プログラム開発の品質を向上

開発効率を向上させ、魅力あるゲームづくりに注力
DeNAが実現したマンパワーに依存しないソフト品質管理

株式会社ディー・エヌ・エー
Japan リージョンゲーム
事業本部
開発基盤部 部長
惠良和隆氏

ビジネス環境の変化により開発現場の作業工数が肥大化

近年、あらゆるビジネスにおいて複数のシステムを連携させ、リアルタイムにデータを処理し、サービスを迅速に展開することが求められている。こうしたビジネスサイドの要求に応える開発現場では、短期間で膨大なコードを書かなければならず、これがバグの増大を招いている。特に、競争が激しくリリースサイクルの短いモバイルゲーム業界は深刻な課題に直面している。

「ファイナルファンタジー レコードキーパー」や「戦魂 -SENTAMA-」などのモバイルゲームで業界をリードする DeNA も、システムの複雑化やコード量の増大、エンジニア不足によるコード品質の低下に悩まされていた。

「フィーチャーフォンからスマートフォンへの移行により、モバイルゲームの開発現場に劇的な変化が起きました。ブラウザゲームからネイティブアプリへゲームの主戦場が入れ替わってしまったのです。モバイル端末のチップやグラフィックは、フィーチャーフォンとは比較にならないほど高性能で、ネイティブアプリ開発はその性能を最大限に生かすために、コンソールゲーム（家庭用据置型や業務用ゲーム）並みのコードの量と質が求められるようになりました」と DeNA の惠良和隆氏は昨今の環境の変化を話す。

さらに、モバイルゲームはリリースして終わりではなく、コンテンツや機能を追加し続けなくてはならない。そのため、運用期間が長くなり、これが開発現場の負担になっているという。

「初期開発の段階では、スキルレベルが高いメンバーをそろえているのでコード品質を担保できるのですが、運用段階でメンバーが入れ替わるとスキルレベルがバラつき、結果的にコード品質が低下して不具合が起きやすくなります」と惠良氏。

いかにして、メンバーが入れ替わってもコード品質を落とさないようにするか。これがモバイルゲーム開発の現場における課題となっていた。

マンパワーによる検証ではコード品質を担保しきれない

プログラムの品質を担保する最良の方法は、最初から安全なコードを書くことだ。しかし、どこの企業でも安全なコードが書けるシニアエンジニアの数は限られており、すべてのプロジェクトで品質を上げることは難しい。スキル不足をカバーするには、シニアエンジニアの書いた安全なコードを手本にすればいいのだが、多忙なゲーム開発の現場では手本を書く余裕さえもない。

「当社もシニアエンジニアのコードをお手本にしながら、タイトルコードの実装量削減や積極的なコードレビューの実施、微妙なコードを残さないようリファクタリング工数をしっかりとるなどの取り組みをしてコード品質の向上に努めてきました。しかし、開発が拡大しエンジニアが 10 人 20 人規模になってくると、スキルレベルの低いエンジニアもメンバーに入れざるを得なくなり、コード品質の低下が避けられなくなりました。これまでは QA（Quality Assurance：品質保証）期間を長くとることで対処してきましたが、もはやマンパワーに依存した検証は限界

に達しており、機械的に不具合を検出する静的コード解析ツールの導入が不可欠と判断しました」と恵良氏は、開発現場の課題を説明する。

同社では、以前からオープンソースの「Cppcheck」を使っていたが、機能不足があったことと解析結果が見づらいこともあり、もっと高機能なツールを必要としていた。また、統合開発環境の「Visual Studio」や「Xcode」にも付属のコード解析機能はあったが、これは特定プラットフォームへの依存が強くモバイルゲーム開発で使うには厳しいと判断。Java 対応の商用ツール「Klocwork Insight」も検討したが、Objective-C などモバイルプラットフォーム向けの言語を一部サポートしていなかったため、最終的に同社の求める要件をすべて満たしていた日本シノプシスが提供する静的コード解析ツール「Coverity」を選択した。

Coverityがプロダクトの品質とエンジニアのスキルを向上させる

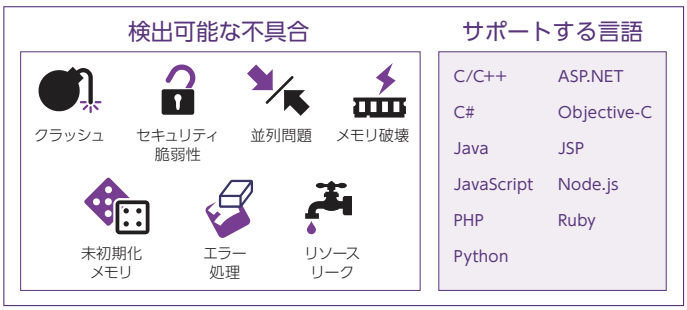
恵良氏は前職のゲーム開発会社で Coverity を使っていたため、個人的には製品の特徴を理解していたが、モバイルゲーム開発に適しているかどうかは未知数だったため、あらためて「DeNA の基準」で評価を行った。

そして、実際のゲームタイトルや内製のライブラリ、サーバーコード、クライアント SDK などを解析した結果、単なる不具合検出にとどまらず、ソースコードを見ながら不具合の起きるパスを確認できることや、人手では発見することが難しいクリティカルな不具合まで検出できることがわかり、現場のエンジニアから「ぜひ使いたい」との声が上がった。

「ROI の観点で行った検証でもスキルレベルの低いエンジニアが増えてもコードの劣化を抑制しプロダクト品質を向上させられることや、ミドルウェアやサーバーなど他の社内プロジェクトも解析対象にできるのでコストパフォーマンスが高いこと、どういうコードを書くと不具合が起きるか指摘してくれるのでエンジニアの教育効果が高いことなど、全般的にポジティブな評価となったので、『Coverity』を導入しました」（恵良氏）

現在、同社では C/C++ 実装のゲームタイトルに加え、独自に開発したゲームエンジンならびにオーディオエンジンと、リアルタイム通信サーバーの静的コードを Coverity で解析している。

「現場での運用は、解析結果にコミットするだけでは意味がないので、git blame コマンドを使って『誰が変更したコードで不具合が起きたのか』まで追跡し、担当者をアサインするところまで行います。もちろん、不具合はそのコード修正が原因とは限りませんが、『誰も関与していない』状態は避けたいので、必ず担当者をアサインするようにしています」と恵良氏は Coverity の運用方針を説明する。



Coverityは面白いゲームを開発するために不可欠なツール

Coverity の導入から約半年。恵良氏は、まだ定量的な効果は示せないと前置きしたうえで「Coverity の導入により、人手をかけずに不具合を検出できるようになり、修正時間が劇的に減りました。不具合の修正時間が短縮されたことにより、ゲームを面白くするという本質的な作業にエンジニアの貴重な時間と情熱を振り向けられるようになりました。バグ修正に時間も体力も削られて面白いゲームがつくれないようでは本末転倒ですから、その効果は非常に大きいですね」と導入効果を話す。

ゲーム開発の現場は業務システムの開発とは異なり、面白くするためなら仕様変更もコードの書き換えも当たり前なので、不具合が起きやすい特殊な環境といえる。不具合の発生が避けられない以上、クリティカルな不具合でも自動検出してくれる Coverity はゲーム開発の質を担保するうえで不可欠な存在となる。

「私自身の体験談ですが、エンジニアは不具合の検出が担保されるだけで新しいコードを書く際に感じるストレスがかなり減るんです。また、Coverity はプロジェクトで出した不具合を数字で示してくれるのですが、この数字を減らすことがスキルアップの実感につながり、『次こそ不具合ゼロを目指す』というモチベーションにもなるんですよ。そういうメンタル面の効果も、案外大きいと思います」（恵良氏）。

今後の展望について恵良氏は、Coverity で検出した不具合の修正率を 100％にすることが目標と話す。これは検出された不具合をすべて直すという意味ではなく、結果的にコードの修正を行わないと判断するケースも含めて、不具合として放置されるものをなくすという意味である。

さらに、Coverity から収集した不具合件数とコード量のデータから相関関係を分析し、不具合が増加しているときにはエンジニアの体調不良やタスク過多になっているかもしれないなど現場の状況を把握するための定量的な情報を得られるため、現場のマネジメントにも使っていきたいと将来の展望を話す。

モバイルゲーム市場は、急成長を遂げており 2018 年にはコンソールゲーム市場を超える規模になると予想されている。この成長市場を勝ち抜くには、良質なゲームタイトルを短期間でリリースすることが求められる。こうしたニーズを満たすには、人手を介さず高精度なコード解析を行うことができ開発期間の短縮に貢献する Coverity は欠かせない存在といえる。グローバル規模で市場の拡大が続くゲーム業界において、Coverity は今後ますます存在感を高めていくに違いない。

SYNOPSYS®
http://www.synopsys.com/JP2/Software
日本シノプシス合同会社 営業本部 ソフトウェア インテグリティ グループ
〒158-0094 東京都世田谷区玉川2-21-1二子玉川ライズオフィス
TEL: 03-6746-3600 Email: sig-japan-sales@synopsys.com