

繰り返し型の効率的なテストを実現するテスト活動データ分析技術

近年、コストを抑えながらソフトウェアを早期にリリースすることへの要求が増えています。そのため、開発全体のコストに占める割合が大きく、ソフトウェアの一定の品質を確保するために必要となるソフトウェアテストは、開発におけるQCD (Quality, Cost, Delivery) の要といえます。本稿では、NTTソフトウェアイノベーションセンタ (SIC) が研究開発を行いOSS化した、テストを飛躍的に効率化する技術について紹介します。

たんの 丹野	はると 治門	きりぬき 切貫	ひろゆき 弘之
かわぐち 川口	たかひろ 敬宏	たじま 但馬	まさき 将貴
おいぬま 生沼	もりひで 守英	むらもと 村本	たつや 達也

NTTソフトウェアイノベーションセンタ

ソフトウェアテストの重要性

ウォーターフォールモデルによるソフトウェア開発は図1のように行われます*。テストで抽出できなかったソフトウェアの不具合はそのままユーザの元へリリースされてしまうため、ソフトウェアの品質確保においてテストは欠くことのできない大事な工程です。しかし、これらを手動ですべて行おうとすると大きなコストがかかります。また、近年ではユーザのニーズの変化や、動作環境となるソフトウェア、ハードウェアの進化のスピードが早いと、必要に応じて素早くソフトウェアを改良し短期で繰り返してリリースすることも求められています。しかし、一定の品質を確保しながらソフトウェアのリリースを繰り返すためには、新規部分のテストに加

えて、新規機能追加等による既存機能の劣化が引き起こされていないかを確認するためのテスト (回帰テスト) を、リリースのたびに旧機能全体に対して行う必要があります。これにも大きなコストがかかります。NTTソフトウェアイノベーションセンタ (SIC) では、ソフトウェア開発におけるQCD (Quality, Cost, Delivery) の要となっているテストを変革し、ソフトウェア開発の生産性を飛躍的に向上させることをめざしています。

ソフトウェアテストの現状

ソフトウェアテストの目的は、ソフトウェアが正常に振る舞うことを確認したり、ソフトウェアの不具合数を減らしたりすることです。テスト工程における主要なタスクはテスト設計、テスト実施となります。テスト設計では、行うべきテストのバリエーションの洗い出しを行い、テストケースを網羅的に設計し、各テストケースそれぞれについて具体的にテスト実施可能な手順への詳細化や、自動実施用のスクリプト作成を行います。そしてテスト実施では、テストケースごとに、入力データを与え、ソフトウェアを動作させ、それぞれのテストケースでソフトウェアの振る舞いが期待どおりであるかを確認します。現状のテストでは大きく3つの課題があると考えています。

1 番目の課題は、従来の事前計画した指標値に基づいた網羅的かつ詳細なテスト設計には大きなコストがかかるという点です。品質向上をめざして網羅性を追求すると膨大な時間がかかります。加えて、テストの再現性や監査性の確保のために、各テストケースについて、テスト手順など詳細に記述するような重厚長大なテストケース表の作成にも大きな稼働がかかります。

2 番目の課題は、テストの実施には膨大な稼働とスキルを要するという点です。多数あるテストケースそれぞれについて、手順に沿って、テスト対象アプリへの操作や入力、そして結果の

作成にも大きな稼働がかかります。

* 図1はウォーターフォール開発におけるプロセスとなっており、アジャイル開発ではこのようなプロセスを短期で繰り返すこととなります。



図1 ソフトウェア開発プロセス

確認を行う必要があります。手順どおりにすべてを実施していくことに多大な時間がかかります。

3番目の課題は、短期で繰り返しリリースを行うために必要となる回帰テストの自動化にコストがかかる点です。JUnit, Seleniumなど、テストを自動実行するための既存のフレームワークやライブラリは多く存在しますが、自動実行のためにはスクリプト作成が必要となり、スクリプト作成に大きな手間がかかります。そのうえ、スクリプトは一度作成したら終わりではなく、テスト対象ソフトウェアの修正に合わせてスクリプトを修正するという保守作業にも大きな手間がかかります。

めざす世界

SICでは、多様化・曖昧化するビジネスの要件とビジネスの進化スピードに対応した超高速開発の実現のため、人の作業の一部を代替・超越するAI

開発技術を確立し、AIと人が協働して開発を行うというアプローチをとっています。超高速開発におけるテストについては図2に示すような世界をめざしています。

私たちのアプローチでは、闇雲に網羅性を追求せず、テストすべき個所を選択し、そこに集中します。そして、どこへ選択・集中するかについては、テスト実施状況・結果のデータを収集、分析して逐次判断します。これにより、従来の網羅的なテストの課題を解決し、テストの飛躍的な効率化を実現します。収集したテスト実施状況・結果のデータに詳細なテスト手順も含まれるため、テストの再現性と監査性も担保することができます（1番目の課題の解決）。加えて、収集したテスト活動データを活用し、保守しやすいテストスクリプトの自動生成も行うことで、回帰テストの自動化を容易に行えるようにし、ソフトウェアを改良したあとに迅速なリリースを可能にしま

す（3番目の課題の解決）。

そして、最終的には、大量に蓄積したテスト活動データを解析することで、過去のテスターの知見を再利用し、テスト実施中に繰り返し行われる手順については自動実施による補助を行うことに加え、テスト品質を向上させるためにテスト推薦を行うことで、テスト実施についても大幅に省力化することをめざします（2番目の課題の解決）。

今回、本稿では、NTTグループ内でも多く開発されている、Webアプリケーションをフロントエンドとして持つ業務アプリケーションにおける、効率化のニーズが高い結合テストを対象とした技術「LatteArt」が、どのように1番目と3番目の課題を解決しているかについて説明します。

繰り返し型の効率的なテストを実現するテスト活動データ分析技術LatteArt

業務アプリケーションは、多くのユー

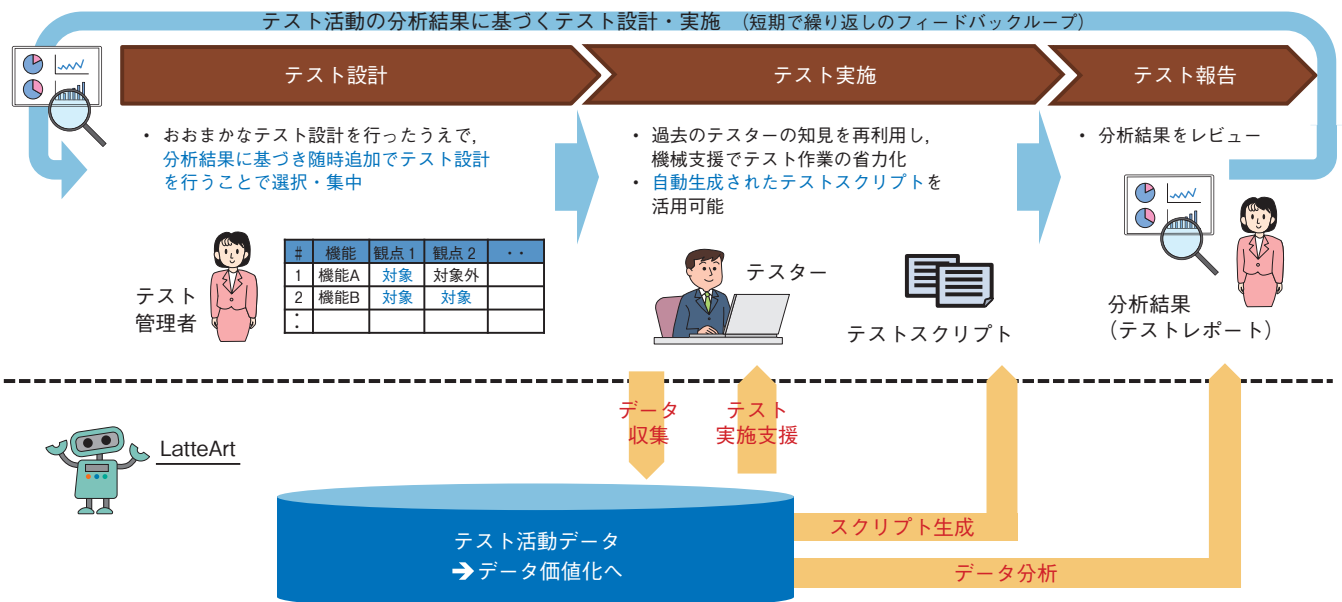


図2 めざす世界

スペースシナリオ、機能、画面があり、各画面における入力のパターンも多くの組み合わせが存在し、従来の網羅的なテストには大きなコストがかかります。なんらかのモデル（例えば、ソフトウェアの設計モデル）から網羅的なテストを自動設計することでテスト設計を支援するツールも存在しますが、ソフトウェアの設計モデルの作成という追加のコストが発生してしまうという課題がありました。また、Webアプリケーションのテストを自動化するためには、画面の操作を自動で行うテストスクリプトの作成が必要となります。このようなテストスクリプトは、例えば、既存のキャプチャ&リプレイツール（例：SeleniumIDE）を使うことで、高いスキルがなくてもスクリプトを作成することはできますが、スクリプト作成の時間はかかりますし、キャプチャ&リプレイツールで記録したテストスクリプトはモジュール化されていないため、修正しづらく保守性が低いという課題もあります。

このような課題を解決するため、私たちは、繰り返し型の効率的なテストを実現するテスト活動データ分析技術LatteArtの開発を進めてきました。LatteArtには以下のような特徴があります。

- ① テスト活動データの収集：テスターのWebアプリケーション画面上の操作ログ、スクリーンショット、およびテスターがテスト実施中に入力したテスト目的、バグ、気付きなどを自動で収集し、テスト活動データとしてデータ化します。
- ② テスト活動データの分析：自動で収集されたテスト活動データを

分析し、さまざまなデータモデルで可視化することが可能です。例えば、図3、4に示すようなシーケンス図、画面遷移図といったモデルで可視化することが可能で

す。これにより、随時人がテストの充分性を確認し、必要最低限のテストを補うことで効率的なテストを実現できます。加えて、テスト活動データに基づき、テスト

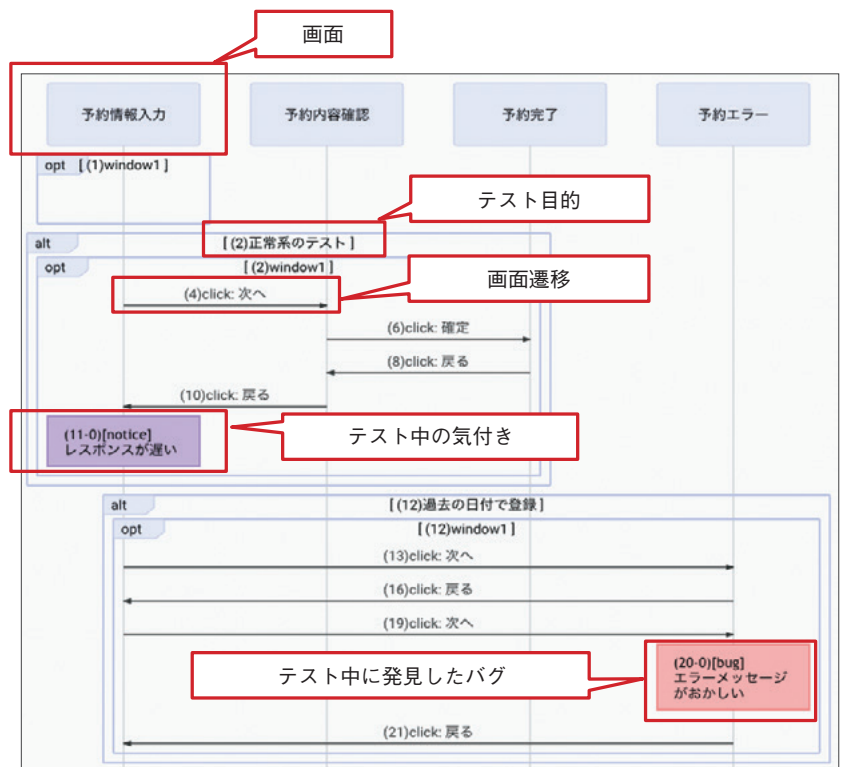


図3 シーケンス図

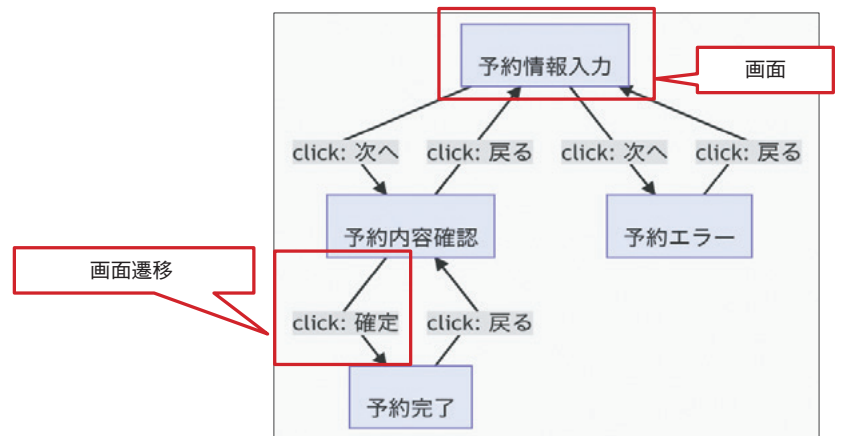


図4 画面遷移図

```
doGotoNext({
  reserveMonth,
  // ...
}) {
  this.reserveMonth.setValue(reserveMonth);
  this.reserveDay.setValue(reserveDay);
  // ...
  return new 予約内容確認();
}
```

図5 テストスクリプト

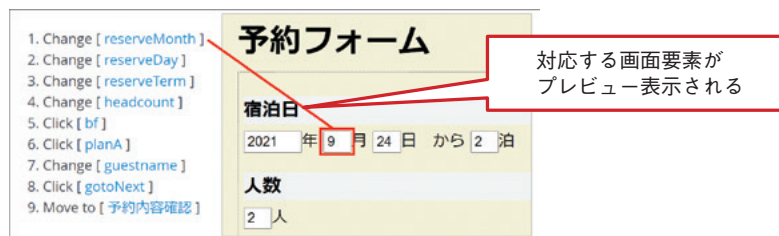


図6 テストスクリプトのドキュメント

ケース表とテスト結果相当の情報をテスト活動データから自動で生成するので、従来作成していた詳細なテスト手順を含む重厚長大なテストケース表も不要となります。

③ 回帰テスト向けテストスクリプト生成：テスト活動データから、ページオブジェクトパターンに基づいてモジュール化した保守性の高いテストスクリプトの雛形を自動生成⁽³⁾します。さらに、ドキュメントも合わせて一緒に自動生成⁽⁴⁾することで、編集の際のテストスクリプトの可読性を向上させます。ドキュメントには、テストがどのような画面遷移を行うかを示す画面遷移図や、操作される要素を示したスクリーンショットも含まれています。図5、6にテストスクリプトの例を示します。これにより、テストスクリプトの実装、保守が容易になり、手間をかけずに回帰テストを自動化することを可能とします。

到達点と今後の展望

現在、NTTグループ会社との共同実験でLatteArtの適用評価を行っており、開発現場から好感触を得ています。また、LatteArtの技術については国内の学会における発表でも高い評価を受け、複数の賞をいただくことができました。LatteArtを用いた短期

繰り返し型のテストは新しいテスト手法であるため、NTTグループ内外のステークホルダ（開発者・システムオーナー）により、新しいテスト手法が理解され、広く受け入れられる土壌をつくり上げることが重要と考えます。そのために、LatteArtのOSSコミュニティを立上げ、ソースコードをGithubで公開しました⁽⁵⁾。

今後は、実践で得たデータも武器に、大量に蓄積されたテスト活動データから、過去のテスターの知見を抽出することにより、テスト実施時に繰り返し行われる事前手順作業の自動化やテスト推薦を行うことで、テスト実施についても大幅な省力化を実現し、テスト全体のさらなる効率化をねらいます。加えて、NTTグループに閉じず、NTTグループ外の企業、大学等とも連携し、産学のさまざまな知見をいただきながら進めることで、LatteArtのOSSコミュニティを中心としたテスト変革のエコシステムを育てていくつもりです。めざす世界の実現のため、一步一步着実に研究開発を進めていきたいと考えています。

■参考文献

- (1) 熊川・峯尾・丹野・切貫・倉林：“SONAR Testing 効率と客観性を両立した新たなテスト手法,” 日本科学技術連盟 ソフトウェア品質シンポジウム2019, 2019.
- (2) H.Kirinuki, T.Kurabayashi, H.Tanno, and I.Kumagawa: “Poster: SONAR Testing - Novel Testing Approach Based on Operation

- Recording and Visualization,” ICST 2020, pp. 410-413, Oct. 2020.
- (3) 切貫・丹野：“手動テストのログを用いた有用なEnd-to-Endテストスクリプトの自動生成,” ソフトウェアエンジニアリングシンポジウム2020論文集, pp. 106-114, 2020.
 - (4) 但馬・切貫・丹野：“ドキュメントの自動生成によるEnd-to-Endテストスクリプトの理解支援,” ソフトウェア工学の基礎28, pp. 177-178, 2021.
 - (5) <https://github.com/latteart-org/latteart/>



(上段左から) 丹野 治門/ 切貫 弘之/ 川口 敬宏
(下段左から) 但馬 将貴/ 生沼 守英/ 村本 達也

私たちはテストを飛躍的に効率化することをめざし、繰り返し型の効率的なテストを実現するテスト活動データ分析技術の研究開発を進めています。今後も、開発現場のフィードバックを得ながら技術を磨いていきます。

◆問い合わせ先

NTT ソフトウェアイノベーションセンター
ソフトウェア開発技術プロジェクト
TEL 03-6712-9638
FAX 03-6712-8372
E-mail iso-tool-support-p-ml@hco.ntt.co.jp