

WHITE PAPER | 2015 年 9 月

テスト・データ 管理

継続的デリバリを実現する唯一の方法

Huw Price
CA Technologies

目次

継続的デリバリとは	3
継続的デリバリを妨げる不十分な TDM	3
TDM に対する要求ベースのアプローチ	5
まとめ	7
参考資料	8
CA Technologies のメリット	8
著者について	9

セクション 1

継続的デリバリとは

ソフトウェア開発の世界では、継続的デリバリという言葉がよく聞かれるようになりました。多くのベンダは継続的デリバリを実現するものとして、プロジェクトの遅延や失敗の既存の原因を修正するツールを提供しています。そのツールを採用すれば継続的な革新が可能になり、スケジュールと予算の範囲内で高品質のソフトウェアをデリバリーできるということです。

継続的デリバリという言葉を使えば注目を集めることができます。現在のようなアプリケーション・エコノミーでは、組織はソフトウェアに依存して顧客に価値を提供しています。企業が市場で優位性を確保するには、ビジネスと IT の整合性を強化して、顧客に日常的に価値を提供する必要があります。そのためには、IT チームには変化する市場と顧客の期待への迅速な対応、ビジネス・クリティカルなニーズの変化に合わせたソフトウェア開発、テストにかかるコストと時間の削減が求められます。

継続的デリバリを実現すると主張するツールは数多くありますが、それらはロジスティクスを主眼とし、開発ライフサイクルの後半でしかデプロイできません。DevOps は運用の観点からのみ捉えられ、ソフトウェアが設計済みであることを前提としたツールによって開発と回帰テストがサポートされます。要求の収集で始まる初期段階で発生したシステムの問題は放置され、継続的デリバリの成功を妨げる高コストの問題やプロジェクト遅延の原因になっています。

特にテスト・データ管理 (TDM) の重要性や、テストのために適時に適切な場所に適切なデータを提供することの重要性は見過ごされがちです。テストケースの設計が不十分で、質の低いデータが非効率的にプロビジョニングされれば、テスト・チームはシステムを完全にテストするのに必要なデータを入手できません。時間と予算の範囲内でソフトウェアを提供しようとすると、品質が犠牲になります。

セクション 2

継続的デリバリを妨げる不十分な TDM

多くの組織とベンダの TDM は、本番データのコピー、マスキング、サブセット化 (可能な場合) に終始しています。このような TDM 戦略は単にロジスティクスにすぎず、データの移動のみを重視して、プロジェクトの要求の設計でデータは見過ごされます。コピーして、開発、テストまたは QA の環境に移動した本番データは通常、「ゴールド・コピー」と呼ばれますが、これは、テスト担当者が必要なすべてのテストを実行して、システムを完全にテストすることのできる完全なテスト・データを意味しています。

マスキングとサブセット化は、インフラストラクチャのコストとコンプライアンスに関する従来の問題の解消に役立ちます。ただし、本番データベースをサブセット化およびマスキングすると、本番データの使用に固有の多くの問題が発生します。これらの問題によって、継続的デリバリの導入を成功させることは不可能になります。

時間の無駄

多くの組織には一元的な TDM チームやプロビジョニング・サービスが存在しないため、データはローカル・チームが検出または作成する必要があります。テスト担当者は作業時間の 50% をデータの検出に費やし、SDLC の総コストの 20% がデータを待つことに費やされています。それによってテストが妨げられ、完璧な機敏性と継続的デリバリの実現は不可能になります。たとえば、Grid-Tools (現 CA Technologies 傘下) の顧客のあるチームでは、3 週間の「スプリント」のために、4 週間もかけてデータを準備していました。

本番データをマスキングおよびサブセット化する場合、テスト・チームは使用するデータを自ら用意する必要があります。これは時間のかかる大変な作業ですが、制御されていないスプレッドシートに一貫性のないままデータが保存されると、さらに多くの時間が必要になります。たとえば、クレジットカードに関する欄は3つなのに、1人の顧客に4つの欄がある場合、個人情報「メモ」欄を追加するなどの作業が発生します。さらに、データの参照整合性も維持する必要があります。データが複雑になればなるほど、これを実現するのはむずかしくなるため、データのマスキングにかかる時間と労力は多くの場合、テスト担当者がゼロからデータを作成するよりも時間がかかります。

ゼロからデータを手作業で作成しても時間がかかることに変わりはありません。また、データは通常、特別なテストケースを念頭に作成されるため、常に変化する現実世界では短期間で陳腐化し、不適切になってしまいます。一般的な例としては、データが1日で古くなる取引パターンと通貨為替レートなどがあります。データが陳腐化したら、さらに時間のかかる更新を行うか、多くの場合は「削除」が必要になります。バージョンが変更された場合、データの更新を依頼された場合、または仮想環境が変更されるたびにデータを作成する必要がある場合なども同じです。

手作業で作成したデータを異なるチーム間で共有したり、再使用することはほとんど不可能です。各チームが労力を費やして作成したデータを活用できないために余分な作業が発生し、組織はコストを削減して開発期間を短縮するか、必要なすべての機能を備えた価値あるソフトウェア・アプリケーションをデリバリーするかのいずれかを選択せざるを得なくなります。

依存性による制約

テスト担当者の時間の大半は、データがプロビジョニングされるか使用可能になるのを待つことで費やされます。多くの組織やベンダのSDLCは、1つのチームが作業を終えたら次のチームにそれを引き渡すような直線的な段階で構成されています。そのため、各チームは上流のチームから「目的に適合する」データが提供されるまで待機し、必要なデータがあっても他のチームが使用しているときはそのデータを使用できません。

これは、継続的デリバリーの中核として、各スタッフの時間と革新の能力をすべて最大限に活用するというアジャイルの並行開発の考え方と対照的です。また、1つのチームがデータを変更すると、そのデータを使用した他のチームが明確な理由もなくアプリケーションのテストに失敗するような悪影響を及ぼす可能性もあります。

チームが手動でデータを作成する必要がある場合、または、データが使用できるまで数日または数週間待つ必要がある場合、要求の変更に迅速に対応したり、完全にテストしたソフトウェアをデリバリーすることは期待できません。

品質上の問題

本番データを本番以外の環境で使用する場合、最も問題になるのは品質です。本番データは通常、10～20%の機能網羅率しか提供しないため、どのようなサンプリングの手法を使用しても、新しいサブシステムを構築するのに必要なすべてのテストケースに対応するデータが提供されることはほとんどありません。本番データの多くは非常によく似ていて、「通常通り」の一般的なトランザクションから抽出され、その性質からサンタイズされ、システムの障害の原因となる不適切なデータは排除されます。そのため、テストでは「ハッピー・パス」が重視され、機能以外の部分やネガティブ・テストは軽視されています。

しかし、このような異常値と境界のシナリオがシステム破綻の原因になるため、ネガティブ・テストはテストの80%程度を占めるべきです。テストでハッピー・パスを焦点にしている限り本番環境で欠陥が発生し、再作業、致命的な遅延およびコストの上昇が回避できないだけでなく、プロジェクトが失敗する可能性さえあります。業界の調査によると、テスト段階でのバグ修正は要求の段階で検出するのに比べて50倍も時間がかかります¹。このような遅延が発生し、変化に対応できなければ、継続的デリバリーを実現することはできません。

本番データの機能網羅率を手作業による作成で補完することはできますが、それでは不正確な上に時間がかかり、あまりに非科学的です。また、最大限の網羅率に対応できたか検証する方法もなければ、データの参照整合性を確認する方法もありません。

高コスト

再作業とデリバリの遅延だけでも高コストになりますが、本番データを実際にコピーすることは非常に時間がかかり、コストも増大します。調査によると、1つのデータベースのコピーを20個も作成した結果、ハードウェア、ライセンス、サポートで高額のコストが発生した組織もあります。

また、ボリュームの大きいデータ・ストレージも高コストになります。関連するサードパーティのデータ・ストレージ調査では、クラウドなどの低コストの階層にデータを移動すると一時的にストレージ・コストは減少しますが、データ増加の根本的な問題が解決するわけではありません。つまり、データの移行、サブセット化、およびマスキングに依存する TDM ポリシーでは、これまで以上に複雑で複合的な最新のアプリケーションで処理したデータを低コストで保存するという課題に対応できません。

また、より高コストのコンプライアンス違反のリスクもありますが、これは本番データをマスキングしても解決できません。ある調査によると、データ漏えいのコストは2014年には平均15%も増加し、1件あたり350万ドルに達しました。また、2016年に施行される欧州連合（EU）の新しいデータ保護指令では、データ漏えいに対する罰金の最高金額は1億ユーロ相当²または総売上高の5%のいずれか高い方になります。

データ漏えいのリスクは人的ミスが最も多く、その50%以上が社内での行為に関連しているため³、本番データをマスキングしてもコンプライアンスは保証されません。また、マスキングする場合、データの参照整合性を維持しないとアプリケーションのテストが失敗する可能性があります。さらに、データが複雑になればなるほど相関関係のある情報が増えるため、容易に不正利用される可能性があります。

インフラストラクチャのコスト上昇を防止し、機密データの漏えいを防止する唯一の現実的な方法は、本番データを使用しないことです。このようなコストは継続的デリバリのフレームワーク導入を目指す企業に固有のものではありませんが、コストの上昇は継続的デリバリの考え方と対照的であることは間違いありません。コストが上昇すると、組織は変化するビジネスニーズに合わせて予算内で迅速に高品質のソフトウェアを提供することができません。

セクション 3

TDM に対する要求ベースのアプローチ

継続的デリバリの導入を目指す組織は、単にテストと開発のプロセスを入れ替えるのではなく、それらについて再考する必要があります。TDM に対するアプローチも再検討の必要があります。要求を中心としたテスト・データ管理のエンドツーエンドの完全なアプローチによって、テストの前倒し、リスク緩和、欠陥の最小化が可能になり、それによって高品質のソフトウェアを迅速かつ低コストで提供することができます。

テストケースごとのデータを考えるのではなく、設計の意思決定の観点からデータを捉えて、要求を中心にテストケースを設計し、データを要求に直接リンクさせる必要があります。テスト・データを要求に合わせてカスタマイズして、「目的に適合する」データをテスト・チームに迅速に提供できれば、ビジネスニーズの変化にも迅速に対応できるようになります。

このようなことから、継続的デリバリは SDLC の初期段階から開始する必要があります。要求から得たアイデアから開始すれば、テスト・チームと開発チームはアイデアの変化をすべて経験しているので、ビジネスニーズの変化に迅速に対応できます。また、両チームはテストケースを作成し、予測される結果と仮想データに直接リンクさせて、要求の有効性確認と検証を簡略化する必要があります。このようなモデルベースのテストこそが「前倒し」であり、ソフトウェアの継続的デリバリを可能にします。開発ライフサイクルのすべての作業を要求収集の段階に凝縮することで、要求が変更されてもその後のすべての作業をスムーズに進行させることができます。

要求作成の改善

テストに必要なシステムの品質に関するすべての情報を要求自体に含めることによって、テスト担当者はユースケースとテストケースを直接最初の「アイデア」から抽出し、両者の間の追跡可能性を確保できます。このようなトレーサビリティは、要求が変更されたときにテスト担当者が迅速にテストを更新するのに必要になります。

適切なツールがあれば、フローチャートによるモデル化を使用して簡単なフローの設計で、テストに必要なシステムの品質に関するすべての情報を含むフローチャートを作成できます。CA Agile Requirements Designer (旧 Grid-Tools Agile Designer) では機能ロジックによってフローチャートが強化され、最大限の網羅率で最小限のテストケースを自動的に抽出することができるため、原因結果モデルやペアワイズ法のように経営サイドが「途方に暮れる」ことはありません²。

また、フローチャートによって要求はより明確かつ完全になります。要求を通常構成している異種混合の「言葉の壁」や複雑なダイアグラムは、小さな理解しやすい固まりに分割されます。これらのプロセスにはシステムの原因結果のロジックが反映されるため、実質的に「what if, then this」ステートメントが構成されます。

これは要求の曖昧さに起因する欠陥の 56%⁴ を排除できるだけでなく、要求チームは制約、制限、境界の条件について、システムをモデル化して考えざるを得なくなります。要求からフローチャートとしてモデル化すると、システム全体の可能なすべてのパスを特定することができます。そこから 100% の機能網羅率に対応したテストケースを抽出できるため、ネガティブ・パスや予想外の結果を含め、システム全体のすべての可能なパスをテストできます³。

アクティブなフローチャートでは、このようなテストケースはさらに複雑なメトリクス、仮想データ、テスト・データ、自動化スクリプト、予測される結果およびバックログにリンクされ、SDLC の作業は要求収集の段階に凝縮されます。

「目的に適合する」データのプロビジョニング

テストケースの設計プロセスを CA Agile Requirements Designer などのツールによって自動化することで、テスト担当者は要求を最適に思い通りにテストするために必要な適切なテストケースを生成できます。また、この文書のテーマでもある継続的デリバリーに必要な TDM を改善できます。100% の網羅率のテストケースを確保するために、テスト担当者は適時に適切な場所に提供される「目的に適合する」データにアクセスする必要があります。

品質

前述のように、本番データではシステムを完全にテストするのに十分な網羅率を確保できません。それに対して、テスト・データを合成して生成すると、イベントが発生する前にすべての可能なシナリオに対応したコンパクトでありながら豊富なデータ・セットを作成できます。実際のシナリオのそれぞれは別のデータ・ポイントとみなすことができるため、新しいシナリオや将来のシナリオに対しても合成したデータを作成できます。CA Test Data Manager では、インテリジェントなデータ・プロファイリングの手法によってデータ・モデルの正確なイメージを把握でき、100% の機能網羅率に対応した豊富で洗練されたデータを生成できます。

要求ベースのアプローチでテストケースに合わせてデータが生成されるため、テスト担当者の個別のニーズに最適なデータが提供されます。このような「テストケースに合わせたデータ生成」によって、テスト・チームは欠陥の大半を 1 回のテスト試行で検出できる可能性が高いため、継続的デリバリーの妨げとなる時間のかかる再作業を回避できます。

時間

手作業でデータを作成または操作するのにかかる時間、あるいは、データが提供されるまでの待ち時間は、前述のとおり継続的開発の考え方に反するものです。

それに対して、CA Test Data Manager のようにデータ作成のために自動化されたツールは、RDBM や ERP API レイヤと直接関係できるため、ユーザは処理能力が許す限り迅速にデータを生成できます。大量のスクリプトによって、データベース・インフラストラクチャの処理と同じ速度で組織のデータ量は倍増します。そのため、「目的に適合する」データが数週間ではなく数時間で収集され、テスト担当者は必要なデータをスプリントで必要なときに使用できます。

また、強力なテスト照合機能によって、データが存在する場合は複数の異なるソースからデータが特定および抽出され、適切なテストケースに照合されて割り当てられます。このテスト照合機能では、手作業に比べるとデータの検出とプロビジョニングにかかる時間が 95% も削減されることが分かっており、また、データに関する問題が原因で失敗するテストを実行前に判別できます。

テスト・データを一元化されたデータ・ウェアハウスに保存すると、下流のチームはデータが使用可能になるまで待機する必要がなくなります。CA Test Data Manager On Demand のテスト・データ・ポータルでは動的なフォーム構築機能が提供されるため、ユーザはクレジット・カードのタイプや地理的な場所などの特定の基準に基づいて必要なデータのタイプを選択できます。このフォームではテストケースに必要なデータが割り当てられ、データが存在しない場合は新しいデータが作成されます。そのため、これまでデータがプロビジョニングされるのを待つのに作業時間の 50% を費やしていたエンジニアは、テストで確認された欠陥の修復にもっと多くの時間をかけられるようになります。また、IT セキュリティ・チームにデータのオーナーシップを集中させて、承認されたスタッフが依頼した場合のみ機密データをプロビジョニングすることができます。

コスト

最後に、優れた TDM ポリシーはコストの超過を防止できます。初期段階で欠陥を検出して修正すると、欠陥の発生を最高 95% も削減でき⁵、欠陥 1 つにつき 5 万ドル以上もコストを削減できます⁵。さらに、コンパクトで豊富なデータ・サブセットを使用すると、インフラストラクチャのコストをデータベース 1 つあたり最高 5 万ドルも削減できるだけでなく⁵、品質に優れたテストを回数を減らして実行できるため、テストの時間とコストをさらに大幅に削減できます。合成されたデータを使用することで、機密データを本番環境から取り込む必要がなくなるため、コスト上昇の原因になるコンプライアンス違反のリスクも排除されます。

セクション 4

まとめ

本番データを「唯一」の情報源として本番以外の環境にコピーすると、継続的デリバリの導入を成功させることはできません。テスト・チームはソフトウェアを完全にテストする必要があるときに、目的に適合するデータを使用できないこととなります。そうすると、ビジネス要件の変更に迅速に対応することはできません。サブセット化とマスキングのツールによって一部の問題を解決できても、本番データは欠陥を明確にして解決するのに十分な品質を備えていません。

エンドツーエンドの要求ベースのソフトウェア開発アプローチを採用し、テスト・データを再使用可能な価値ある資産として活用することは、継続的デリバリの導入を成功させる上で最も重要です。「アイデア」からすべての作業を行うことで、チームはビジネスニーズの変化に迅速に対応できるようになり、「目的に適合する」データを適切な場所へ適切な時間に提供して、テストの実行回数を最小限に抑えながらソフトウェアを完全にテストすることができます。

また、このアプローチでは、テスト・データのプロビジョニングが効率化され、継続的デリバリの実現を妨げる遅延、再作業およびコスト上昇のリスクを緩和できます。その結果、重要なビジネスニーズに対応した価値あるソフトウェアを、少ないリソースでスケジュールどおりに提供する継続的デリバリが実現します。

セクション 5

参考資料

- 1 <http://www.softwaretestingclass.com/why-testing-should-start-early-in-software-development-life-cycle/>
- 2 <http://www.agile-designer.com/resources/test-case-generation-bloor-market-report/>
- 3 フローチャートによるモデル化のメリットのコンセプトと数学的処理については、Llyr Wyn Jones の「Primer of A Critique of Testing」 (<http://www.agile-designer.com/resources/critique-testing-primer/>) を参照してください。

セクション 6

著者について

Grid-Tools の共同創設者 /CA Technologies 副社長

QA Guild の「IT Director of the Year 2010」に選出された Huw Price は、米国とヨーロッパのソフトウェア企業数社で技術アーキテクトの責任者を務めていました。テスト自動化ツールを専門に、ソフトウェア業界のテスト・モデルを変革する革新的な製品を多数手がけています。





ca.com/jp/でCA Technologiesにアクセスしてください



CA Technologies (NASDAQ:CA) は、企業の変革を推進するソフトウェアを作成し、アプリケーション・エコノミーにおいて企業がビジネス・チャンスを獲得できるよう支援します。ソフトウェアはあらゆる業界であらゆるビジネスの中核を担っています。プランニングから開発、管理、セキュリティまで、CA は世界中の企業と協力し、モバイル、プライベート・クラウドやパブリック・クラウド、分散環境、メインフレーム環境にわたって、人々の生活やビジネス、コミュニケーションの方法に変化をもたらしています。詳細については ca.com/jp/ をご覧ください。

- 1 Ponemon Institute : 2014 Cost of Data Breach
- 2 EU Proposed Data Protection Regulation (EU データ保護案)
- 3 2013 Data Breach Investigations Report
- 4 Bender Report : Requirements Based Testing
- 5 Grid-Tools の実装経験で得たメトリクス