

WebAPIの利用時品質を定量化するメトリクスの考察

中村 匡秀¹

概要: WebAPI を活用してソフトウェアを開発する場合、ソフトウェアとしての API の品質は、開発者が当該 API を選択・使用する価値に大きく影響する。本稿では、ソフトウェア品質モデル SQuaRE の利用時の品質に着目し、クラウドサービスや WebAPI の品質を定量化するメトリクスを考察する。

1. はじめに

昨今のソフトウェアビジネスでは、**API エコノミー**という新たな経済圏が形成されている。企業のリソースやサービスを API (Application Program Interface) を通して社外に公開し、外部のサードパーティーから利用・連携してもらうことで価値を創出し、利益を生み出そうとするものである。従来のビジネスでは開発したソフトウェアを利用者に納品し対価を受け取るのに対し、API エコノミーでは利用者が API を利用して（呼び出して）初めて利益が生まれる。そのため、API の提供者は、魅力的な API を開発・提供し、多くの利用者から利用してもらう必要がある。

API の利用者は主にアプリケーション開発者、あるいは、それを用いたサービスの提供者である。彼らにとっての API の選択・利用は何らかの**価値観**によって決定されるが、開発するアプリケーションや展開するビジネスによって、非常に個別的である。筆者は、API のソフトウェアとしての**品質**が、API の価値に大きく影響すると考え、ソフトウェア品質モデル SQuaRE [1] を API エコノミーの文脈で解釈し、価値に関連する品質特性を洗い出した [2]。

本稿では、API 価値に関連するソフトウェア品質のうち、**利用時の品質**に焦点を当て、API の利用者からみた品質を定量化するメトリクスについて考察を行う。ワークショップでは、これらのメトリクスの妥当性や計測可能性、問題点、代替案等を議論したい。

2. API エコノミーにおける価値分析モデル

筆者は [2] において、API のどのような品質が利用者にとっての価値に影響するかを分析する **API 価値分析モデル**を提案した。提案モデルは、SQuaRE ソフトウェア品質モデルを API エコノミーの文脈で解釈し、各品質特性が

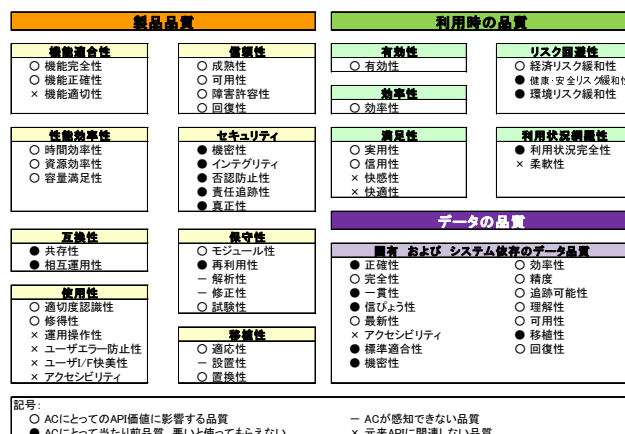


図 1 SQuaRE 品質モデルに基づく API 価値分析モデル

ACが見出す API 価値に影響するかどうかを分類する。

図 1 に提案する API 価値分析モデルを示す。SQuaRE の 3 種類の品質 (製品品質, 利用時の品質, データの品質) における品質特性を、4 つの観点で分類している。

3. WebAPI の利用時の品質とメトリクス

WebAPI の利用時の品質とは、API 利用者が自身のニーズに照らして、API を利用できる度合いである。図 1 に示す通り、5 つの品質特性と 11 の副特性で性質づけられる。以降では、API 利用者 u がある API (x とする) を利用してアプリを開発すると想定し、 u にとっての x の品質 $Q(u, x)$ を定量化するメトリクスを考察する。

3.1 有効性メトリクス

有効性は、明示された目標を利用者が達成する上での正確さおよび完全さの度合いである。該当 API を利用することで、開発するアプリケーションの機能をどの程度正確かつ完全に実現できるかを表す。いま、 u が開発するアプリの機能のうち、 u が x を利用して実現しようとしているもの

¹ masa-n@cs.kobe-u.ac.jp, 神戸大学 Kobe University, 1-1 Rokkodai, Nada, Kobe 657-8501, Japan

を $f_1, f_2, \dots, f_n$ とする. また各 f_i の要求を $r_{i1}, r_{i2}, \dots, r_{ik_i}$, 各要求の重要度を w_{ij} , $a_x(r_{ij})$ を x によって r_{ij} が達成された時 1, それ以外は 0 を返す関数とする. 各機能 f_i の実現度合いを次のように定義する:

$$\text{achieved}(x, f_i) = \frac{\sum_{j=1}^{k_i} w_{ij} \cdot a_x(r_{ij})}{\sum_{j=1}^{k_i} w_{ij}} \quad (1)$$

u にとっての x の有効性は, x の利用によって機能 f_i ($1 \leq i \leq n$) がどれだけ実現できたかに帰着できるので, 有効性メトリクスを以下のように定義できる:

$$\text{Effect}(u, x) = \sum_{i=1}^n \text{achieved}(x, f_i) \quad (2)$$

3.2 効率性メトリクス

効率性は, 利用者が特定の目的を達成するために使用した資源の度合いである. 該当 API を利用することで, 利用者がどの程度の資源を節約できたかで表すことができる. API 利用によって節約できる最も代表的な資源は, 開発にかかる労力 (=工数) が挙げられる. その他にも, 運用・保守の費用, 機能の実行に必要な計算資源が存在する. いま, u が x を利用して (せずに) f_i を開発した場合の工数を $E(u, f_i, x)$ (それぞれ $E'(u, f_i)$) とする. また, u が x を利用して (せずに) f_i を運用した場合の工数を $M(u, f_i, x)$ (それぞれ $M'(u, f_i)$) とする. さらに, f_i の実行に必要な計算資源を $R(f_i)$ とする. u にとっての x の効率性は, x の利用によってどれだけ資源が節約されたかで表現できるので, 効率性メトリクスを以下のように定義する:

$$\text{Efficiency}(u, x) = e \sum_{i=1}^n \{E'(u, f_i) - E(u, f_i, x)\} + m \sum_{i=1}^n \{M'(u, f_i) - M(u, f_i, x)\} + r \sum_{i=1}^n R(f_i) \quad (3)$$

ここで, e, m, r はそれぞれ開発工数, 運用保守工数, 計算資源にかかる係数である.

3.3 満足性メトリクス

満足性は, API の利用によって利用者がどの程度満足するかの度合いである. 図 1 に挙げた通り, API エコノミーの文脈では, 実用性と信用性の 2 つの副特性が特に重要である. 実用性は「この API は本当に使える」と開発者が評価できるかどうかを, 信用性は「この API は安心して使える」と評価できるかどうかである. 実用性は有効性に深く関連するが, API がアプリの要求を単に達成するだけではなく, 期待以上の働きをしてくれるかどうかがかギとなるのではないだろうか. その意味で API の呼び出しやすさやデータ操作のしやすさ等の使用性にも影響するだろう. 信用性は, 利用者が意図した通りに API の呼び出しが行われ, 意図した結果が得られることによって形成され, 製品品質の信頼性に大きく影響するであろう. このように, 満足性は複数の要素が複雑に絡み合うが, u が x に満足して

いるかどうかは, u が x をどれだけ長く, 頻繁に使い続けているかで定量化できるのではないかと. u の x の使用時間を $T(u, x)$, 呼び出し回数の総数を $C(u, x)$ とするとき, u の x に対する満足度を以下のように定義する:

$$\text{Satisfaction}(u, x) = t \cdot T(u, x) + c \cdot C(u, x) \quad (4)$$

3.4 リスク回避性メトリクス

リスク回避性は, API の利用によって出会うかもしれない様々なリスクが低く抑えられているかを表す度合いである. リスクの種類によって, 経済リスク緩和性, 健康・安全リスク緩和性, 環境リスク緩和性の 3 種類の副特性が定義されている. いま, 機能 f_i の実現に x を使うことによって生じる潜在的なリスクを $pr_{i1}, pr_{i2}, \dots, pr_{ik_i}$, 各リスクの重要度を rw_{ij} , $rm_x(r_{ij})$ を x が pr_{ij} を考慮しているとき 1, それ以外は 0 を返す関数とする. このとき, 各機能 f_i のリスク回避性は次のように定義できる:

$$\text{mitigated_risk}(x, f_i) = \frac{\sum_{j=1}^{k_i} rw_{ij} \cdot rm_x(r_{ij})}{\sum_{j=1}^{k_i} rw_{ij}} \quad (5)$$

u にとっての x のリスク回避性は, 各機能においてどれだけ潜在的なリスクが回避されているかで表現可能である:

$$\text{RiskMitigation}(u, x) = \sum_{i=1}^n \text{risk_mitigated}(x, f_i) \quad (6)$$

3.5 利用状況網羅性メトリクス

利用状況網羅性は, 様々な利用状況において API が意図したとおりに利用できるかを表すものである. API エコノミーにおいては, 利用状況完全性が特に重要である. これは, 最初から意図した範囲においては様々な利用状況の違いに関わらず品質を保持するというものである. いま, x が利用されるコンテキストを $c_1, c_2, \dots, c_n$ とし, それぞれの出現回数を $\text{freq}(c_i)$ で表す. また $\text{covered}(c_i, u, x)$ を, コンテキスト c_i において $\text{Effect}(u, x)$, $\text{Efficiency}(u, x)$, $\text{Satisfaction}(u, x)$, $\text{RiskMitigation}(u, x)$ がそれぞれ水準以上の時 1 を, それ以外は 0 を返す関数とすると, 利用状況完全性を次のように定義する:

$$\text{ContextCompleteness}(u, x) = \sum_{i=1}^n \text{covered}(c_i, u, x) / n \quad (7)$$

謝辞 この研究の一部は, 科学技術研究費 (基盤研究 B 16H02908, 15H02701, 基盤研究 A 17H00731, 萌芽研究 15K12020), および, 立石科学技術振興財団の研究助成を受けて行われている.

参考文献

- [1] ソフトウェア高信頼化センター: つながる世界のソフトウェア品質ガイド, 情報処理推進機構 (2015).
- [2] 中村匡秀: API エコノミーにおける開発者視点からの API 価値に関する一考察, 信学技報, Vol. 117, No. 184, SC2017-21, pp. 45-50 (2017).