

超短期開発プロジェクト/ 非クリティカルシステムに向けた軽量品質保証 概念とその最上流プロセスの提案



SQiPシンポジウム2013 事例発表
松木 晋祐

軽量品質保証はDevQUTへ



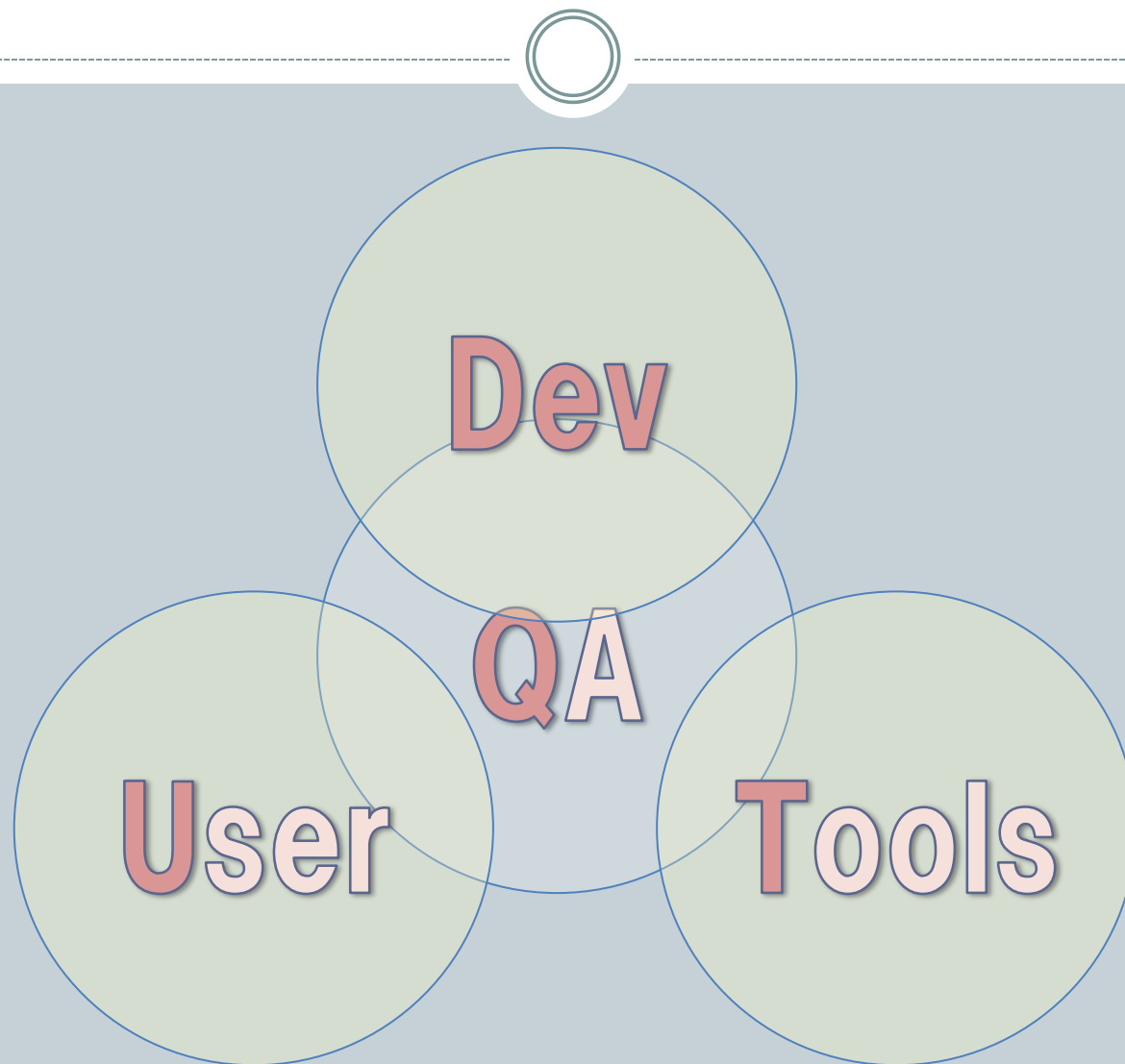
- ダメな、重厚な品質保証は壁を作る
- 開発を信頼しないという壁
 - 開発者の技術力を信頼しないから、バグを怖れて単純作業的なテストケースを大量に消化しないといけない
- ユーザを信頼しないという壁
 - ユーザを信頼しないから、クレームを怖れて不必要に細かいバグも取り除かないといけない
- ツールを信頼しないという壁
 - ツールを信頼しないから、とにかく人海戦術でテストをしないといけない

壁を壊せ！ -DevQUT-



- **Communitaization**(造語)
：コミュニティ化する
- 対立しそうなものを融合して同じ方向に向けること
 - 提供と消費の壁を壊して成功している例
 - ✦ 初音ミク
 - ✦ AKB48
 - ソフトウェア開発への応用
 - ✦ DevOps
 - ✦ DevQUT

壁を壊せ！ -DevQUT-



Dev+Q 開発者とQAの融合



- 開発者が気にしていないところをテストする
 - 性能、障害対応性、操作性、データの組み合わせなど、開発者が気にしていないかどうかを質問してテストする
- 開発者はどこがヤバそうか知っているので、その知恵をフルに活用してテストを行う
 - 開発者が経験したバグパターンを探索してテストする
 - 開発者が難しいと感じているところを集中的にテストする
 - プラットフォームや規格の組み入っている部分を聞いて、集中的にテストする
- 何が大事か、何がヤバいか最初に

User+Q ユーザとQAの融合



- ユーザが大事に思っている品質に注力する
 - 品質モデルや非機能特性をしっかりと考えて、対応するテストを行う
- ユーザを出荷後のQAプロセスに巻き込んでしまう
 - ユーザは自分が使いたい機能やシステムの価値を知っているのだから、リリースを早めていち早く使ってもらって、優先的にバグを潰すべき機能を教えてもらう
 - ユーザは自分が困ることは何かを知っているのだから、優先的に直すバグやバグの致命度を教えてもらう
 - ただしコミュニケーションが未熟な場合は、厳しいクレームになってしまうかもしれない

Tool+Q ツールとQAの融合



- ツールに入れ込まれた知恵を積極的に使う
 - 例えばコードインスペクションツール
- 人間は知恵を使う仕事をし、知恵を使わない作業をツールに任せる
 - フロントローディング型キーワード駆動テスト
 - ✦ 開発時にテストスクリプトライブラリを作ること、GUIベースの自動テストスクリプトを自動生成し自動実行する
 - コールスタックフィンガープリント
 - ✦ QAチームがバグを登録する際、コールスタックの重複を調べることで高確率で同根のバグを発見できる

Dev + QA + User + Tools



- DevQUTが進むとできること
- クラッシュボタン式自動回帰テスト
 - システムに不具合が起きた際にユーザがレポートボタンを押すと、不具合発生操作が自動テストツールスクリプトとして送付され、自動で再現テスト・回帰テストを行う
- ビジネスレベルキーワード駆動テスト
 - 仕様であるユーザストーリーをユーザの言葉で書いたり変更したら、自動テストスクリプトに変換され、自動実行される
 - Cucumber + Open2test + Selenium
 - それなんてBDD

Communitaization度を高めるために



- U以外は社内文化の方向付け、またはどちらかの歩み寄りで向上できる
 - 後半にDev + Qの事例をご紹介します
 - Toolを必要以上に恐れない。フェルミ推定で「もしこのツールがやってくれることを手でやったら」を計算してみる
 - ✦ そもそも手でもやってないことはツールでは解決しない！
- 壁を壊そう！
 - 「なぜ？」と訊けなければ壁がある

Communitaization度を高めるために



- Userとどのようにコミュニケーションするか？
 - 「顧客エンゲージメント」などとも呼ばれる
- ソーシャルメディアの活用
 - 頻繁なフィードバック、協調フィルタリングの形成
- 公式エキスパートユーザの設定
 - Google API Expert等
 - ✦ ベンダ資格と異なり、コミュニティから活躍の度合いに応じて選出される

● 図3 エンゲージメントのレベル

	Low (低)	Medium (中)	High (高)	Highest (最高)
状態	受容	協調フィルタリング	コンテンツ作成	ソーシャル
行動(例)	ブックマークする、タグをつける	投票に参加する、コメントする	ブログで発信する、同じ愛好者のコミュニティに参加する	同じ愛好者のコミュニティを作る

- ザッポス式顧客対応KPI指標の設置
 - 捌いた件数ではなく、時間の長さ(どれぐらい長い時間ザッポスと接してくれていたか?)で測る

出典: 図3 週刊トラベルジャーナル誌 執筆・鶴本浩司

<http://www.marketingvoice.jp/columns/column/090413-01.php>

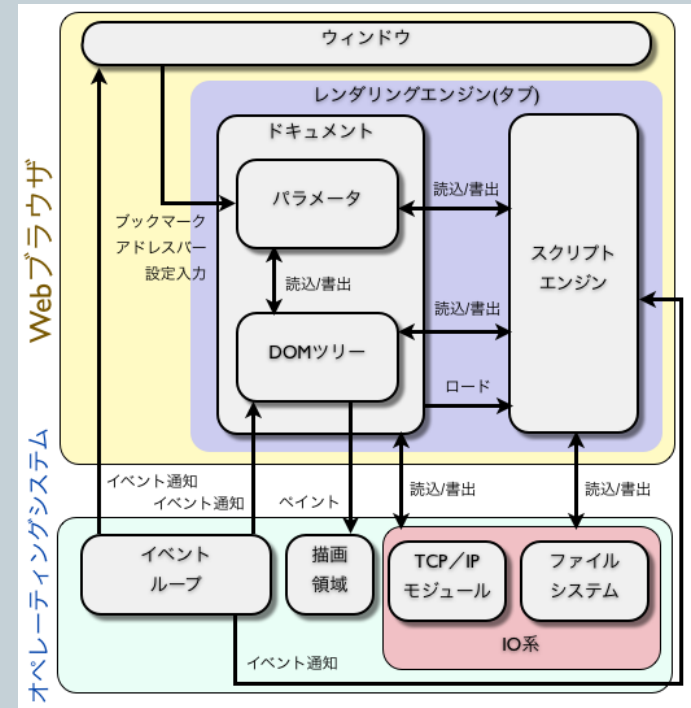
その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013

- ウェブブラウザの“品質”をどうやって共有するか？
(レンダリングエンジン+スクリプトエンジン)

- ポイント

- 入力と出力はそれぞれ基本的に無限
- その無限の入力を更にスクリプティングできる
- 入力は基本的にネットワーク経由
- ユーザ環境にも著しく依存する



その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013



- STEP1 : QAから予め、品質モデルを対象製品に対してある程度選定、咀嚼した形で提示する
 - パーツはウェブブラウザのとある版の例

Changeability

Reliability

Function & Fill Standard

Time and Resource
performance

SDK API Usability

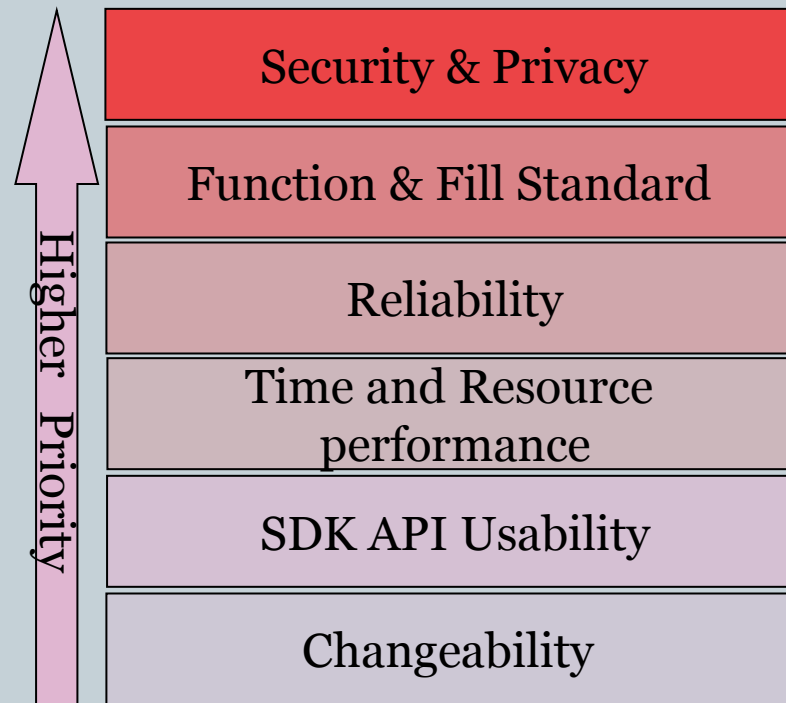
Security & Privacy

その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013



- STEP2 : 開発チームをまじえてその優先度を議論する



その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013



- STEP3 : 上から順に「それが満たされるということは例えばどういうこと？」を議論、共有していく

Security & Privacy

Function & Fill Standard

Reliability

Time and Resource
performance

SDK API Usability

Changeability

Confidentiality?

Integrity?

Availability?

Infinity loop site?

Random site access?

その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013



- STEP4 : 「例えばどういうこと？」(メタ水準)
が固まった時点で、Dev+Qの外側のビジネス上のス
テークホルダーを交えて合意を取る
 - 逆に、これ以上細かくしても判断できないという粒度として
はギリギリの線

Security & Privacy

Confidentiality?

Integrity?

Availability?

Infinity loop site?

Random site access?

その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013



- STEP5 : メタ水準から更に、定量的な水準を導き再度開発チームと合意する
 - これがQAを交えた開発チームが達成すべき“定量的な”ゴールとなる
 - テストと開発が合流したとき(システムテスト工程など)に、ゴールを達成したか否か？を厳密に判断できる

Infinity loop site?



Major browser
crasher site list covered

Random site access?



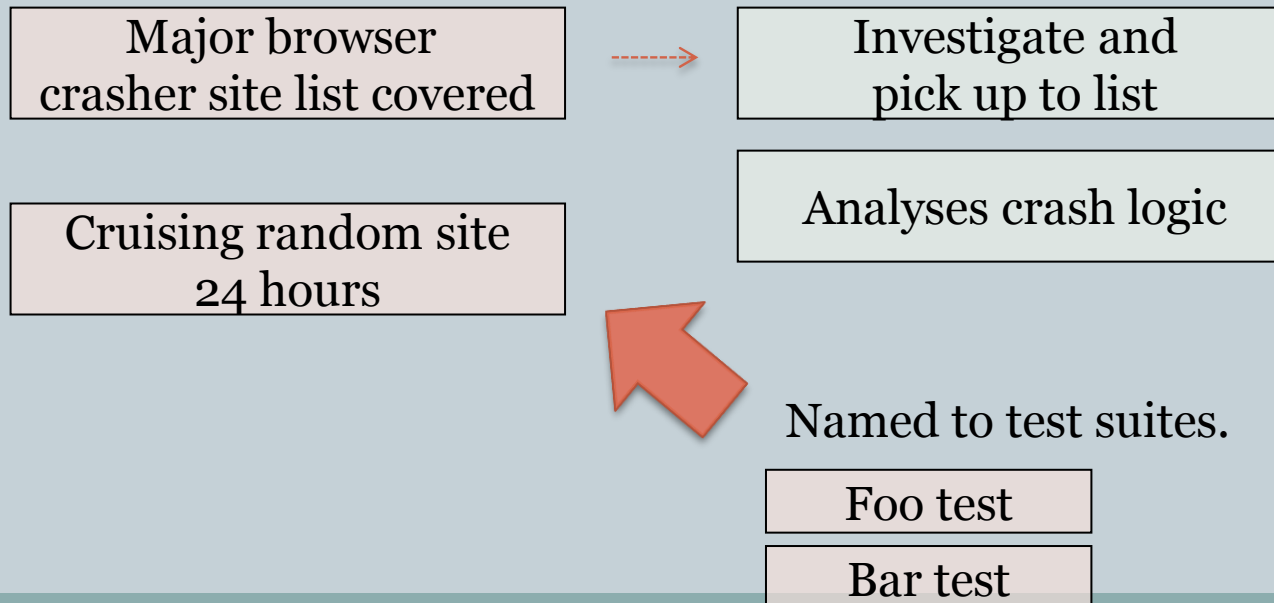
Cruising random site
24 hours

その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQiP2013



- STEP6 : 定量的な水準が合意できたら、開発チームは「製品」を、QAチームはその製品を「できるだけ素早く正確に測れるテスト」を開発する

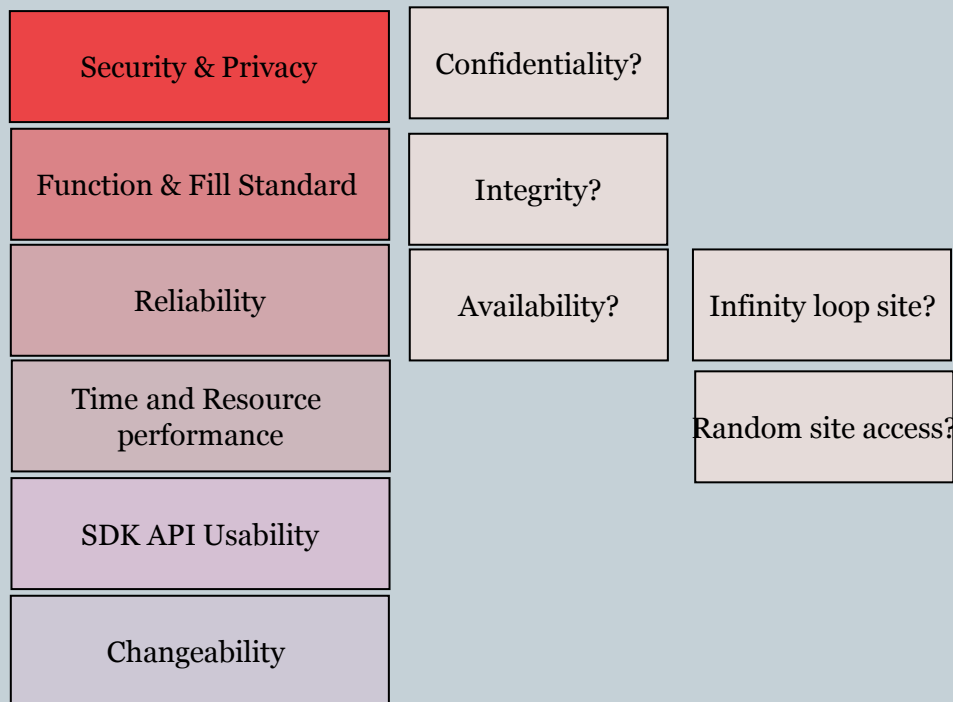


その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

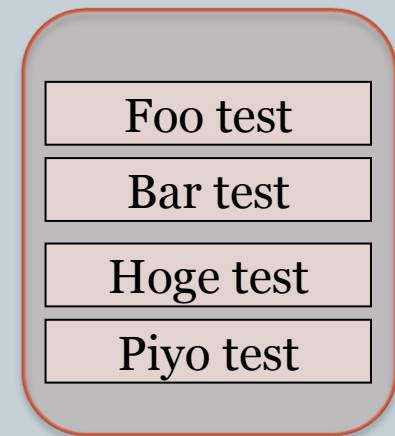
SQIP2013



- STEP7：出来上がった製品に対しテストを実行し、予め設定した水準をどれくらい満たせたかを計測する



Applying Named Test Suites
& judge to each characteristics



その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQIP2013



● 結果と効能

- 開発チーム、QAチーム共に成果物が明確になった
- バグ管理システムの“重要度”を合意に基いて適切に設定できるようになった
- 開発チームに納得感が生まれた
 - ✦ 組織目標にまでしてくれた

● 改善ポイント

- 最後に結合する形式だと間に合わない場合があった
- 品質モデルを対象製品に変換し「例えば」を述べるSkillが意外と高い＝要研修
- 合意を数レイヤで何度も取るためコストが高い

その最上流プロセスとして Dev+Qによる品質モデル定義の実践事例

SQiP2013



- 今後の展望

- 合意を取る部分をシステム化、合理化しコストを下げる
- 前述の合意形成コスト削減と併せてテスト開発部分を加速し、イテレーション型の開発にも適合しやすくする

Special thanks to Yasuharu Nisi
DevQUT part.