

2012年度 第1回特別講義 レポート

日時	2012年5月11日(金) 10:10~12:00
会場	日本科学技術連盟・東高円寺ビル 地下1階講堂
テーマ	「ソフトウェア品質技術研究～あなたを変える、組織を変える、社会を変える～」
講師名・所属	野中 誠氏(東洋大学)
司会	SQiP 研究会運営委員長 秋山 浩一氏 (富士ゼロックス株式会社)
アジェンダ	<ol style="list-style-type: none">1. ようこそ、SQiP 研究会へ2. ソフトウェア品質技術が生み出す「価値」3. ソフトウェア技術者への期待:品質技術の研究4. ソフトウェア品質技術研究の例:メトリクスの基礎と応用研究
アブストラクト	<p>ソフトウェア品質技術は、さまざまな「価値」を生み出す有用な技術です。すなわち、顧客にビジネス価値をもたらし、安心・安全な社会を実現し、組織の強みを増幅させ、そして、あなた自身の成長につながります。</p> <p>しかし、ソフトウェア品質技術にはまだ未成熟な部分があり、研究の余地が多く残されています。より優れた「価値」のためには、ソフトウェア技術者である皆様が、現場のニーズを踏まえて品質技術を考え、実践することが求められます。</p> <p>この講義では、SQiP 研究会のキックオフとして、まずは研究員の皆様に期待したいこととお話しします。また、ソフトウェア品質技術の一つであるメトリクスを取り上げ、その基礎と、定量的管理への応用について、私の研究成果や技術動向などを交えてご説明します。</p>
<講義の要約>	
◆ようこそ、SQiP 研究会へ	
<p>SQiP は、実践的で実証的なソフトウェア品質技術・施策の研究・普及を目的として設置された、ソフトウェア品質向上のための推進組織である。お金を払った人だけが関われる閉じた世界ではなく、「ソフトウェア品質を良くしたい」という思いを共有する方なら誰でも参加できるオープンな場で、コミュニティもある。</p>	
<p>SQiP の「P」には、高度な教育訓練を受けた専門的職業を意味する「プロフェッション (Profession)」という語をあてている。ソフトウェア技術者は聖職者・弁護士・医師に次ぐ「第4のプロフェッション」となるべく自覚を持って行動する時期にきている。</p>	

SQIP 研究会は、その先にある、SQIP シンポジウムや世界ソフトウェア品質国際会議(WCSQ)などの広範なソフトウェア品質活動への「入口」である。また、オープンイノベーションを起こせる場でもある。オープンイノベーションとは、企業内部と外部のアイデアを有機的に結合させ、価値を想像することである。ソフトウェア品質もオープンイノベーションが必要である。自社の取り組みや情報を社外には出せないため論文が書けない、という話を聞くが、取り組みを社外に出したところで、他社はそう簡単には真似できない。実際、組織に根付かせ、効果がでるまで長い歳月がかかる。研究会は外部のアイデアを得るよい場である。是非ベストプラクティスの共有をしていただきたい。

◆ソフトウェア品質技術が生み出す「価値」

ソフトウェア品質に対する取り組みは、顧客に価値を提供する全社的な取り組みである。顧客にどのような価値を提供するかを定義し、その価値により高い顧客満足度を維持し続けるために、組織として、また個人単位で取り組む「必要な活動」を定義し、実践する。必要な活動として、顧客価値を提供／創造できるソフトウェアは何か？を考える。

開発スピードを加速させるために、欠陥の検出・除去・混入予防に必要な活動を定義する。単にレビューやテストに時間をかけるだけでなく、そもそも設計をどうするか？を考えることで混入予防につながる。また、混入をさせてしまうメカニズムはいろいろなところにある。単に作りこみ過程だけでなく、組織体制などにも問題があるだろう。品質部門と開発部門の役割が分かれ過ぎていると、「あとはやってくれるだろう」と甘えがちになる。

この段階のテスト、レビューでは、どういう観点で行うべきだろうかを考えることも必要である。

価値提供の結果を評価し、欠陥に学ぶことで、意味のあるフィードバックができる。一連のフィードバックを積み重ねることで組織は賢く、強くなれる。賢く、強くなった組織は、幸せになる。

品質にしっかりと取り組めば、開発コストを下げられる。安易な品質コストの削減は、取り返しのつかないところまで組織能力を衰退させる。

◆ソフトウェア技術者への期待: 品質技術の研究

東洋大学の「社会人カノート」に記載されている、社会人心得と技術者心得についての紹介。

社会人心得には、発信力、傾聴力が大切であることと、柔軟性、状況把握力、規律性、ストレスコントロール力が必要であることが書かれ、技術者心得には、手と機械と心を一体化させることや、知識と智慧を身につけること、世界観を持つこと、心で現象を見ることの大切さなどが書かれている。

ソフトウェア品質技術・施策を研究するにあたり、「社会人心得・技術者心得」を心に留めてほしい。

研究会での皆さんのアイデアや貢献を、SQuBOK へ位置づけたい。

◆ソフトウェア品質技術研究の例:メトリクスの基礎と応用研究

なぜメトリクスが必要なのか?「よそが測っているから」ではなく、「必要な情報」を得た上で意思決定をするために必要である。「必要な情報」として、主観的情報だけではエビデンスとして説明責任が果たせない。メトリクスを定めることで、必要な情報を効果的に一貫して収集でき、組織やメンバーに説明可能な根拠情報として利用できる。

しかしメトリクスは万能ではない。実世界で測定できることは、測定したい概念のごく一部にすぎない。メトリクスの限界を理解した上で“*Informed decision*”に役立てる。

測定には信頼性と妥当性が求められる。具体的には、信頼性は「誰が何度測定しても同じ結果が得られるか」、妥当性は「測定したい概念を正しく捉えているか」「意思決定や予測に役立つか」「測定したい概念を十分カバーしているか」が挙げられる。また、メトリクスを用いて何を評価したいのか、目的を意識すること。実体把握に用いるなら、目的によって要求される精度は異なる。予測や計画に用いるなら、精度は粗くてもよいが手順の再現性を重要とする。評価に用いるのなら、高い精度が必要となる。

上流工程までの欠陥除去率を評価しようとして、上流工程までの欠陥除去数を欠陥総数で割ると、上流工程での欠陥除去率が実際より低くなる場合がある。これは下流工程で混入した欠陥数も総数に含めてしまった結果である。下流工程で除去した欠陥については、混入工程を把握しておく必要がある。また、目標値を意識するあまり、実際の欠陥の粒度を変えて件数を増やし数字合わせをするというような、期待外れな行動にでてしまうことがある。欠陥の粒度のガイドラインを定めておく必要がある。欠陥を管理している組織は多いが、欠陥の粒度定義と測定精度の確認を行なっている企業は少ない。

欠陥の数え方は、組織によって様々である。また、欠陥1件の粒度について、IEEE 標準やISO規格では特に言及していない。ここでソフトウェア欠陥1件の数え方の例を挙げる。開発プロセス全体で欠陥の粒度に一貫性をもたせ、欠陥除去工程のパフォーマンスを正しく評価したい場合、プロダクトの原因部分で集約して1件と数える。プロセス単位で集約してしまうと、何もかも1件として集約されてしまう可能性が高まる。あくまで見逃した欠陥はいくつか?を考える。

欠陥識別のタイミングをどこにするか?非公式レビューにおける指摘も欠陥数に含めるのか?欠陥識別のタイミングについて「原則」を定めることは難しい。

欠陥測定の信頼性を高めるには、開発プロセスと欠陥メトリクスを紐付けることも重要である。欠陥の混入と除去に関連するアクティビティを定義する。

欠陥の数を扱う前に考えなければならないことは、「何に関する欠陥なのか」と「どこまでを欠陥とするのか」である。欠陥には計算ミスなど現時点で修正が必要な機能性欠陥と、コードの読みやすさなど将来の機能性欠陥につながる発展性欠陥がある。欠陥除去のフロントローディングを議論するときは機能性欠陥に着目する。また、発展性欠陥は、早い段階で除去する。欠陥のタイプに応じて、どのタイミングで指摘し、欠陥除去するべきかを考えることが重要である。

DRE(Defect Removal Efficiency)は各工程またはプロセス全体の欠陥除去率で、欠陥除去能力の低い工程を特定し重点的に改善する目的で用いる。PDCE(Phase Defect Containment Effectiveness)はフェーズ欠陥阻止率(各々の欠陥除去工程で除去すべき欠陥をどれだけ除去できたか)で、見逃した欠陥に着目し、レビュー技法やチェックリストの改訂を行う目的で用いる。どちらも終わったプロジェクトに対して測定する。プロジェクト中に介入し判断したい場合、欠陥除去率の分母は予測値になる。

ソフトウェア欠陥数の予測に、欠陥数と相関の高い変数を用いて予測精度を高めることは可能であるが、因果関係の有無を確認しないまま用いている場合もある。なぜ相関が高いのかまで分析して用いることが大事である。

有用そうな変数について、箱ひげ図を描いて欠陥数に影響する要因を探ってみる方法もある。ただし、相関係数を高くすることにこだわりすぎて、変数の入れ替えに専念してしまっはいけない。

fault-prone(欠陥が含まれていそうな)モジュールの予測のイメージを説明すると、コーディング／単体テストが終わった時点で、ソースコードから測定可能なプロダクトメトリクスやプロセスメトリクスを用いて、機能テスト／システムテストで欠陥が見つかりそうなモジュールに対してレビューやテストの優先度／重要度を決め、効率／効果を高める。

欠陥修正の有無を何らかのメトリクスで予測する事例の紹介。

データ分析により、複雑度重み付き規模(WMC)、結合度(CBO)、凝集度の欠如度合い(LCOM)の3つの変数により、欠陥修正の有無を高い精度で予測できた。100%ではないが、どのモジュールに対し欠陥が発生したのか、ある程度説明できるものがでてる。

メトリクスをとる上で大事なことは、分析をしたところから踏み込めること。レビューやテストなどのアクションや品質に繋がられること。そこから根っこのしっかりした議論ができる。

John Maynard Keynes の言葉に「精密に誤るよりも、漠然と正しくありたい」というものがある。メトリクスを取り組みで忘れてはいけない基本姿勢である。取り組んでいるうちに、精密に相関係数を見だしたり、相関係数が高くなることにこだわったりしてしまうが、それが狙いではない。あくまで「このメトリクスでこういうことが得られたから、ではこういうアクションを取ろう」という正しい意思決定に役立つことが目的である。

◆質疑応答

Q. いろいろなメトリクスをとろう、という活動を、これからはじめる会社もあると思う。

その場合、どこから手を付けるのがよいか？なにを測るのがよいか？

メトリクス初心者用のおすすめのメトリクスはあるか？

A. まずはテストでみつけた現象や、レビューの指摘数を測り、「その件数が少なくなること」を見ることから始めてみてはどうか。

Q. 欠陥の数え方について、重み付けがあると思う。もっと複雑ではないか？

同じ「1個」ではなく、どこでみつけたか、で、重みが違う。(あとで見つけると大変だから今見つけてよかった、というものは重みがある。)

A. DRE を用いて見てみると、圧倒的に上流で除去されるほうがインパクトが抑えられるのがわかる。しかし、重みが表に出ない。道理を解釈して数字を見る。

あるいは、実際いくらかかったのか？を計算する。システムテストで出たバグについて、工数を計算することでインパクトの大きさが明らかにでてくる。そのような値と紐付けるという手もある。

Q. マネージャに対して、メトリクスとは何なのか(メトリクスは何のためにとるのか)を、説明するには？

A. 意思決定に役立たせたい。これがベースだと思う。例えば、なにがどうまずいのか？わかりやすく伝えられる。過去の統計からすると、絶対無理だろう、ということが示せる。

そのために、意思決定に使えるようなものをとる。何を見ておけば、説明できるだろうか？を考えておく。

<講義の感想>

欠陥の測定を中心に、様々なメトリクスの測りかたや測る上で留意することを紹介いただき、大変参考になりました。また、研究を行う上で、目標や目的を意識することと根拠や効果を示すことの大切さを改めて気付かせてくださる講演内容でした。情報を集めて整理することで満足してしまったり、メトリクスをとることに必死になってしまったりで、本当は何を示したいのか、どんな成果を伝えたいのか、そのあとどうしたいのか、ということを忘れがちになると思います。今後研究を進めていく中で、常に意識していこうと思います。