MSR 2012 @ ICSE

目录

- Mining Software Repository 2012 @ ICSE
  - MSR(MicroSoft Research) talk @ MSR(Mining Software Repositories)
  - Towards Improving BTS with Game Mechanisms
  - GHTorrent
  - Topic Mining
  - SeCold

- The evolution of software
- Do Faster Releases Imporve Software Quality?
- Security vs Performance Bugs in Firefox
- 一些感想
    - 基于自然语义分析的commit分割
    - 关于这次发表中大家用的slides系统
    - 微软是个腹黑娘！

Mining Software Repository 2012 @ ICSE

参加了今年的MSR，会场在University of Zurich。一大早来到大学，注册有点 小插曲，显然瑞士人搞不清楚中国人的名字，3个杨（Yang）姓的中国人的名牌 被搞错了。然后堀田学长的所属被写作了“Japan, Japan”，成为了全日本的代表。

MSR(MicroSoft Research) talk @ MSR(Mining Software Repositories)

首先是来自微软亚洲研究院（MicroSoft Research @ Asia, MSR Asia）的Keynots， 于是就变成了MSR在MSR的演讲。MSR的张冬梅（Dongmei Zhang）女士的演讲 分为关于Software Analysis和XIAO的两部分。XIAO是MSRA开发的Code Clone Detector，似乎我要给井上研做的就是这个。想更多了解Xiao的细节，不过张女士 演讲结束的时候的鼓掌导致了话筒的小故障。

Towards Improving BTS with Game Mechanisms

感觉这篇的内容基本上就是关于

http://www.joelonsoftware.com/items/2008/09/15.html

这里写到的东西，然后说同样的理论是否可以用于Issue Tracking之类的事情上。 个人感觉这个意义不大，stackoverflow之所以成功是因为它把开源社区本身就 具有的名誉体系具现化了，本着大家都喜欢被别人奉为大牛的心态，就如同 wikipedia一样。同样的理论如果用于公司内部的Issue Tracking系统上，会得到 完全不同的东西吧。就像MSDN的组织方式虽然和wikipedia是一样的，但是在MSDN 里找信息的感觉和在wikipedia完全不一样。个人不太看好这个方向。

GHTorrent

这篇的slide在这里可以看到：http://www.slideshare.net/gousiosg/ghtorrent-githubs-data-from-a-firehose-13184524

Data exporter for github. Github的主要数据，代码，已经可以通过git接口 获得了，wiki是git的形式保存的。所以这个项目的目的就是暴露别的数据，主要 是issue tracking，code comments，这种。代码访问github api，然后用分布式 实现以克服api的限制，然后提供torrents形式的history下载。github api获得 的json数据以bson的形式保存在MongoDB里，解析过的有了Schema之后的数据保存 在MySQL里并可以导出SQL。

个人的想法，觉得数据如果能够更统一，全部存在Git里或许更好，像Wiki一样。 同样是要暴露全部历史记录的目的，用Torrent自己实现的历史远不如用Git的 接口实现的历史记录方便吧，git blame之类的也更方便追踪code comment之类的 作者信息。当然对git的raw date直接读写，需要对git的内部原理有足够的理解， 或许只有github的人有这种能力了。

Topic Mining

用得两个参数， DE 和 AIC，完全不能理解，过后研究。实验针对了Firefox, Mylyn, Eclipse三个软件。试图从Repo中分析源代码的identifier和comments， 找到topic和bug之间的关系，比如怎样的topic更容易导致bug。得出的结论似乎 也很暧昧，只是说核心功能被报告的bug更多，但是不知道原因。这只能表示核心 功能受到更多关注和更多测试吧，并不能说明核心功能就容易产生bug。

不过这个的Slide做得很漂亮，很容易理解。

SeCold

A linked data platform for mining software repositories

没听懂这个项目的目的。

The evolution of software

第二天的Keynotes，关于将Social Media和Software Development相结合的想法。 或许就是Github赖以成功的基础。讲到代码中的comment, Tags, uBlog, blog之类 的social的特性和IDE的融合的趋势。

Do Faster Releases Imporve Software Quality?

使用Firefox作为例子。

结论是快速发布导致bug更多，更容易crash，但是bug更快得到修复，并且用户 更快转向新的发布。

Security vs Performance Bugs in Firefox

Performance bugs are regression, blocks release.

一些感想

基于自然语义分析的commit分割

经常工具（比如git）的使用者并没有按照工具设计者的意图使用工具，这给MSR 带来很多困难。举个例子，git有非常完美的branch系统，通常期望git的使用者

能够在一次commit里commit一个功能，比如一个bug的修复，或者一个feature的 添加，但是事实上经常有很多逻辑上的commit被合并在一个里面了。

或许这不是使用者的错，而是工具仍然不够人性的表现。或许我们可以自动把 一次的commit按照语义分割成多个。

分割之后，可以更容易地把issue和commit关联，也更容易组织更多的研究。

关于这次发表中大家用的slides系统

题目为``Incorporating Version Histories in Information Retrieval Based Bug Localization''的人用的slide是beamer的。公式很多，overlay很多，列表 很多，图片很少，典型的beamer做出的slide。思维导图用得很不错。今天一天 有至少3个slide是用beamer做的。

题目为``Towards Improving Bug Tracking Systems with Game Mechanisms'' 的人用了prezi，图片很多，过度很多。但是比如没有页号没有页眉页脚，正式 会议的场合不太方便。

至少有六个以上用了Apple Keynotes，Keynotes做出来的东西真的和Powerpoint 做出来的很难区别，其中两个人用了初始的主题所以才看出来。

剩下的自然是PPT。MSRA的张女士做的虽然是PPT，倒是有很多beamer的感觉， 比如页眉页脚和overlay的用法。这些如果都是PPT做出来的，会多很多额外的 人力吧。

值得一提的是有一个题目为``Green Mining: A Methodology of Relating Software Change to Power Consumption''的人的slide全是``劣质''的手绘漫画，效果意外地好，很低碳很环保很绿色很可爱。具体效果可以参考下面的动画，虽然 现场看到的不是一个版本：

http://softwareprocess.es/a/greenmining-presentatation-at-queens-20120522.ogv

微软是个腹黑娘！

嘛虽然这也不是什么新闻了。MSR2012的Mining Challenge的赞助商是微软，管理 组织者来自微软研究院，奖品是Xbox和Kinect。然后今年的题目是：

```
Mining Android Bug
```

我看到了微软满满的怨气……