IDE，名字叫RIDE

. RIDE长期只支持python 2.

. 2019年初时推出了1.7.3版本,加入了对python 3.6的支持。

wxpython针对于python 2不支持64位版本的python， wxPython2.8→2.8.12.1

ride=1.5.2.1

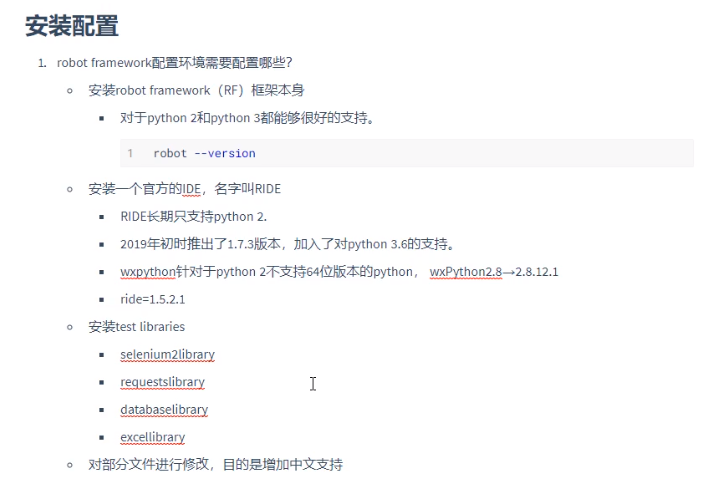
。安装test libraries

■selenium2library

■reguestslibrary

■databaselibrary

■excellibrary



###### 3月18日

代码分层

写好自动化测试用例的原则

➢

要做到流程和数据分离

➢

通过不断抽象消除冗余

➢

测试用例应当尽量简单易读，避免复杂逻辑

➢

建立用例分层架构并坚守

➢

自动化测试用例必须非常健壮，避免误报

9.循环语句

:FOR

IN RANGE/IN

4.补充内容

1.如何运用命令行来执行f的脚本

2.自定义库的方法

简单做法

■复杂做法

■库在项目中的做法

3. testdoc工 具的应用

4. libdoc工具的应用

5.数据驱动、关键字驱动、行为驱动三者的来历与应用

6.数据驱动在rf中的实现方法

7.行为驱动在rf中的实现方法

1. Given 假设一个场景

2. When 满足的条件

3. Then 当满足条件后就可以做一些事情了

4. And表示并且的作用

命令行运行RF文件:进入robot文件所在目录,启动CMD,输入robot 空格 加文件名,会自动执行文件内RF脚本并生成相应的report和log的html文件.

导入自己的写的库:1,直接在site-packages目录下创建py文件,2在该目录下创建文件夹,里面可以包含init文件和其他多个py文件,另外还可以生成自己写的库的说明文档:在需要生成文档的目录下运行cmd,输入命令python -m robot.libdoc 库名 需要生成的文档名.html

KDT:关键字驱动,DDT:数据驱动,BDD:行为驱动

###### 3月19日

Loadrunner简称LR

社区共享版和企业版功能一样,但是虚拟用户数只有50个.

Loadrunner组成四部分

。Vuser Generator

主要用于开发性能测试脚本。

。Controller

主要用于设计性能测试场景、用于执行性能测试脚本、用于监控脚本执行时的系统状态。

。Analysis

主要用于对性能测试执行后产生的结果进行分析。

。Load Generator

负载生成器

安装loadrunner的时候系统提示缺少KB文件,需要去官网下载相应的KB文件后双击打开更新,有一个KB2919355无法安装更新,需要修复:

**方法1：修复系统组件**

使用收件箱部署映像服务和管理 (DISM) 工具修复更新

在“搜索”框中，键入“命令提示符”，右键单击“命令提示符”，然后单击“以管理员身份运行”。如果您收到管理员密码提示或确认提示，请键入密码或单击“允许”。

键入以下命令，并在键入每个命令后按 Enter。注意：每个命令操作可能需要几分钟才能完成。

•DISM.exe /Online /Cleanup-image /Scanhealth

•DISM.exe /Online /Cleanup-image /Restorehealth(如果出现报错,再根据响应报错信息进行修复)

3. 关闭命令提示符，然后再次运行 Windows Update。

###### 3月20日

Vuser Generator

1.脚本编写

。GET和POST请求的实现

。检查点

。关联

。思考时间

。事务

。参数化

。集合点

注册函数web\_reg\_find()必须要出现在其所需要处理的函数之前.

通常，注册类函数名称中包含\_reg\_的字眼，例如，web\_reg\_find(),web\_reg\_save\_param()等。

web\_find是在返回的页面中进行内容查找，web\_reg\_find是在HTML源文件中查找所需内容.

web\_find只能只用在基于HTML模式录制的脚本中，而web\_reg\_find没有此限制 。

web\_find在执行效率上不如web\_reg\_find

web\_find是普通函数，web\_reg\_find是注册函数

###### 3月23日

| **代码** | **说明** |
| --- | --- |
| . | 匹配除换行符以外的任意字符 |
| \w | 匹配字母或数字或下划线或汉字 |
| \s | 匹配任意的空白符 |
| \d | 匹配数字 |
| \b | 匹配单词的开始或结束 |
| ^ | 匹配字符串的开始 |
| $ | 匹配字符串的结束 |

常用限定符

| **代码/语法** | **说明** |
| --- | --- |
| \* | 重复零次或更多次 |
| + | 重复一次或更多次 |
| ? | 重复零次或一次 |
| {n} | 重复n次 |
| {n,} | 重复n次或更多次 |
| {n,m} | 重复n到m次 |

常用的反义代码

| **代码/语法** | **说明** |
| --- | --- |
| \W | 匹配任意不是字母，数字，下划线，汉字的字符 |
| \S | 匹配任意不是空白符的字符 |
| \D | 匹配任意非数字的字符 |
| \B | 匹配不是单词开头或结束的位置 |
| [^x] | 匹配除了x以外的任意字符 |
| [^aeiou] | 匹配除了aeiou这几个字母以外的任意字符 |

思考时间:lr\_think\_time

打印:lr\_output\_message

取值:lr\_eval\_string