

ソフトウェア品質管理の現状と将来

～ソフトウェアテストシンポジウム2008東京 Capers Jones氏基調講演～



2008年1月30日(水)31日(木)、東京・目黒雅叙園にて、『JaSST '08 Tokyo(ソフトウェアテストシンポジウム2008東京)』が開催されました。

JaSST は、Japan Symposium on Software Testing の略称で、NPO 法人 ASTER (ソフトウェアテスト技術振興協会) が主催するシンポジウムです。スタートは2003年から。現在では、ソフトウェアのテスト、レビューなど品質、信頼性に特化した国内最大級のイベントとして、マネジメント、開発、研究などの分野・立場を問わず、ソフトウェア業界全体のテストやソフトウェア品質に関する活発な議論と交流が行われています。

6年目を迎える今回も、さまざまなチュートリアルや成果発表が行われ、2日間で延べ1,700人も参加者が集まりました。その中でも注目を集めたのは、ASTERが招聘した『ソフトウェア開発の定量化手法』など多数の著書を持つソフトウェア品質の第一人者、Capers Jones (ケイパーズ・ジョーンズ) 氏の基調講演です。ジョーンズ氏はその経験と鋭い視点から、ソフトウェア品質管理の現状と将来について提言し、現場の技術者はもちろん、経営者や政府のトップに対し、ソフトウェア品質の向上は人類にとって極めて重要な取り組みであり、品質にかける努力、コストは、結果として会社全体、そして社会に好影響を与えることを訴えました。

ここ500年の人類の歩みの中で 最も大きなリスクを生むもの。それがソフトウェア



ケイパーズ・
ジョーンズ氏

ジョーンズ氏の講演は、まず自身の身の回りで起きたトラブルのエピソードから始まりました。

「1ヶ月前、私は航空機を予約しました。しかしボーイング 747 に搭乗する予定が、予約ソフトウェアの不具合のために、ボーイング 727 に変更されたのです。幸い私の座席はありましたが、小さい飛行機になったことで、多くの乗客は自分の席がなくなってしまいました。また、ここ10～15年の間で、航空業界で起こったトラブルで最大のもは、デンバー空港での出来事です。1年にわたって発着の遅延が続いたのですが、原因は、手荷物システムに関するソフトウェアの不具合だったのです」

このように身近で起こった、しかし放っておけば大きな問題につながりかねない、ソフトウェアに起因するトラブルの実例をジョーンズ氏は次々に挙げていきます。

「金融機関におけるソフトウェア品質の問題もあります。例えばフランスでの70億ドルもの不正取引による損失。ソフトウェアが不正をきちんと検出していれば、これほどの事

態にはなりません。しかし、これは異例なことではありません。世界中を見ればもっと多くの不正が隠れていることでしよう」

さらに、X線の過剰照射や間違った処方箋など、ソフトウェアの不具合が生死に直結する医療関係のトラブルから、保険業界、地方自治体、政府レベルでの個人情報漏えい、過去に起こったチェルノブイリ、スリーマイルアイランドなど原子力発電所をはじめとする公共サービスでのトラブル、そして西海岸、ニューヨークでの大停電。大停電については、発生原因が配線にあるように思われていますが、ここでも原因はソフトウェアの品質であると、ジョーンズ氏は指摘。ソフトウェアの品質の低さが引き起こす問題の実例を挙げます。

なぜこのような事例を挙げるのか。ジョーンズ氏は理由を以下のように述べます。

「ここ500年の人類の歩みの中で、最も大きなリスクを生む要因は、ソフトウェアであると私は考えています。あらゆるトラブルの原因として、まずソフトウェアの不具合が疑われます。そしてここに関わる皆さんは、会社の社長に、政府のトップに、あまり敬意を表されない職業といえるでしょう。もちろん私も含めて。それはソフトウェア開発が他の製造業よりも経営にインパクトを与える大きなリスク要因となるからです。それを理解したうえで、品質を向上させることがプロフェッショナルとして最も重要な取り組みなのです。これは世界レベルで行わなければならない」

品質管理の現状は

非常に憂慮すべきものがある

では、プロフェッショナルたちはどのように取り組んでいかなければならないのでしょうか。

「私の考える品質管理とは、欠陥率を低くして、お客様満足度を高めることです。そのために気をつけることは、①欠陥の事前防止、②除去、そして③バッド・フィックス（バグ修正時の誤修正、および新たなバグの混入）、この3点が基準です」

この3つの基準によって、欠陥を最低限に抑えること。そう考えるととてもシンプルです。しかし、このシンプルなことが難しいのも事実。実際、ジョーンズ氏が提示した資料には、うまくいっているとはいいがたい数字が並びます。

「アメリカでのソフトウェアの品質を見てみましょう。平均で、最終的にファンクション・ポイント（FP：ソフトウェアの規模を測定するための単位）あたり、0.75個の欠陥が出

表1. アメリカにおけるソフトウェア品質の平均値

（データはファンクション・ポイントあたりのエラー発生数）

| エラー発生時期 | 潜在的エラー | 除去率 | 納品・リリース後に顕在化したエラー |
|-----------|--------|-----|-------------------|
| 要求 | 1.00 | 77% | 0.23 |
| 設計 | 1.25 | 85% | 0.19 |
| コーディング | 1.75 | 95% | 0.09 |
| ドキュメント化 | 0.60 | 80% | 0.12 |
| バッド・フィックス | 0.40 | 70% | 0.12 |
| 合計 | 5.00 | 85% | 0.75 |

（ファンクション・ポイントごとのカウントは、コーディング段階のエラーだけでなく、すべてのエラー要因を浮かび上がらせません。）

ています。これは非常に恥ずかしい数字です。そして注目いただきたいのは設計段階までに、85%しか欠陥が除去されていないことです（表1）。これは他の製造業の世界では許されない数字です。それがソフトウェアでは許されています。ベスト・イン・クラスの企業は、最終的に96%の欠陥を除去しています（表2-1）。どの企業もこのレベルを目指すべきです。欠陥除去率62%、1FPあたり3.05個という低品質（表2-2）では成功は収められません。命を預かるような医療機器ならば、99.99%以上の欠陥を除去してほしい。あなたの会社の除去率がどれくらいなのか、ぜひ調べてみてください」

表2-1. ベスト・イン・クラス企業のソフトウェア品質

（データはファンクション・ポイントあたりのエラー発生数）

| エラー発生時期 | 潜在的エラー | 除去率 | 納品・リリース後に顕在化したエラー |
|-----------|--------|-----|-------------------|
| 要求 | 0.40 | 85% | 0.08 |
| 設計 | 0.60 | 97% | 0.02 |
| コーディング | 1.00 | 99% | 0.01 |
| ドキュメント化 | 0.40 | 98% | 0.01 |
| バッド・フィックス | 0.10 | 95% | 0.01 |
| 合計 | 2.50 | 96% | 0.13 |

所見：ほとんどのエラーはシステム・ソフトウェア段階で発見されます。これはSEI CMM® Level3以上に相当します。

表2-2. 低品質企業のソフトウェア品質

（データはファンクション・ポイントあたりのエラー発生数）

| エラー発生時期 | 潜在的エラー | 除去率 | 納品・リリース後に顕在化したエラー |
|-----------|--------|-----|-------------------|
| 要求 | 1.50 | 50% | 0.75 |
| 設計 | 2.20 | 50% | 1.10 |
| コーディング | 2.50 | 80% | 0.50 |
| ドキュメント化 | 1.00 | 70% | 0.30 |
| バッド・フィックス | 0.80 | 50% | 0.40 |
| 合計 | 8.00 | 62% | 3.05 |

所見：ほとんどのエラーは5,000FP以上の巨大なクライアント・サーバー・プロジェクトにおいて発見されています。

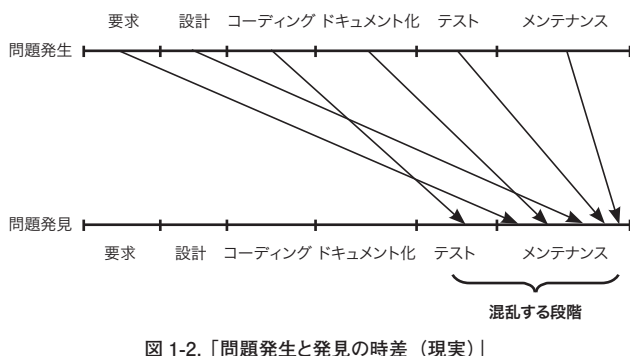
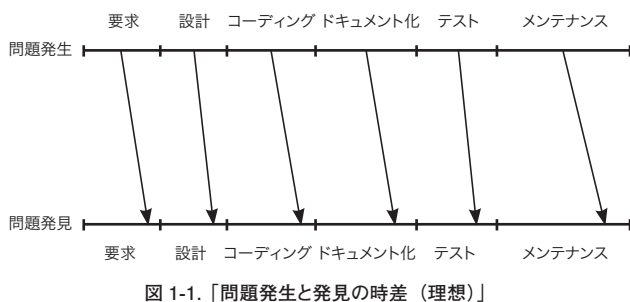
ジョーンズ氏が現在調査をしている1,500プロジェクトのうち、除去99%を超えるものは6件のみ。わずか0.4%です。

「ソフトウェア業界の品質への意識や取り組みを、製造業など他業種・業界と同レベルにすること」を目指すべきとするジョーンズ氏にとってこの現状は見過ごすことができない問題です。

除去率99%を目指すために 現場と経営層が取り組むべきこと

続いて具体的な改善策へジョーンズ氏の話は進みます。高いレベルでの品質管理を目指し、予防、除去を確実にしていくためのメソッドが紹介されました。

「一つは図1-1で示すように、問題の発見、除去を、要求(Requirements)、設計(Design)、コーディング(Coding)、テスト(Testing)、メンテナンス(Maintenance)の各段階で発見するのが理想であるという考え方です。「問題はテスト・フェーズで発見すれば良い」という考え方が一般的かもしれませんが、テスト段階で発見されたものは発生原因があいまいで特定できないことも多く、また、テスト自体がしっかりできていないと多くの問題を見逃してしまう可能性が高まります。結果、コストと時間の損失につながってしまうのです。図1-2のように、要求や設計での問題が後で発見されるのは“プア・クォリティー(粗悪な品質)”に陥る大きな原因となるということです。



各フェーズでの予防、除去を行うメリットについて、ジョーンズ氏は図2のグラフを使ってわかりやすく解説しました。

「健全な品質管理というのは、コーディングまでの段階にコストをかけていきます。一方、不健全な場合は初期段階では欠陥除去コストは小さく、後で問題が発見されることによってコストが増大していきます。これがソフトウェア品質管理の経済学です」

最初の6カ月、12カ月にコストをかけることに対してためらうケースが多いようですが、しかし、初期段階で問題を見出し、改善していくことによって品質の信頼性は高まり、結果、10年、15年後までコストは低い水準に抑えられるのです。一方で、不健全なものはリリース後、使えば使うほど、メンテナンスのコストや不具合の修正といった膨大なコストが発生します。場合によっては、訴訟の対象にさえなりかねません。

高品質なソフトウェアを作ることはとても難しく、特効薬はない、というのがジョーンズ氏の考えです。したがって高い除去率を目指すためには、品質向上のために良いとされるあらゆるテスト手法、検証、取り組みを、多数组み合わせて着実に実施するべき、と説きます。そのためには、個人の努力による部分に加えて、長期にわたる組織的な取り組みが必要であり、経営層、管理層も、ここにコストを投下する決断が必要です。

「一人の開発者ができることには限界があります。確かに、経験を積むことで個人のデバック精度は高まります。しかし、個人では2つに1つ、50%しか発見できません。組織的なノーマル・テストを組み合わせると75%、さらに厳格なレビューを組み合わせれば99%を超える可能性があります。この段階になれば、開発スケジュールが短くなり、運用コストが下がる。正確なクォリティーの効果測定で、品質が上がり、生産効率も最大化できます。お客様満足も上がり、運用コストが下がり、メンテナンス、テスト、QAにかかわる人のモチベーションが上がる。そして訴訟になることも回避できます」

品質管理にける努力、コストは、結果として会社全体に、そして社会に大きな好影響を与える。そうなればソ



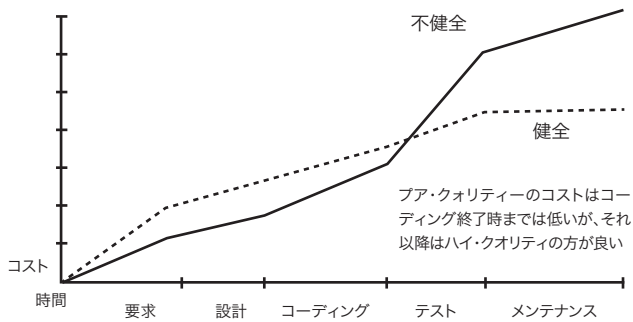


図2. 品質がソフトウェアのコストに与える影響

ソフトウェア技術者の地位も上がる。これが、ジョーンズ氏の訴えです。

そして、日本に期待するもの

知日派でもあるジョーンズ氏。ソフトウェアの品質管理における日本の役割、期待についても熱く語りました。

「日本に来るのは3度目。この美しい国に来られて幸せです。今回は、京都、奈良、下田、前は宮本武蔵のお墓を訪ねました。平家物語、葉隠れなど日本の歴史も興味をもっています。

日本は世界のどの国と比較しても優れた技術を持っています。自動車、テレビ、電子技術、など日本の製品は世界でトップレベルです。しかし残念ながら、ソフトウェアに関してはそのレベルではない。私はここで警鐘を鳴らします。ソフトウェア分野で、より高いレベルを達成している国はインドです。インドでは、日本が製造業で実現している品質レベルをソフトウェアに求めています。日本が経済大国であり続けるためには、インドを見習い、ソフトウェアの品質を高めるべきです。ものづくりの力をソフトウェアにも生かすべきだと思います。みんなでそれをやっていきましょう」

1970年代からソフトウェアの品質管理に携わり、自身がソフトウェア・エンジニアの一人でもあったジョーンズ氏。現在でこそスポットライトをあてられているソフトウェア品質管理の大切さを当時から認識し、最適な答えを探求してきたのです。何十年もの経験の中で編み上げられてきた考え方が現在のソフトウェア品質管理にも多大な影響を与えているという事実。ジョーンズ氏が日本の技術者に向け、この不変ともいえるメッセージを語るとき、淡々とした口調の中にも熱い想いが溢れていました。最後にシンポジウム最終日に行われたスペシャル・セッションでの一言をご紹介します。

「次の世代を担う技術者たちに、将来にかける期待



をお聞かせください」

という質問に対して

「宮本武蔵の五輪書から引用できることも多いと思います」

との回答。ジョーンズ氏の真意はどこにあるのか？探ってみてはいかがでしょうか。

今回取材を快諾いただいた ASTER とソフトウェアテストシンポジウムの今後ますますの発展をお祈りします。

【参考文献】

- [1] Capers Jones氏の基調講演資料
<http://www.jasst.jp/archives/jasst08e/pdf/A1.pdf>