

IssueTracker产品设计方案

张刚2018.7.31

用户群体A：管理者和开发者（IssueTrackerWeb）

缺陷、克隆、
架构问题
等单版本缺陷检测



面向演化的
问题持续跟踪
管理



IssueTracker-持续问题合并、跟踪

用户群体B-研究者及未来产品演进(IssueTrackerCore)



IssueTracker Web

用户场景-登录

- 用户名、密码方式
- 手机号、验证码方式
- 社交账号登录方式
(微信、github等)

统一的用户ID

说明：注册、登录场景是通用场景；在初始研发阶段优先级不高；确保用户能登录即可；暂时仅实现用户名、密码登录即可；

如有空闲研发资源（如其他工作处于等待状态），可完善登录、注册系统的开发

账号

用户

登录

用户名

密码

验证码登录



社交账号登录

注册

- （待设计，仅限于用户名，密码的账号系统）

账号关联

- 如果是社交账号登录，则允许在登录后关联其他账号类型；
- 社交账号、验证码等登录方式，会自动分配一个用户名；
- 用户可以更改自己的用户名；

登录后的主页

项目管理

仪表盘

个人设置



项目名称 代码库类型 监控URL地址 最后更新日期 检测状态 动作

有没有可能把仪表盘和项目管理合二为一？会不会更简洁？
对于大多数用户，不会同时关注太多项目

说明：动作可以是“立即检测”、“查看详情”，“管理”

添加项目

- 项目可以对应到代码库，也可以对应到一些代码快照的序列（在无法直接访问代码库的情况下）；
- 添加项目的本质是建立了一个“用户”-“代码库”的映射关系。代码库自身如果存在，不会在系统中按照不同的用户存在不同的代码库实体；
- 主体结构是类似于jenkins的项目账号添加方式；
 - 如果有用户的github账号，则此处的项目添加可以有快捷方案；
 - 如果是上传代码的方式，则允许此处没有代码库对应。

添加项目

项目名 为空时，可在项目url更新时自动推算

代码库类型 ☐ git ☐ svn ☐ github ...

代码库地址

说明：后台会自动根据URL和代码库地址归并实际的代码库；项目名不会自动化归并，这个是用户的助记符，不同用户可以不一样

项目仪表盘

昨日

10

新增问题

本周

本月

30

消除问题

1000

剩余问题

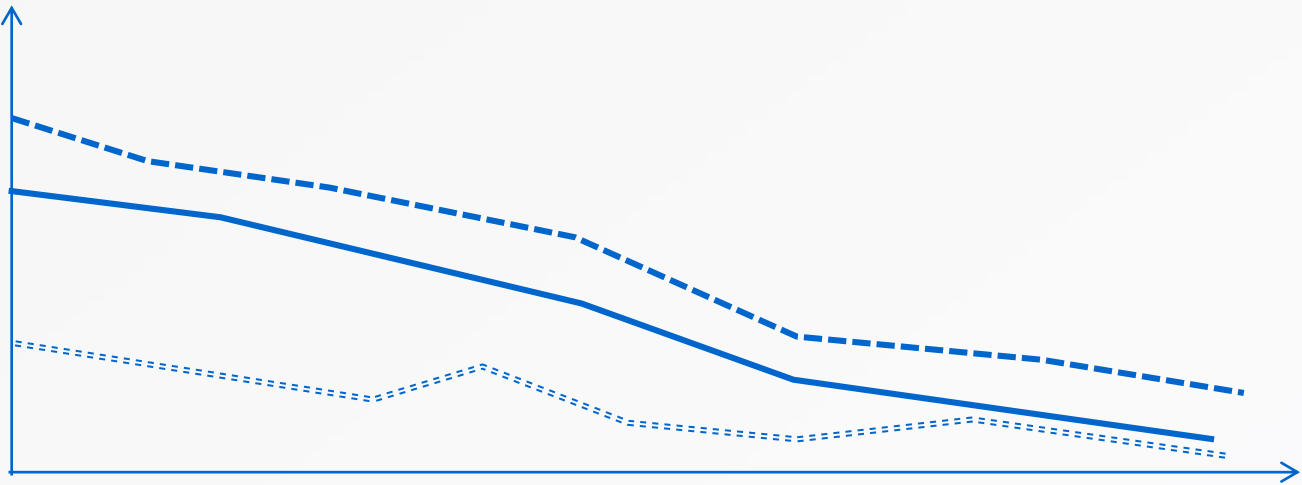
缺陷

克隆

...

折叠

趋势



(无变化的不列出)

版本 时间 提交者 文件位置

版本跟踪区

可整页显示

```
7- @Override
8- public Object sendScanMessage(JSONObject requestParam) {
9-     JSONObject result=new JSONObject();
10-    if(requestParam.getString("projectId")==null||requestParam.getString("c
11-        result.put("msg","请求参数不全!");
12-    return result;
13-
14-    String projectId=requestParam.getString("projectId");
15-    Project project=projectDao.getProjectByProjectId(projectId);
16-    project.setScanStatus("Scanning");
17-    project.setLast_scan_time(DateUtil.formatedDate(new Date()));
18-    projectDao.updateProjectStatus(project);
19-    kafkaTemplate.send("Scan",requestParam.toJSONString());
20-    logger.info("send message to topic -> Scan " +requestParam.toJSONString
21-    result.put("msg","scanMessage send success!");
22-    return result;
23- }
```

可整页显示

id	类型	位置	引入日期	消除日期	标签及 动作
----	----	----	------	------	--------

说明：动作可以是“不再关注等各种标签；图标表示”

列表区

第X页 共XX页 跳转到

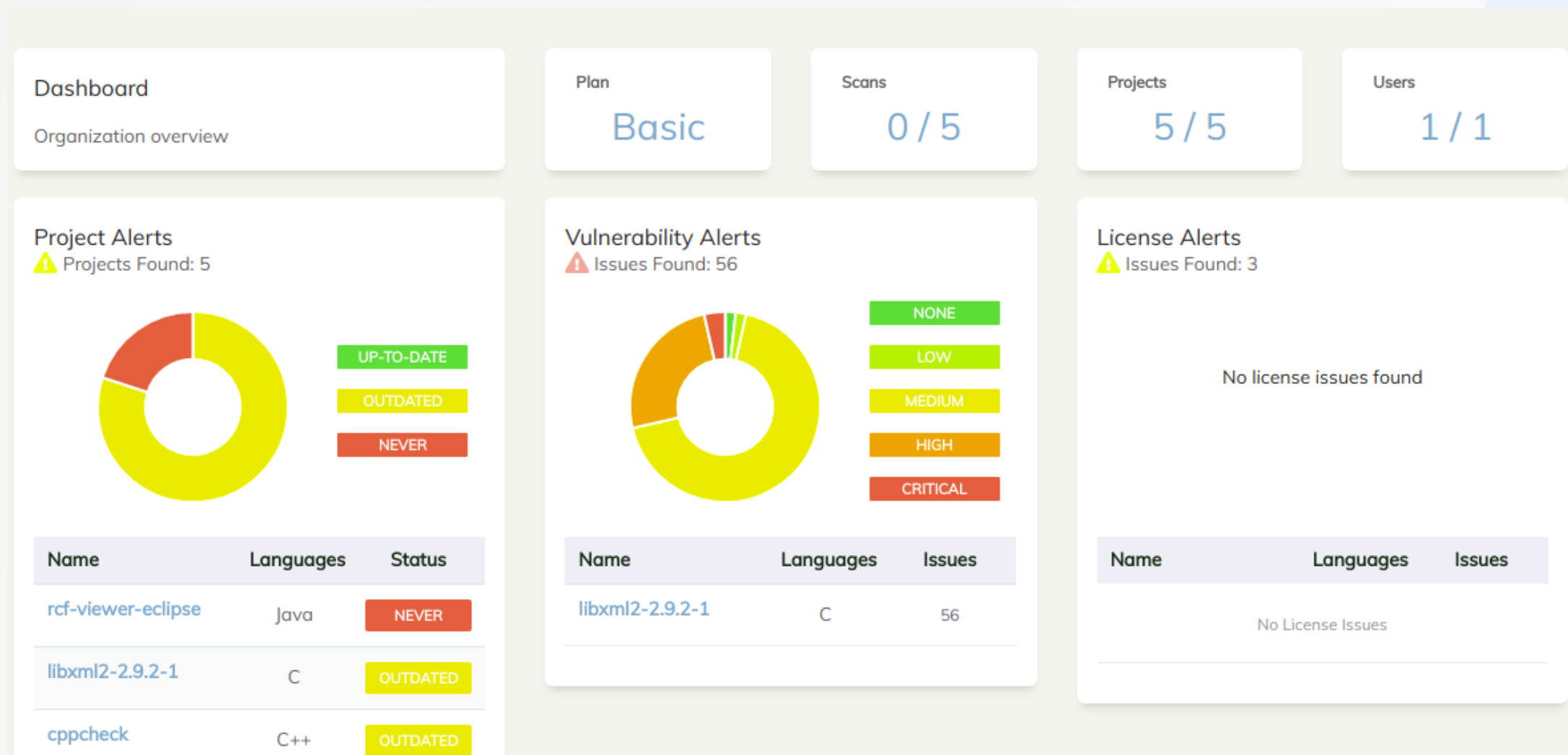
手动扫描

版本id	提交日期	提交者	扫描状态	选中
------	------	-----	------	----

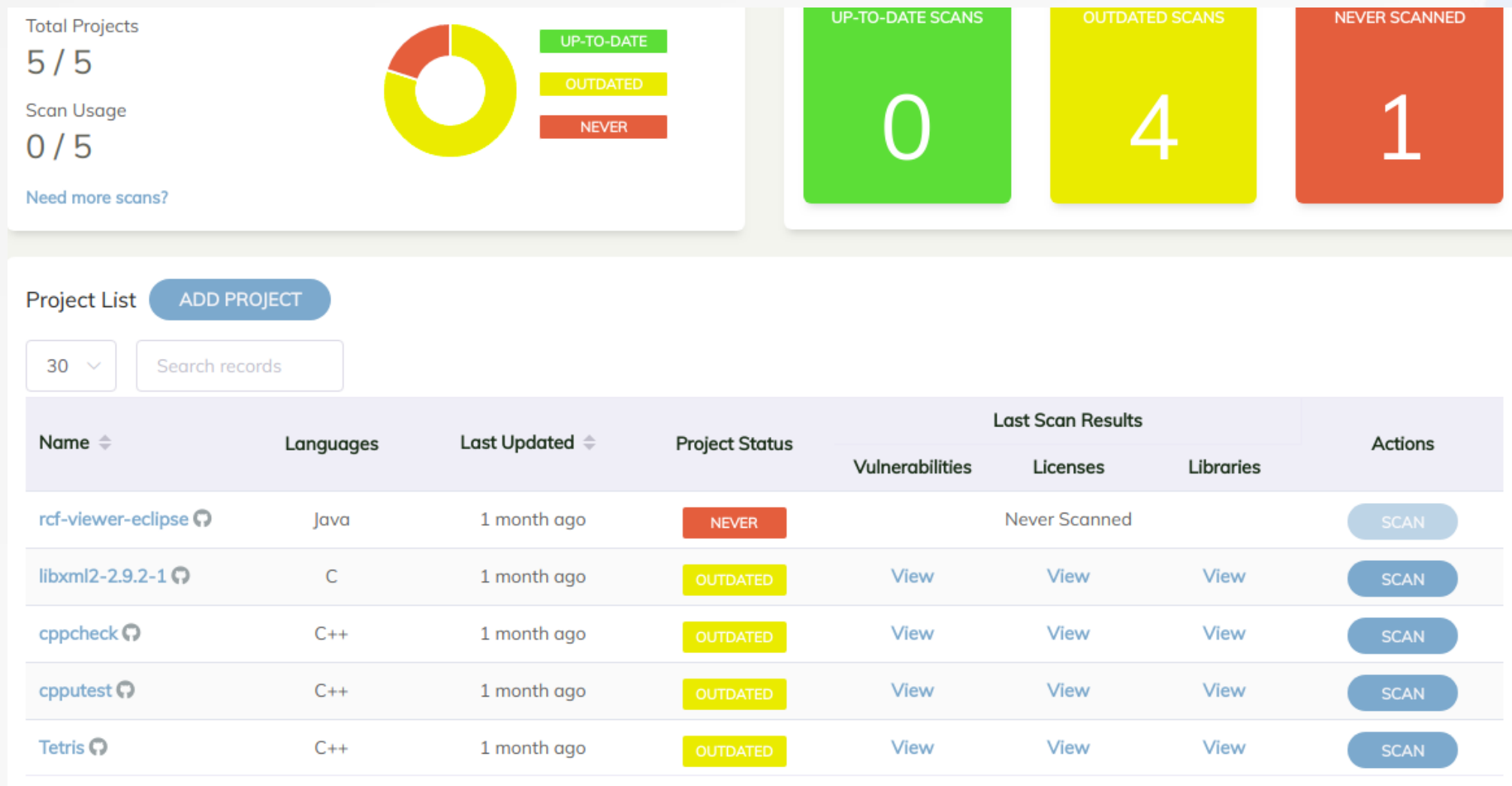
扫描

备用

仪表盘参考-其他方案



项目管理-其他方案



IssueTracker Core

The background features a gradient from dark blue at the top to light blue at the bottom. Overlaid on this are several wavy, horizontal bands of color: a thin light blue band, a thick yellow band, and a thin light blue band, all curving across the lower half of the image.

IssueTracker Open API

- `add_monitor_repo(repo, issue_type)`
- `remove_monitor_repo(repo, issue_type)`
- `retrieve_tracker_handle(repo, issue_type)`
- `get_issue_list(handle, offset, limit)`
- `retrieve_issue_raws(issue_id)`
- `retrieve_code(repo, commit, file)`
- `force_start_scan(repo, commit)`
- `remove_scan(repo, commit)`

Issue Tracker DB (和领域模型一致，略)

