

如何通过精准化测试来解决效率黑洞

主讲人：王鹏

搜狗 资深高级测试开发工程师

目录

Contents

01、影响效率的因素

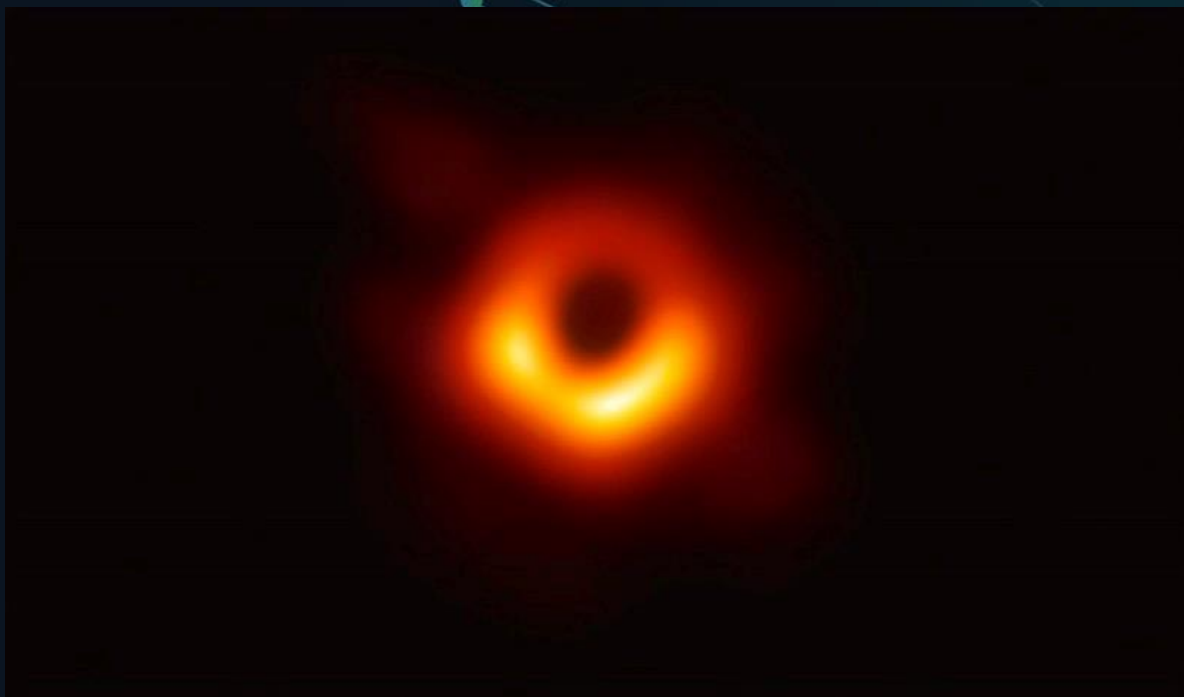
02、精准化测试思想

03、提升效率的手段

01

章节 PART

影响效率的因素



效率的黑洞

- 投入产出比低

R: 工作效率低下

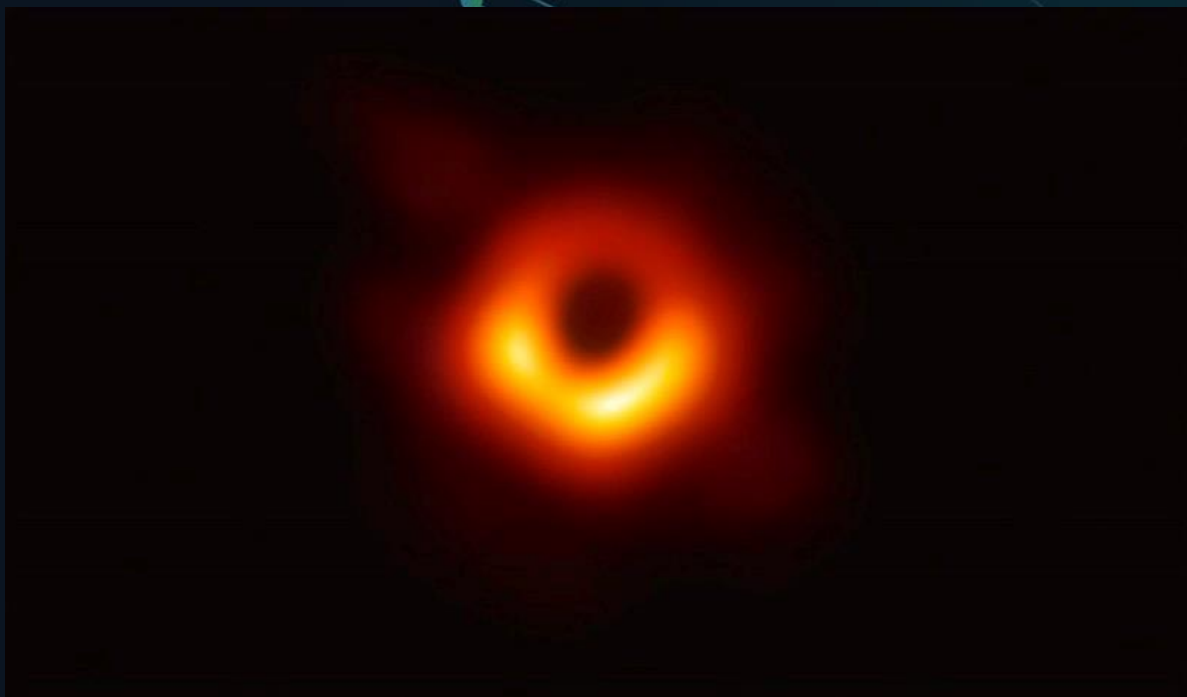
R: 创新性工作启动难

- 成效不可衡量

R: 没有一套被认可的解决方案

R: 所测项目输出质量被质疑

黑洞: 质量大体积小 工作: 付出多成效少



黑盒测试:

- 过程
 - ✓ 多猜测
 - ✓ 不稳定
 - ✓ 难控制
- 效果
 - ✓ 个人素质
- 管理
 - ✓ 成本高
 - ✓ 难度大

白盒测试:

- 门槛高
- 单兵作战
- 目标单一
- 分析之殇

02

章节 PART

精准化测试思想

用非常**精准**和**智能**的软件来解决软件测试的问题，并从根本上引领软件测试从**经验型方法**向**技术性方法**的转型。

函数调用关系计算

测试用例

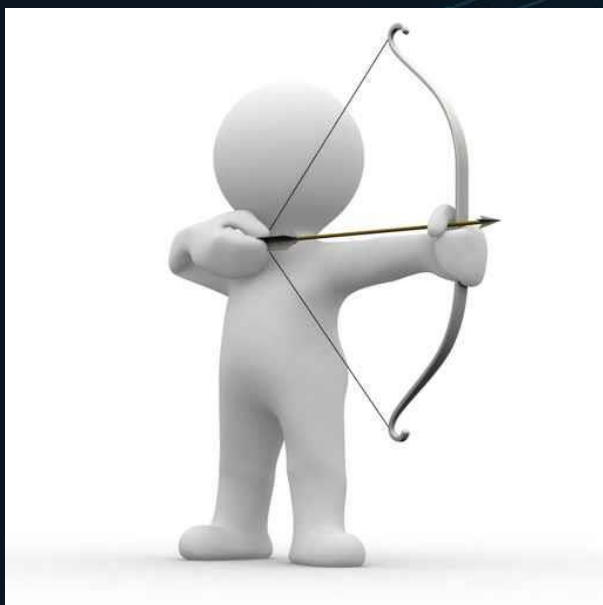


代码逻辑

代码染色

A

测试用例到代码逻辑的精准记录的双向追溯



B

精准的代码级的缺陷定位和崩溃分析

C

精准的测试充分度分析



A

回归用例自动筛选

B

自动化用例筛选与执行

C

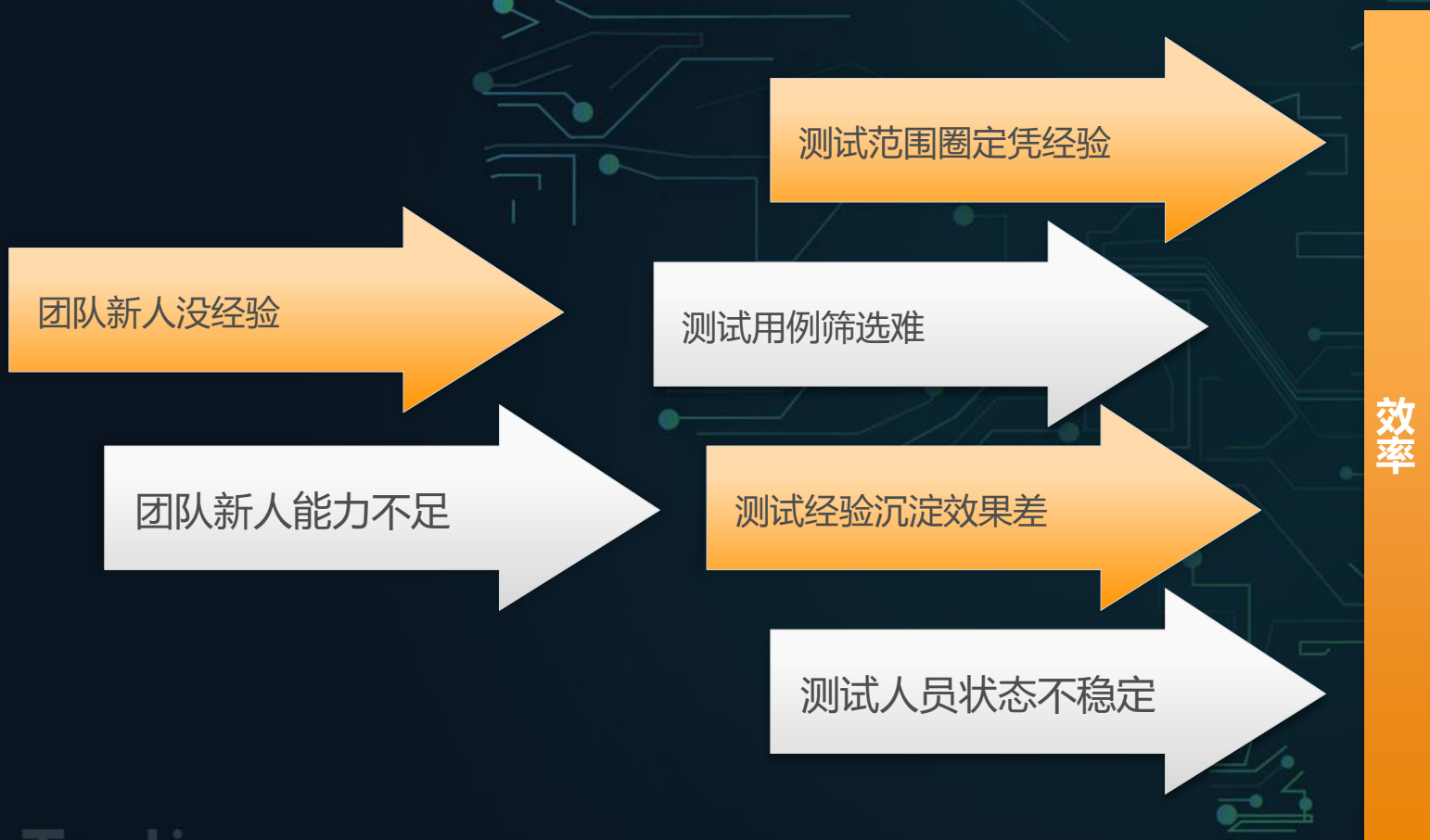
持续集成

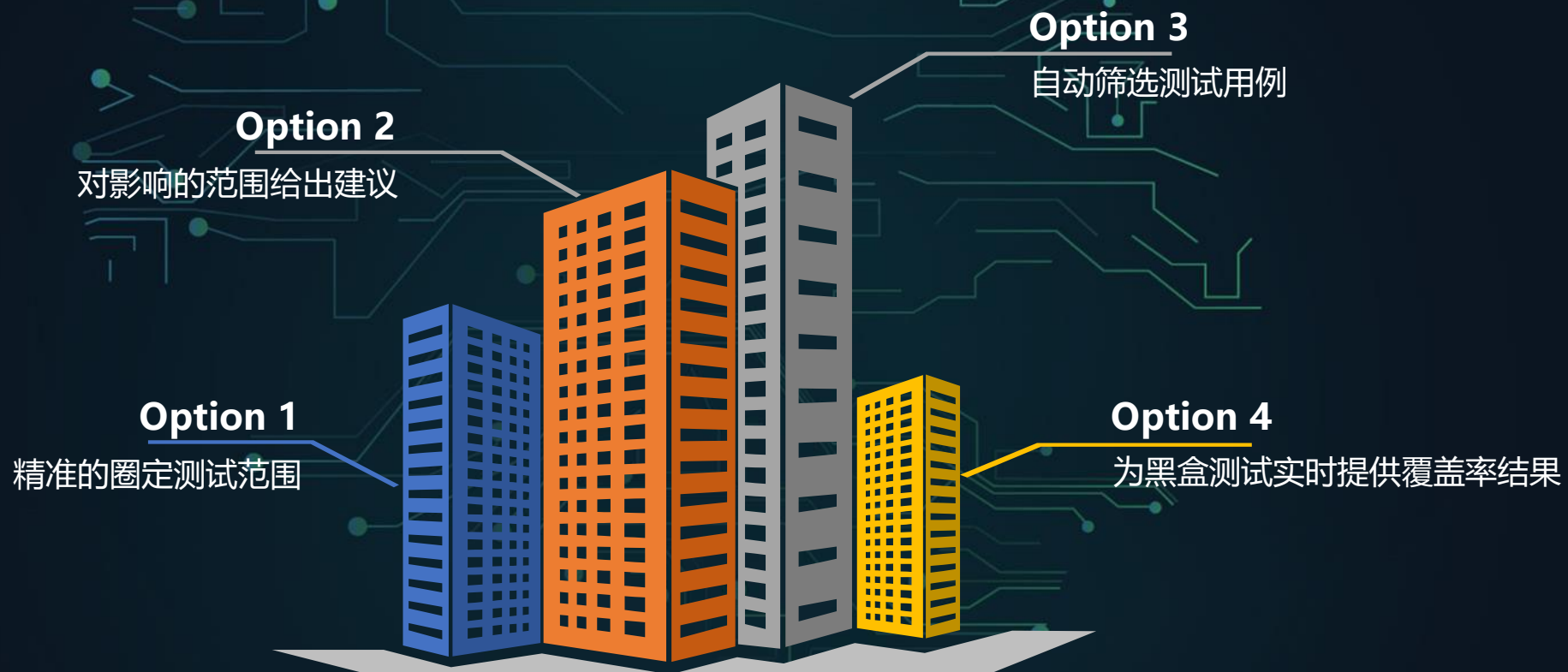
03

章节 PART

提升效率的手段

影响效率的因素





01

版本提测



提测版本信息
线上版本信息

Diff结果解析



变更函数计算
函数调用关系计算

测试用例生成

03



版本测试用例
Sanity test用例

example

开发为了实现功能1，同时修改了函数A和函数C，其中函数之间调用关系如下。

功能1: C->A

功能2: B->A

问题：在做Sanity测试时，如何能够筛选出功能2的测试用例。

应用语言: java

应用工具: javacg

目的：通过对class文件的解析，计算函数之间的调用关系。

```
java.  
javax.  
com.wenwen.jutil.  
com.wenwen.log.  
com.google.common  
org.  
utils.
```

01

变更代码段

解析Diff结果文件，计算变更文件名和变更代码段位置信息。

变更函数

扫描源文件，通过变更代码段的位置信息，计算出变更函数和类。

02

03

影响范围

结合函数调用关系圈定受影响的函数范围。

Diff结果解析

A.Java
Change
Code

Code Segment1

StartPos1

EndPos1

Code segment2

StartPos2

EndPos2

Code segment3

StartPos3

EndPos3

A.Java

Source

Code

Method1

StartPos1

EndPos1

Method2

StartPos2

EndPos2

Method3

StartPos3

EndPos3

Diff结果解析

```
def handlerDiffFile(self):
    diff_map={}
    groups=self.splitTXTByInde
    nowClassInfo=''
    for block in groups:
        start=0
        offset=0
        lines=block.split('\r
        if ((len(lines)==0) o:
            continue
        for line in lines:
            if self.isClassIn:
                nowClassInfo=
                diff_map[nowC:
            else:
                if self.isDif:
                    start=int
                    offset=0
                else:
                    if line.s:
                        diff_
                    if line.s:
                        offse:
    return diff_map
```

```
packageName = packageName + key.split('/')[0]

path = self.path + '\\\' + key.replace('/', '\\\'')
lines = open(path, 'r').readlines()
```

```
for line in lines:
    if "class" in line:
        temp = self.getClassName(line)
        if temp <> None:
            className = temp
            classLineInfo.setdefault(className, i)
        if "public" in line or "private" in line or "protect" in line:
            if '(' in line:
                temp = self.getMethodName(line)
                if temp <> None:
                    methodName = temp
                    methodLineInfo.setdefault(className+ ':' + methodName, i)
            i = i + 1
```

```
classLineInfo = sorted(classLineInfo.iteritems(), key = lambda asd:asd[1], reverse = False) #按行数排序，升序排序
methodLineInfo = sorted(methodLineInfo.iteritems(), key = lambda asd:asd[1], reverse = False)
lineNums = self.diffFileInfo[key]
#lineNums = [61, 81, 112, 125, 164]
```

```
for num in lineNums:
    className = ''
    methodName = ''
    temp = ''
    for node in methodLineInfo:
        if num >= node[1]:
            temp = node[0]
            continue
        else:
            methodName = temp
            break
    if methodName == '' and temp <> '':
        methodName = temp
```

```
if methodName <> '':
    name = packageName + '.' + methodName
    try:
        sql = 'insert into javadiffresult (projectId, method) values (%s, "%s")' % (str(projectId), name)
        cur.execute(sql)
    except MySQLdb.Error, e:
        print e
```

Index: adapter/AppUserRankAdapter.java

```
--- adapter/AppUserRankAdapter.java      (版本 150306)
+++ adapter/AppUserRankAdapter.java      (版本 149887)
@@ -3,7 +3,6 @@
import android.content.Context;
import android.net.Uri;
import android.support.design.widget.BottomNavigationView;
```

```
{
}) {
    a.getPortraitUrl());

    Name().trim());
    tPortraitUrl());
    ().trim());
    aeOf(data.getRecvPraiseNum());
    +4));
    错误爬 "+data.getRecAnswerNum());
```

```
=====
306)
3887)
```

```
stRecyclerView;
[temJoinUsersView;
[temTagView;
[temUserRoleView;
RankView;
```

```
idLayoutManager;
```

```
t<>());
```

```
public interestRelatedListAdapter(Context context,String cid) {
```

[精准化信息](#)[版本Diff](#)[函数依赖](#)[历史信息查询](#)[用例详情](#)[SVN部署](#)[覆盖率](#)[统计](#)

选择项目: 1 : 百科

选择工程: baike-image

选择Diff类型: trunk diff trunk

请输入tag1名称

请输入tag2名称

输入需要比对的版本revision 194884 181976

[生成diff内容](#)[查看变更函数](#)[查看diff内容](#)[查看用例集](#)[下载用例集文件](#)

有变动的函数

com.sogou.baike.BaseUtils.java.BaseUtils:generateOldFileName

com.sogou.baike.BaseUtils.java.BaseUtils:getNewFileName

com.sogou.baike.BaseUtils.java.BaseUtils:generateNewFileName

com.sogou.baike.include.SchemeUtils:modifyImageGroupDescriptorScheme

20%

人工录入

人工阅读源码，梳理与业务相关函数，并与测试用例建立相关性。

低效

30%

注释抽取

推动开发撰写注释，使用工具自动抽取函数和注释信息，人工建立与测试用例的相关性。

中效

50%

代码染色

根据测试人员行为，自动录入测试用例和函数的对应关系。

高效

分类

精准化

选择项目:

选择包:

选择类:

选择上传S

查看记录

id		
1753	onO	
1754	OnO	
1755	qqF	
1756	qqL	
1757	sog	
1758	wei	
1759	noLogin	游客模式登陆
1760	getSogouTokenData	获取搜狗登陆的验证码
1761	handleLoginSuccess	判断是否首次登陆,如果是首

```
item.setModule(feed.getType() == Feedtype.ZZ_QUESTION ? MsDataModule.brilliantAnswer : MsDataModule.lueDongHotTopic);
item.setTitle(feed.getTitle());
item.setPic(ImageUtil.conventImgUrl(feed.getPic_url()));
item.setUrl(String.valueOf(feed.getQid()));
item.setId(item.getUrl());
item.setPraiseNum(feed.getPraise_num());
items.add(item);
}
} else {
    logger.error("getHotQuestionItems(req:{{}} ## rsp:{{}}", req, r
}
return items;
}
```

```
private List<MsDataItem> getAllDataResult(MsDataRequest request) thr
    String qid = request.getQidStr();
    MsDataResult result = new MsDataResult();
    List<MsDataItem> items = Lists.newArrayList();
    //从redis里取 百科 指南
    MsDataItem relatedBaiKe = redisService.getBaike(qid);
    MsDataItem relatedGuide = redisService.getGuide(qid);
```

精准化信息编辑:

方法名

getAllDataResult

功能简介

请输入功能简介

功能描述

关联case

请输入关联case,用;号分隔

case链接:

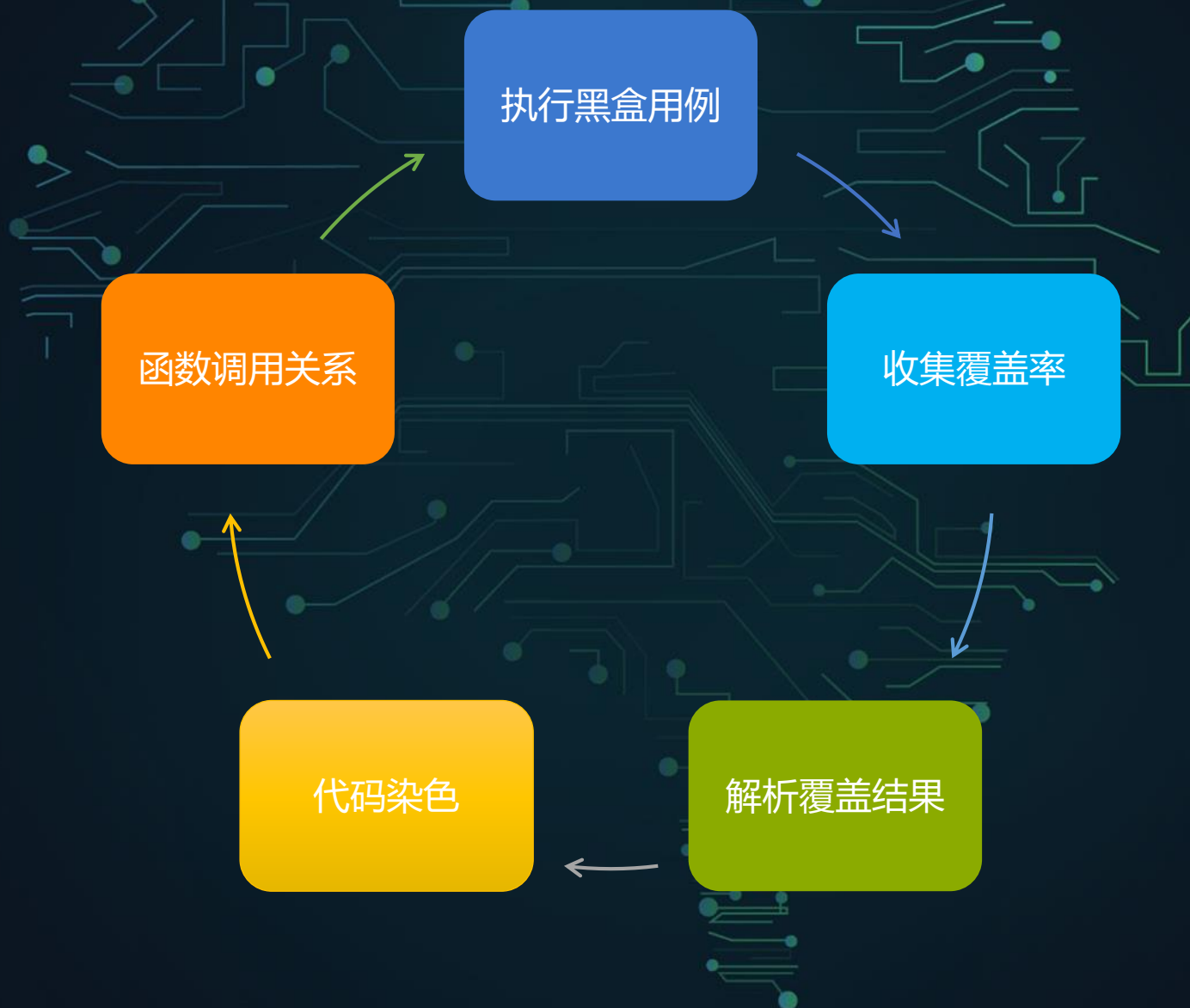
关闭

保存

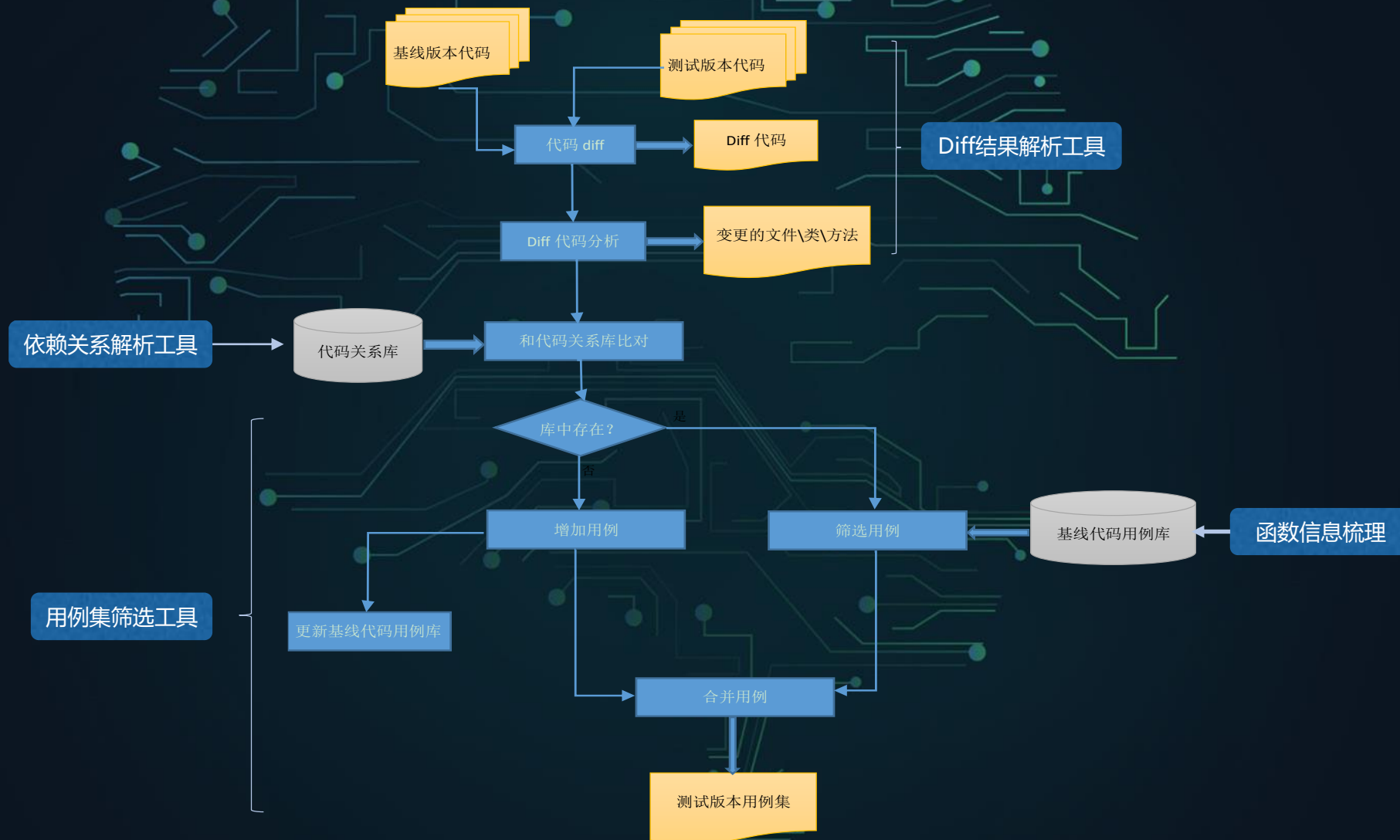
关闭


提交

id	名称	操作
5	游客模式	编辑 删除
5	获取搜狗登陆的token	编辑 删除
0	首次登陆判断和保存	编辑 删除
0		编辑 删除
8		编辑 删除
2		编辑 删除
	游客模式	580783 编辑 删除
	获取搜狗登陆的token	574948 编辑 删除
	首次登陆判断和保存	582228 编辑 删除
		580232 编辑 删除
		580086 编辑 删除




Testin
助力产业智能化



 精准化信息 版本Diff 函数依赖 历史信息查询 用例详情 SVN部署 覆盖率 统计

选择项目: 1 : 百科

选择包: all

 精准化信息 函数依赖 版本Diff 历史信息查询 用例详情 SVN部署 覆盖率 统计

用例集Id 查看

测试用例集:580143

问题详情页未登录态验证

问题详情页展示验证

问题详情页操作验证

问题审核未通过验证

步骤

预期结果

1.问题区

1.1) 返回按钮；
+2) 关注按钮；
+3) 分享按钮；
+4) 兴趣标签：默认展示1行（一级兴趣显示在首位）；
+超1行显示展开按钮，点击显示全部。（可收回，不满1行不显示该按钮）；
+5) 问题主体：展示全部内容，正方形展示全部（0-5张）图片；
+6) 提问者：头像、昵称、身份标识（最多两个，与内容主标签匹配：如动漫下的内容只展示动漫下的身份标识）

2.回答区

+2.1回答不为空

2.
+2.1+推荐回答、最新回答分别按回答时间倒叙展示（显示对应数量，上滑吸顶）
+1) 回答人信息：头像、昵称、身份标识（最多两个，与内容主标签匹配：如动漫下的内容只展示动漫下的身份标识）
+2) 回答主体：5行截断展示，展示图片或视频；
+3) 【。。。更多】按钮



- Jacoco
- On-the-fly
- class

Java
覆盖率

JS
覆盖率

- 基于JSCover二次开发
- 插桩+上报+proxy
- 源文件、编译文件 (ES6)

代码插桩

- 源文件插桩
- 编译文件插桩

数据上报

- 事件上报数据

文件映射

线上->proxy->本地

生成报告

- 解析jscoverage.json
- 生成html报告

通过捕获事件进行覆盖率实时上报

- 页面切换
- 鼠标移动

.....

```
//无线
//$(window).on("touchmove", throttle(reportData, 10000));
window.addEventListener("touchmove",throttle(reportData, 10000));
//PC
//$(window).on("mousemove", throttle(reportData, 10000));
window.addEventListener("mousemove",throttle(reportData, 10000));
```

```
def rewriteInstrumentFiles(self, fileName):
    os.chdir(self.basePath + '\\target')
    sourceFile = open(fileName, 'r')
    codeTop = open('codeTop.o', 'r')
    sourceLines = sourceFile.readlines()
    sourceFile.close()
    count = len(sourceLines)
    templines = ""

    insertCode = ''
    var request = jscoverage_createRequest();
    if (typeof(_jscoverage) === "undefined" && (typeof(Storage) !== "undefined") && typeof(localStorage["jscover"]) !== "undefined")
        _jscoverage = jscoverage_parseCoverageJSON(localStorage["jscover"]);
    if (typeof(jscoverbeforeunload) === "undefined") {
        var jscoverbeforeunload = (window.onbeforeunload) ? window.onbeforeunload : function () {};
        window.onbeforeunload = function () {
            jscoverbeforeunload();
            if ((typeof(_jscoverage) !== "undefined") && (typeof(Storage) !== "undefined")){
                request.open('POST', '/ui/js', true);
                request.setRequestHeader('Content-Type', 'application/json');
                request.send(jscoverage_serializeCoverageToJSON());
            }
        };
    }

    i = 0
    for eachline in sourceLines:
        if i == 0:
            if fileName in eachline:
                templines = templines + insertCode + eachline
                i = 1
            else:
                templines = templines + eachline

        elif i == 1:
            templines = templines + eachline

    sourceFile = open(fileName, 'w')
    codeTopLines = codeTop.readlines()

    sourceFile.writelines(codeTopLines)
    sourceFile.writelines(templines)

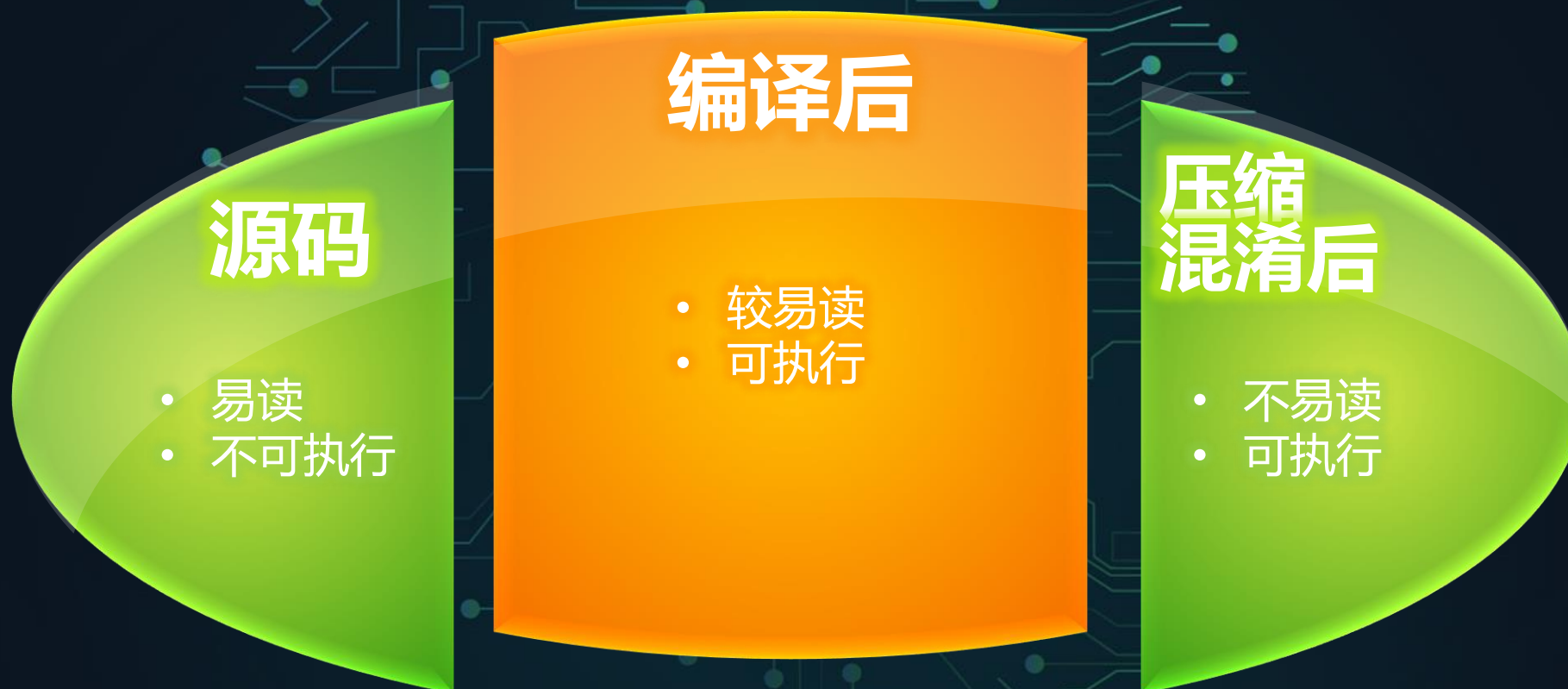
    codeTop.close()
    sourceFile.close()
```

兼容ES6标准

```
module_domestic_college.vue x module_episodes_link.vue x dialog.vue x login.vue x module_search.vue x
ESLint: Please specify ESLint package

script
1 <template>
2   <div>
3     <!-- 剧集选集列表 -->
4     <div class="top-episodes-pop" v-if="isAddMore" style="...">
5       <div class="mask-white"></div>
6       <div class="top-episodes-content">
7         <div class="header header-mod">
8           <div class="header-main">
9             <div class="header-close" @click="closeSelectLayer()"><span>关闭</span></div>
10            <h1>剧集</h1>
11          </div>
12          <div class="header-box"></div>
13        </div>
14        <div class="top-episodes-less">
15          <div v-if="episodeUpdateNum != episodeAllNum" class="top-episodes-txt">更新至<span
16            v-text="episodeUpdateNum"></span>集/共<span v-text="episodeAllNum"></span>集
17          </div>
18          <div v-else class="top-episodes-txt">已完结 共<span v-text="episodeAllNum"></span>集</div>
19
20          <div class="top-episodes-box">
21            <ul class="top-episodes-list">
22              <li v-for="(itemLink, index) in linkAll">
23                <a href="itemLink.episodeUrlWap" target="_blank"
24                  :class="['top-episodes-item', showNew && index == 0 ? 'new' : '']"
25                  onclick="stget('newwap.tvfc.dj', 'wapbaike');DT.reportCLK('newwap.tvfc.dj', 'wapbaike')">
26                  <span v-text="itemLink.episodeName"></span>
27                </a>
28              </li>
29            </ul>
30          </div>
31        </div>
32      </div>
33    </div>
34  </template>
```

```
style.css x common_new.js — sogousvn\...\.js x episodes_link.bundle.js x common_new.js — JSCover-1.0.23\target\onlineFiles x and
1 webpackJsonp([12],{
2
3   /**/ 116:
4   /**/ (function(module, exports, __webpack_require__) {
5
6     module.exports={render:function () {var _vm=this;var _h=_vm.$createElement;var _c=_vm._self._c||_h;
7       return _c('div', [_c('div', {
8         staticClass: "top-episodes-pop",
9         staticStyle: {
10           "visibility": "visible"
11         },
12       }, [_c('div', {
13         staticClass: "mask-white"
14       }, _vm._v(" ")), _c('div', {
15         staticClass: "top-episodes-content"
16       }, [_c('div', {
17         staticClass: "header header-mod"
18       }, [_c('div', {
19         staticClass: "header-main"
20       }, [_c('div', {
21         staticClass: "header-close",
22         on: {
23           "click": function($event) {
24             _vm.closeSelectLayer()
25           }
26         },
27       }, [_c('span', [_vm._v("关闭")])]), _vm._v(" ")), _c('h1', [_vm._v("剧集")])]), _vm._v(" ")), _c('div'
28         staticClass: "header-box"
29       })]), _vm._v(" ")), _c('div', {
```



选择较易读、可执行的编译后文件作为插桩文件

黑盒测试实时查看覆盖率



精准化信息 版本Diff 函数依赖 历史信息查询 用例详情 SVN部署 覆盖率 统计

```
24. * creat on 2017/11/22 ...
25. *
26. * 关于“Feed类热门内容填充”需求开发中:【20180329 - ytt】
27. * 下次Feed类不多弄几天
28. * 1: 去掉原来的Feed类热度和首页banner20条的加载
29. * 2: 去掉Feed类中的total和relatedQuestionTotal等无用字段
30. * 3: 考虑平滑的将mongo中存储的格式转换为List<Item>类型 (原本过期的数据继续使用, 所有的都用list)
31. * 4: 在3的基础上将DataService里方法的返回结果改为List类型
32. * 5: dubbo接口的返回结果也应为List类型
33. *
34. * ps: 涉及到的工程除了feed类的两个之外还有webapp_search, 每天定时加载 首页banner20条
35. */
36. @Service
37. public class MainSiteDataAction {
38.     private static Logger logger = Logger.getLogger(MainSiteDataAction.class);
39.
40.     @Resource
41.     private MongoTemplate mongoTemplate;
42.
43.     /**
44.      * 返回new MDataResult()mongo过期 OR 没有数据 (没存进mongo)
45.      * 返回null:数据已用完
46.      */
47.     public MDataResult getMDataResult(MDataRequest request) throws Exception {
48.         String qidStr = request.getQidStr();
49.         MDataResult bean = null;
50.         try {
51.             bean = mongoTemplate.findOne(qidStr, MainSiteFeedBean.class);
52.             // catch (Exception e) {
53.             //     logger.error("mongoTemplate.findOne() err", qidStr, e);
54.             // }
55.             if (bean == null) return new MDataResult();
56.
57.             //判断过期的时间 过期则重新获取
58.             Calendar createTime = Calendar.getInstance();
59.             createTime.setTime(bean.getCreate_time());
60.             Calendar expireTime = Calendar.getInstance();
61.             expireTime.add(Calendar.HOUR_OF_DAY, -Util.is.getFeedMongoExpireTime());
62.             if (expireTime.compareTo(createTime) > 0) {
63.                 return new MDataResult();
64.             }
65.             //过期, 取mongo里的result
66.             MDataResult result = bean.getResult();
67.             int start = request.getDataStart();
68.             List<MDataItem> items = result.getItems();
69.             if (start >= items.size() || start < 0) {
70.                 return new MDataResult();
71.             }
72.         }
73.     }
74. }
```

白色为不计算覆盖

蓝色为代码已覆盖

黄色为代码逻辑未覆盖全

红色为代码未覆盖

全屏显示 差异显示

置顶

收起该函数最新测试结果

查看js覆盖率报告

43.0

3.0

3054

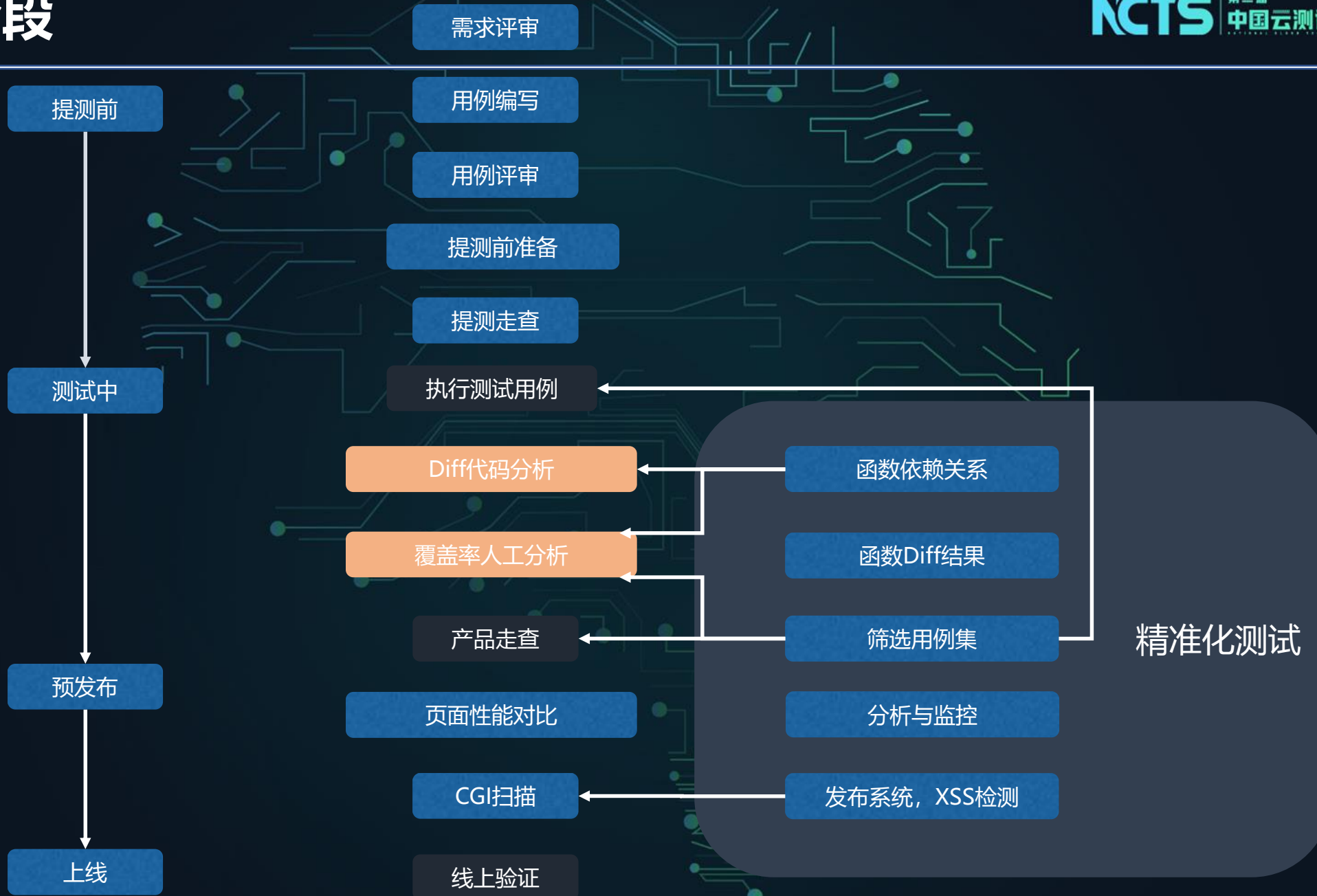
1482

查看分析结果

RED : 未覆盖 REDC : 未覆盖变更 REDC++++ : 未覆盖变更中新增部分 REDC---- : 未覆盖变更修改部分 C++++ : 覆盖变更新增部分 C---- : 覆盖变更修改部分

```
1 2 webpackJsonp([3],[
2
3     /**/ 10:
4     /**/ (function(module, exports) {
5
6
7     /**
8      * When source maps are enabled, `style-loader` uses a link element with a data-uri to
9      * embed the css on the page. This breaks all relative urls because now they are relative to a
10      * bundle instead of the current page.
11      *
12      * One solution is to only use full urls, but that may be impossible.
13      *
14      * Instead, this function "fixes" the relative urls to be absolute according to the current page location.
15      *
16      * A rudimentary test suite is located at `test/fixUrls.js` and can be run via the `npm test` command.
17      *
18      */
19
20 1 module.exports = function (css) {
21     // get current location
22 0 var location = typeof window !== "undefined" && window.location;
23
24 0 if (!location) {
25 0 throw new Error("fixUrls requires window.location");
26     }
27
28     // blank or null?
29 0 if (!css || typeof css !== "string") {
30 0 return css;
31     }
32
33 0 var baseUri = location.protocol + "//" + location.host;
34 0 var currentDir = baseUri + location.pathname.replace(/[\/\\]+$/, "/");
35
36     // convert each url(...)
37     /**
38      * This regular expression is just a way to recursively match brackets within
39      * a string.
40      */
```


引入阶段





The background of the slide is a dark blue-grey color. Overlaid on this is a complex, glowing green circuit pattern. The pattern consists of numerous thin, interconnected lines that form a web-like structure, with several small, solid green circles acting as nodes or solder points at various intersections and endpoints. The overall shape of the pattern is somewhat irregular, filling most of the frame.

THANK YOU

感谢聆听

精准化信息平台

