

ICSEMでのトレンド

大平 雅雄

和歌山大学 システム工学部

自己紹介

- 出身：大阪府
- 学位：博士（工学）
- 学歴

- 1998.3 京都工芸繊維大学・電子情報工学科 卒業
- 2000.3 奈良先端科学技術大学院大学・博士前期課程 修了
- 2003.3 奈良先端科学技術大学院大学・博士後期課程 修了

- 職歴

- 2003.04-2003.9 科学技術振興事業団 さきがけ研究「協調と制御」領域 グループメンバ（ポスドク）
- 2003.10-2004.9 奈良先端科学技術大学院大学・EASEプロジェクト 産官学連携研究員（ポスドク）
- 2004.10-2012.3 奈良先端科学技術大学院大学・情報科学研究科 助手（2007年より助教）
- 2012.04-現在 和歌山大学・システム工学部・情報通信システム学科 講師（2014年より准教授）

- 研究分野 (Research Area)

- ソフトウェア工学 (Software Engineering)
- エンピリカル（実証的）アプローチ (Empirical Approach)
- リポジトリマイニング (Mining Software Repository)
- オープンソースソフトウェア (Open Source Software)



本日の内容

- ICSMEでのトレンド

- 概要と統計
- セッション解説
 - 基調講演
 - パネル
 - 研究発表
 - Research Track
 - Industry Track
 - Early Research Achievements Track

- 和歌山大学での取り組み

- ソフトウェア保守に関する研究の紹介
 - リポジトリマイニング
 - IPAソフトウェア工学分野の先導的研究支援事業
 - IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築

ICSME (INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE AND EVOLUTION) 2014

@Victoria, BC, Canada (10月1～3日)



ICSME (INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE AND EVOLUTION) 2014

- ソフトウェア保守を専門とするトップカンファレンス
 - 今年で30周年を迎えた老舗会議
 - 今年から名称変更：ICSMからICSMEへ
 - ソフトウェア保守における進化（Evolution）の側面を反映
 - 約300名の参加者
- 3種類のセッション構成
 - 基調講演
 - パネル討論
 - 研究発表
 - Research Track（フルペーパー）
 - Early Research Achievements Track（ショートペーパー）
 - Industry Track（企業論文）
 - Tools and Demonstrations（ツールとデモ）

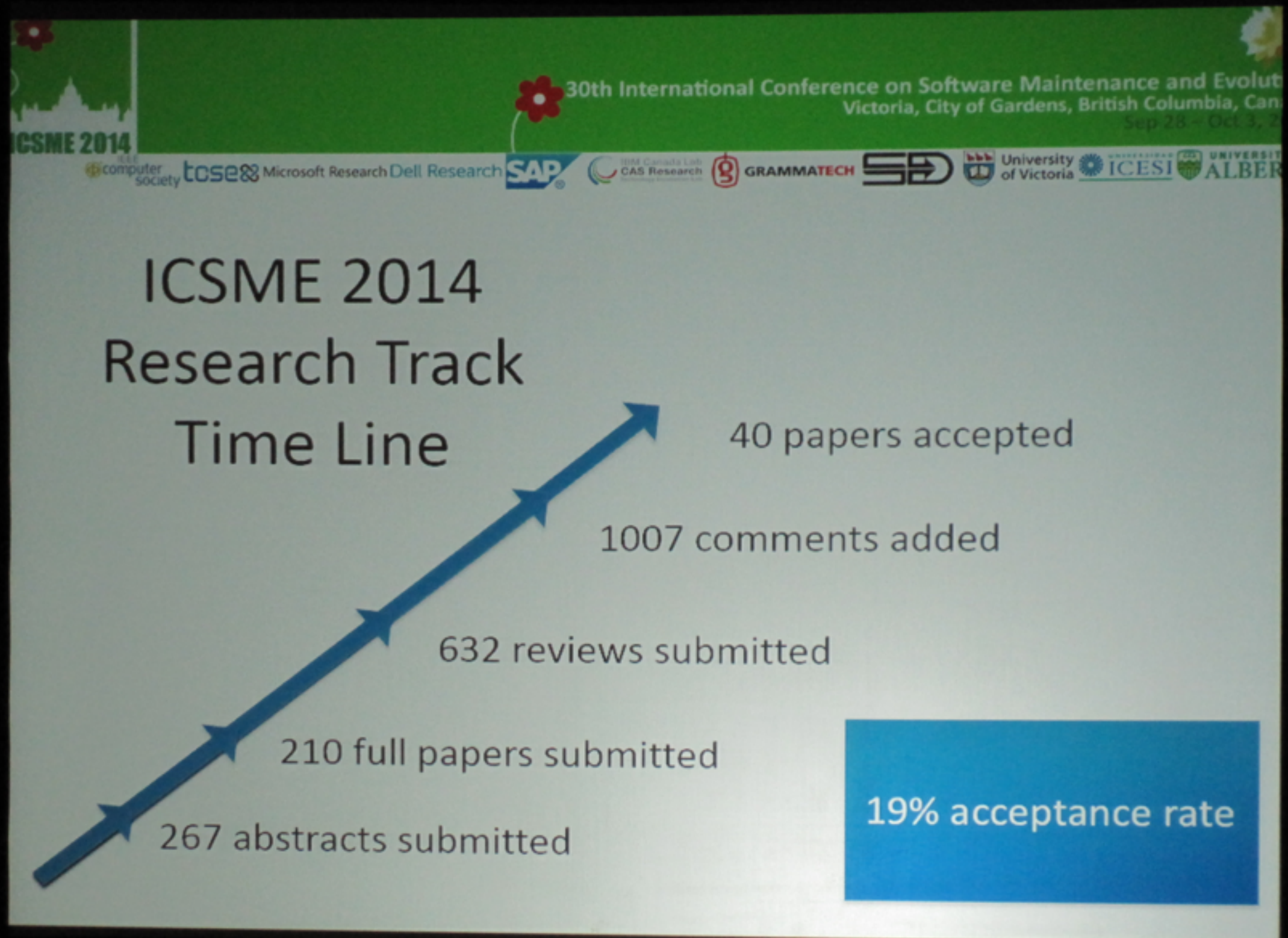
参加者内訳



フルペーパー：投稿数／投稿者数



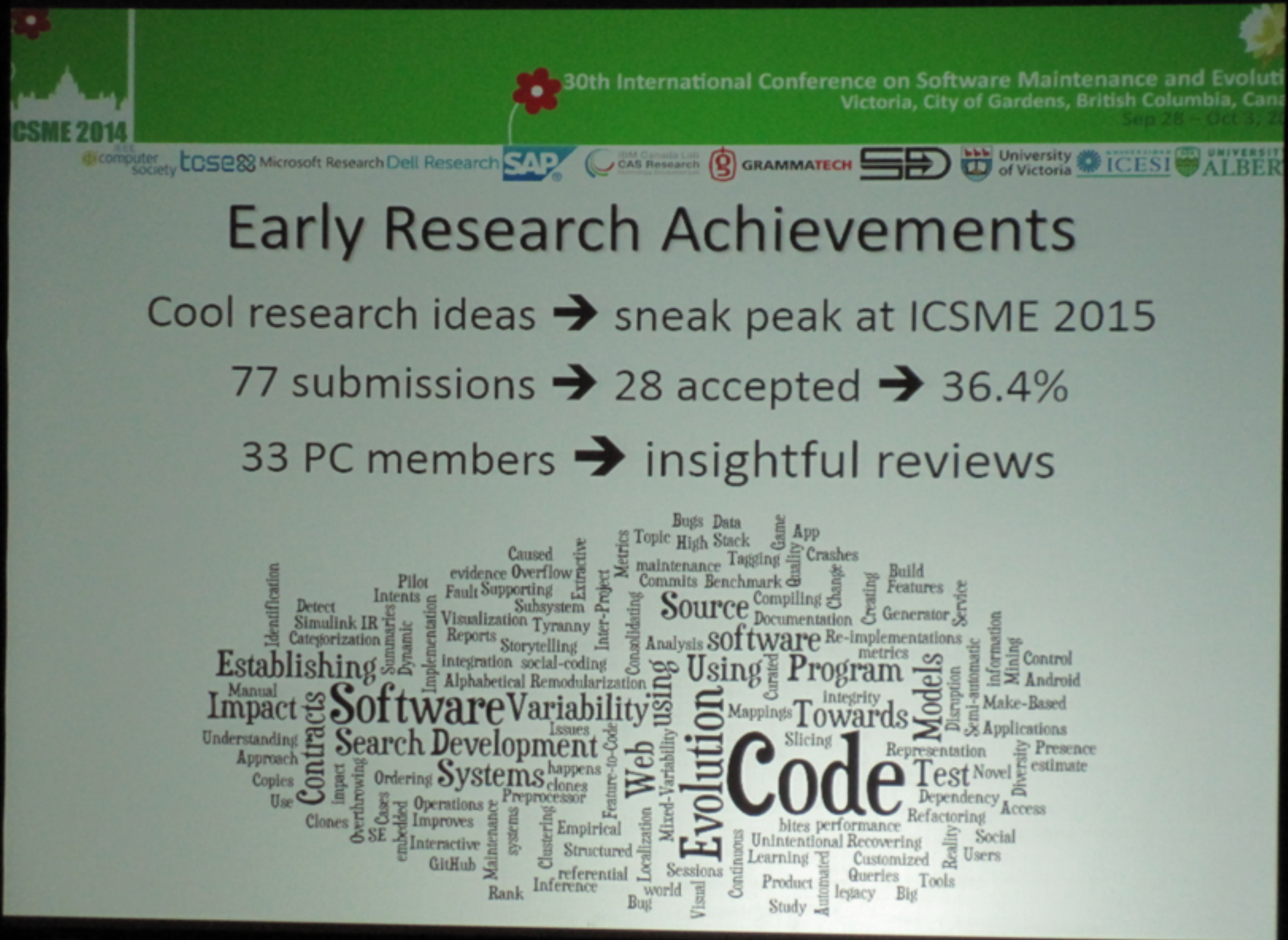
フルペーパー：採択率



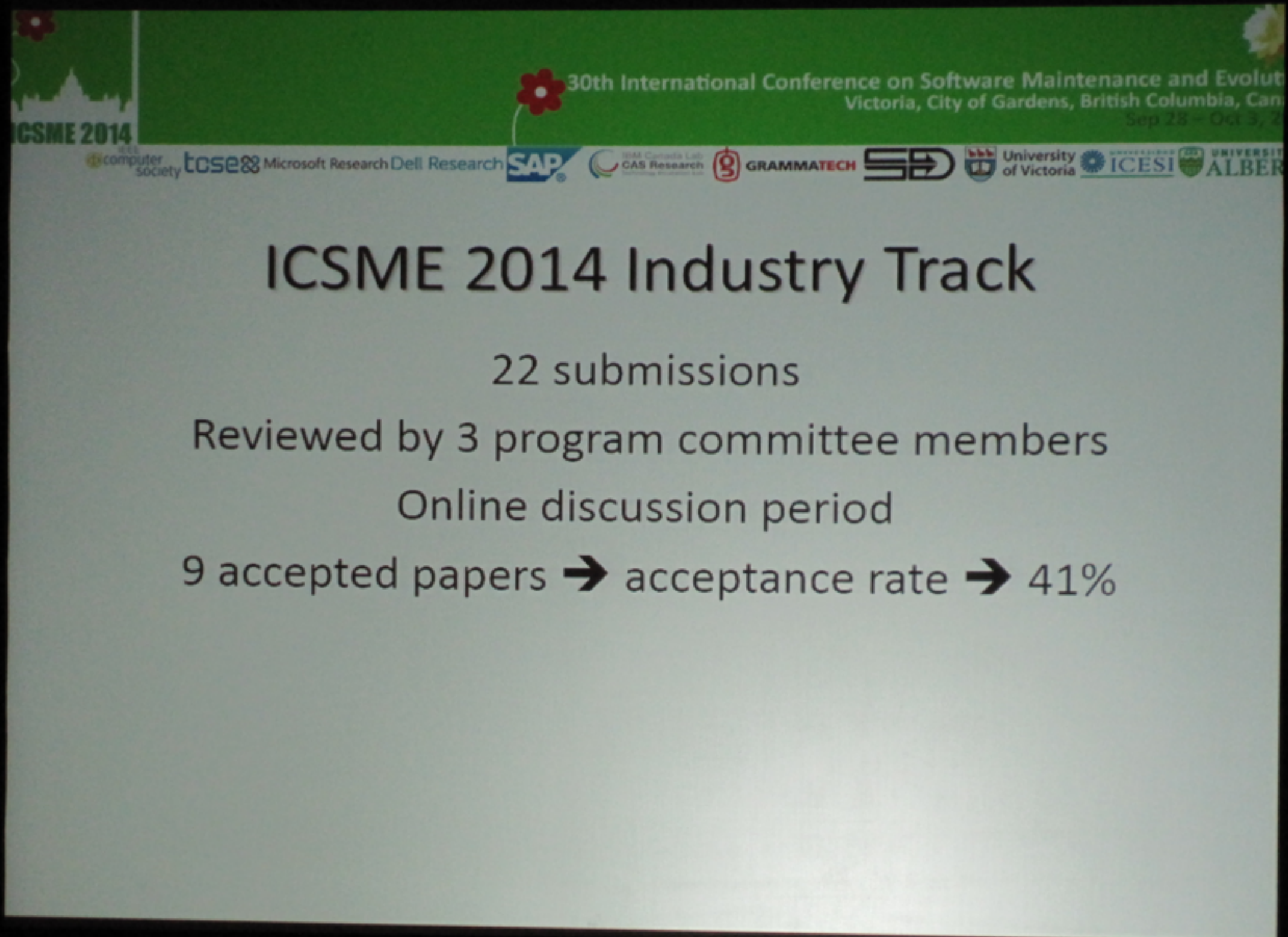
フルペーパー：国別採択論文数



ショートペーパー：採択率



企業論文：採択率



The poster for the ICSME 2014 Industry Track features a green header with a red flower icon. The text in the header includes 'ICSME 2014', '30th International Conference on Software Maintenance and Evolution', 'Victoria, City of Gardens, British Columbia, Canada', and 'Sep 28 - Oct 3, 2014'. Below the header is a row of logos for sponsors: IEEE Computer Society, tcase, Microsoft Research, Dell Research, SAP, CAS Research, GRAMMATECH, S&D, University of Victoria, ICESI, and University of Alberta. The main content area is white and contains the following text:

ICSME 2014 Industry Track

22 submissions

Reviewed by 3 program committee members

Online discussion period

9 accepted papers → acceptance rate → 41%

ツールとデモ：採択率



ICSME 2014 Tool Demo Track

27 submissions

Reviewed by 3 program committee members

14 were accepted → 52% acceptance rate

11:00 to 12:30 Wed/Thu/Fri

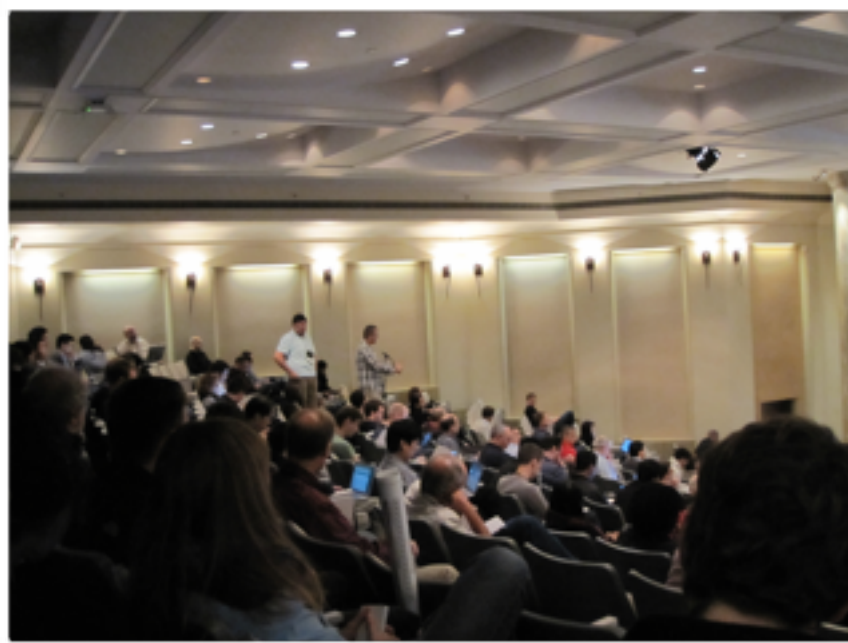
Saanich — Presentations

Foyer — Demos and posters

会場の様子



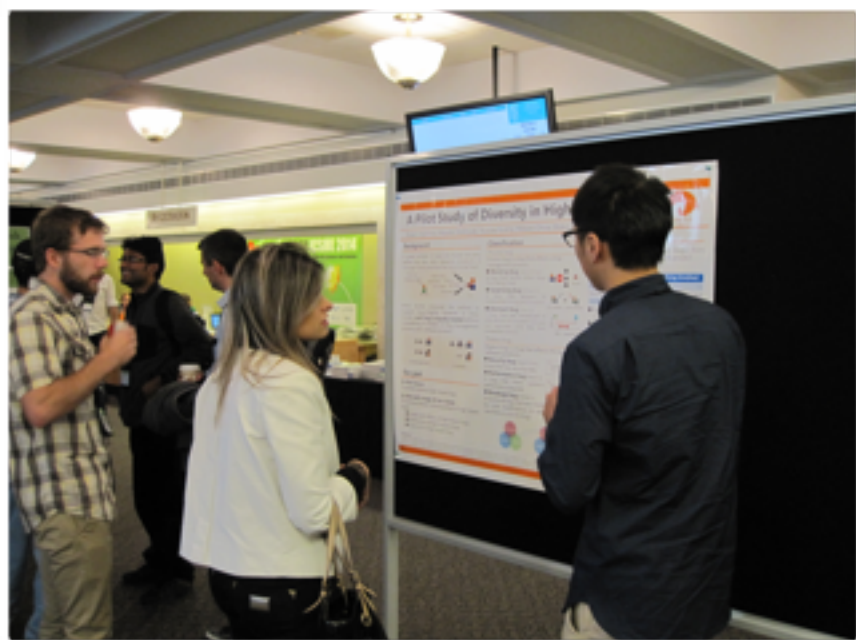
パネル@メインホール



質疑@メインホール



研究発表@サブホール



ポスター発表@コーヒースタンド



コーヒースタンド



バンケット

ICSME WEEK

	Saturday Sep 27	Sunday Sep 28	Monday Sep 29			Tuesday Sep 30				Wednesday Oct 1	Thursday Oct 2	Friday Oct 3
Rooms		Salon C	Salon C	Esquimalt	Oak Bay	Esquimalt	Saanich	Oak Bay 1	Oak Bay 2	Lecture Theater, Saanich, Esquimalt, Oak Bay		
07:45 09:00		Breakfast Salon C	Breakfast in Conference Centre Foyer									
07:00 09:00										ICSME Bike Tour 7:30-8:30	ICSME Bike Tour 7:30-8:30	
09:00 10:30		SCAM Keynote	SCAM	VISSOFT Keynote	MESOCA Keynote	VISSOFT	MTD	MUD Keynote	DOC	ICSME Keynotes Awards: 2014 MIP, Harlan Mills		
10:30 11:00		Nutrition Break in Conference Centre Foyer										
11:00 12:30		SCAM	SCAM	VISSOFT	MESOCA	VISSOFT	MTD	MUD	DOC	ICSME Research, ERA, Industry, Tool Demo Papers		
12:30 13:30	ICSME Hike Mt. Douglas Park 13:00-16:00	Lunch in Fairmont Empress Crystal Ballroom										
13:30 14:00		SCAM	SCAM	VISSOFT	MESOCA	VISSOFT	MTD	MUD	DOC	ICSME Technology Briefings Frontiers of Software Practice		
14:00 15:30		SCAM	SCAM	VISSOFT	MESOCA	VISSOFT	MTD	MUD	DOC	ICSME Research, ERA, Industry, Tool Demo Papers		
15:30 16:00		Nutrition Break in Conference Centre Foyer										
16:00 17:30		SCAM	SCAM	VISSOFT	MESOCA	VISSOFT	MTD	MUD	DOC	ICSME Research, ERA, Industry, Tool Demo Papers		
17:30 18:30										Soccer at Heywood Beacon Hill Park	Open ICSME Steering Committee Meeting	
19:00 21:30		SCAM Banquet Canoe Brewpub		VISSOFT Banquet Empress Palm Court			MTD Dinner !! Terrazzo			Starts at 19:30 ICSME Banquet Crystal Ballroom	ICSME Reception Royal BC Museum	

ICSME 2014: MAIN PROGRAM

午前

	Tuesday	Wednesday	Thursday	Friday	Location
Breakfast 07:45–09:00	Registration	Registration	Registration	Registration	Foyer
Plenary 09:00–09:30	Pre-Doctoral Symposium	Welcome Mix Hausi Muller	Harlan Mills Award Patrick & Radhia Cousot	2004 MIP Award Radu Marinescu	Lecture Theatre
Keynotes 09:30–10:30	By Invitation Only	Judith Bishop Microsoft Research	Gabby Silberman Dell	Arie van Deursen TU Delft Slides	Lecture Theatre
Break 10:30–11:00	ERA & DOC Posters	ERA & DOC Posters	ERA & DOC Posters		Foyer
Sessions 11:00–12:30	Pre-Doctoral Symposium - By Invitation Only	Research Developer Knowledge	Research Fault Localization 1	Research Recommenders	Oak Bay
		Research Evolving Systems	Research Empirical Studies	Research Patches	Esquimalt
		ERA Commits & Visualization	ERA Reengineering	ERA Maintenance	Lecture Theatre
		Tool Demo Software Analyzers	Tool Demo Recommenders	Tool Demo Context & Environments	Saanich
Lunch 12:30–13:30	Middle Eastern Delights Buffet	Great Canadian Escape Buffet	Orient Express Buffet	Lunch to Go	Crystal Ballroom

午後

Technology Briefings 13:30–14:00	Post-Doctoral Symposium - Open to all ICSME Registrants	Wiggelsworth, IBM BlueMix and Openstack	Marin Litoiu York University Adaptive Cloud	McGeer SAP Labs The Lively Web	Lecture Theatre
		Hilton Conquer Mobile VB Medical Simulation	Maletic, Kent State srcML Metaprogramming	Collard, Akron srcML Metaprogramming	Oak Bay
		Sukow, Terapeak eBay Analytics	Anderson, GrammaTech CodeSonar	Tilley, FIT Web System Evolution	Esquimalt
Sessions 14:00–15:30		Research Developer Support	Research Software Quality	Research Refactoring	Lecture Theatre
		Research Technical Debt	Research Fault Localization 2	Research Clones	Oak Bay
		ERA Code & Traceability	ERA Clones & Applications	ERA Bugs & Issues	Saanich
	Industry Software Quality	Industry Cross-cutting Concerns	Research Managing Change 2	Esquimalt	
Break 15:30–16:00	ERA & DOC Posters	ERA & DOC Posters	ERA & DOC Posters		Foyer
Sessions 16:00–17:00	Post-Doctoral Symposium - Open to all ICSME Registrants	CSM Pioneers Panel Lecture Theatre	Technical Debt Panel	Closing & Future ICSMEs	Lecture Theatre
			Research Managing Change 1		Saanich
17:00 - 17:30			ICSME Open Steering Committee Meeting		Lecture Theatre
17:30 - 18:00		Soccer Game in Beacon Hill Park			
18:00 - 18:30					

基調講演

- Judith Bishop

Director of Computer Science
Microsoft Research, Redmond, USA

- Software and Hardware – can Abstraction be the Key to Evolution?



- Gabriel (Gabby) Silberman

Executive Director, Technology Strategy
Dell, USA

- The Power of Prediction: Understanding and Preparing for Disruptive Information Technologies



- Arie van Deursen

Professor
Delft University of Technology, The Netherlands

- On Software Changes, Large and Small



Maintenance and Evolution



Evolution

Curiosity-driven research

Help to *understand* the world:

- How do people manage to change their software?

Nurtured by challenges faced during software maintenance

Maintenance

Engineering research

Help to *change* the world

- How can we support people changing their software?

Nurtured by software evolution research results

パネル討論

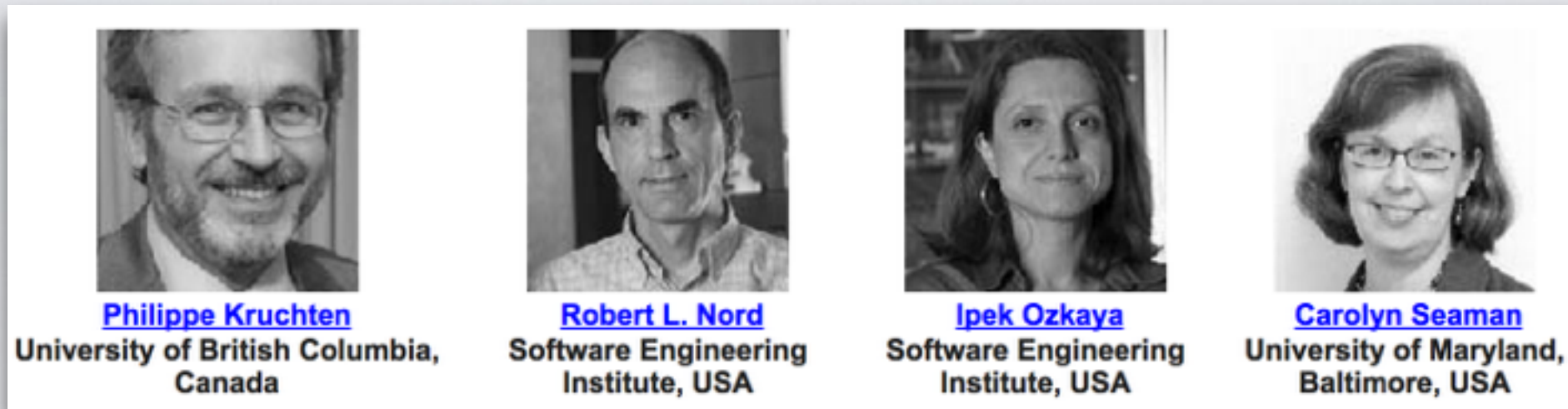
- Panel on CSM Pioneers



- Conference on Software Maintenance (CSM) First Decade
 - <http://www.icsme.org/program/pioneers/csm-first-decade>
- Audio Recording
 - <http://youtu.be/DrMoFKwbURs>

パネル討論

- Technical Debt: From Metaphor to Theory and Practice



- 技術的負債 [wikipedia]
 - ・ 行き当たりばったりなソフトウェアアーキテクチャと、余裕のないソフトウェア開発が引き起こす結果のことを指す新しい比喩である。「設計上の負債(design debt)」とも言う。
 - ・ 開発の中で先送りされるのは、文書化、テストコードの記述、ソースコード中の積み残し(TODO)項目の解決やコンパイラの警告、静的コード解析ツールの解析結果への対応などである。他にも、技術的負債の例として、組織で共有されない知識や、複雑すぎて変更が難しいコードなどがある。
 - ・ 関連領域：software aging and decay, software metrics, prediction and estimation, release planning, and architecture

研究発表

- 今年は特別（？） 全論文がダウンロード可能
 - <http://www.icsme.org/2014/proceedings>
- 全体の傾向
 - リポジトリマイニング（MSR）技術の開発, あるいは, MSR技術を用いた分析が主流
 - オープンソース開発の分析, あるいは, オープンソース開発データを用いた手法の評価が目立つ

研究発表：RESEARCH TRACK

- セッション概要

- ソフトウェア保守および進化に関して独創性が高く重要な成果を含む論文発表の場

- セッション数：14

- Developer Knowledge
- Evolving Systems
- Developer Support
- **Technical Debt**
- Empirical Studies
- **Fault Localization 1**
- **Fault Localization 2**
- Software Quality
- Patches
- Recommenders
- Clones
- Refactoring
- **Managing Change 1**
- **Managing Change 2**

研究発表：RESEARCH TRACK

- セッション概要

- ソフトウェア保守および進化に関して独創性が高く重要な成果を含む論文発表の場

- セッション数：14

- Developer Knowledge
- Evolving Systems
- Developer Support
- **Technical Debt**
- Empirical Studies
- **Fault Localization 1**
- **Fault Localization 2**
- Software Quality
- Patches
- Recommenders
- Clones
- Refactoring
- **Managing Change 1**
- **Managing Change 2**

セッション：TECHNICAL DEBT

- An Exploratory Study on Self-Admitted Technical Debt
 - 著者：Aniket Potdar and Emad Shihab.
 - 概要：開発者自ら認めている技術的負債についての分析
- Do they Really Smell Bad? A Study on Developers' Perception of Bad Code Smells
 - 著者：Fabio Palomba, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto and Andrea De Lucia.
 - 概要：code smellに関する開発者の認識の違いについての分析
- Smelling Faults in Spreadsheets
 - 著者：Pedro Martins, Rui Abreu, Alexandre Perez, Jácome Cunha, João Paulo Fernandes and João Saraiva.
 - 概要：表計算アプリケーション利用時のcode smell (fault)の検出手法

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- 著者：Aniket Potdar and Emad Shihab
- 概要
 - 技術的負債は開発者の悪意のないミスが原因とされてきた
 - 開発者が意図的に技術的負債を生みだすことはないのだろうか？
 - 自ら認めた(self admitted)技術的負債を調査した初の研究
- 目的：自己容認の技術的負債に関する以下のRQに答えること
 - RQ1: 自己容認の技術的負債はどの程度存在するのか？
 - RQ2: なぜ自己容認の技術的負債が混入するのか？
 - RQ2-1: 経験によるものか？（新人は混入しやすい？）
 - RQ2-2: 時間的プレッシャーによるものか？（リリース間際は混入しやすい？）
 - RQ2-3: 複雑さによるものなのか？（コードが複雑なほど混入しやすい？）
 - RQ3: 技術的負債が混入した後、どれくらい除去されるのか？

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- 調査方法：ソースコード内のコメントに自己容認の技術的負債に関する記述がないかどうかを調査
 - 対象プロジェクト：長期間存続する大規模プロジェクト 4 件
 - Eclipse, Chromium OS, ArgoUML, Apache httpd
 - 全ソースコードから合計101,762件のコメントを抽出し目視により確認
 - 自己容認の技術的負債の存在を示す62種類のキーワードを抽出
 - *hack*
 - *fixme*
 - *is problematic*
 - *this isn't very solid*
 - *probably a bug*
 - *hope everything will work*
 - *fix this crap*

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- RQ1: 自己容認の技術的負債はどの程度存在するのか？

Type	Project	Total	Self-Admitted Technical Debt	%
Files	Eclipse	6,389	152	2.4%
	Chromium OS	14,328	574	4.0%
	ArgoUML	1,846	77	4.2%
	Apache httpd	255	79	31.0%
Classes	Eclipse	11,993	151	1.3%
	Chromium OS	18,249	82	0.4%
	ArgoUML	2,304	76	3.3%
	Apache httpd	-	-	-%
Methods/ Functions	Eclipse	48,382	145	0.3%
	Chromium OS	153,514	626	0.4%
	ArgoUML	12,845	71	0.6%
	Apache httpd	4,361	112	2.6%
Constructors	Eclipse	5,561	151	0.2%
	Chromium OS	18,988	82	0.1%
	ArgoUML	1,920	76	0.5%
	Apache httpd	-	-	-%

2.4~31.0%のファイルに自己容認の技術的不在が存在する

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- RQ2: なぜ自己容認の技術的負債が混入するのか？
 - RQ2-1: 経験によるものか？（新人は混入しやすい？）
 - RQ2-2: 時間的プレッシャーによるものか？（リリース間際は混入しやすい？）
 - RQ2-3: 複雑さによるものなのか？（コードが複雑なほど混入しやすい？）

Project	Developer	Total Commits	Self-Admitted Technical Debt	Experience Rank	Total Committers
Eclipse	E1	1,231	21	8	221
	E2	2,513	20	2	
	E3	1,508	12	5	
	E4	4,436	10	1	
	E5	1,190	7	9	
Chromium OS	C1	4,853	28	1	1,784
	C2	410	24	219	
	C3	1,034	21	64	
	C4	1,930	21	26	
	C5	388	21	232	
ArgoUML	AU1	2,544	17	4	53
	AU2	4,879	7	1	
	AU3	3,088	6	3	
	AU4	3,192	4	2	
	AU5	480	3	8	
Apache	AH1	4,908	14	1	145
	AH2	3,214	13	3	
	AH3	721	11	26	
	AH4	1,219	11	13	
	AH5	1,239	10	11	

経験豊富な開発者の方が自己容認の技術的負債を混入させやすい

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- RQ2: なぜ自己容認の技術的負債が混入するのか？
 - RQ2-1: 経験によるものか？（新人は混入しやすい？）
 - **RQ2-2: 時間的プレッシャーによるものか？（リリース間際は混入しやすい？）**
 - RQ2-3: 複雑さによるものなのか？（コードが複雑なほど混入しやすい？）

Project	Time to Release	Self-Admitted Technical Debt	Total Changes	% Over Total Changes	% Self-Admitted Technical Debt
Eclipse	>= 6month	55	16,682	0.33%	41.7%
	>= 3month & <= 6month	48	7,452	0.64%	36.4%
	>= 1month & <= 3month	22	5,591	0.39%	16.7%
	<= 1month	7	1,777	0.39%	5.3%
ArgoUML	>= 6month	16	7,973	0.20%	33.3%
	>= 3month & <= 6month	20	5,917	0.34%	41.7%
	>= 1month & <= 3month	8	4,315	0.19%	16.7%
	<= 1month	4	1,708	0.23%	8.3%
Apache	>= 6month	51	23,579	0.22%	37.5%
	>= 3month & <= 6month	43	13,960	0.31%	31.6%
	>= 1month & <= 3month	28	11,670	0.24%	20.6%
	<= 1month	14	6,614	0.21%	10.3%

リリース 3 ヶ月以上前に混入されることが大半（リリース間際のプレッシャーによるものではない）

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- RQ2: なぜ自己容認の技術的負債が混入するのか？
 - RQ2-1: 経験によるものか？（新人は混入しやすい？）
 - RQ2-2: 時間的プレッシャーによるものか？（リリース間際は混入しやすい？）
 - **RQ2-3: 複雑さによるものなのか？（コードが複雑なほど混入しやすい？）**

Project	Cyclomatic Comp.	Fan-in	Fan-out
Eclipse	-0.1992618 **	0.3159485	-0.08568448
Chromium OS	0.1840126 **	-0.01470615 *	-0.2529673 *
ArgoUML	-0.03449873 *	-0.01459957 *	0.1686399
Apache	0.2386505 **	0.3333192 ***	0.1108314 *

(p < 0.01 ***; p < 0.1 **; p < 1 *)

Cyclomatic複雑度と技術的負債の混入との間に（正あるいは負の）弱い相関は認められるものの、相関係数は小さく直接的な原因となっているとは言えない

AN EXPLORATORY STUDY ON SELF-ADMITTED TECHNICAL DEBT

- RQ3: 技術的負債が混入した後, どれくらい除去されるのか?

Eclipse

Version	Release Date	Removed Admitted Technical Debt	Self-	Change
3.0 – > 3.0	June 2004	-		-
3.0 – > 3.1	June 2005	26.3%		26.3%
3.0 – > 3.2	June 2006	27.9%		1.6%
3.0 – > 3.3	June 2007	29.9%		2%
3.0 – > 3.4	June 2008	35.6%		5.7%
3.0 – > 3.5	June 2009	35.6%		0%
3.0 – > 3.6	June 2010	36.9%		1.3%
3.0 – > 3.7	June 2011	36.9%		0%

- 複数のバージョンリリース後でも多くの技術的負債は除去されない
- 除去された技術的負債の大半は, 混入後のリリースで除去される

研究発表：RESEARCH TRACK

- セッション概要

- ソフトウェア保守および進化に関して独創性が高く重要な成果を含む論文発表の場

- セッション数：14

- Developer Knowledge
- Evolving Systems
- Developer Support
- **Technical Debt**
- Empirical Studies
- **Fault Localization 1**
- **Fault Localization 2**
- Software Quality
- Patches
- Recommenders
- Clones
- Refactoring
- **Managing Change 1**
- **Managing Change 2**

セッション：FAULT LOCALIZATION 1

- On the Use of Stack Traces to Improve Text Retrieval-based Bug Localization
 - 著者：Laura Moreno, John Joseph Treadway, Andrian Marcus and Wuwei Shen.
 - 概要：テキスト検索ベースのバグ特定の精度を向上させるためにスタックトレースを用いる手法を提案
- On the Effectiveness of Information Retrieval based Bug Localization for C Programs
 - 著者：Ripon Saha, Julia Lawall, Sarfraz Khurshid and Dewayne E Perry.
 - 概要：Javaのみに対応するテキスト検索ベースのバグ特定手法をCにも対応できるかを検証
- Compositional Vector Space Models for Improved Bug Localization
 - 著者：Shaowei Wang, David Lo and Julia Lawall.
 - 概要：複数の派生モデルを持つベクター空間モデル(VSM)の最適な組み合わせを導出するために遺伝的アルゴリズムを用いたバグ特定手法

セッション：FAULT LOCALIZATION 2

- Boosting Bug-Report-Oriented Fault Localization with Segmentation and Stack-Trace Analysis
 - 著者：Chu-Pan Wong, Yingfei Xiong, Hongyu Zhang, Dan Hao, Lu Zhang and Hong Mei.
 - 概要：セグメント化されたソースコードとスタックトレースを不具合票に対応づけることでバグを特定する手法を提案
- Learning to Combine Multiple Ranking Metrics for Fault Localization
 - 著者：Jifeng Xuan and Martin Monperrus.
 - 概要：実行トレースを用いた動的なバグ特定手法の精度を改善するために、コードの性質を表す複数のメトリクスをランキングする手法
- Precise Isolation of Concurrency Bugs via Delta Triaging
 - 著者：Bo Liu, Bin Wang and Zhengwei Qi.
 - 概要：平行プログラムで発生する平行性不具合（共有メモリ通信に関連する不具合など）の根本原因を発見するのを支援する手法

研究発表：RESEARCH TRACK

- セッション概要

- ソフトウェア保守および進化に関して独創性が高く重要な成果を含む論文発表の場

- セッション数：14

- Developer Knowledge
- Evolving Systems
- Developer Support
- **Technical Debt**
- Empirical Studies
- **Fault Localization 1**
- **Fault Localization 2**
- Software Quality
- Patches
- Recommenders
- Clones
- Refactoring
- **Managing Change 1**
- **Managing Change 2**

セッション：MANAGING CHANGES 1

- Mining Co-Change Information to Understand when Build Changes are Necessary
 - 著者：Shane McIntosh, Bram Adams, Meiyappan Nagappan and Ahmed E. Hassan.
 - 概要：ビルドシステムの変更を必要とするコード変更は、不慣れな開発者にとって大きな障害となる（また、開発の停滞につながる）。コード変更がビルドシステムの変更を必要とするものかどうかを知るために、ビルドシステムの変更が必要となるソースコードとテストコードをマイニングした。
- How Developers' Collaborations Identified from Different Sources Tell us About Code Changes
 - 著者：Sebastiano Panichella, Gabriele Bavota, Massimiliano Di Penta, Gerardo Canfora and Giuliano Antoniol.
 - 概要：ML, BTS, IRC chatの3つのコミュニケーションチャンネルを調査し、重要人物の特定やその人物とコード変更との関係を調べた結果、チャンネル間での重複は少なく、チャンネル毎に異なる結果が得られた。

セッション：MANAGING CHANGES 2

- Maintenance patterns of large-scale PHP Web applications
 - 著者：Panos Kyriakakis and Alexander Chatzigeorgiou.
 - 概要：スクリプト言語は特に保守の観点から大規模システムには不向きであると批判されている。実際はどうなのか？を調べた結果、言語によらずシステムは系統的に保守されていることがわかった。
- Enhancing Clone-and-Own with Systematic Reuse for Developing Software Variants
 - 著者：Stefan Fischer, Lukas Linsbauer, Roberto Erick Lopez-Herrejon and Alexander Eyed.
 - 概要：既存システムをベースに（コピー）してカスタムメイドのシステムを開発するClone-and-Own方式（派生開発？）が広く普及する一方、保守の効率は低下している。Clone-and-Ownを支援する手法を提案している。

研究発表：RESEARCH TRACK

- セッション概要

- ソフトウェア保守および進化に関して独創性が高く重要な成果を含む論文発表の場

- セッション数：14

- Developer Knowledge
- Evolving Systems
- Developer Support
- **Technical Debt**
- Empirical Studies
- **Fault Localization 1**
- **Fault Localization 2**
- Software Quality
- Patches
- Recommenders
- Clones
- Refactoring
- **Managing Change 1**
- **Managing Change 2**

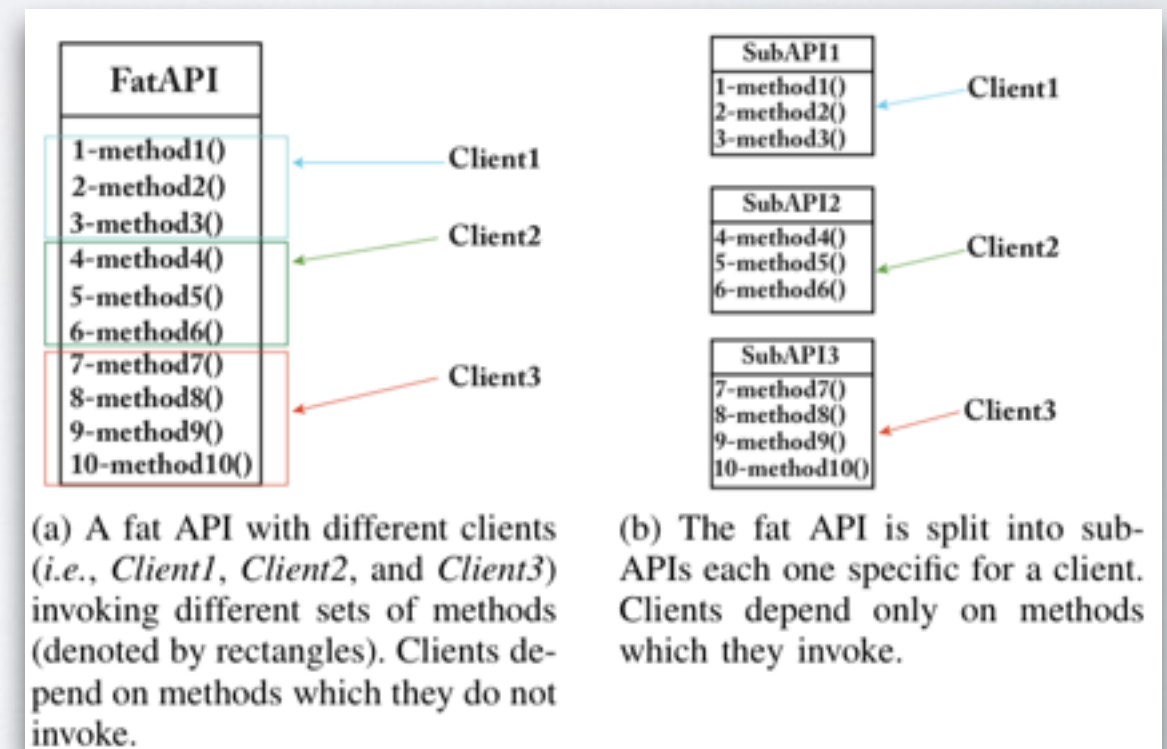
REFACTORING FAT INTERFACES USING A GENETIC ALGORITHM (BEST PAPER)

- 著者

- Daniele Romano, Steven Raemaekers and Martin Pinzger

- 概要

- 肥大化したAPI (Fat API) は、多機能であるが故に頻繁に変更される
- Fat APIの品質は低くなりがちのため、結果的に保守コストを増大させる
- すべてのクライアントの要求を満たすFat APIよりも、各クライアントの目的を満たす程度にAPIを分割する必要がある
- 本研究ではFat APIを適度なサイズのAPIに分割するための手法を提案している



REFACTORING FAT INTERFACES USING A GENETIC ALGORITHM (BEST PAPER)

• 目的

- Interface Usage Cohesion (IUC) が最も高くなるような method の組み合わせを求める

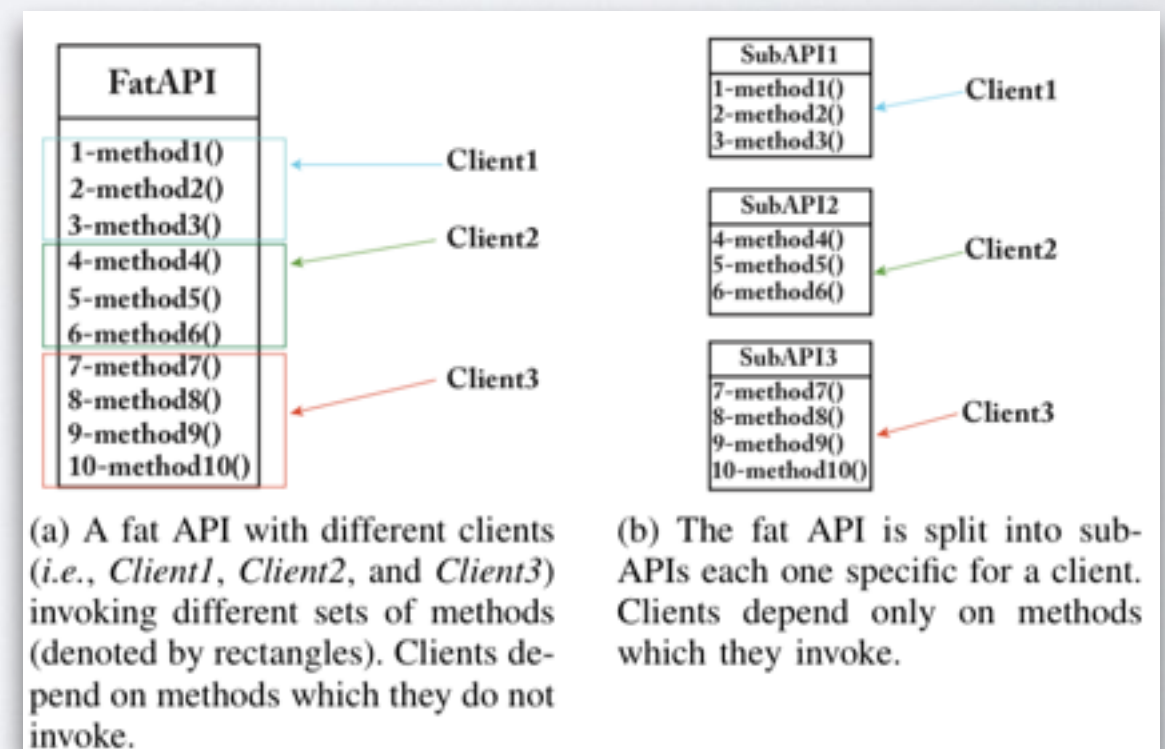
$$IUC(i) = \frac{\sum_{j=1}^n \frac{used_methods(j,i)}{num_methods(i)}}{n}$$

- ただし、多数のクライアントに利用されている Fat API を適切なサイズの sub-API 群に分割するために総当たりで IUC を求めるのは計算量的に現実的ではない

• アプローチ

- 遺伝的アルゴリズムを用いる

1	2	3	4	5	6	7	8	9	10
1	1	2	3	2	4	5	3	6	1

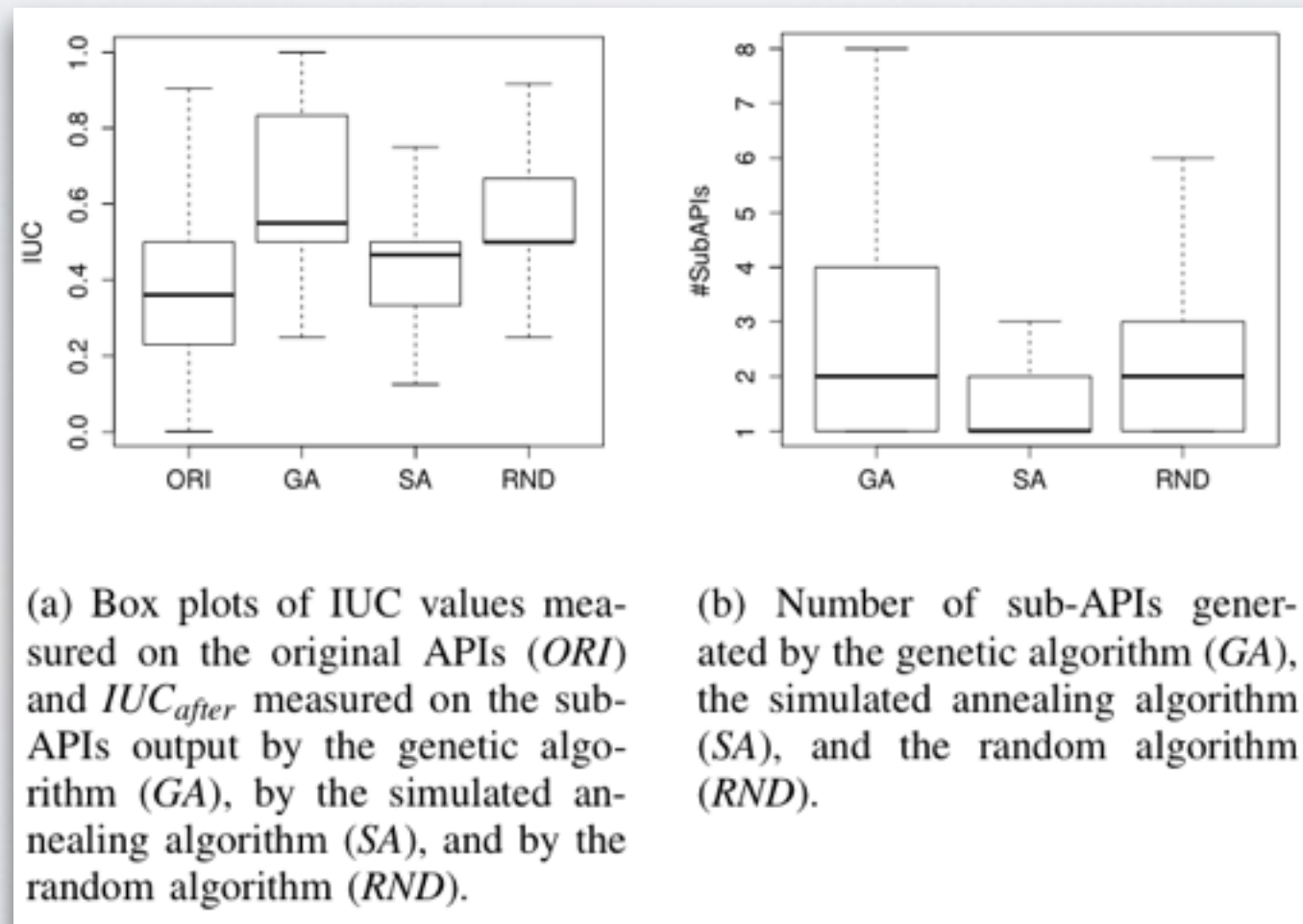


REFACTORING FAT INTERFACES USING A GENETIC ALGORITHM (BEST PAPER)

• 評価実験

- 目的：提案手法(GA)がランダム法や焼きなまし法(SA)より優れているかを確かめること
- 実験データ：Maven repositoryから収集した42,318個のJava API

• 結果



研究発表：INDUSTRY TRACK

- セッション概要

- 産業界の実務的な問題の解決や経験に関する論文を発表する場

- セッション数：2

- Cross-cutting Concerns

- Improving Low Quality Stack Overflow Post Detection
 - Evolutionary Improvements of Cross-cutting Concerns: Performance in Practice
 - Autonomic Software Systems: Developing for Self-Managing Legacy Systems
 - Reverse Engineering PL/SQL Legacy Code: An Experience Report
 - A Heuristic-Based Approach to Refactor Crosscutting Behaviors in UML State Machines

- Software Quality

- Continuous Software Quality Control in Practice
 - Software Defect Prediction for LSI Designs
 - Improving Review of Clustered-code Analysis Warnings
 - Applying ISO/IEC 25010 Standard to prioritize and solve quality issues of automatic ETL processes

研究発表：ERA TRACK

- セッション概要
 - 研究の初期段階ではあるが、将来有望な技術や知見をもたらす可能性のある研究の発表の場
- セッション数：5
 - Commits & Visualization
 - Code & Traceability
 - Reengineering
 - Clones & Applications
 - Maintenance
 - Bugs & Issues

ICSME 2015

- 9月29~10月1日@Bremen, Germany



Die Weserpromenade Schlachte bei Nacht

© BTZ / 2006



和歌山大学での取り組み

本日の内容

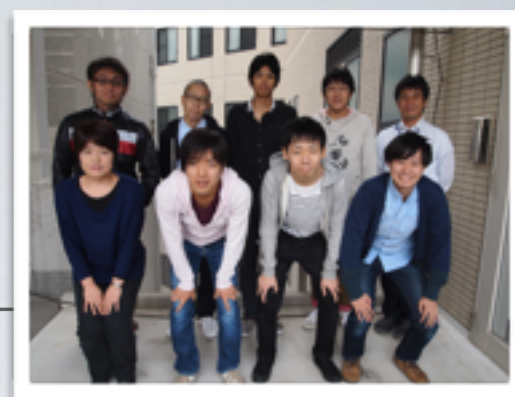
- ICSMEでのトレンド

- 概要と統計
- セッション解説
 - 基調講演
 - パネル
 - 研究発表
 - Research Track
 - Industry Track
 - Early Research Achievements Track

- 和歌山大学での取り組み

- ソフトウェア保守に関する研究の紹介
 - **リポジトリマイニング**
 - バグトリアージ：不具合修正タスクの割り当て支援@SES2014最優秀論文賞
 - ソーシャルマイニング：開発者の社会的関係と生産性／品質の関係@ICSM2012
 - 遅延相関分析：ソフトウェア進化の予兆検出@DICOMO2013優秀論文賞
 - ハイインパクトバグ：開発スケジュール／顧客満足度に大きな影響を与えるバグの分析@ICSME2014
 - IPAソフトウェア工学分野の先導的研究支援事業
 - IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築

研究室紹介： オープンソースソフトウェア工学研究室



- OSSを利活用したソフトウェア開発の支援
- OSS開発を「見える化」するための技術の開発

コミュニティ成熟度の計測／不具合解決時間の予測／協調・競争領域の判別／組織構造分析／バグトリージ手法, など

OSSコミュニティで発生している問題

1. 開発リソースが不足している
2. 一部の開発者に負担が集中し過ぎている
3. 不具合管理が限界にきている
4. 有能な開発者がすぐに離脱する
5. ユーザ企業からの支援がほとんどない



安心・安全にOSSを開発・
利用できる環境の実現！

OSS導入の際のユーザ企業の懸念

1. 緊急時の技術的サポートが得にくい
2. 利用しているOSSがいつまで存続するかわからない
3. バグの改修や顧客からの要請対応に手間がかかる
4. 将来のバージョンアップの計画が把握しにくい
5. セキュリティホールに対するコミュニティの対応に不安

ユーザ企業



エンドユーザ



ソフトウェアリポジトリマイニング

- 蓄積された開発ログデータからソフトウェア開発に有用な知見を導き出すための技術の総称



- Software Intelligence (SI) の実現手段
 - Business Intelligence (BI) にちなんだ造語
 - SW開発における意思決定プロセスを強力に支援 (したい)

WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES (MSR)

- ICSE併設の会議として2004年から続くリポジトリマイニング専門の国際会議
 - MSR2013での主なセッション
 - **Bug Triaging** : 不具合修正担当者の推薦
 - MSR Goes Mobile : モバイルアプリ開発のためのMSR技術
 - **Changes and Fixes** : 変更／不具合修正の支援
 - **Software Evolution** : ソフトウェア進化の分析
 - **Analysis of Bug Reports** : 不具合報告の分析
 - Ecosystems, Big Data : SWエコシステム／大規模リポジトリの分析
 - **Bug Classification** : 不具合箇所の特定
 - **Social Mining** : SW開発における社会的関係／コミュニケーションの分析
 - **Search-Driven Dev** : SWコンポーネント検索エンジン
 - Mining Unstructured Data : 非構造データ（テキスト）マイニング
 - **Predictor Models** : 工数見積り／バグモジュール検出のための予測モデル



リポジトリマイニングでできること (1)

- 製造（プログラミング）支援
 - ソフトウェア部品検索エンジン（SPARSなど）
 - 使用頻度に基づくOSSライブラリの安定バージョンの判別
 - コードクローン検出, など
- テスト／デバッグ支援
 - バグモジュール予測 (Fault-prone Bug Prediction)

MSR技術概説資料として門田暁人准教授（奈良先端大）の講演資料がオススメ！

「門田暁人, ソフトウェアリポジトリマイニングの 技術動向とその応用, ソフトウェアジャパン2013, ITフォーラムセッション」

www.ipa.go.jp/files/000030061.pdf

リポジトリマイニングでできること (2)

- 保守支援

- 不具合箇所の特特定 (Bug Localization)
 - ・ 不具合の原因となるソースコードを特定する
- バグトリージ (Bug Triage)
 - ・ 不具合修正に最も適した開発者を推薦する
- Reopen bugの予測
 - ・ 後日、再度修正が必要となる不具合かどうかを予測する
- 不具合修正時間の予測, など

リポジトリマイニングでできること (3)

- ライセンス管理
 - OSSのライセンス違反を自動的に検出する, など
- 組織構造／コミュニケーションの分析支援
 - 開発プロジェクトの潜在的な組織構造を数値化・可視化する
 - 開発プロジェクト内での情報流通を数値化・可視化する, など
- OSSユーザ支援
 - 修正予定の不具合の特定
 - OSSの不具合は必ずしもすべての不具合が修正される訳ではない
 - 将来, 修正が行われそうな不具合を特定し, ユーザ自ら修正するコストを削減
 - OSSの品質／成熟度の評価, など

TECHNIQUES BEHIND MSR

ソフトウェア（コード）解析

静的／動的解析，形式手法，ソフトウェアメトリクス

統計解析

平均，最頻値，中央値，分散，標準偏差，共分散，相関係数，誤差，偏差値，無作為抽出（ランダムサンプリング）期待値，不偏分散，有意差，尤度関数，回帰分析，重回帰分析因子分析，主成分分析，判別分析，共分散構造分析

データマイニング

頻出パターン（相関ルール分析），判別分析（単純ベイズ分類器，決定木，サポートベクターマシン），回帰分析（線形回帰、ロジスティック回帰），クラスタリング

テキストマイニング

形態素解析，N-gram，共起関係，tf-idf，LSA (LSI)，LDA，主成分分析，クラスター分析

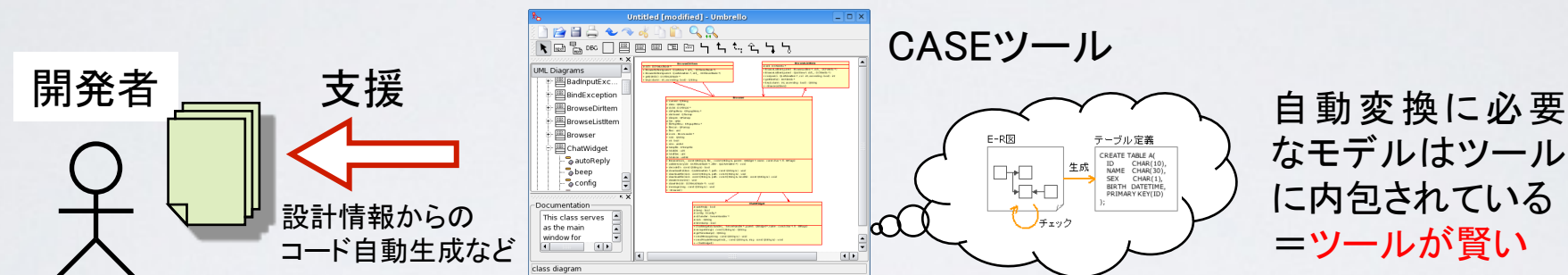
ソーシャルネットワーク分析

グラフ理論，次数分布，平均経路長，クラスタ係数，スケールフリー，次数中心性，近接性，媒介性

これまでのSW開発支援技術との違い

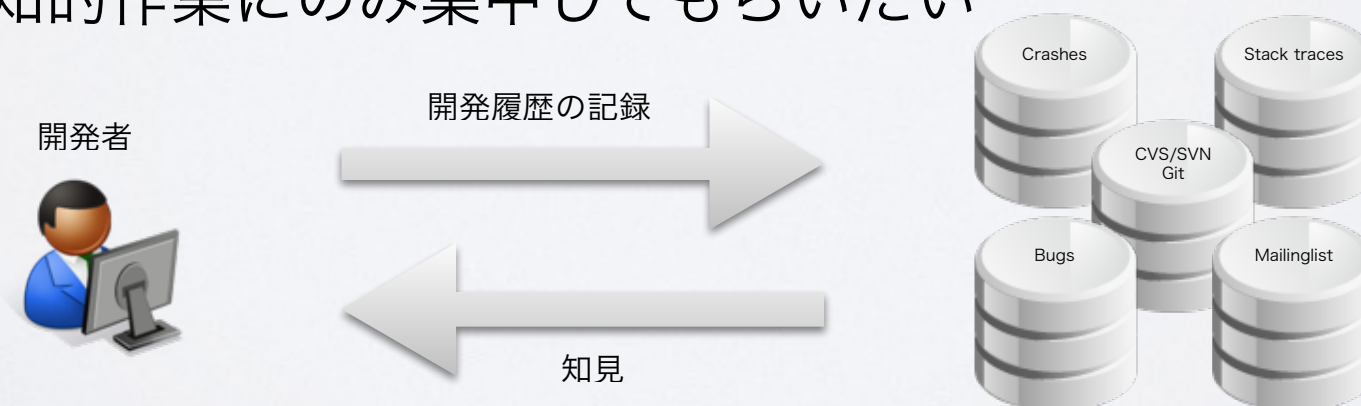
- 伝統的アプローチ

- 例) CASE (Computer Aided Software Engineering) ツール



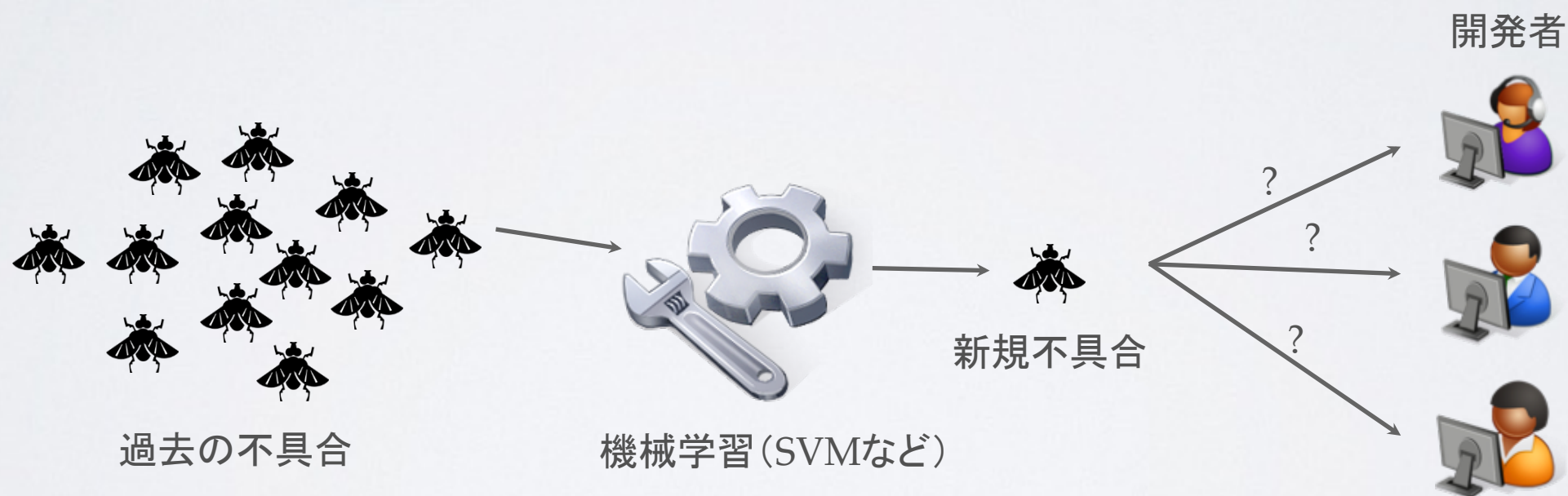
- MSR的アプローチ

- 前提：蓄積された開発ログデータには開発者の創意工夫が埋もれている
 - ・ 隠されたパターンや傾向を掘り起こし、SW開発を自動的に支援したい
 - ・ 開発者には知的作業にのみ集中してもらいたい



BUG TRIAGE : 適任の開発者推薦

- 過去の不具合票(テキストデータ)を機械学習し, 新規不具合に適任の開発者を推薦するアルゴリズム
 - 整数計画法との併用によりタスク割り当てを最適化
 - 不具合修正効率の改善に寄与する技術



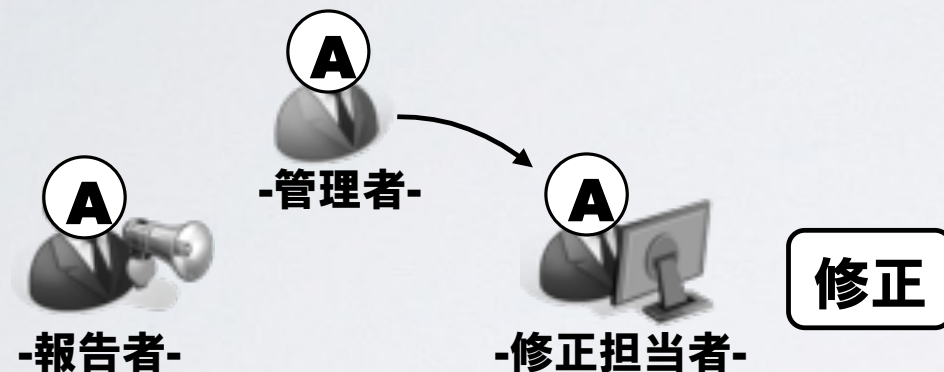
柏 祐太郎, 大平 雅雄, 阿萬 裕久, 亀井 靖高, "大規模OSS開発における不具合修正時間の短縮化を目的としたバグトリージ手法," ソフトウェアエンジニアリングシンポジウム2014論文集, pages 66--75, 2014年8月. (SES2014最優秀論文)

SOCIAL MINING :

開発者の社会的関係と生産性／品質の関係

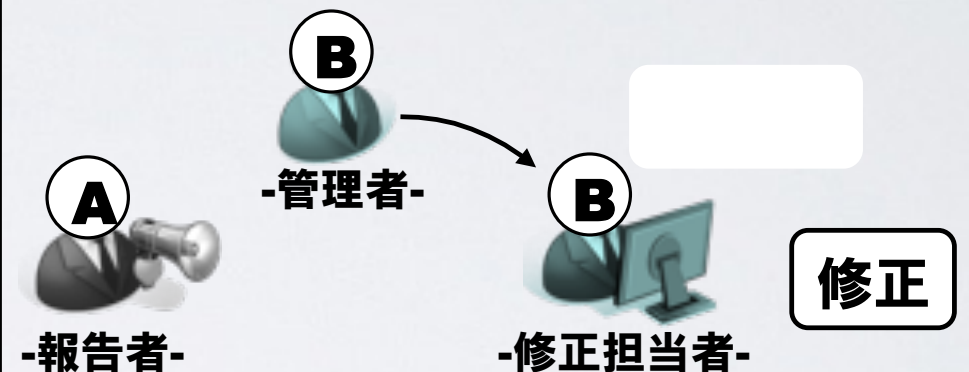
自己完結型

報告者, 管理者, 修正担当者が同一



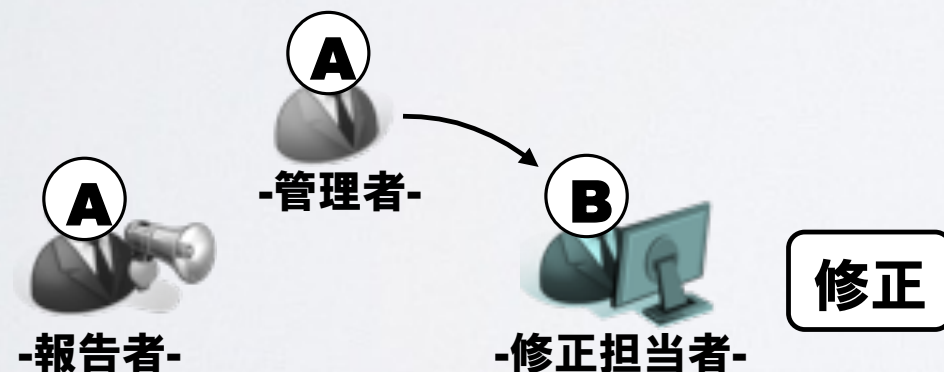
受託型

管理者, 修正担当者が同一



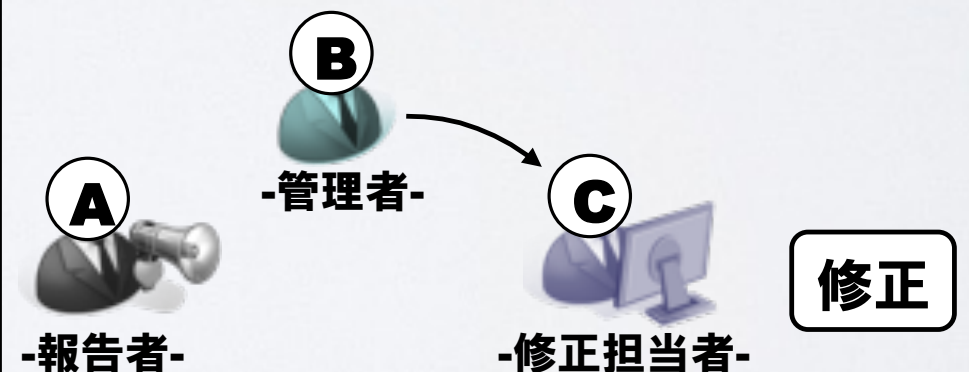
委託型

報告者, 管理者が同一



分業型

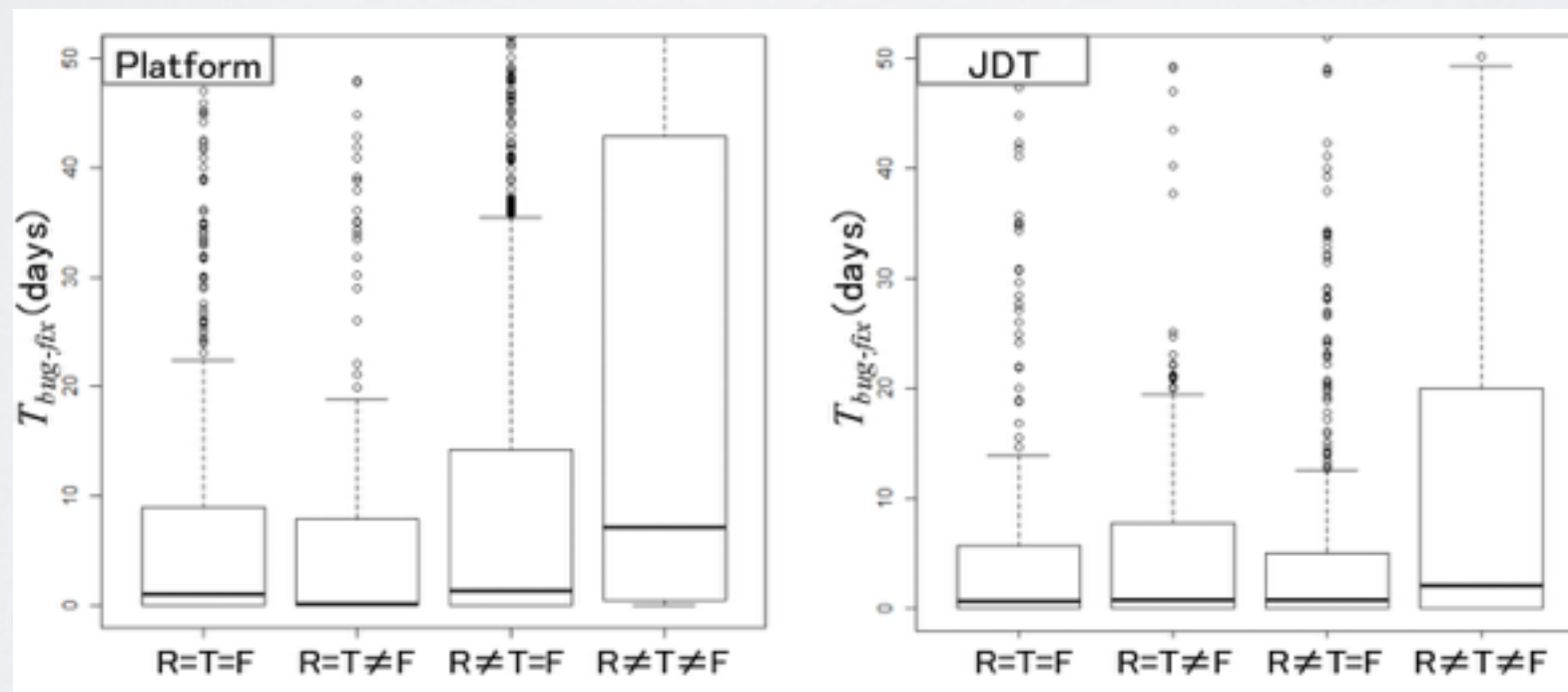
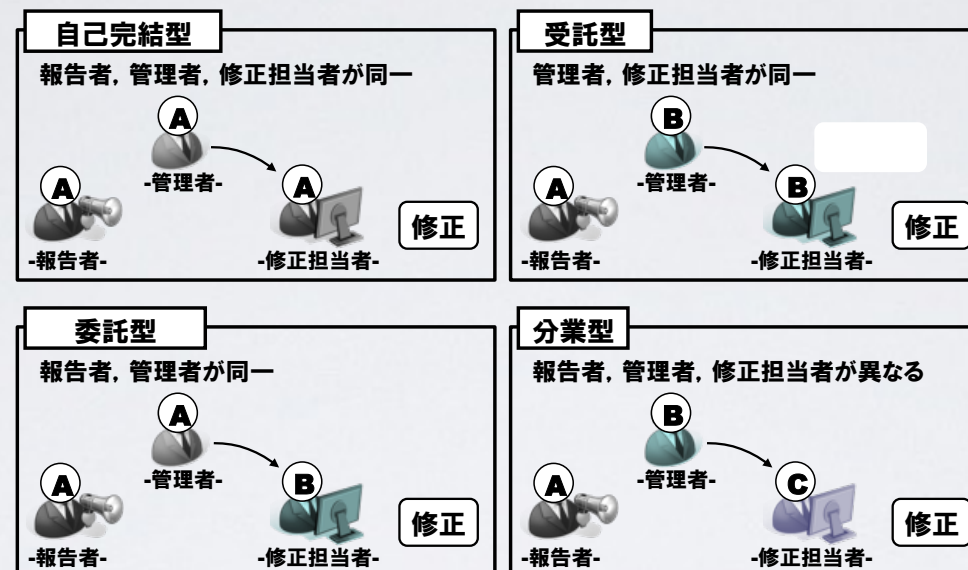
報告者, 管理者, 修正担当者が異なる



Masao Ohira, Ahmed E. Hassan, Naoya Osawa, and Ken-ichi Matsumoto, "The Impact of Bug Management Patterns on Bug Fixing: a Case Study of Eclipse Projects," In Proceedings of 28th IEEE International Conference on Software Maintenance (ICSM2012), pages 264--273, September 2012.

SOCIAL MINING :

開発者の社会的関係と生産性／品質の関係



Masao Ohira, Ahmed E. Hassan, Naoya Osawa, and Ken-ichi Matsumoto, "The Impact of Bug Management Patterns on Bug Fixing: a Case Study of Eclipse Projects," In Proceedings of 28th IEEE International Conference on Software Maintenance (ICSM2012), pages 264--273, September 2012.

SOCIAL MINING :

開発者の社会的関係と生産性／品質の関係

ユーザ／開発者双方のコミュニティに活発に関与

Apache

Netscape

2005/12 (Ver.2.2.0 released)

2002/08 (Ver.7.0 released)

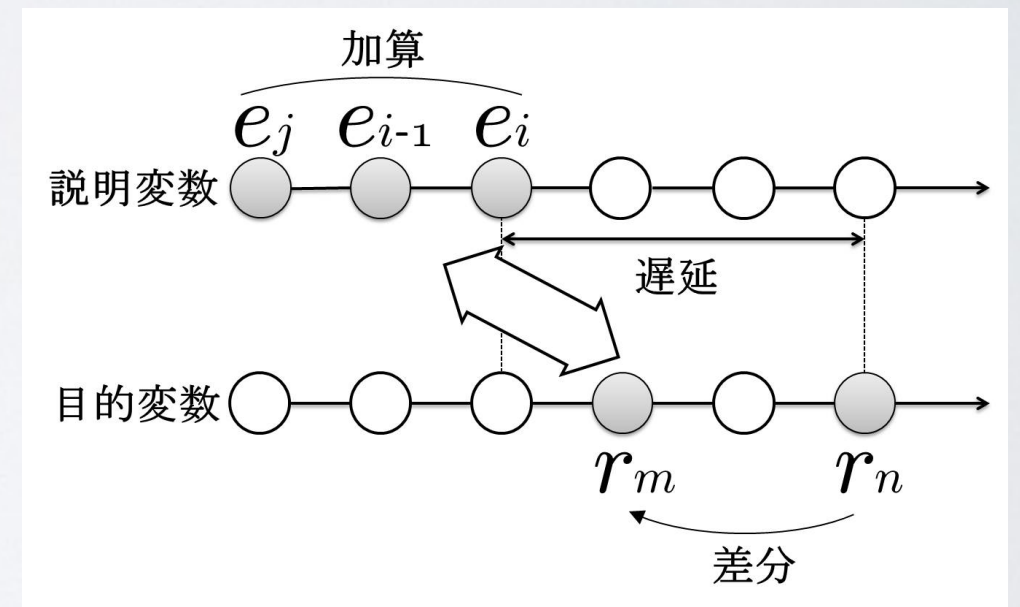
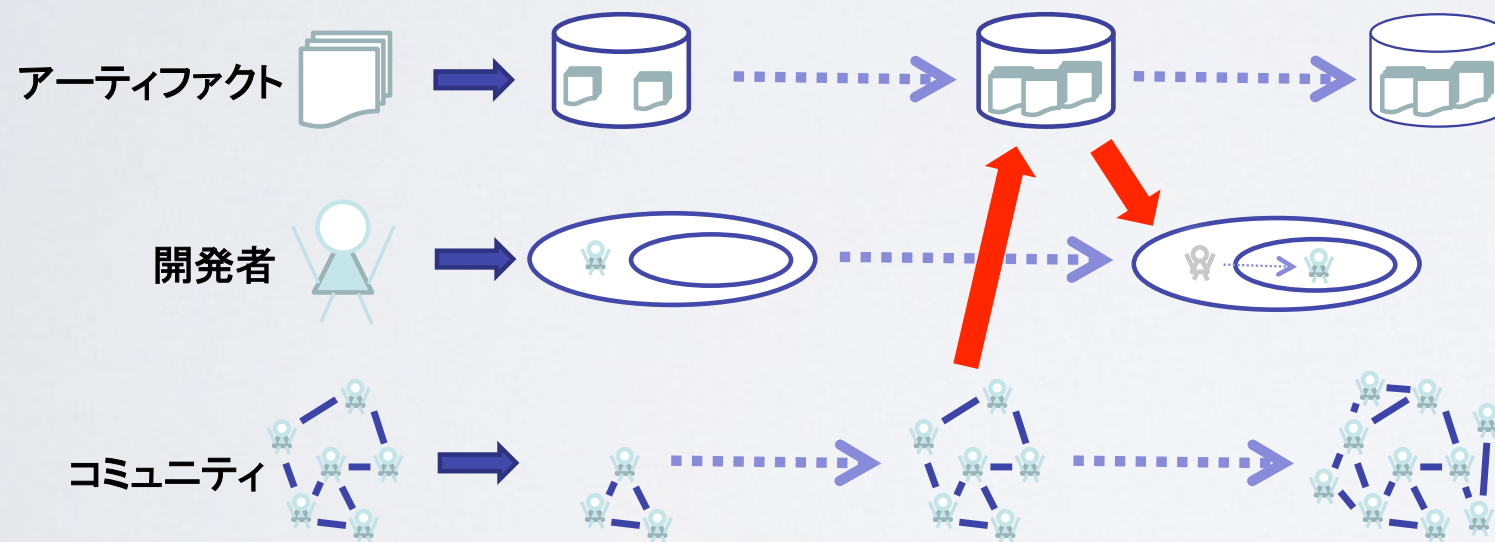
all participants

ほとんど関与しない

遅延相関分析：ソフトウェア進化の予兆検出

● 時系列データ間の遅延を考慮した相関分析

- 竹内らによって提案されさまざまな分野に用いられている
 - ・ 摂取カロリーが体重に与える影響を分析[7]
- 説明変数の**累積**が**遅延**し目的変数の**変化**に影響を与える



山谷 陽亮, 大平 雅雄, Passakorn Phannachitta, 伊原 彰紀, "OSSシステムとコミュニティの共進化の理解を目的としたデータマイニング手法," 情報処理学会 マルチメディア, 分散, 協調とモバイル (DICOMO2013) シンポジウム論文集, pages 1695--1703, 2013年7月. (優秀論文賞)

ハイインパクトバグ：開発スケジュール／顧客満足度に大きな影響を与えるバグの分析@ICSME2014

- **Blocking bug:** 他の不具合の修正完了を滞らせる不具合
- **Surprising bug:** 予想外の時期や場所に現れる不具合
- **Dormant bug:** あるバージョン開発時に埋め込まれ、後バージョンのリリース後に発生した不具合
- **Security bug:** 認証エラーやセッションタイムエラー、暗号化の不足など、セキュリティリスクを持つ不具合
- **Performance bug:** 製品のパフォーマンスを低下させる不具合のこと.
- **Breakage bug:** 新機能の追加や不具合の修正により、以前使えた機能が使えなくなる不具合.

Qpid	# of HIBs	Fixing-time
Surprising	18	130.9
Dormant	7	26.4
Blocking	2	7.2

Derby	# of HIBs	Fixing-time
Security	6	110.9
Performance	7	136.4
Breakage	19	99.7

Yutaro Kashiwa, Hayato Yoshiyuki, Yusuke Kukita, and Masao Ohira, "A Pilot Study of Diversity in High Impact Bugs," In Proceedings of 30th International Conference on Software Maintenance and Evolution (ICSME2014), pages 536--540, 2014.

独立行政法人情報処理推進機構 (IPA)

2013年度ソフトウェア工学分野の先導的研究支援事業

- IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築—一次世代の定量的プロジェクト管理ツールとリポジトリマイニング研究基盤—

1. 背景・課題

IPA EPM-X

- IPAが公開している定量的プロジェクト管理ツール
- EASEプロジェクトのEPM (Empirical Project Monitor)の後継ツール

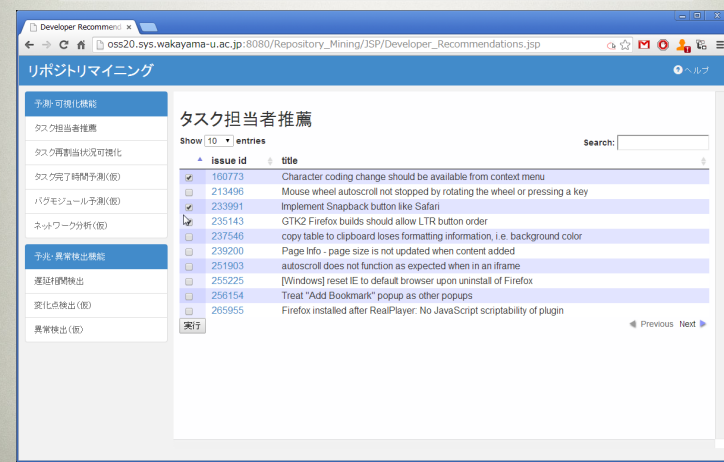


2014/1/29 IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築 (新潟大学)

4. 中間目標の達成状況と成果

達成状況と成果(研究課題A)

Bug Triage: 適任の開発者推薦

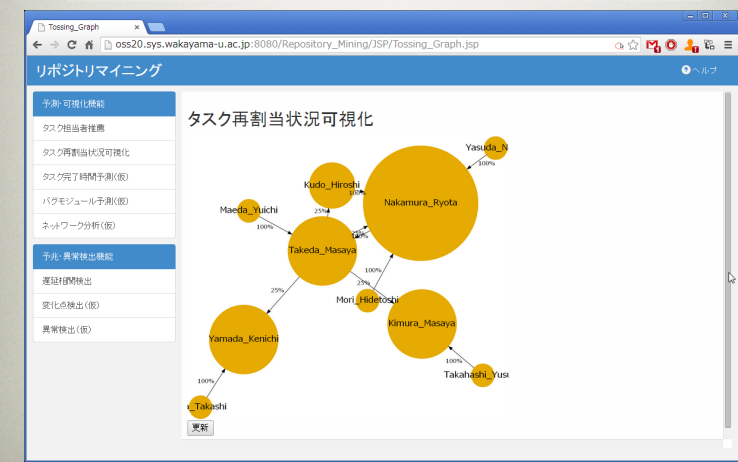


2014/1/29 IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築 (新潟大学)

4. 中間目標の達成状況と成果

達成状況と成果(研究課題A)

Bug Tossing Graph: 再割当ての回避



2014/1/29 IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築 (新潟大学)

2013年度ソフトウェア工学分野の先導的研究支援事業

- IPA EPM-Xの機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築—一次世代の定量的プロジェクト管理ツールとリポジトリマイニング研究基盤—

- demo



CONTACT

- email
 - masao@sys.wakayama-u.ac.jp
- 個人HP
 - <http://masao-ohira.org/>
- 研究室HP
 - <http://oss.sys.wakayama-u.ac.jp/>
 - 論文等をダウンロード頂けます
- EPM-XプラグインHP
 - http://oss20.sys.wakayama-u.ac.jp:8080/Repository_Mining/
 - IPAとの契約の関係により期間限定（11月まで）で公開中
 - リクエスト頂けましたら再度公開も可能です