

# MSR 2012 @ ICSE

---

## 目錄

---

### Contents

- 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 Improve Software Quality?
- Security vs Performance Bugs in Firefox
- 一些感想
  - 基於自然語義分析的commit分割
  - 關於這次發表中大家用的slides系統
  - 微軟是個腹黑娘！

# Mining Software Repository 2012 @ ICSE

參加了今年的MSR，會場在University of Zurich。一大早來到大學，註冊有點小插曲，顯然瑞士人搞不清楚中國人的名字，3個楊（Yang）姓的中國人的名牌被搞錯了。然後堀田學長的所屬被寫作了“Japan, Japan”，成為了全日本的代表。

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

首先是來自微軟亞洲研究院（MicroSoft Research @ Asia, MSR Asia）的Keynotes，於是就變成了MSR在MSR的演講。MSR的張冬梅（Dongmei Zhang）女士的演講分爲關於Software Analysis和XIAO的兩部分。XIAO是MSRA開發的Code Clone Detector，似乎我要給井上研做的就是這個。想更多瞭解Xiao的細節，不過張女士演講結束的時候的鼓掌導致了話筒的小故障。

## Towards Improving BTS with Game Mechanisms

感覺這篇的內容基本上就是關於

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

這裏寫到的東西，然後說同樣的理論是否可以用於Issue Tracking之類的事情上。個人感覺這個意義不大，stackoverflow之所以成功是因爲它把開源社區本身就具有的名譽體系具現化了，本着大家都喜歡被別人奉爲大牛的心態，就如同wikipedia一樣。同樣的理論如果用於公司內部的Issue Tracking系統上，會得到完全不

同的東西吧。就像MSDN的組織方式雖然和wikipedia是一樣的，但是在MSDN 裏找信息的感覺和在wikipedia完全不一樣。個人不太看好這個方向。

## GHTorrent

---

這篇的slide在這裏可以看

到：<http://www.slideshare.net/gousiosg/ghtorrent-githubs-data-from-a-firehose-13184524>

---

Data exporter for github. Github的主要數據，代碼，已經可以通過git接口獲得了，wiki是git的形式保存的。所以這個項目的目的就是暴露別的數據，主要是issue tracking, code comments, 這種。代碼訪問github api, 然後用分佈式實現以克服api的限制，然後提供torrents形式的history下載。github api獲得的json數據以bson的形式保存在MongoDB裏，解析過的有了Schema之後的數據保存在MySQL裏並可以導出SQL。

個人的想法，覺得數據如果能夠更統一，全部存在Git裏或許更好，像Wiki一樣。同樣是要暴露全部歷史記錄的目的，用Torrent自己實現的歷史遠不如用Git的接口實現的歷史記錄方便吧，git blame之類的也更方便追蹤code comment之類的作者信息。當然對git的raw data直接讀寫，需要對git的內部原理有足夠的理解，或許只有github的人有這種能力了。

# Topic Mining

---

用得兩個參數，DE 和 AIC，完全不能理解，過後研究。實驗針對了Firefox, Mylyn, Eclipse三個軟件。試圖從Repo中分析源代碼的identifier和comments，找到topic和bug之間的關係，比如怎樣的topic更容易導致bug。得出的結論似乎也很曖昧，只是說核心功能被報告的bug更多，但是不知道原因。這只能表示核心功能受到更多關注和更多測試吧，並不能說明核心功能就容易產生bug。

不過這個的Slide做得很漂亮，很容易理解。

## SeCold

---

A linked data platform for mining software repositories

沒聽懂這個項目的目的。

## The evolution of software

---

第二天的Keynotes，關於將Social Media和Software Development相結合的想法。或許就是Github賴以成功的基礎。講到代碼中的comment, Tags,

uBlog, blog之類 的social的特性和IDE的融合的趨勢。

## Do Faster Releases Improve Software Quality?

使用Firefox作為例子。

結論是快速發佈導致bug更多，更容易crash，但是bug更快得到修復，並且用戶 更快轉向新的發佈。

## Security vs Performance Bugs in Firefox

Performance bugs are regression, blocks release.

---

## 一些感想

## 基於自然語義分析的commit分割

經常工具（比如git）的使用者並沒有按照工具設計者的意圖使用工具，這給MSR帶來很多困難。舉個例子，git有非常完美的branch系統，通常期望git的使用者能夠在一次commit裏commit一個功能，比如一個bug的修復，或者一個feature的添加，但是事實上經常有很多邏輯上的commit被合併在一個裏面了。

或許這不是使用者的錯，而是工具仍然不夠人性的表現。或許我們可以自動把一次的commit按照語義分割成多個。

分割之後，可以更容易地把issue和commit關聯，也更容易組織更多的研究。

## 關於這次發表中大家用的slides系統

題目爲 ``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，圖片很少，典型的beamer做出的slide。思維導圖用得不錯。今天一天有至少3個slide是用beamer做的。

題目爲 ``Towards Improving Bug Tracking Systems with Game Mechanisms" 的人用了prezi，圖片很多，過度很多。但是比如沒有頁號沒有頁眉頁腳，正式會議的場合不太方便。

至少有六個以上用了Apple Keynotes，Keynotes做出來的東西真的和Powerpoint做出來的很難區別，其中兩個人用了初始的主題所以才看出來。

剩下的自然是PPT。MSRA的張女士做的雖然是PPT，倒是有很多beamer的感覺，比如頁眉頁腳和overlay的用法。這些如果都是PPT做出來的，會多很多額外的人力吧。

值得一提的是有一個題目爲` ` Green Mining: A Methodology of Relating Software Change to Power Consumption"的人的slide全是` `劣質"的手繪漫畫，效果意外地好，很低碳很環保很綠色很可愛。具體效果可以參考下面的動畫，雖然現場看到的不是一個版本：

<http://softwareprocess.es/a/greenmining-presentation-at-queens-20120522.ogv>

## 微軟是個腹黑娘！

嘛雖然這也不是什麼新聞了。MSR2012的Mining Challenge的贊助商是微軟，管理組織者來自微軟研究院，獎品是Xbox和Kinect。然後今年的題目是：

Mining Android Bug

我看到了微軟滿滿的怨氣……



