ICST 2016の振り返りから見るICST 2017の狙い目

～ 君に決めた！ と思える論文・セッションの探し方 ～

ICST 2017 Organization
International Software Testing Contest Chair
朱峰 錦司

自己紹介

- 朱峰 錦司（あけみね きんじ）@kjstylepp
  - 業務
    - ソフトウェアテストに関するR&D (2009.04〜2014.03)
      - テストプロセスの標準整備／技術支援
      - テスト自動化ツールの開発
    - アジャイル開発に関するR&D (2014.04〜)
      - CERTIFIED ScrumMaster
      - アジャイル開発・管理プロセスの標準整備／技術支援
  - 社外活動
    - 主にソフトウェアテスト分野で活動中
      - ICST 2017 Organization
      - テスト自動化研究会
      - WACATE実行委員会

宣伝

- WACATE
  - Workshop for Accelerating CApable Testing Engineers
  - 年2回（6/12月）、神奈川県は三浦海岸にてテストに関する1泊2日のワークショップ合宿を主催
  - 若手中心ですが、ベテランの参加も大歓迎
  - JaSST’ 17 Tokyoにてコミュニティブースも出展予定

- WACATE 2017 夏
  - 2017.06.17（土）～18（日）

0. 本題

こんなお悩みありませんか？

- テストケースが多すぎて実行にすごく時間かかる・・・
- テストケースのメンテナンスが大変・・・
- テストケースやテストデータを自動生成したい・・・
- 非機能テストってどうやったらいいかわからない・・・

そのお悩み！

最新の研究で

解決できるかも！

このセッションのゴール

1. 研究論文の読みはじめ方を理解する

2. ICST 2016で発表された論文の（超）概要を理解する
3. ICST 2016で発表された論文のうち、自身が普段抱いている問題意識に近いものを発見する

4. ICST 2017で発表される論文の（超々々）概要を理解する
5. ICST 2017で発表される論文のうち、自身が普段抱いている問題意識に近いものを発見する

お品書き

1. 論文の読みはじめ方
2. ICST 2016
3. ICST 2017

1. 論文の読みはじめ方

1.1 論文とは

- あるテーマについて、筋道をたてて記した文章
  - 学術研究の成果
  - 業務改善の成果

- ソフトウェア工学関連の論文では、現場の悩みに近い事柄がテーマ、解くべき課題として設定されていることも多い

1.2 論文の構成

- IMRAD, IMRa(nd)D
  - Introduction
    - 論文の導入として、コンテキストの定義や問題提起、研究の位置付けを行う。
  - Methods
    - 解くべき課題に対して、手法を提案する。
  - Results
    - 実験等、手法適用によって得られた結果を述べる。
  - Discussion
    - 得られた結果に対して、論証を行う。

1.3 読みはじめ方 (1/3)

Title	表題
Abstract	概要
Introduction	導入
Methods	手法
Results	結果
Discussion	考察
Conclusion	結論

1. 明らかに関心ごとと関係しないものを除外
2. 論文の背景、目的、成果をつかむ
3. 後半1/4ぐらいを読んで、目的について
   より深く理解する
4. 実験・考察の結果の概要をつかむ

1.3 読みはじめ方 (2/3)

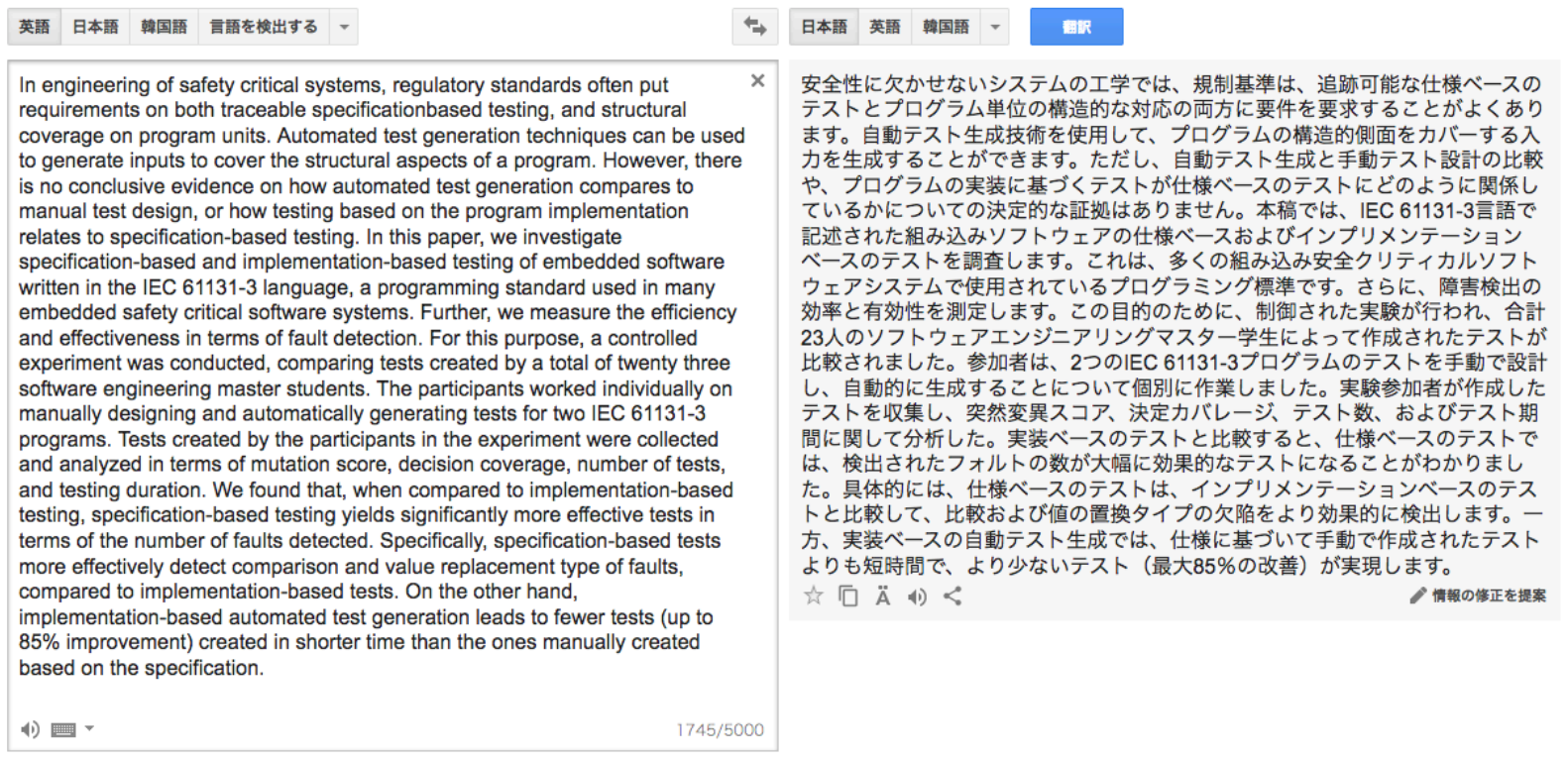
Title	表題
Abstract	概要
Introduction	導入
Methods	手法
Results	結果
Discussion	考察
Conclusion	結論

超々々概要：ICST 2017

超概要：ICST 2016

1.3 読みはじめ方 (3/3)

- Abstract把握の強い味方：Google翻訳
  - Abstract全文をGoolge翻訳に乱暴に突っ込む



※ A Controlled Experiment in Testing of Safety-Critical Embedded Software - http://ieeexplore.ieee.org/document/7515454/

1.4 論文の探し方 (1/2)

- 基本的には有料
  - 発表された会議に参加して配布資料として手に入れる
  - 会員登録した論文検索サイトで手に入れる
    - 海外
      - IEEE Xplore Digital Library
      - ACM Digital Library
    - 国内
      - CiNii

- 意外と著者が自身のサイトで無料で公開している
  - 著者名での検索も忘れずに！

1.4 論文の探し方 (1/2)

- Google Scholar
  - 論文検索に特化したGoogle検索
  - お目当の論文を引用している論文も見つけてくれる優れもの

1.5 心構え

■英語から逃げない

■数学から逃げない

■怪しいダイエット法と思って接する

2. ICST 2016

2.0 ICST 2016の紹介 (1/2)

- 計34本のResearch Papers
  1. Test Generation (4)
  2. Constraint Solving and Search (4)
  3. Debugging (4)
  4. Concurrency and Performance (4)
  5. Web Applications (4)
  6. Test Analysis (4)
  7. Regression Testing (3)
  8. Mutation (3)
  9. Unit Testing (4)

2.0 ICST 2016の紹介 (2/2)

- 各論文について、以下の2点を紹介します
  - どのような問題意識、モチベーションなのか？
  - それに対してどのようなゴールを設定（≒達成）したか？

- 皆さんの普段の問題意識と照らし合わせながら、気になる研究があるかどうか聞いてみてください

2.1 Test Generation

1	A Controlled Experiment in Testing of Safety-Critical Embedded Software	
	ソースコードから自動生成したテストってイケてるの？	人間が設計したテストとソースコードから自動生成したテストをいろんな観点で比べてみたよ。
2	Repeated Combinatorial Test Design – Unleashing the Potential in Multiple Testing Iterations	
	アジャイルで何回も組み合わせテスト生成してると毎回結果が微妙に変わってつらいよね。	前回の結果をいい感じにふまえた繰り返し適用前提の組み合わせテスト生成技法を考えて評価したよ。
3	A Framework to Evaluate the Effectiveness of Different Load Testing Analysis Techniques	
	負荷テスト分析・設計っていろんなやり方あるけど、優劣の比較がされてないよね。	オープンソース分析して見つけた複数の負荷テストアプローチをもとに、それらを横並びで比較するやり方考えて実際に比較したよ。
4	Automatically Discovering, Reporting and Reproducing Android Application Crashes	
	モバイルアプリってセンサーとかが予期せぬ動作してクラッシュしたり、いろいろつらいよね。	Android向けのイケてるクラッシュ可能性検知ツール作って、既存ツールと比較したよ。

2.2 Constraint Solving and Search

1	Symbooglix: A Symbolic Execution Engine for Boogie Programs	
	Boogieで記号実行できるようにしてみたらおもしろそうじゃない？	やってみた！
2	Nonconformity Resolving Recommendations for Product Line Configuration	
	大規模なプロダクトラインに基づく製品構成を人力でやると不整合起こして死ぬよね。	裏で制約解決をしながら対話的に製品構成をサポートするツールつくって評価したよ。
3	Using Exploration Focused Techniques to Augment Search-Based Software Testing: An Experimental Evaluation	
	多数目的最適化が必要なSearch-Based Testingのやり方ないよね。	探索的アプローチで実現した手法を考えて、実際に商用案件で評価したよ。
4	Detecting Assumptions on Deterministic Implementations of Non-deterministic Specifications	
	本来非決定な仕様なものを誤解して設計されたイケてないテストってあるよね。	そんなイケてないテストを見つける手法を確立してJava向けにツール実装して評価したよ。

2.3 Debugging

1	An Empirical Study on Detecting and Fixing Buffer Overflow Bugs	
	バッファオーバーフローを見つける手法いろいろあるけど、最近あまり横並び評価されてないよね。	めっちゃ頑張って横並び評価して結果も公開したよ。
2	Automatic Detection and Removal of Conformance Faults in Feature Models	
	プロダクトラインにおけるフィーチャーモデルのデバッグって大変だよね。	ミューテーション分析にもとづいた自動デバッグ手法考えて評価したよ。
3	Debugging Without Testing	
	「より仕様に則している」ことを評価する手法がのぞまれている！	「正しさ」を相対的に評価・比較する手法を考えたよ。（実験はこれからだよ。）
4	Properties of Effective Metrics for Coverage-Based Statistical Fault Localization	
	バグ予測メトリクスってどれがいいの？	いろんなメトリクス横並び評価して、よかったやつの特徴を分析したよ。

2.4 Concurrency and Performance

1	Mysteries of Dropbox Property-Based Testing of a Distributed Synchronization Service	
	ファイル同期をするようなサービスのテストオラクルってなんだろう？	Dropboxの人たちと協力してサービスのあるべき姿をモデル化したよ。
2	Effective Partial Order Reduction in Model Checking Database Applications	
	データベースを扱うアプリのモデル検査って状態爆発起こしてしんどいよね。	既存研究を参考にしながら、データベースを扱うのに特化した状態数をいい感じにおさえる手法考えて評価したよ。
3	Canopus: A Domain-Specific Language for Modeling Performance Testing	
	パフォーマンステストって設計から実行まで大変だよね。	パフォーマンステスト用のDSL作って、モデルベーステストできるようにしたよ。商用案件で評価もしたよ。
4	Predicting Testability of Concurrent Programs	
	並行プログラムのテストって大変だよね。	プログラムのテスト可能性を定量化して、テスト可能性低いところに集中できるようにしたよ。手法考えて評価もしたよ。

2.5 Web Applications

1	**Why do Record/Replay Tests of Web Applications Break?**	
	キャプチャリプレイ型のテストケースが壊れるケース大杉	頑張って1000件以上の事象調査して壊れる原因を分類したから役立ててね。
2	**Using Visual Symptoms for Debugging Presentation Failures in Web Applications**	
	Webのデザイン崩れを自動検出・修正する手法ってないよね。	画像認識と統計手法にもとづいた自動検出・修正手法を考えて評価したよ。
3	**Detecting and Localizing Internationalization Presentation Failures in Web Applications**	
	国際化で単にテキストを翻訳しただけだと文の長さとかでレイアウト崩れるよね。	翻訳文章がレイアウトを崩しちゃうのを自動検知・修正の提案をする手法を考えて実サイトで評価したよ。
4	**Selecting the right topics for industry-academia collaborations in software testing: an experience report**	
	ソフトウェア工学の文脈での産学連携って少ないよね。	カナダ・トルコでのソフトウェアテストでの成功例分析をして、産業界の人がどうやってアカデミックな手法を取り込むべきかの考察をしたよ。

2.6 Test Analysis

1	**Test Set Diameter: Quantifying the Diversity of Sets of Test Cases**	
	テストの十分性ってどう示すんだろう？	テスト入力セット間の距離を定量化する手法を考えて、それらの多様性を数値で示すことで十分性の指標にしてみよう。
2	**Interpreting Coverage Information Using Direct and Indirect Coverage**	
	コードカバレッジはいろんな理由で100%にできないよね。	カバレッジ未達のうち、間接的に達成できるものを除外して、真にフォーカスしないといけないところを特定する手法を考えたよ。
3	**How well are your requirements tested?**	
	モデルベースドテストツールってイケてるの？	試してみたらけっこうしょぼかったけど、そこからトレサビ改善とかテストデータ設計とかに活かすことはできそうだよ。
4	**Empirical Evaluation of Test Coverage for Functional Programs**	
	コードカバレッジって関数プログラムでも本当に有効なの？	いろんなカバレッジとって、有効性を評価してみたよ。

2.7 Regression Testing

1	**Test Case Prioritization for Compilers: A Text-Vector Based Approach**	
	コンパイラみたいなプログラムには既存のテスト入力データにもとづくテストケース優先順位判定がつかえないよね。	新しくベクトルベースのテスト入力評価手法を考えてGCCとかLLVMに適用評価したよ。
2	**Tedsuto: A General Framework for Testing Dynamic Software Updates**	
	DSUのパッチは注意深くテストしないと本末転倒になるよね。	DSUのためのテスト設計フレームワークを考案し、Javaで実装して評価したよ。
3	**Model-based Regression Test Selection for Validating Runtime Adaptation of Software Systems**	
	DSUにおいてテストに使える時間やリソースは非常に限られているから効率よくやりたいよね。	JavaのDSUフレームワークであるFiGA上でモデルベースドでテストケース選択する手法を考えて評価したよ。

2.8 Mutation

1	A Theoretical Framework for Understanding Mutation-Based Testing Methods	
	ミューテーションテスト手法ってあやしくない？	テスト手法の根拠を数学的に論じるためのフレームワークを構築したよ。
2	Generating Evil Test Strings for Regular Expressions	
	正規表現が正しいかどうかのテストは難しいよね。	正規表現にマッチする意地悪な文字列を生成する手法を考えて評価したよ。
3	MuVM: Higher Order Mutation Analysis Virtual Machine for C	
	ミューテーションテスト手法は有用だけど、ミュータント生成が遅いよね。	頑張って早くする手法を考えて評価したよ。

Japan

2.9 Unit Testing

1	**Atrina: Inferring Unit Oracles from GUI Test Cases**	
	リッチなUIを実現するJavaScriptのユニットテスト、とくにアサーションかくのしんどいよね。	JavaScriptコードを解析してアサーションの自動生成支援を行う手法を考えて、ツール化とのその評価をしたよ。
2	**Automatically Documenting Unit Test Cases**	
	ユニットテストケース文書は書くのしんどいし、実施書かれないよね。	いろんな技術組み合わせて自然言語の文書生成する技術を考えて評価したよ。
3	**Profiting from Unit Tests For Integration Testing**	
	結合テストって特有のバグがいっぱい出るよね。	バグの情報やユニットテストコードを活用して、バグ発見能力の高い結合テストを自動生成する手法を考えて評価したよ。
4	**Coordinated Collaborative Testing of Shared Software Components**	
	複数グループ間で共有されるコンポーネントの互換性確認のための回帰テストはコストが高いよね。	コストを最小化するためのグループ間協調プロセスを考えて、その評価をしたよ。

2.10 まとめ

- いろいろなタイプの論文がある
  - 技術的な問題・課題の解決
  - 明確な根拠のない事象や疑問に対する評価の実施
  - 調査
    - ある程度の範囲のものを横並び評価
    - 広く調べて分類
  - おもしろそうだからやってみた

- セッション名だけで判断するのは危険
  - せめてタイトル見て論文単位でお目当を決めよう

3. ICST 2017

3.0 ICST 2017の紹介

- ICST 2017で発表される予定の全36本の各論文について、タイトルと、その中に含まれるキーワードをピックアップしながら紹介します

- 前章と同様、自分がICST 2017に参加するならどれを「推し研」にするかを考えながら聞いてみてください

3.1 論文一覧 (1/4)

※ http://aster.or.jp/conference/icst2017/program/accepted.html より抜粋

1. Are there any Unit Tests? An Empirical Study on Unit Testing in Open Source Python Projects
2. Coveringcerts: Combinatorial Methods for X.509 Certificate Testing
3. The Theory of Composite Faults
4. Using Semantic Similarity in Crawling-based Web Application Testing
5. FIFA: A Kernel-Level Fault Injection Framework for ARM-based Embedded Linux System
6. Statistical Model Checking Meets Property-Based Testing
7. Model-Based Testing IoT Communication via Active Automata Learning
8. The Fitness Function for the Job: Search-Based Generation of Test Suites that Detect Real Faults

3.1 論文一覧 (2/4)

※ http://aster.or.jp/conference/icst2017/program/accepted.html より抜粋

9. A Search-based Testing Approach for XML Injection Vulnerabilities in Web Applications
10. Prephecy: Performance Regression Test Selection Made Simple but Effective
11. System Testing of Timing Requirements based on Use Cases and Timed Automata
12. Model-based API Testing of Apache ZooKeeper
13. Symbolic Complexity Analysis using Context-preserving Histories
14. Automated Testing of Definition-Use Data Flow for Multithreaded Programs
15. Error Propagation Analysis of Multithreaded Programs Using Likely Invariants
16. JavaScript: The (Un)covered Parts
17. Mining Sandboxes for Linux Containers

Japan

3.1 論文一覧 (3/4)

※ http://aster.or.jp/conference/icst2017/program/accepted.html より抜粋

18. A Selection Method for Black Box Regression Testing with a Statistically Defined Quality Level
19. Dynamic Inference of Timed Automata
20. Incremental Deductive Verification for Relational Model Transformations
21. CBGA-ES: A Cluster-Based Genetic Algorithm with Elitist Selection for Supporting Multi-objective Test Optimization
22. Broadcast vs. Unicast Review Technology: Does it Matter?
23. Localizing Faults in SQL Predicates
24. Non-Semantics-Preserving Transformations For High-Coverage Test Generation Using Symbolic Execution
25. Verifying Concurrent Programs using Contracts
26. Prevalence of Single-Fault Fixes and its Impact on Fault Localization

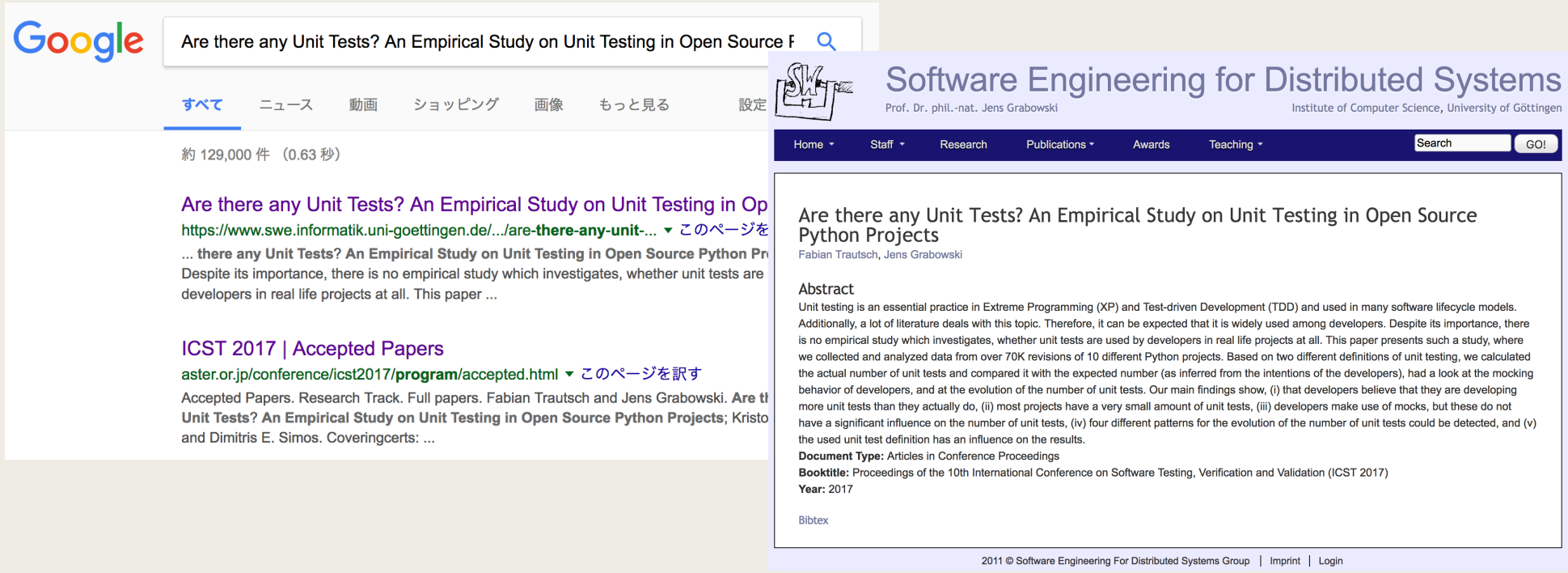
3.1 論文一覧 (4/4)

※ http://aster.or.jp/conference/icst2017/program/accepted.html より抜粋

27. Recovering Semantic Traceability Links between APIs and Security Vulnerabilities: An Ontological Modeling Approach
28. Uncertainty-Driven Black-Box Test Data Generation
29. Behavioral Execution Comparison: Are Tests Representative of Field Behavior?
30. Private API Access and Functional Mocking in Automated Unit Test Generation
31. Barista: A Technique for Recording, Encoding, and Running Platform Independent Android Tests
32. Efficient Incrementalized Runtime Checking of Linear Measures on Lists
33. Automated Testing of Alloy Models
34. ATOM: Automatic Maintenance of GUI Test Scripts for Evolving Mobile Applications
35. Automated Random Testing in Multiple Dispatch Languages
36. Using Delta Debugging to Minimize Stress Tests for Concurrent Data Structures

3.2 Abstractを読んでみよう

- ググれば2017の論文でも意外と出て来ます
  - たとえば1つめの論文



※ https://www.swe.informatik.uni-goettingen.de/publications/are-there-any-unit-tests-empirical-study-unit-testing-open-source-python-projects

おわりに：このセッションのゴール（再掲）

1. 研究論文の読みはじめ方を理解する

2. ICST 2016で発表された論文の（超）概要を理解する
3. ICST 2016で発表された論文のうち、自身が普段抱いている問題意識に近いものを発見する

4. ICST 2017で発表される論文の（超々々）概要を理解する
5. ICST 2017で発表される論文のうち、自身が普段抱いている問題意識に近いものを発見する


Let's take part in
ICST 2017