

クラウド型サービス品質管理の新潮流

山下 克司

The new way of service quality management on the Cloud-ready environment

Katsushi Yamashita

クラウドや WEB アプリケーション分野でのサービス品質管理は、Service Level Agreement (SLA) だけに頼らない多様なメトリクスを管理することで、ユーザーの感じるサービスの品質を管理し、ビジネス指標との連携を実現しようとしている。最近注目されている開発と運用が一体となった DevOps 環境では、その品質を向上させるためのサービス品質のメトリクスとその測定法が重要となる。本技術解説では WEB サイトやクラウドで活用されている先進的なサービス品質のメトリクス評価を検証し、ユーザー・エクスペリエンスを評価する仕組みを考えることで開発と運用の共通のメトリクスによるサービス品質管理の新たな潮流を分析する。クラウド時代のサービス品質管理を基礎とした品質改善体制を考察することで、これまで難しかったビジネス、アプリケーション、インフラ運用に共通したサービス品質管理が実施できるようになる。

The current age of service quality management, which is based on the techniques of the Cloud Computing and the WEB application sites, try to find new way to manage user level quality by measuring various metrics instead of simple Service Level Agreement (SLA). In the DevOps environment, which unites development and operations, the "Kaizen" activity is the effort to improve service quality through the feedback mechanism of the measuring the business and site metrics. The key questions are what are the metrics and how to measure them. This paper will discuss about metrics that are using in the modern WEB sites, and will analyze new way of the common service quality management between development and operations based on the user experience management. Finally, this paper concludes the scheme of the development and the improvement based on cloud age of quality management techniques.

Key Words & Phrases : DevOps, クラウド, サービス品質, SLA, ユーザー・エクスペリエンス
DevOps, Cloud, Service Quality, SLA, User Experience

1. はじめに

日本の列車は正確なダイヤで運行されている。大都市や大規模な新幹線網などの列車運行の正確さは際立っている。しかし、極めて正確に列車は運行されている一方で、ラッシュアワーの不快さも同時に際立っている。ダイヤの正確さを表す数字とユーザーの感じている品質には大きな乖離がある。

GTO2009 [1] で IBM リサーチは「サービス品質」について、提供者が計測可能な品質と受益者の感じているサービス品質には乖離がある、と指摘している。これはサービスが成功したか、失敗に終わったかを割合で示すことに不合理があるからである。例として挙げられているのが空港の手荷物サービスである。空港の手荷物サービスの品質を「正しい荷物が正しい持ち主に一定時間内に届けられた」こととしたら、一見合理的に感じるだろう。手荷物サービスの正確さを計測することで顧客満足度を計測することができる

かもしれない。しかし、実際は空港の手荷物サービスの正確さは 4 シグマ、99.379% の正確さで行われているが顧客の不平の第 2 位に上げられている [2]。

これには次のような理由が考えられる。成功したトランザクションと失敗したトランザクションではトランザクション当たりの満足度の変化に違いがあるからである。飛行機が無事に到着した時に手荷物が正しく届いていることは極めて当たり前で顧客価値は少ない、すなわち満足ではあるが満足度が大きく増加するわけではない。しかし、逆に荷物が紛失した場合には直接的な航空会社への不平につながり、かつ顧客は大変困ってしまう。つまり、1 つの荷物が届くという価値よりも 1 つの荷物が届かないという損失の方がより大きい。そのため、1 つの損失によって損なわれるサービス価値を当たり前前の品質と同じ単位量で判断してはいけない、ということが分かる。

最新のクラウド環境におけるサービス品質管理はどうか。クラウド時代の運用開発手法であるとされる DevOps 環境では、開発と運用が一体となって小規模のリリースを定期的に行うリズムを刻みつつ、継続的にシステム全体の品質を向上することを目的としている。サイトの健康状態に何らかの

提出日:2012年8月20日

変化が現れた時、開発と運用は同時に1つの問題に対処して根本原因を改善するように活動する。改善活動はテスト・評価をしながらサイトの提供するサービス品質を向上させる努力である。ここでテスト・評価されるサービス品質のメトリクスとその測定法にはこれまでのSLA評価とは異なった視点が含まれている。

本論文では、2章でSLAの課題を述べた後に、3-4章で最新のサイトで用いられるサイトの健康状態を表すメトリクス管理について論じ、5-6章でユーザビリティを基本としたビジネス・メトリクス管理とサイトの健康状態を結びつける方法について解説する。最後にDevOps環境におけるビジネス、アプリケーション、インフラ運用に共通したサービス品質のメトリクス管理を提言する。

2. サービスにおけるSLAの機能不全

SLAの定義を式(1)に示す。

$$SLA = \frac{(MTBF - MTTR)}{MTBF} \quad (1)$$

MTBF: 平均故障間隔
MTTR: 平均修理時間

この計算は誰でも理解でき、かつ計測しやすい。99.99%の稼働率であれば、1年間に52分、99.999%の稼働率であれば5分の停止が許容される。自宅のパソコンであっても1年に何時間も止まるといわれたら困る。最低でも99.99%でなくては、と感じるだろう。しかしシステムはそんなに単純ではない。インターネットにつながっているプロバイダーやルーター、家庭内の無線LANもすべてが動いていなくてはパソコンも機能を果たさない。これらの可用性をすべて掛け合わせてすべてが完全に動いている確率を求めて初めてSLAが算出できる。これらの確率計算は信頼性の計算に基づいて計算するべきである。

ハードウェアが提供する基本的信頼性性能はMean Time Before Failure (MTBF)である。MTBFは故障するまでの平均時間、つまり前の故障から次の故障までの平均時間である。故障の発生確率を示す信頼度Rは、故障率λ(単位時間あたりの故障発生件数)と時間tから式(2)により求められる[3]。

$$R = e^{-\lambda t} \quad (2)$$

(Rは信頼度, λは故障率, tは時間)

MTBFが30年の高信頼性ハードウェアの5年間の信頼度は、その期間に1件故障する故障率λが1/30(0.033)、時間tが5であるから5年間の信頼度R = $e^{-(0.033 \times 5)}$ = 0.85(5年間の利用での故障率は15%)となる。

MTBFが30年のハードウェアの平均修理時間が8時間であった場合の可用性は99.997%で、稼働時間の0.003%が、停止しているということになる。しかし、同じハードウェアを5年利用していて故障に出会う確率は0.003%ではなく

15%である。SLAとユーザーの感じる品質の差はこうした基本的な数学からも実感できる。

ここまで述べて来たように、SLAは本来信頼性計算による確率と修理にかかわる時間を決定する保守体制などによって決定される。また、損失の非対称性にも配慮しなくてはならない。例えばSLA 90%を下回った場合、10%の不都合に対して月額料金の100%を返金するというような非対称なSLAも存在している[4]。これは評価に値する。しかし、サービスの市場で利用されているSLAはこれほど厳密ではないことが多い。例えば、「サービス品質保証制度において、規定のサービス品質を下回った場合においてはサービスの停止時間に相当する月額料金を返金します。尚、サービス停止については弊社からの通知は行いません。」というような制度の場合、ユーザーは自力で作成した1年間のサービス停止記録とともに返金請求をしなくてはならない。その労力の結果、1時間分の返金となっても月額料金の0.14%に過ぎない。月額10万円としたら140円足らずである。SLA条項は高い可用性への信頼性を保証する大事な条項であり、サービス提供者の義務を条項化しているべきだが、利用者の印象ほどプロバイダー・ビジネス上の制約条項となっていないことに気付く。つまりユーザーのSLAへの期待は「動いてほしい」だが、一部の提供者の論理が「請求されたら返金する」ということではユーザーが感じる品質との乖離は埋まらない。

3. CloudやWEBにおけるメトリクス管理

米WEB分析会社であるBitcurrentの社長アリスティア・クローラ氏は著書「WEBオペレーション」[5]の中で以下のように指摘する。「かつてITオペレーター達はインフラの健康状態だけを気にしていた。アプリケーションが稼働するプラットフォームが健康ならばユーザー・エクスペリエンスも問題ないと考えていたからである。今、それが事実ではないことを我々は知っている。我々が構築するのは膨大な数の、見ず知らずのユーザーを相手にするWEBアプリケーションである。エンドユーザーのメトリクスはバックエンドのメトリクス同様に必要不可欠なものである。」同じ著作の中で、Flickrのインフラ構築を担当したジョン・オルスポー氏は「メトリクスを収集することはWEBサーバーをネットワークに繋ぐのと同じくらいに重要である。」と述べている。

さらに、メトリクスの効用について、「資源の中長期的な予測は科学ではなく芸芸である。メトリクスの収集だけに頼ることは不可能ではないにしても、ソーシャル・ウェブ・アプリケーションの有機的な成長や、新しい機能をリリースしたときにユーザーが指数関数的に利用を始める非有機的な成長を、自信を持って予測することはできない。」と前提を置きながらも「メトリクス収集ツールはインフラの目であり耳である。トラブルの兆候があれば、真っ先にかじりつくところであるし、原因や問題解決につながるイベントの繋がりがや相関が見つけられるところである。」と言う。

クラウド環境のシステム構築ではどうか。クラウド環境が社

会的に進展して行く中で、これまでのような従量課金の資源提供によるコストダウンという視点だけではなく、クラウド・コンピューティングのテクノロジーをより積極的に活用する事例が生まれてきている。ガートナーが 2011 年 7 月に発表したハイプサイクル [6] において、黎明期に位置付けられている「Cloud Optimized Application Design」という技術がある。ここで述べられているクラウドに最適化されたアプリケーション設計の特徴は、水平分散型のスケーラビリティ、非停止システム、高性能で高効率なシステム、複数のクラウド・プラットフォームをまたがった高い移植性などがあげられている。これらがクラウドに最適化されたアプリケーション設計における主要な基本理念として提唱されている。これらの基本理念をアプリケーション設計の前提として捉えて、クラウドに対して完全に準備できているシステムを構築することである。また、IaaS タイプのクラウド・スタックが提供するシステム管理におけるプログラミング・インターフェースを積極的にアプリケーションから活用し、システム管理を予めアプリケーション設計に組み込むという方向性が示唆されている。まさに、インフラ運用がコード化されていく方向性である。ただし、インフラ運用のコードとアプリケーションのコードは明確に分けられた層で実装され、ログを中心とした連携がとられるべきである（後述）。ガートナー・レポート内でも参照されている米国でオンラインのビデオ・レンタルを行う NETFLIX 社の CIO であるエイドリアン・コッククロフト氏は 2011 年 QCon Tokyo パネル・ディスカッションに筆者とともにパネリストとして登壇し、以下のように述べた。「ビジネスのインフラとしてクラウド環境を用いることで、例えば日本でビジネス的な環境を整えば、10 時間後には日本でビデオ・レンタルのビジネスを開始できる。」モデル・ドリブン・アーキテクチャーと呼ばれる NETFLIX のアーキテクチャー [7] は、クラウド上で NETFLIX のビジネス・プラットフォーム全体を素早く実装することを可能にしている。クラウドの提供するインフラ管理用 API を最大限活用してインフラ運用のコード化を進めるとともに、アプリケーションの処理構造やシステム管理機能をクラウドに対応したパッケージングをする、クラウドに最適化されたデザインと実装の方法論がこの迅速性を支えている。クラウドに最適化された環境においてはシステムが単純に（単体で）プロビジョニングされるだけではなく、サービスが開始されるまでのすべてのプロセスにおいてクラウドを前提とした判断がされる。また、システムの機能を提供するために必要とされる、システム間の接続やシステム管理のコンポーネントなどすべての要素が統合されてパッケージングされていることで、メトリクス評価を含めたシステム管理を統合したサイトを提供することができる。

4. サイトの健康状態を測るメトリクス管理

Twitter では 30,000 以上のメトリクスを収集グラフ化している、と指摘するのは Twitter の運用担当者であるジョン・アダムス氏 [8] である。メトリクスを収集することでサイトの健康状況とデブロイによる安定性の影響が理解できる。メ

トリクスが表すグラフの変化は、サイトのパフォーマンスの影響を示す。こうしたデータは収集、処理、警報に用いられ、インフラに対する新たな知見をもたらしてくれる。Twitter の代表的なインフラのメトリクスは表 1 のとおりである。

表1. Twitterのメトリクス

秒間の Tweet 数	プロセスの再起動率
帯域幅	キューの深さ
応答時間	サーバー負荷、プロセス数
HTTP エラー	DBMS の負荷
ログの中の例外の比率	メモリー消費

メトリクスの管理と報告においてはグラフによる傾向値の表現も重要だが、SLA より誤解が少なく、かつ経営者やユーザー部門に理解できるメトリクスの単位が必要となる。そこで、様々なサービス・レベルを統一的に示すには Application Performance Index (Apdex) [9] を用いる。これはアプリケーションのパフォーマンスや可用性が与条件を満たしたことを頻度で表した指標である。

Apdex では図 1 に示すように目標となる性能指標 (t: Tolerant) を決定する。t 以内に収まったトランザクション数を（満足の数）とする。次に許容限界となる性能指標 (f: Frustrated) を決定する。一般的に f は t の 4 倍の値をとる。f を超えてしまったトランザクション数を（不満足だった数）とする。そして、性能目標 t と f の間にあったトランザクション数を（許容の数）として式 (3) に示す計算式を用いる。

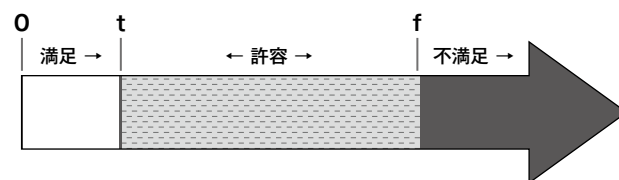


図1. Apdexの指標

$$\text{Apdexスコア} = \frac{(\text{満足の数}) + (\text{許容の数}) \div 2}{(\text{全体の数})} \quad (3)$$

Apdex 指標はすべてのトランザクションに対して、半数が時間内の応答を返し、半数が許容できる応答を返した場合、スコアは 75% をとる。Apdex はユーザーの満足に直結する指標を簡単に数値化するのに利用でき、応答時間とエラーの発生比というような異なる数値も比較対象とすることができる。なにより簡単に理解できて誤解がないことが良い。SLA の誤解に見られるように、経営者はゴルフのスコア以上に複雑なメトリクスは理解できないものだからである。

5. ユーザー・エクスペリエンスとファネル分析

ビジネスの成果に直結したメトリクス監視はさらに重要であ

る。IBMのユーザー・エクスペリエンス・デザインセンターでデザインリーダーを務めた吉武良治氏は、2010年に行われた日本人間工学会公開講座においてユーザー・インターフェース・デザインとユーザー・エクスペリエンス・デザインの違いを述べるにあたり、次のような例を挙げた [10]。

「ユーザー・エクスペリエンスをデザインすることは、製品の見た目だけではなく、ユーザーが体験するすべてのできごとを考えてデザインすることである。製品に興味を持ち、注文し、配達され、それに支払って、梱包を解き、使い始め、そして使い込んで、質問して、回答を得て、そして…アップグレードするまでのユーザーの体験すべてをデザインするのがユーザー・エクスペリエンス・デザインである。」そこで指摘されるのが、果たすべき当たり前の機能や無機質なSLAを満足しているということはゼロレベルでの水準であり、よりよいユーザー・エクスペリエンスを通じてユーザー体験を提供することこそ積極的な購買行動につながる、ということである。ISO 9241-11/JIS Z 8521 [11] [12]のユーザビリティの定義によると、ユーザビリティとはサービスや製品が特定の状況下で目的を達成するために用いられる際の、有効さ、効率及びユーザーの満足の度合いであり、それを構成しているのは、効率や目的への合致とともにユーザーのコンテキスト（利用に至る経緯）であるとしている。

WEBやCloud上で用いられているサービス品質の基礎的な解析モデルではユーザーのコンテキストを「ファネル理論」で解析する。「ファネル理論」は広告や販売促進の効果について、実際にトランザクション（購買）に至るまで（あるいは購買後のロイヤルティ獲得まで）のプロセスにおいて、その歩留まりを測定し、マーケティングのゴールに対して問題なのはどのプロセスで、最終的なゴールを挙げるためにはどこを改善するのが一番いいのかを発見する考え方である [13]。

図2にあるように、訪問者が何かのリンク（Referrer）からウェブサイトに到着（Landing）する。すぐに他のサイトに

離れて（Bounce）しまうか、あるいはサイトにとどまる（Visit）。サイト内では訪問者を利用しているブラウザであるとか、アドレスから割り出した国であるとか、リンク元などのシステム情報の特性によって分類（セグメンテーション）している。その後、訪問者のサイト内での活動を通じ、よりパーソナルなコンテキストを収集して、細かなセグメンテーションを行い、それぞれにおいて分析を行い、分類に応じた特別なオファーやプロダクトを表示することも可能になる。

訪問者は、サイト内で商品を購入したり、ユーザー登録をしたり、あるいはフォロワーとして登録をしたりする顧客、ユーザー、メンバーにコンバージョンすることになる。途中で放棄してサイトから離れることもある。ファネル理論では、アテンションからコンバージョンへの成功率が最も重要な指標であり、ユーザー・エクスペリエンスのコンテキストを監視するやり方である。

4章で述べたサイトの健康状態のメトリクス監視と本章で述べたビジネス成果のメトリクス監視を結び付けたい。次に紹介するのはWEBサイトの最適化を行って、サイトの健康状態を向上させた結果、ファネル分析の各指標がどう変化するかを計測したものである [14]。この実験ではWEBサイトの最適化アプライアンス（WEBキャッシュを用いた高速化装置）を用い、ユーザー・セッションごとに最適化する対象と最適化しない対象についてファネル解析を行ったものである。

ファネルの各段階のメトリクスは、表2のとおりである。この実験では最初のLandingは最適化グループに優位に結果が出たが、New Visitの比率において劣位の結果が出ている。これはリピーターの多さを示していると考えられることができる。さらに、それ以外のすべての指標において、WEBサイトの健康状態がファネルのメトリクスに良い影響を及ぼしていることがわかる。

表2. サイトの健康状態とファネル分析

	Optimized	Unoptimized
Landing	8,505	4,740
New Visit Rate	10.85%	13.61%
Bounce Rate	13.38%	14.35%
Page/Visit	15.64	11.04
Avg. time on Site	0:30:10	0:23:50

最終的にユーザー登録が行われたコンバージョン率は16.07%、購買活動の記録は5.51%の改善が見られ、サイトの健康状態とビジネス指標には相関があるものと考えられることができる。

サイトの健康状態とビジネスの成果を結び付けることができたら、それによって優れた監視、堅牢なインフラ、無駄の無いサイト・デザインの有効性などをいつでも検証できる。例えばサイト遅延の急増やコンバージョン率の低下が同時に起こっていたら、優先度を最大にして調査するだろう。十分な量のデータがあれば、相関分析も可能かもしれない。しかし、ここで述べたような人為的な実験環境でなければ、サイトの健康状態とビ

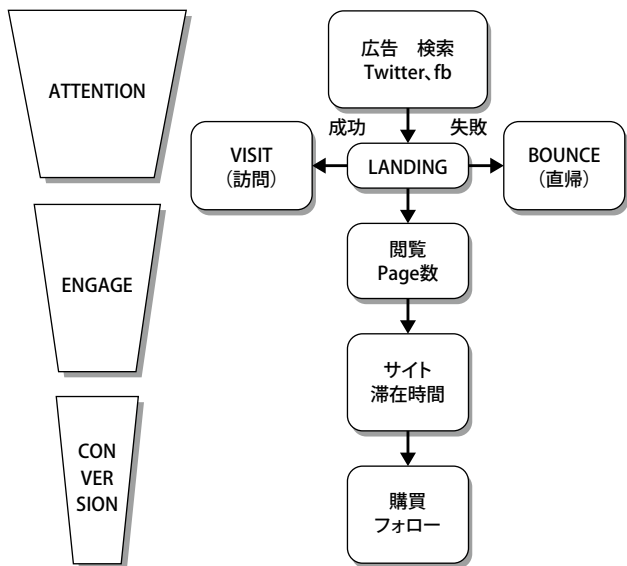


図2. ファネル分析によるシンプルな解析モデル

ビジネスの成果を結び付ける結果を得るのは容易ではない。

6. ログがすべて～ETL/BIツールの活用

早稲田大学大学院の丸山不二夫教授は2012年4月17日に行われたクラウド研究会の「FacebookのリアルタイムBig Data処理」[15]と題された講演において、Real Time AnalyticsのアーキテクチャーとしてWrite Ahead Log (WAL)を利用したTailingアーキテクチャーが使われている、と解説した。簡単に特徴を抜粋すると、WAL-Tailing ArchitectureではWEBから到着した新しいイベントは(Scribeで)ログ・ファイルに格納され、ログは(PTailで)後ろから処理される。システムはログからイベントを巻き取って(Pumaで)処理し、(HBaseの)ストレージに書き出す。ユーザー・インターフェースはストレージからデータを引き出してユーザーに表示する。ログは非常に簡単な形式であり、それがコンパクトなものであればあるほど多くをメモリー上に格納することができる。(本文中Scribe, PTail, Puma, HBaseはそれぞれFacebookで用いられているソフトウェアである。)

前出の書籍「WEBオペレーション」にも特にFacebookというわけではないが、同様の図式が掲載されている(図3)。システムのパフォーマンスとビジネスの成果を関連付ける方法として、ユーザーのすべてのセッションを記録するものである。図中のWEB AnalyticsはWEBページ単位のアクティビティ(リクエストと送信, エラーなど)、WIA: Web Interaction Analyticsはページ内でのユーザーのアクション、RUM: Real User Monitoringでは実際のユーザーがページを受信する速度などリアルなユーザーの活動を監視する。VOC: Voice of Customerはサイト内のフォームなどに寄せられた意見である。

システム上ではクッキーやソースIPアドレスなどをユーザーの識別としてユーザー・セッションとみなし、分析を行う。上記で解説した解析ツールが保存しているデータは、ETL

(Extract Transform Load) ツールでデータ・ウェアハウスに格納されBIツールによって解析される。本来この処理が難しいのは、それぞれの解析ツールが別々にデータを格納し、それぞれの基準で集計しているからである。例えばRUMデータは都市別、URL別でまとめられているが、ファネルデータは訪問者のセッションでまとめられていたりする。それぞれの指標もばらばらである。そこで、前述したようなApdexを用いた指標化を行いながら、大量のレコードをすべてそのまま利用することでデータは解析可能になる。BIツールを用いて大量のデータから傾向と相関を導き出すことができるようになる。想像だが、もしインフラの指標(JVMのヒープサイズは常に気になるだろう)が閾値を超えたアラートを発行しても、ビジネス・メトリクスに変化の兆候が無かったら、対応への優先度を下げ、定期保守で対応するなど無駄な緊急対応を避けることもできるかもしれない。

こうしてこの2つの例を見てみると、Facebookがログを中心にシステムを構築している理由が分かってくる。Facebookには大量のトランザクションがやってくる。その大量のトランザクションを確実に処理するには、これまでのようにデータの入力時に複雑で確実な(冗長な)処理をしている時間はない。まずログに記録してから、後に複雑な処理を行えば障害は発生時の例外処理として扱うことができる。これまでのトランザクション処理とは異なる考え方である。また、Facebookのビジネス・メトリクスはユーザーが登録するコンバージョン率と広告収入に直結するページ・クリック数であると推測できる。この2つの指標が計測、管理できなくてはサイトの健康状態を守るための指標を失い、目標を見定められずWEBの宇宙をさまよってしまう。すべてを統一されたコンパクトなログを中心にシステムを構築することで大容量のトランザクションを処理しながらも、リアルタイムに大量のデータを解析しサイトの健康状態とビジネスの関係を導き出している。

7. 提言～DevOpsにおけるサービス品質管理

ここまで述べてきたように、これからのシステム管理において最も重要な取り組みは、サイトの健康状態とビジネスの成果を結び付けることだと主張したい。この二者を結び付けてサービス品質を管理し改善して行くためには開発と運用が統合されたDevOps環境が必要であり、その一つの要素が統一されたログを中心としたシステム運用だということができる。

本論は、将来のシステム構築のあるべき姿について次のような提言をすることで考察に代えたいと思う。まず、開発はアプリケーションを作りっぱなしにするのではなく、実行環境のパフォーマンスや運用中のユーザー、データの増加にもっと気を払うべきである。これまでの社内向けシステムとは違う想定外の処理量の増加が想定されるからである。そのような事態には運用面でのインフラストラクチャーの増強だ

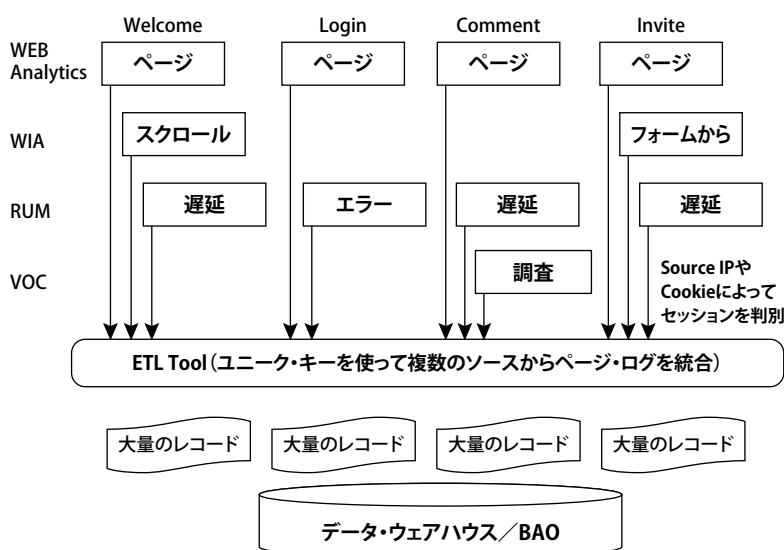
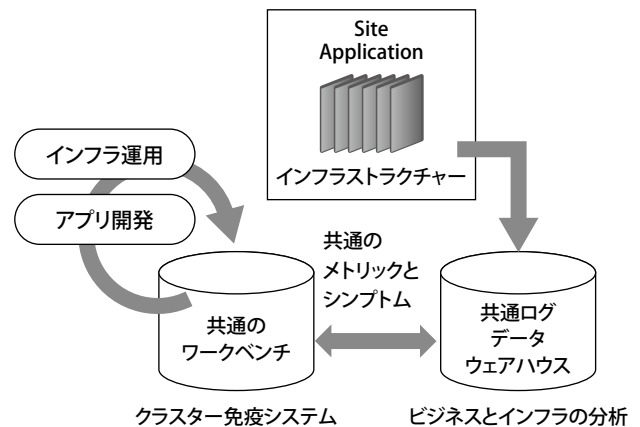


図3. ユーザーのセッションをすべて記録する[5]

けでは支えきれないこともある。こうした課題は、アプリケーションの変更やデータ保持の変更などを一体で行うことで、解決していくべきである。一方のインフラストラクチャーの管理は、SLA の単純な指標に隠れてデータセンター内部の健康状態しか気にしないのではなく、サービス全体が健全に機能を果たしているかどうかビジネス指標やインフラの健康状態を総合的に管理したい。仮想化不能な固定的なインフラストラクチャーもしばらくは残るかもしれない、しかし、インフラ運用をソフトウェアから制御するためのコード化は必須である。コード化されないインフラストラクチャーには高度な管理や分析が通用しないからである。

こうしたシステム運用が力を発揮するには、開発と運用の相互のコミュニケーションが円滑である必要がある。その上で共通理解されたビジネス・メトリクスの目標が必要となる。インフラ運用のコード化はソフトウェアで活用されてきた Rational Team Concert のようなワークベンチを共用できる可能性を高めている。このような共通のワークベンチが開発と運用の共通したメトリクス管理を実現すると考えられる。DevOps が提唱するクラスター免疫システム (表 3) [16] は、開発と運用の共通のワークベンチの強力な指針であると考えることができる。

図 4 に DevOps における共通のワークベンチの配置図を示した。ビジネスとインフラの統合化されたログ管理システムと相関分析が共通のメトリクスとメトリクスの示す症状であるシンプトムを提供する。これにより、システム全体の統合的な改善ポイントが明確になる。共通のワークベンチでは、改善



ポイントに対してインフラ、アプリの両面での検討が行われることになる。このような共通のワークベンチは運用だけではなくITシステム全体に適用でき、インフラストラクチャーや構築プロセスの標準化を進めるプラットフォームとして活用することが可能になる。こうしたことで、今までは計測しにくかったプログラムやインフラストラクチャー構築の生産性もチェックイン/アウトの活動記録から客観的に計測できる。

9. おわりに

ここまで、過去サービス品質の管理において SLA がいかにユーザーの感じる品質ということからかけ離れているかという点を指摘し、WEB やクラウドの上で行われているシステム管理のアプリケーションへのパッケージングについて論じてきた。クラウドを前提にすることで、システムの柔軟性はこれまでのレベルを遥かに超えてきたことも事実である。インフラ運用をコード化するというクラウド特有の環境がもたらした劇的な変化であると言える。

これまで挙げてきたシステムの特徴を次に示す。

- 多くのメトリクスを収集する
- 異なるメトリクスは Adpex を用いて指標化する
- インフラ運用は仮想化自動化を進めることでコード化する
- ユーザー・エクスペリエンスに基づいたビジネスに直結したメトリクスを管理する
- サイトの健康状態とビジネスの成果を統合的に管理できるログ管理システムを用いる。
- クラスター免疫システムに基づいた共通のワークベンチを用いて、a. ~ e. を統合する。

ここで a. と b. は成熟技術であり既存システムの品質メトリクスにも適用でき、c. はクラウド対応技術の適用によって実現することができる。次に d. は提供するサービスの内容に対する依存性が高いためマーケティングなどの専門家を巻き込んだ取り組みが不可欠であると考え、最後の f. は e. を前提として IT サービスを提供する組織全体の改革を伴う取り組みであり、今後もっとも力を入れるべき領域である。

表3. クラスター免疫システム

クラスター免疫システム	
(1) 局所的にテストする	開発者がサンドボックスを持っていること。 継続的なビルド環境=Build Bot テスト(検査工程)を行う →エラーがあれば「ラインを止める」 ドラムロールに合わせて作業する →早すぎず、遅すぎず
(2) 逐次デプロイメント	システム全体の状態とビジネス指標をモニター 指標の変化に対して敏感に反応し、ロールバックする
(3) アラートと予測的なモニタリング	インフラ、アプリ、ビジネスの多層的な モニタリング →しきい値管理を行う 許容境界値の予測には統計的な手法を用いる (≠直感)
(4) 顧客が失敗に気付いたら (VOC)	問題を顧客のために直す すべてのレベルで防御を改善する

前出のアリステア・クロール氏は同書の中でクラウド環境について「ルーフバック問題」があると指摘している。「車を買うと標準で付いているものとオプションで取り付けるルーフバックのようなものは明らかに分かる。その点クラウドにおいて複雑なのは PaaS タイプのクラウドには監視やメトリクス収集の仕組みは標準である場合があるが、IaaS タイプのクラウドにおいてはこうしたオプションは自分なりに組み合わせて作らなくてはならない。」

こうした課題に直面するクラウド時代のシステムの構築にはこれまでのシステム構築と大きく異なる点がある。プログラミング、リリース管理、自動運用、データの分散、ハイブリッド環境、ビジネスの計測と改善のプロセスなどすべてを考えるのは 1 人の手に余る。そうなのであれば、この構築／運用手法はチームの果たしてきた経験の蓄積によって達成されるものだと思う。開発と運用が統合された DevOps チームとその経験がその答えに近づくことができる。こうした取り組みを通じて、これまでの硬直した IT システムと構築手法が、クラウドの力を借りて大きく飛躍して、IT ユーザーの変革を促すことができる日を望んでいる。

謝辞

スマータークラウド事業の紫関昭光氏、新島智之氏には常に大きな刺激を受けている。ユーザー・エクスペリエンスについて吉武良治氏の洞察は本論の基礎を成すものである。ここに深い感謝を捧げます。

参考文献

- [1] IBM : Global Technology Outlook ., http://www-06.ibm.com/software/jp/academic/skills/resource/gto_overview.pdf
(筆者注: 2009 年の Global Technology Outlook の内容については IBM 社員にお尋ねください。)
- [2] Nikos Anerousis : Services Innovation, <http://www.summersoc.eu/wp-content/uploads/2012/07/2012-Crete-SOC-Anerousis.pdf>
- [3] 電気通信主任技術者試験 資格取得のための信頼性計算, <http://www8.plala.or.jp/ap2/shinraisei/>
- [4] NTT 西日本: NTT 西日本のサービス品質保証制度 (SLA) について, <http://www.ntt-west.co.jp/datatop/sla.html>
- [5] John Allspaw and Jesse Robbins : Web Operations – Keeping the data on time, pp.315, O'Reilly & Associates Inc (2010).
- [6] David Mitchell Smith : Hype Cycle for Cloud Computing, 2011, Gartner, <http://www.gartner.com/id=1753115>
- [7] Netflix : Adrian Cockcroft, <http://www.slideshare.net/adrianco>
- [8] John Adams : Operations on Twitter, <http://www.usenix.org/event/lisa10/tech/slides/adams.pdf>
- [9] APDEX: Apdex Alliance, <http://apdex.org/>
- [10] 吉武良治: 魅力的な商品企画・開発のためのユーザー・エクスペリエンス・デザイン -UCD (ユーザー中心設計) アプローチ, <http://www.ergonomics.jp/official/wp-content/uploads/2010/04/yoshitake2010.pdf>
- [11] U-Site 編集部: ユーザビリティの定義, <http://www.usability.gr.jp/whatis/definitions/>
- [12] ISO : ISO 9241-210 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems, http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075
- [13] 横山隆治: ファネル理論とネットマーケティング, <http://adv.yomiuri.co.jp/ojo/02number/200811/11integ.php>
- [14] Alistair Croll : Proof that speeding up websites improves online business. <http://www.watchingwebsites.com/archives/proof-that-speeding-up-websites-improves-online-business/>
- [15] 丸山不二夫: Facebook のリアルタイム Big Data 処理, <http://www.slideshare.net/maruyama097/facebook-big-data>
- [16] Eric Ries: STARTUP LESSONS LEARNED, <http://www.startuplessonslearned.com/>



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス事業
ITS デリバリー
ディステイングイッシュト・エンジニア
技術理事

山下 克司 Katsushi Yamashita

【プロフィール】

1987 年、日本 IBM 入社。適用業務パッケージの開発などを経てネットワーク分野のテクニカル・リーダーを務める。2007 年にネットワーク仮想化技術などの貢献を評価されディステイングイッシュト・エンジニアに認定され技術理事に就任。2010 年から 2012 年まで日本 IBM クラウド・コンピューティング事業の技術統括をするチーフ・テクノロジー・オフィサーに就任。現在はグローバル・テクノロジー・サービスのデリバリー部門の技術理事。