

社内リソースを活用した一斉テスト



社内一斉テストで製品の品質を高めよう！

株式会社ワークスアプリケーションズ
アドバンスト・テクノロジー＆エンジニアリング本部
品質エンジニア 高橋サラ

AGENDA

1. 紹介・背景
2. クラウドソースドテスト及び探索的テストとの出会い
3. クラウドソースドテスト + 探索的テスト + α = ?
4. 一斉テストの効果
5. 一斉テストをやってみよう！
6. 今後の展望



AGENDA

1. 紹介・背景
2. クラウドソースドテスト及び探索的テストとの出会い
3. クラウドソースドテスト + 探索的テスト + α = ?
4. 一斉テストの効果
5. 一斉テストをやってみよう！
6. 今後の展望



企業紹介:会社概要

株式会社ワークスアプリケーションズ



事業内容： 大手企業向け**ERP**パッケージ「**COMPANY®**」の
開発・販売・サポート

企業理念： 「日本の情報投資効率を世界レベルへ」
「日本のクリティカルワーカーに活躍の場を」

企業紹介: COMPANY®シリーズラインナップ

AC Series

財務 / 管理会計

COMPANY.
Financial Management

- 一般会計
- 連結会計
- 予算管理
- 管理会計
- 開示書類作成
- 経費管理

債権 / 債務管理

COMPANY.
Business Management

- 請求処理
- 入金・消込
- 受取手形管理
- 支払予定管理
- 支払処理
- 支払手形管理

資金管理

COMPANY.
Cash Management

- 資金繰り管理
- 借入金管理
- CMS

資産管理

COMPANY.
Assets Management

- 固定資産管理
- リース資産管理
- 減損会計

一般購買管理

COMPANY.
Purchase Management

- 購入申請
- 在庫管理
- 検収支払管理

EC Series

サイト構築

COMPANY.
CMS

- サイト構築・編集
- コンテンツ配信
- 公開管理

通販 / 問合せ対応

COMPANY.
E-Support

- 問合せ対応
- 電話・FAX注文
- コールセンター連携

電子商取引

COMPANY.
E-Commerce

- 受注・配送・決済
- 顧客情報管理
- 商品情報管理

分析 / プロモーション管理

COMPANY.
E-Marketing

- アクセス解析
- 売上・顧客分析
- メールプロモーション

SCM Series

販売管理

COMPANY.
SCM 販売管理

- 受注管理
- 出荷管理
- 売上管理

調達管理

COMPANY.
SCM 調達管理

- 発注管理
- 入荷・検査管理
- 仕入・検査

SCM共通

COMPANY.
在庫管理
棚卸

原価管理

輸出管理

背景

所属部署で扱っている製品「COMPANY®」はE-Commerceのパッケージソフトウェア。

E C サイトの構築・運営業務をサポートする。

*E-Commerce（E C）：電子商取引
商品やサービスをインターネット上で売買する



背景

大きく二つに分けることができる。

- ・サイト構築・運営をするための管理画面
- ・エンドユーザーがお買い物をするＥＣサイト

背景

品質エンジニアとして
管理画面及びECサイトの評価を行う必要があり、
特にエンドユーザーが直接利用する
サイト側の品質が特に重要。

ECサイトに不具合有り
= サイトに対する不信感
= サイトからの離脱の可能性大

背景

管理画面

- ・利用者が限られている
- ・利用方法をある程度指定することが可能

E Cサイト

- ・利用者は不特定多数
- ・利用方法は人それぞれ、指定不可能
- ・あらゆる利用方法の想定が必要

背景

だが、とにかく検証・評価にかけられる時間が限られている！

プロジェクトの性質

- ✓ 短期導入（1～2ヶ月）
- ✓ デザイン会社の介在
- ✓ ECは流行りもの



問題を解決するためのポイント

以下のポイントを達成すれば

「時間がない！」という問題を解決できるのでは？

短期導入に耐えられるテスト手法を確立する

短時間でも様々な操作方法や操作環境で検証する

短時間でも検証内容に網羅性を持たせる

不具合を短時間で発見し、

開発者が修正にかけられる時間を確保する

問題を解決するためのポイント

ただ、そんな簡単に答えが出るはずもなく．．．

AGENDA

1. 紹介・背景
2. クラウドソーステスト及び探索的テストとの出会い
3. クラウドソーステスト+探索的テスト+ α = ?
4. 一斉テストの効果
5. 一斉テストをやってみよう！
6. 今後の展望



JaSST'11での出会い①：クラウドソースドテスト

英語で Crowd Sourced Testing、
つまり群衆でのテスト。

外部の不特定多数の人にテストを委託すること。



JaSST'11での出会い①：クラウドソーステスト



JaSST'11での出会い①：クラウドソースドテスト

メリット

数百人規模での同時テストが実施可能

フレキシブルな工数確保が可能

多種多様な視点からの洞察を得ることができる

JaSST'11での出会い①：クラウドソースドテスト

だが、色々と問題点もある。

弊社のセキュリティ上、NDA（秘密保持契約）をテスターと結ぶ必要があるが、不特定多数のテスターとの急速な締結はできない

検証内容の把握が困難である

公開環境作成にかかる工数をテストに回した方が効率が良い

JaSST'11での出会い②：探索的テスト

テスターが製品について探索をしながらテストを設計し、実行するというテスト手法。

チャーター

探索的テストを行う際、「チャーター」を用いる場合がある。
これはテストのガイドラインとなるもの。

時間制限

濃密なテストにするため、探索的テストは60～120分という短時間で行うのが理想的。

AGENDA

1. 紹介・背景
2. クラウドソースドテスト及び探索的テストとの出会い
3. クラウドソースドテスト+探索的テスト+ α = ?
4. 一斉テストの効果
5. 一斉テストをやってみよう！
6. 今後の展望



クラウドソーステスト+探索的テスト+ α = ?

クラウドソーステストと探索的テストの
いいとこ取りをしてみよう！



大勢の社員を集め、一斉にテストしてもらう

+

チャーターを用意してテストの方向性を決め、
後は探索的にテストをしてもらう

+

テスト時間を30分間に限定！

=

一斉テスト

AGENDA

1. 紹介・背景
2. クラウドソースドテスト及び探索的テストとの出会い
3. クラウドソースドテスト + 探索的テスト + α = ?
- 4. 一斉テストの効果**
5. 一斉テストをやってみよう！
6. 今後の展望



30分間限定の効果：テスト工数削減

6つのサイトで一斉テストを行った結果（平均）

- テスト参加人数 : **26名**
- テスト1回の工数 : **1.6人日**
- 総報告件数 : **116件**
- 重複を省いた不具合件数 : **99件**

30分間限定の効果：テスト工数削減

品質エンジニア 1人が1.6人日掛けて評価をしても、ここまで数の不具合を出すことは難しい

→集中力が持続しない

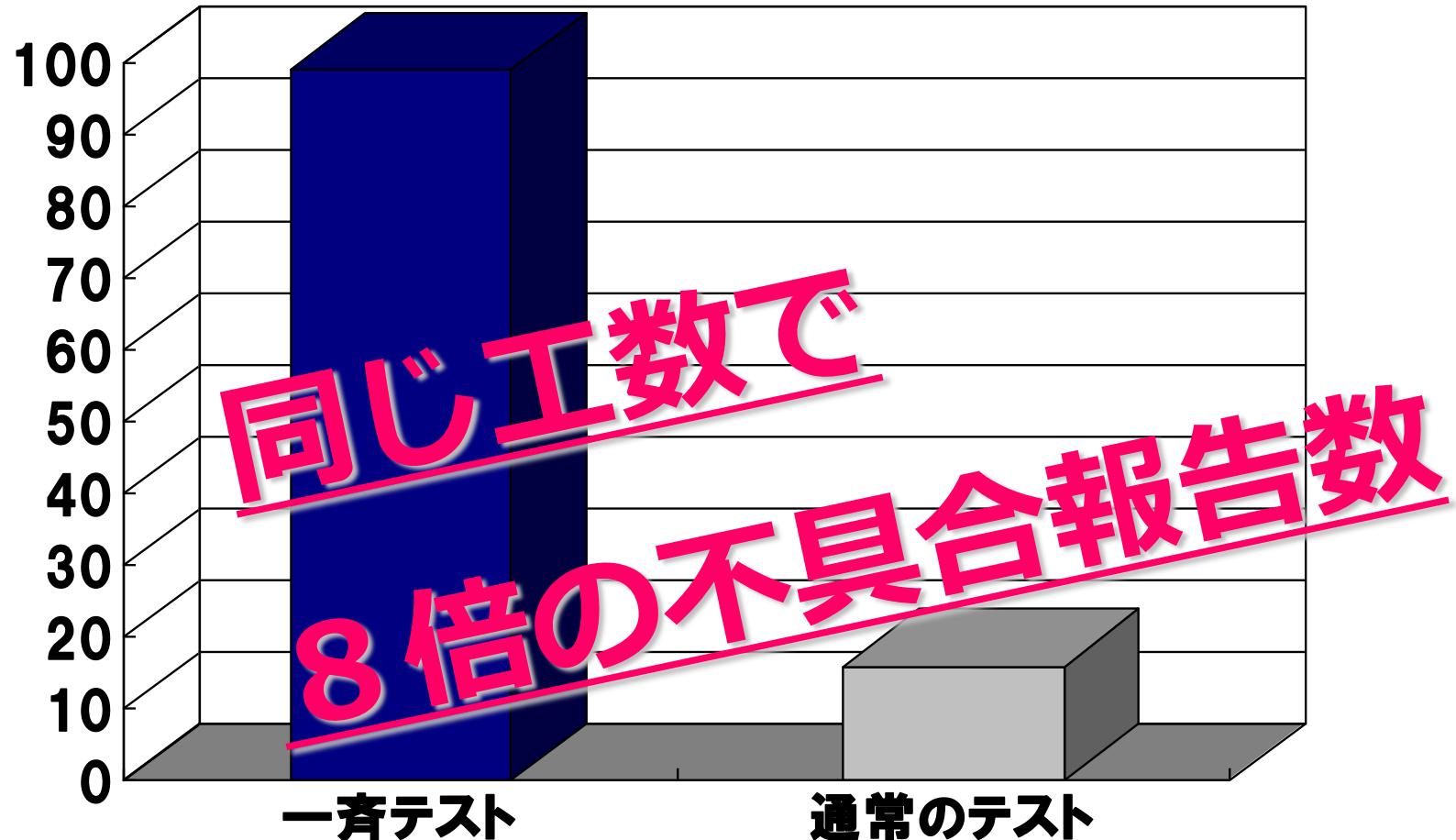
→何度も同じ操作をしていると、飽きる

→ブラウザを変えて再テストしなければならない

人を集めて作業を分散し、

一気に不具合を出してしまう方が効率的！

30分間限定の効果：テスト工数削減



30分間限定の効果：テスト工数短縮

テストが短時間で終了する

= 開発者に不具合を速く報告できる

= 修正期間を多く確保できる

チャーターの効果：報告された不具合の重複の減少

チャーターを用意し、
それに沿って探索的にテストしてもらいました
*チャーター=ガイドライン

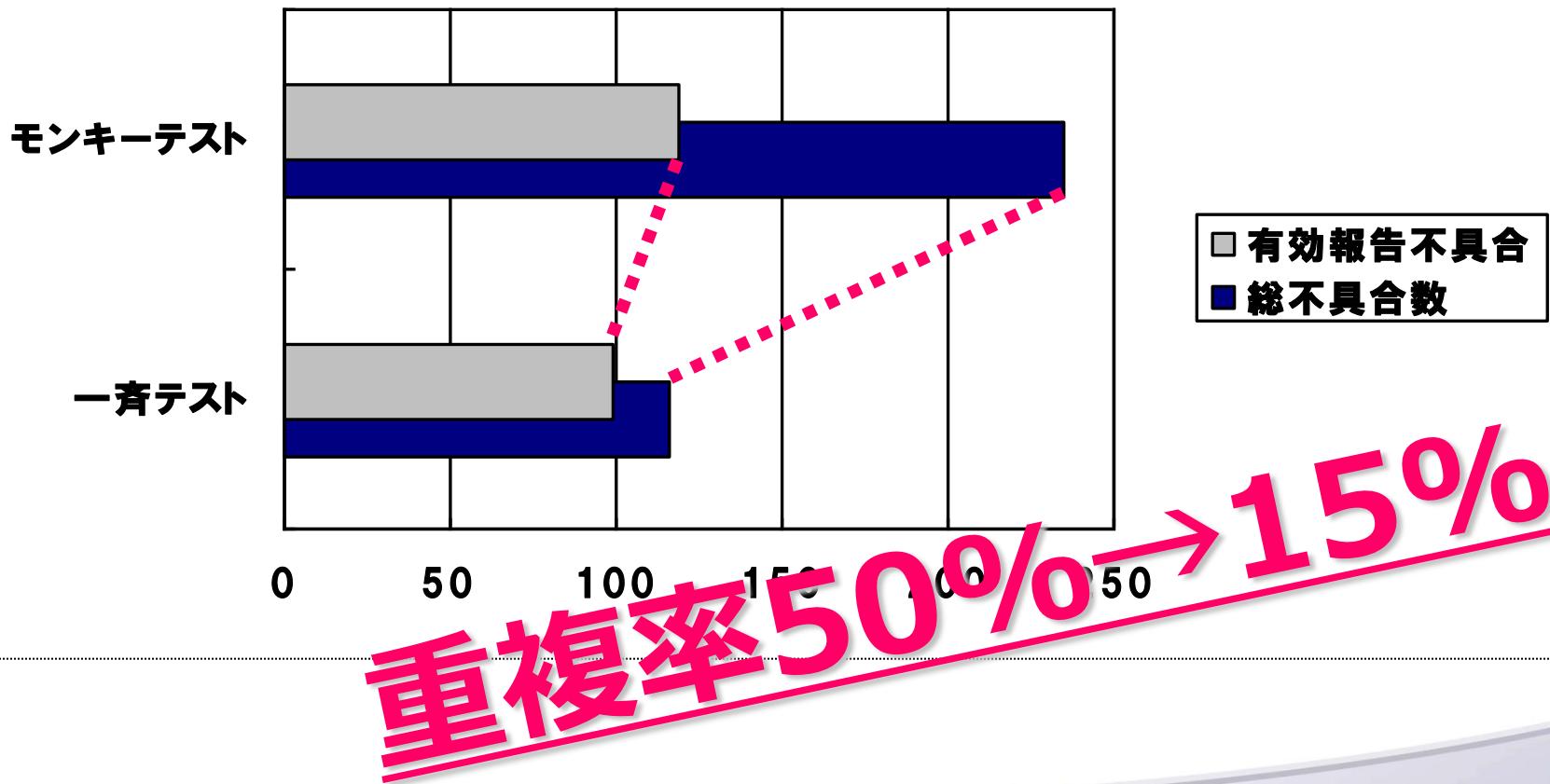
実際に使ったチャーターの例

- 1.会員登録→マイページいじり倒す→退会する
- 2.ひたすら商品検索・カートに出し入れ
・お気に入りリスト追加・削除
- 3.自分の為に商品を購入する
- 4.自分以外の人の為に商品を購入する



チャーターの効果：報告された不具合の重複の減少

結果、テスト範囲を上手く分散することができ、報告された不具合の重複件数を抑えることに成功



チャーターの効果：報告された不具合の重複の減少

なぜ重複が少ない方がいいのか？

複数人が同じ不具合を発見し、
起票をしていることになる。

→折角工数を割いてもらっているのに、
その作業が無駄になってしまうのはもったいない

一斉テストの効果：まとめ

短時間で一気に不具合を検出することが可能

多くの人が多様な環境でテストをする

→様々なブラウザ＆バージョンでのテストが可能

→様々な操作方法でのテストが可能

高負荷時のパフォーマンスを確認することが可能

テスターが社内にいるので、再現依頼が容易

AGENDA

1. 紹介・背景
2. クラウドソースドテスト及び探索的テストとの出会い
3. クラウドソースドテスト + 探索的テスト + α = ?
4. 一斉テストの効果
5. 一斉テストをやってみよう！
6. 今後の展望



一斉テストをやってみよう！

一斉テストの手順紹介

準備編→実施編→実施後編

一斉テストをやってみよう！

一斉テストの準備編

1. テスト環境を整える



一斉テストをやってみよう！

一斉テストの準備編

2. チャーターを用意する



一斉テストをやってみよう！

一斉テストの準備編

3. 参加者を募る



一斉テストをやってみよう！

一斉テストの準備編

4. 参加者にチャーターを割り振り、説明資料を配布する



一斉テストをやってみよう！

一斉テストの実施編

5. 一斉テスト開始！
各自担当チャーターを元にテスト



一斉テストをやってみよう！

一斉テストの実施編

6. 30分経ったら、テスト終了



一斉テストをやってみよう！

一斉テストの実施編

7. 不具合を専用のBTS（TRAC等）に起票してもらう



一斉テストをやってみよう！

一斉テストの実施後編

8. 不具合の起票が完了したら、
重複などを省く精査作業を開始



一斉テストをやってみよう！

一斉テストの実施後編

9. 精査後、担当開発者に不具合を報告



一斉テストをやってみよう！

一斉テストの実施後編

10. 担当開発者による不具合修正作業



一斉テストをやってみよう！

一斉テストの実施後編

11. 不具合の修正確認



どんなソフトウェアに向いている？

E Cサイトの様な、Webアプリなど
利用者数が多く、また利用者によって使い方が
異なるものに向いている。

E Cの管理画面のテストには向いていない。

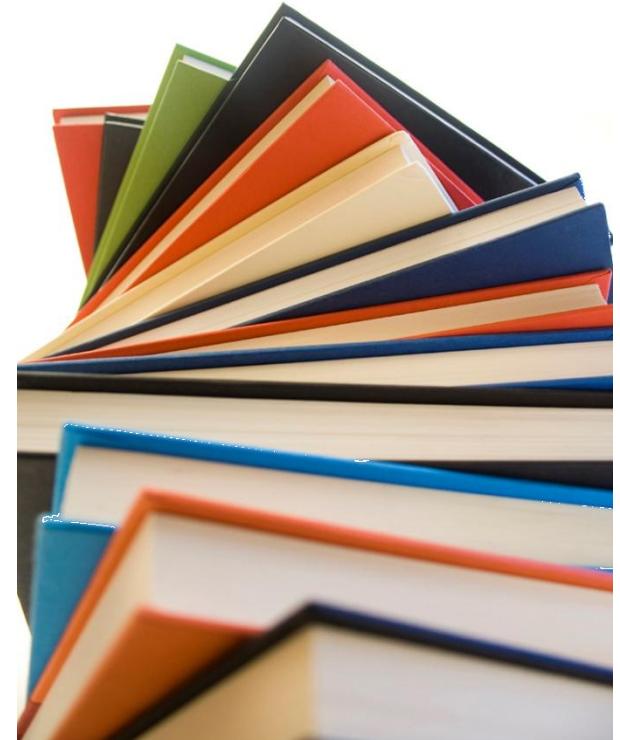
AGENDA

1. 紹介・背景
2. クラウドソースドテスト及び探索的テストとの出会い
3. クラウドソースドテスト + 探索的テスト + α = ?
4. 一斉テストの効果
5. 一斉テストをやってみよう！
6. 今後の展望



今後の展望

- ・実施タイミングの検討
- ・重複チェック方法の検討
- ・最適な参加人数の把握
- ・社内でのさらなる協力体制の確保



おしまい

ご清聴ありがとうございました。

ご質問・ご感想等は

takahashi_sa@worksap.co.jpまでお願いします。