

On Reducing the Pre-release Failures of Web Plug-in on Social Networking Site

Xingliang Yu, Jing Li, Hua Zhong



Institute of Software
Chinese Academy of Science

Outline



- I. Background
- II. Related Works
- III. Our Solution
- IV. Experiment
- V. Analysis

Background (1/6)



- Defect
 - an error in the source code[1]
- Failure
 - an observable error in the program behavior
- Pre-release failures
 - generally occur before a software release, in the course of testing [2]
- Post-release failures
 - found by users after the release

□ End User Development/Programming (EUD/EUP)

- a set of activities or techniques that allow people, who are non-professional developers, at some point to create or modify a software artifact—Lee D., Mehandjiev N.

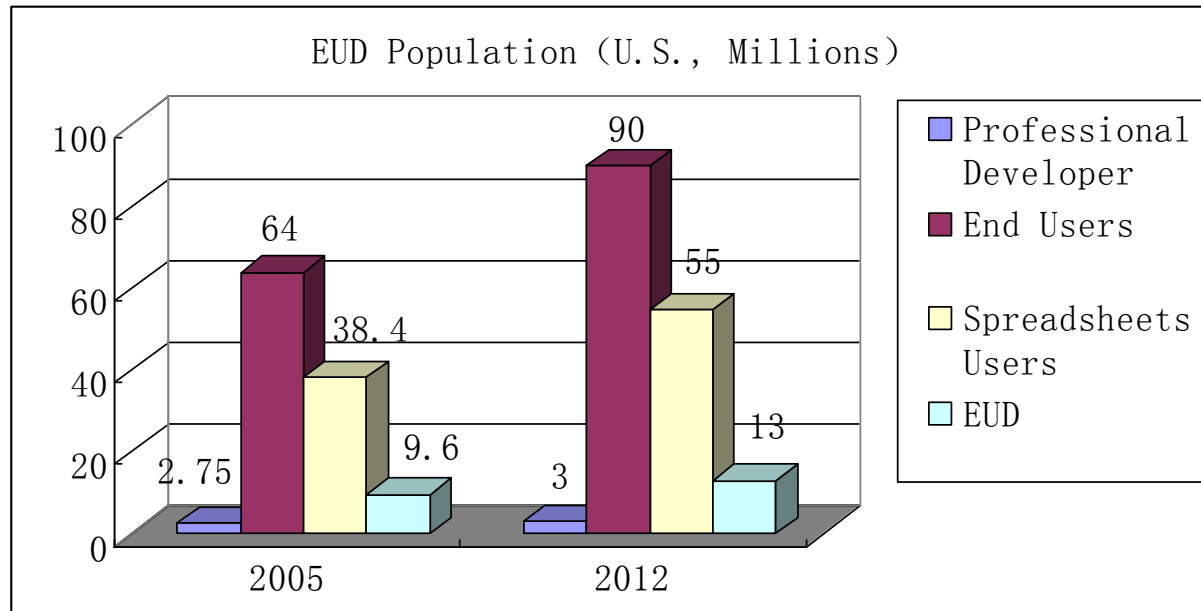
□ Typical EUD Activities

- MS Excel formulae
- MS Word Macro
- HTML Page Authoring
- ERP Customization
- Using of Intelligent Appliance
- Scientific Computation
-

Background (3/6)



□ The pervasiveness of EUD



□ Effectiveness problem in EUD

- 96% spreadsheet contain errors [16]
- EUD error may result in sever impact [17]

□ Social Networking Site (SNS)

■ Facebook

- Growth with participation expanding at rates 20% a month[16]
- Encourage end users to develop web plug-ins and Shares these plug-ins across the whole site [6]

■ Xiaonei.com

- Friend Trade: a web plug-in developed by five college students, which has been installed by more than three millions of Chinese users in 42 days[7]

Background (5/6)



- The difficulty of research on EUD effectiveness
 - Size of programming task is small
 - Distribution of End User is disperse
 - Knowledge needed for communication is scars
- EUD of Web Plug-ins on SNS
 - No professional background knowledge
 - No strict discipline
 - No standard process
 - Developed by one, used by many
- A special type of EUD
 - A large number of users in a single group
 - Mainly young and strongly interested in programming
- Pre-release failure cause severe problem

□ In summary

- Pre-release failure is a key problem of Social Networking Site
- Pre-release failure is also a important case of End User Development (EUD) Effectiveness
 - We introduced a technique, called Release Waiting Farm, borrowed from professional software engineering, and got some significant results.
- EUD is now far from mature, especially there are little research carried on about EUD processes.
 - We hope to attract more researchers to pay attention to this problem.

Outline



- I. Background
- II. **Related Works**
- III. Our Solution
- IV. Experiment
- V. Analysis

Related Research (1/2)



- On SNS stability & dependability
 - Facebook: outlines a list of prohibited plug-in categories and seek for law protection [6].
 - John: ISN technique that only allows a user's friends to view his privacy information [8].
 - Leon: a complex access-matrix, as well as a long list of checking rules [9].
 - Ohlsson et al: CVS-based source code verification, and automatic method [10].
 - Ostrand et al: a formal quality assurance methodology customized for end user development [11].
- All fail to avoid the pre-release failures of plug-ins, or mix pre-release failures with post-release ones.

Related Research (1/2)



- On reducing pre-release failure
 - Chidamber and Kemerer: eight metrics to cover the properties of object-oriented programs[11]
 - Mockus: used them in an industrial software development process to explore what driving software quality[8]
 - Nachiappan: validate these metrics, and found that these metrics appeared to be useful for predicting defect density[3]
- Problem
 - All these metrics and methods assume that formal test phase is the key step to discover pre-release failures, and can't be as casual as the one in SNS.

Outline

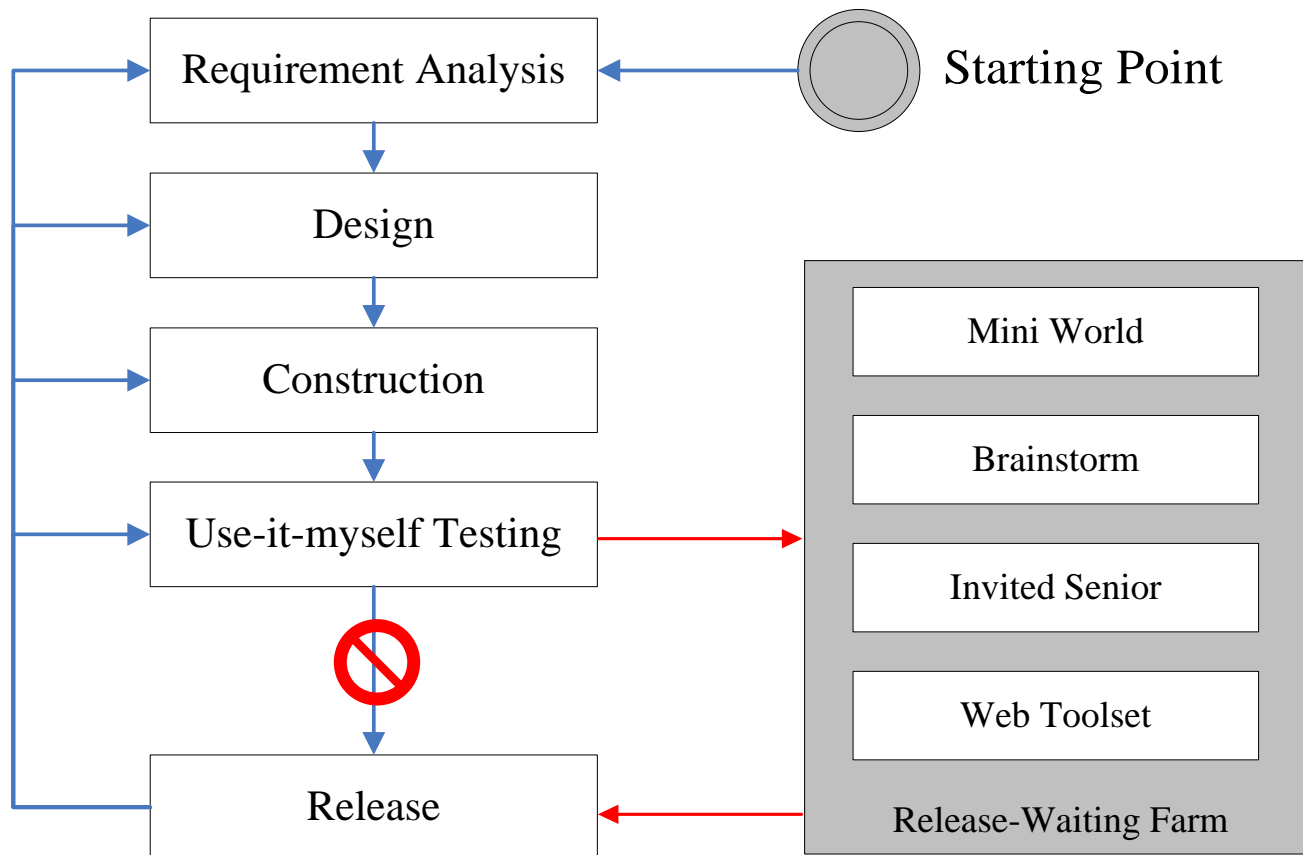


- I. Background
- II. Related Works
- III. **Our Solution**
- IV. Experiment
- V. Analysis

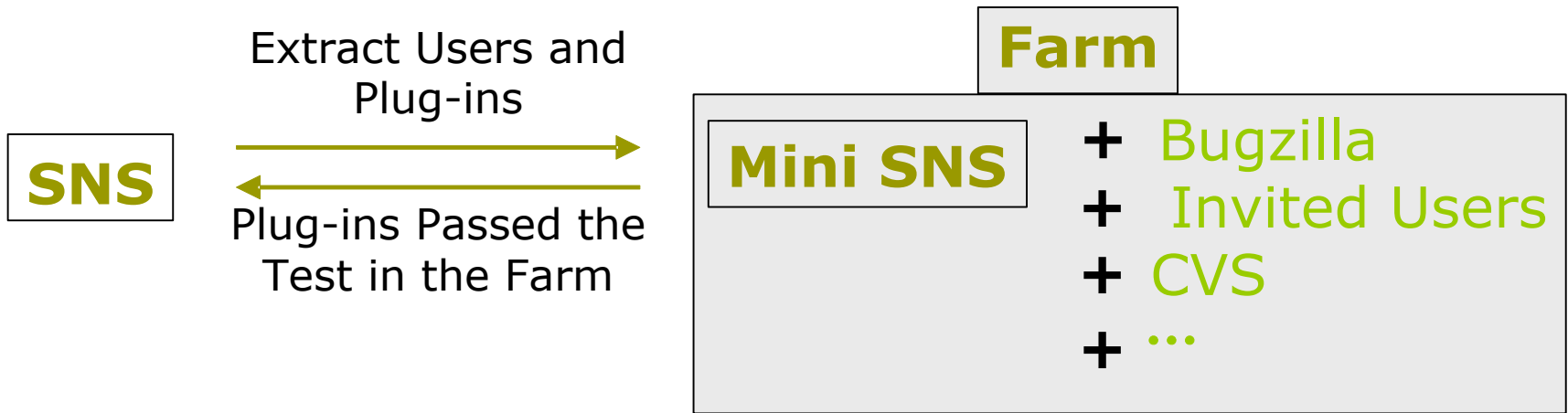
Our Solution (1/2)



□ Release Waiting Farm



Our Solution (2/2)



Outline



- I. Background
- II. Related Works
- III. Our Solution
- IV. **Experiment**
- V. Analysis

Experiment (1/4)



- Experiment
 - Open source SNS on exp.gucas.ac.cn (EXP for short)
 - Feb. 15, 2008 - Jun. 12, 2008
 - provides JSP develop environment
 - Java API
 - URL callback mechanism
 - Apache Tomcat 5.5
- A snapshot of EXP APIs [7].

Class	Function	Description
auth	startAppSession	start an app after system checking
	appLogin	only register app can login
user	getInfo	get open info of a user with his ID
	isAppInstalled	check if the user install this app
friend	getAllFriendID	get all friends' ID of a user
	getOnlineFriends	get all online friends' ID of a user
	areFriends	are two users friends?
	getAppUsers	all users who install this app
invite	addOutsite	invite outside friends to SNS
	addBonus	give bonus to a user for invitation
notify	sendMsg	send notification to a user

Experiment (2/4)



□ Registered users

Type	Sub-type	No.	Senior
CS-related major student	Master	210	12
	PhD	38	2
	Post-Doc	9	0
other major student	Master	80	8
	PhD	23	2
	Post-Doc	3	0
campus staff	Age 23-26	70	5
	Age 27-30	45	1
	Age 30-	23	1
outside campus friend	Age 23-26	18	3
	Age 27-30	22	1
	Age 30+	6	1
All		547	36

Experiment (3/4)



□ Top 10 EXP plug-ins

Rank	Plug-In	Description
1	Buy-Sell Friends	Buy a friend, and punish him
2	Pick Music to Friend	Pick a music for you
3	Vote	You like football, tennis, or...
4	Footmark	I have been to Shanghai...
5	Name-Matching	Our name match?
6	Touch You	Give you a kiss, embrace...
7	Give You a Gift	Give my friend a gift
8	Movie Review	I have watched this movie...
9	Friend Impression	Bill is a strong handsome boy
10	Daily Account	I pay 10\$ for breakfast...

Experiment (4/4)



□ Snapshot of EXP AS rwf.exp.gucas.ac.cn

```
#1 //stop receiving new pre-release plug-ins
#2 stop_receive_newcomer();
#3 //store existing plug-ins and bug info into database
#4 store_beta_plugin();
#5
#6 // top 10% active users
#7 // users who install top 50% plug-ins
#8 users = top10pActiveUsers() && usersInstall50pApp()
#9 loadInfo(users);
#10 loadPlugin(users);
#11
#12 //since not all users will be in farm,
#13 //delete bad connections from to other users
#14 DelBadConnection(users);
#15
#16 load_beta_plugin();
#17 start_receive_newcomer();
```

Outline



- I. Background
- II. Related Works
- III. Our Solution
- IV. Experiment
- V. **Analysis**

Analysis (1/6)



□ The failure analysis of top 10 plug-ins

Plug-In	Found in farm	Found in main server	Not found in main Server
Buy-Sell Friends	122	120	2
Pick Music to Friend	104	100	4
Vote	133	123	10
Footmark	67	67	0
Name-Matching	45	45	0
Touch You	44	44	0
Give You a Gift	59	56	3
Movie Review	65	65	0
Friend Impression	77	76	1
Daily Account	56	56	0
All	772	752	20

Analysis (2/6)



□ Bug status of top 10 plug-ins

Plug-in	Brainstorm		Invited Senior	
	Open	EF	Open	EF
Buy-Sell Friends	190	50	97	72
Pick Music to Friend	145	55	99	49
Vote	233	65	131	67
Footmark	80	17	69	50
Name-Matching	90	35	41	20
Touch You	66	18	37	26
Give You a Gift	98	35	31	24
Movie Review	79	34	30	21
Friend Impression	104	22	87	55
Daily Account	120	30	29	26
All	1205	308	651	410

Analysis (3/6)



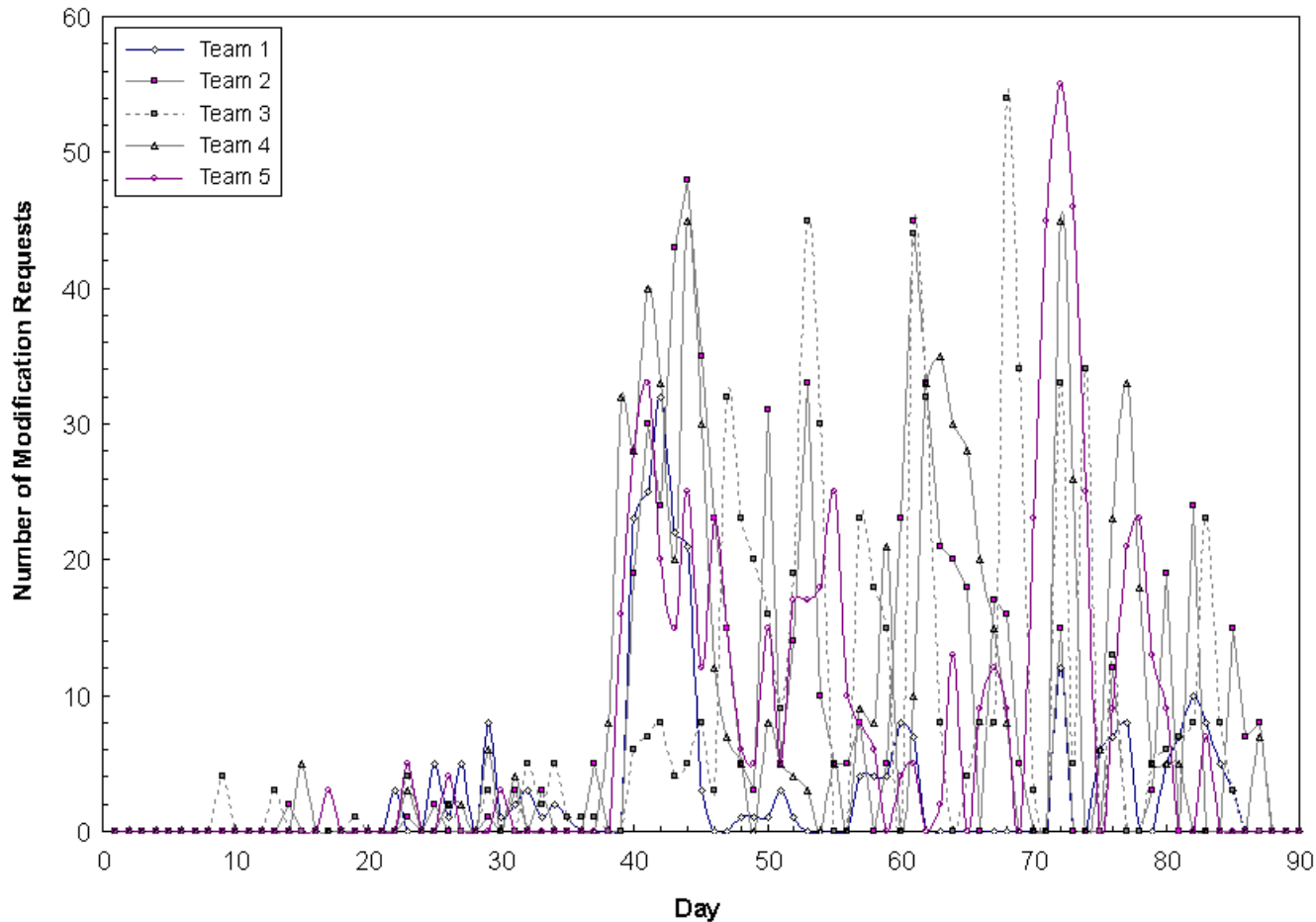
□ Comparison of main server and the farm

Parameter	Main Server	Farm
CPU	Intel Core 2 E4600	Intel P4 2.4 GHz
Memory	2G, 800MHz	512M, 400MHz
Hard Disk	SCSI 72G	SCSI 8G
NIC	Intel 8492 1000Mbps	Intel 8391 100Mbps
Peak Bandwidth	58.9Mbps	22.3Mbps
Network Cost	\$150/m	\$30/m
Maintainer Needed	3	1

Analysis (4/6)



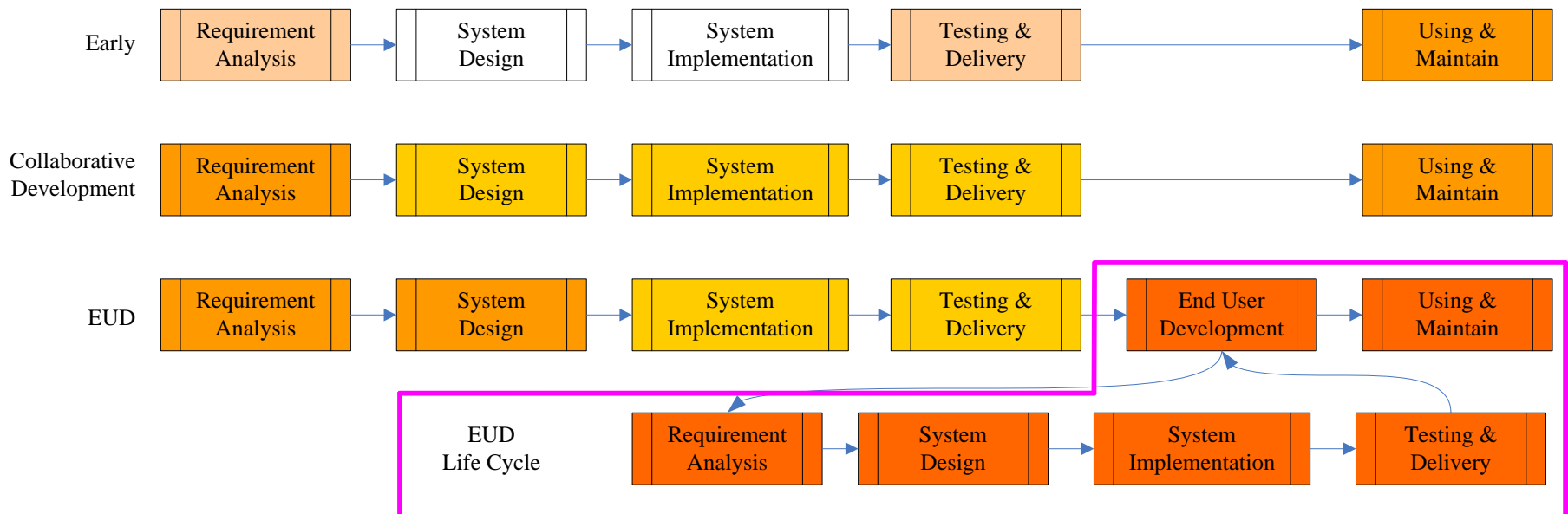
Modification request of teams



Analysis (5/6)



- Formalization of EUD process may mean the importance of EUD Process
- We assumed them as following and practiced them in another research project



Analysis (6/6)



- ❑ Identified the problem of effectiveness of EUD in SNS
- ❑ Release-waiting farm method
 - the farm is big enough to test plug-ins
 - brainstorm and invited senior methods work effectively
 - the farm costs low when compared with the main server, and it is worthy to build a farm
 - formalize the development process of end user, raise plug-in quality
- ❑ As a part of our research on EUD effectiveness, we used this result in our EUD systems
 - It's extended and verified in another paper, published in W23 of ICSE 2009, "Software Engineering for Computational Science and Engineering", at May 23.

References



- [1] Humphrey, W.S.: "The Personal Software Process". Technical Report, CMU/SEI-2000-TR-022, 2000
- [2] Luo Li, Jim Herbsleb, Mary Shaw, "Finding Predictors of Field Failure for Open Source Software Systems in Commonly Available Data Sources: A Case Study of OpenBSD", Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS'05), 2005, pp.32-52.
- [3] N Nagappan, T Ball, A Zeller, "Mining metrics to predict component failures", Proceedings of the 28th international conference on Software engineering, 2006, pp. 452-461.
- [4] R. Subramanyam and M. S. Krishnan. "Empirical analysis of ck metrics for object-oriented design complexity: Implications for software failure". IEEE Trans. Software Eng., 2003, 29(4), pp.297-310.
- [5] T. Zimmermann and P. WeiBgerber, "Preprocessing CVS data for fine-grained analysis", Proceedings of International Workshop on Mining Software Repositories, 2004, pp. 2-6.
- [6] Facebook Inc. <http://www.facebook.com>
- [7] Xiaonei Inc. <http://www.xiaonei.com>
- [8] A. Mockus, Zhang, P., Li, P., "Drivers for customer perceived software quality", Proceedings of International Conference on Software Engineering (ICSE), St. Louis, MO, pp. 225-233, 2005.
- [9] N. Nagappan, Ball, T., "Use of Relative Code Churn Measures to Predict System Defect Density", Proceedings of International Conference on Software Engineering (ICSE), St. Louis, MO, pp. 284-292, 2005.
- [10] N. Ohlsson, Alberg, H., "Predicting fault-prone software modules in telephone switches", IEEE Transactions in Software Engineering, 22(12), pp. 886 - 894, 1996.
- [11] T. Ostrand, Weyuker, E., Bell, R.M., "Predicting the location and number of faults in large software systems", IEEE Transactions in Software Engineering, 31(4), pp. 340 - 355, 2005.
- [12] J. Sliwerski, Zimmermann, T., Zeller, A., "When Do Changes Induce Fixes?", Proceedings of Mining Software Repositories (MSR) Workshop, 2005.
- [13] R. Subramanyam and M. S. Krishnan, "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", IEEE Transactions on Software Engineering, 29(4) pp. 297-310, April 2003.
- [14] X. Yu, J. Li, H. Zhong, "Release-waiting Farm: An Original Framework for Reducing the Pre-release Failures of Web Plug-in on Social Networking Site", Proceedings of CSIE 2009.
- [15] T. Zimmermann, Weigerber, P., Diehl, S., Zeller, A., "Mining Version Histories to Guide Software Changes", IEEE Transactions in Software Engineering, 31(6), pp. 429-445, 2005.
- [16] Raymond R. Panko, What We Know About Spreadsheet Errors, University of Hawai'i, College of Business Administration
- [17] Brad A. Myers, Margaret Burnett, End Users Creating Effective Software, CHI'2004 Special Interest Group, see <http://eecs.oregonstate.edu/EUSES/>

Thank You Very Much!

Questions?

Xingliang Yu, Jing Li, Hua Zhong



Software Institute
Chinese Academy of Science