

# **Oracle® Enterprise Repository**

メトリック コンフィグレーション ガイド

10g リリース 3 (10.3)

2008 年 10 月

Copyright © 2007, 2008, Oracle. All rights reserved.

原著者 : Vimika Dinesh

原協作者 : Jeff Schieli, Sharon Fay

このプログラム(ソフトウェアおよびドキュメントを含む)には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段(電子的または機械的)、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかける目的で使用する場合、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle, JD Edwards, PeopleSoft, Siebelは米国Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性があります。

このプログラムは、第三者のWebサイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者のWebサイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行(製品またはサービスの提供、保証義務を含む)に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

## メトリック コンフィグレーション ガイド

### 目次

- はじめに
- **メトリック要素**
  - 概要
  - メトリックの追加
  - アセットの種類<sup>1</sup>の要素の有効化
- **データベースに対するクエリの実行**
  - データベース クエリ
- **SOA 成功の評価**
  - 再利用可能なアセット
  - 評価プロセス

### はじめに

このガイドでは、Oracle Enterprise Repository で使用可能な強化されたメトリックについて説明します。このガイドの情報は、メトリック フィールドのないアセットの種類にメトリック フィールドを追加する場合に役立ちます。このガイドで取り上げるメトリック フィールドは、**合計開発時間 (TDH)**、**生成投資 (Pinv)**、**消費係数 (Cfac)**、**予測年間再利用回数 (n)**、および**時間負荷率 (B)** です。これらのフィールドは通常、Asset Editor の [Metrics] タブに表示されます。

このドキュメントで説明するアップグレード処理では、ユーザのサイトの DBA、システム管理者、およびレジストラが必要です。

# メトリック要素

## 概要

このドキュメントでは、種類にメトリック要素を追加する方法について説明します。これらのメトリックは、各アセットの価値の見積もり、およびアセット ポートフォリオの ROI の判定に使用されます。

合計 5 つのメトリック要素を追加する必要があります。**Development Hours** 要素は種類にすでに存在する必要があります。この要素が種類にない場合、または削除された場合は、この要素も追加する必要があります。**Type Manager** とデータベースで必要な変更を行います。両方の手順についてここで説明します。新しいメトリック要素は以下のとおりです。

- **合計開発時間 (TDH)** - このアセットの作成に費やされた開発努力の合計。
- **生成投資 (Pinv)** - アセットを再利用可能にするために、アセットの構築または収集、およびアセットのパッケージ化および文書化に費やされた追加の時間または努力。この値は、1 回の使用を目的としたアセットを構築するために最初に必要となる時間に対する割合で表されます。
- **消費係数 (Cfac)** - アセットの使用者がアセットの検索、評価、および使用に費やす時間。アセットを最初から構築するために必要な時間の割合で表されます。
- **予測年間再利用回数 (n)** - アセットが 1 年間に使用されると予想される回数を示します。
- **時間負荷率 (B)** - アセットを生成したユーザの 1 時間あたりのオーバーヘッドコスト。

メトリックは、10 個の基本のアセットの種類、およびユーザのシステムに追加されているその他のアセットの種類に含まれている必要があります。

- [Application]
- [Business Process]
- [Communication Adapter]
- [Component]

- [Environment]
- [Framework]
- [Pattern]
- [Process]
- [Service]
- [XML Schema]

## メトリックの追加

この手順は **Asset Editor** で行われます。

このテキストの要素名を切り取って貼り付けます。

- 合計開発時間 (TDH)
- 生成投資 (Pinv)
- 消費係数 (Cfac)
- 予測年間再利用回数 (n)
- 時間負荷率 (B)
- **注意 :**
  - データベースに対して `validateMetrics.sql` ファイルを実行する必要があります。

1. [Actions] メニューで [Manage Types] を選択します。

**Type Manager** が起動します。

2. **Type Manager** のサイドバーから、メトリックを追加する種類を選択します。

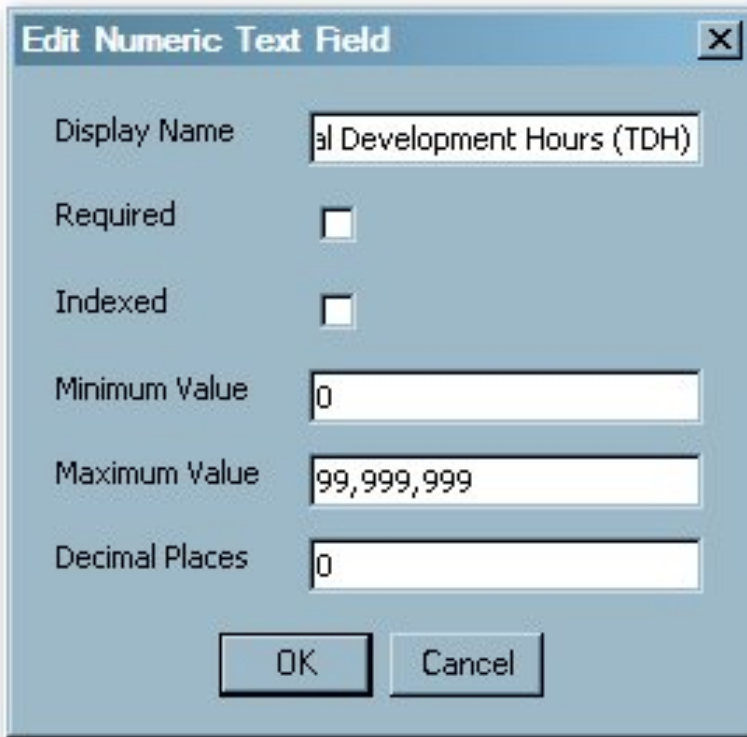
3. [Editor] ビューを選択します。

4. [Metrics] というタブを追加します。[Miscellaneous] タブの上に表示されます。

5. [Elements] セクションの [Add] をクリックします。

6. ドロップダウンから [Numeric Text Field] を選択します。

7. [OK] をクリックします。
8. 要素名をコピーして、[Edit Text Field] ボックスの [Display Name] ボックスに貼り付けます。
9. [Decimal Places] の値を「0」に変更します。[消費係数]要素の小数位のみ 2 です。



The image shows a dialog box titled "Edit Numeric Text Field". It contains the following fields and controls:

- Display Name: al Development Hours (TDH)
- Required:
- Indexed:
- Minimum Value: 0
- Maximum Value: 99,999,999
- Decimal Places: 0
- Buttons: OK, Cancel

10. [OK] をクリックします。
11. 各メトリックについて手順 4 ~ 9 を繰り返します。
12. すべてのメトリックを種類に追加したら、[Save] をクリックします。
  - ここでは [Viewer] タブに要素を追加しません。
13. 基本のアセットの種類それぞれについて、この手順を繰り返します。

## アセットの種類要素の有効化

1. 直前に変更したアセットの種類 1 つに属するアセットを選択します。そのようなアセットを最も簡単に見つけるには、**Asset Editor** の [Registered] フォルダを右クリックして、アセットの種類で並べ替えます。

2. アセットの各メトリック要素にユニークな値を追加します。値 1、2、3、4、5 を使用することをお勧めします。

Sample Component J2EE - Order EJB (2.0)  
Asset Type : Component

Overview	Taxonomy	Architecture	Documentation		
Relationships	Tests	Support	Metrics	Miscellaneous	Administration
Total development hours (TDH)		1			
Production investment (Pinv)		2			
Consumption Factor (Cfac)		3.00			
Predicted number of annual reuse opportunities (n)		4			
Hourly burden rate (B)		5			

Metrics

Approved By: User, Joe Approved Date: 1 Jan 2005

3. 編集したアセットのステータス バーで、アセットの名前、バージョン、およびアセットの種類の名前を確認します。
4. アセットを保存します。

## データベースに対するクエリの実行

次のメトリック クエリを、[後述](#)の手順に従って、データベースに対して実行します。

```
-- Metrics SQL (Oracle、 UDB、 および MsSQL Server)
```

```
-- 合計開発時間
```

```
select a.name as AssetName, a.version as Version, at.name as  
AssetTypeName, axi.stringvalue as tdh from assetxmlindex axi,  
assets a, assettypes at where axi.assetid = a.id and  
axi.assettypeid = at.id and axi.fieldid in  
    (select id from assetxmlindexmappings where fieldname =  
'/custom-data/total-development-hours--tdh-') order by a.name,  
a.version;
```

-- 生成投資

```
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as Pinv from assetxmlindex axi,
assets a, assettypes at where axi.assetid = a.id and
axi.assettypeid = at.id and axi.fieldid in
(select id from assetxmlindexmappings where fieldname =
'/custom-data/production-investment--pinv-') order by a.name,
a.version;
```

-- 消費係数

```
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as Cfac from assetxmlindex axi,
assets a, assettypes at where axi.assetid = a.id and
axi.assettypeid = at.id and axi.fieldid in
(select id from assetxmlindexmappings where fieldname =
'/custom-data/consumption-factor--cfac-') order by a.name,
a.version;
```

-- 予測年間再利用回数

```
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as n from assetxmlindex axi,
assets a, assettypes at where axi.assetid = a.id and
axi.assettypeid = at.id and axi.fieldid in
(select id from assetxmlindexmappings where fieldname =
'/custom-data/predicted-number-of-annual-reuse-opportunities--n-')
order by a.name, a.version;
```

-- 時間負荷率

```
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as B from assetxmlindex axi,
assets a, assettypes at where axi.assetid = a.id and
axi.assettypeid = at.id and axi.fieldid in
(select id from assetxmlindexmappings where fieldname =
'/custom-data/hourly-burden-rate--b-') order by a.name, a.version;
```

-- 予測節約正味時間

```
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as PNHS from assetxmlindex axi,
assets a, assettypes at where axi.assetid = a.id and
axi.assettypeid = at.id and axi.fieldid in
```

```
(select id from assetxmlindexmappings where fieldname =  
'/custom-data/development-hours') order by a.name, a.version;
```

-- Metrics SQL の終了

## データベース クエリ

1. データベースへの接続を開いて、前述の 6 つのクエリを実行します。結果として、アセットの名前、バージョン、アセットの種類、およびメトリックが返されます。メトリックが返されない場合は、メトリック名が正しくコピーおよび貼り付けられていない可能性があります。

各メトリックに独自のクエリがあります。これらのクエリは個別に実行することをお勧めします。想定される結果を返すメトリックが有効化されています。

2. いずれかのメトリック クエリが失敗する場合は、その要素を削除してから、手順に従ってメトリック要素を作成し、有効化して、データベースに対して実行します。選択したアセットの種類に PNHS 要素がない場合は、次の手順に従って追加する必要があります。

1. 「Development Hours」という文字列を追加します。

2. [OK] をクリックします。

3. **Development Hours** 要素を編集して、名前を「**予測節約正味時間 (PNHS)**」に変更します (メトリックの要素名は、**Development Hours** というプログラム名に基づきます)。 **Asset Editor** に [予測節約正味時間] ラベルが表示されます。

3. すべてのメトリックが有効化されたら、Oracle Enterprise Repository の **Type Manager** でそのアセットの種類を開きます。

4. [Viewer] タブを選択します。

- **注意：**

- 追加したすべての要素が [Hidden Elements] 領域に表示される必要があります。
- アセット使用者にとっては、**予測節約正味時間 (PNHS)** メトリックだけが意味を持ちます。
- **予測節約正味時間 (PNHS)** が [Overview] グループの最初の列に表示されていることが推奨されます。

5. アセットの種類を保存します。

6. 手順 1 ~ 5 を繰り返して、他のアセットの種類にメトリック要素を追加します。
7. クエリをテストするために追加したサンプル データを削除して、データを保存します。

## 概要

IT 組織は、処分対象のソフトウェア アセットに熱心に取り組んでいます。これらのソフトウェア アセットは、サービス指向アーキテクチャ (SOA) への変換および SOA の発展の中で組織的に再利用されれば、開発の生産性を上げ、ソフトウェアのメンテナンス コストを減少させることで年間数十万ドル、場合によっては数億ドルの節約になる可能性があります。その結果としてパフォーマンスが向上すると、ビジネス アジリティや、競合する勢力に迅速に対応する能力に重大な効果がもたらされる場合があります。


SOA への変換では、巨大なアプリケーションを、目立たず、結合が弱く、再利用性の高いサービスの集合に分解します。この変換を管理し、サービスと関連アーティファクトを確実に最大限に再利用するための適切なガバナンス方法には、その再利用の価値を評価、追跡、および伝達する手段が必要です。これによって、SOA が組織の目標を達成するための軌道に乗っているかどうかを判断するのに重要なフィードバックが提供されます。提供された情報は、SOA の発展に関する投資やその他の決定の指針となります。

組織のソフトウェア アセットのポートフォリオには、SOA を構成するサービス、ビジネス プロセス、およびその他の補助アーティファクトが含まれる場合があります。サービスおよびその他のアセットの再利用に関する投資利益を評価するには、組織はポートフォリオの主要なアセットの潜在価値を評価し、開発の生産性に対する再利用の効果を継続的に測定し、最終的な効果を判断する必要があります。

このガイドでは、SOA ライフサイクルでのサービスの設計および作成のステージについて重点的に説明します。一般的なポートフォリオに含まれるさまざまなソフトウェア アセットの価値を評価する方法を示します。この方法は、IT 管理者、エンタープライズ アーキテクト、およびその他の関係者が、サービス指向アーキテクチャのビジネス ケースを構築し、そのビジネス効果を判断する際に使用できます。ただし、SOA の全体的な価値の計算には、SOA ライフサイクルの他のステージを対象とするメトリック (運用環境での個々のサービス パフォーマンスを含む) も含める必要があります。そのような評価については、このガイドでは取り扱いません。アセット評価プロセスは、単純なタスク (SOA にとっての関係および価値が最も明白なアセット) から始めて、徐々に拡張する必要があります。たとえば、複数のアプリケーションやプロジェクト間で一般的なセキュリティ コンポーネントまたは顧客情報サービスをすでに使用しているとします。アセット評価プロセスは、これらのサービスから開始して、次にその他のアセットに範囲を拡張する必要があります。

これによって、アセット ポートフォリオの全体的な価値の実態が、次第に包括的かつ正確に示されるようになります。

このドキュメントで説明されているプロセスと計算によって、顧客に関する長年の経験に加えて、業界で認められるベスト プラクティスが引き出されます。これらのプロセスと計算は、アセット ポートフォリオの各アセットの潜在的な再利用価値を評価し、予想投資利益 (ROI) を推定するように設計されました。個々の基準に基づきますが、このドキュメントで説明されているアセット評価プロセスを使用するいくつかの組織が、再利用によって数百万ドルの節約効果を記録しています。

 [Measuring\\_SOA\\_Success\\_workbook.xls](#) は、アセット ポートフォリオの価値の構築に役立ちます。後述の手順で、ワークシートの入力方法と計算の論理について説明します。各手順で、計算に使用されている変数は、どの組織でも固有の状況を反映した値に変更できます。

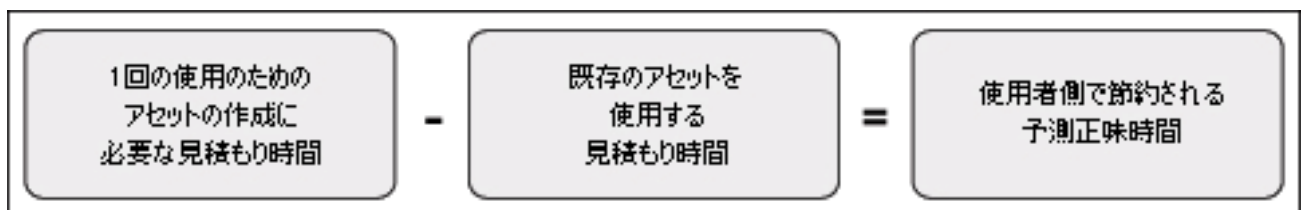
## 再利用可能なアセットとは

再利用可能なアセットは、ソフトウェア開発または SOA ライフサイクルのアーティファクトと定義できます。これには、サービス (Web サービス、データ サービスなど)、ビジネス プロセス、アーキテクチャ モデル、アプリケーション、フレームワーク、コンポーネント、WSDL ファイル、XML スキーマ、DTD (まさに文字通りすべてのコード)、メタデータ、または SOA に関連する補助アーティファクトが含まれます (ただし、これらに限定されません)。アセットに価値があれば、その価値は再利用によって拡大できます。そして、その再利用価値を、追跡および評価する必要があります。

## ポートフォリオのアセットの価値の見積もり

アセットの見積もり価値は、作成しなおすのではなく再利用することで回避される開発コストに基づきます。ここではそのような測定値を**予測節約正味時間**と呼びます (「**予測節約正味時間**」を参照)。組織は、既存のアセットを再利用することで、同じ機能を何度も開発するコスト、および同じ機能の複数の実装を維持するコストの両方を回避します。再利用によって、機能は統一され、余剰は減り、再利用が増えるほど節約効果が上がります。

回避コストの計算に使用する式はごく簡単です。方程式の最初の部分で、1 回の使用を目的としたアセットの作成に開発者が費やす時間を評価します。次に、プロジェクトで既存のアセットを再利用するために費やす時間を差し引きます。その結果が、再利用により節約される正味時間の見積もりです。



次に、各アセットの年間使用状況のレベルを考慮し、予測節約正味時間の合計をドル値に変換します。その結果が、組織のポートフォリオに存在するアセットの潜在価値の見積もりです。

# 評価プロセス

評価プロセスの手順は次のとおりです。

- **手順 1: 使用可能なアセットの一覧の作成**
- **手順 2: 適切なワークシートへのアセットの入力**
- **手順 3: アセットの生成にかかった合計時間の見積もり**
- **手順 4: 1 回の使用を目的としたアセットの構築に必要な時間の見積もり**
- **手順 5: 既存のアセットを使用するための時間の見積もり**
- **手順 6: 各アセットの年間使用状況のレベルの見積もり**
- **手順 7: 予測節約正味時間の合計のドル値への変換**
- **手順 8: 潜在的な再利用価値の計算**
- **手順 9: 進捗状況の追跡**
- **まとめ**

## 手順 1: 使用可能なアセットの一覧の作成

評価プロセスの最初の手順は、使用可能なアセットを一覧にして、カテゴリ別に体系化することです。このドキュメントで説明されているアセットと補助アーティファクトは、3 つの基本的なカテゴリに分類されます。

- サービス (およびインタフェースとして公開されるその他のアセット)
- ブラックボックス コード アセット
- ホワイトボックス コード アセット

サービス、およびインタフェースを通じて公開されるその他のアセットは、生成コストとオーバーヘッド コストが高いのが一般的です。たとえば、要件を収集して一般化し、インタフェースをパラメータ化して、広範な使用事例の下でテストするには時間がかかります。ただし、サービスや同様のアセットは、最も重要な、再利用による全体的な節約効果も提供します。

開発者は、サービスのインタフェースに簡単にアクセスし、サービスを理解して、その機能をテストでき、他の種類のアセットに必要な設計、開発、およびテストの大部分を回避できます。また、サービスの機能の実装が 1 つであるため、継続的なメンテナンス コス

トが大幅に低減されます。

ブラックボックスという用語は、アセットのコード実装がエンドユーザから隠されていることを示します。

一般に複数のユーザから呼び出される 1 つの展開インスタンスを持つサービスとは異なり、ブラックボックス アセットは開発者のコード ベースに組み込まれており、複数のインスタンスが展開されます。

比率計算機とセキュリティ フレームワークは、ブラックボックス アセットの典型的な例です。これらは、バイナリまたは難読化されたオブジェクトとして配布されます。明確に定義および文書化されたインタフェースは、アセットと周辺環境の間の通信手段として機能します。ブラックボックス アセットの開発にかかる生成コストとオーバーヘッド コストはサービスのコストと同様ですが、再利用による節約は、開発者がブラックボックス アセットをコード ベースに組み込むと発生するテスト コストおよびメンテナンス コストによって減少します。

ホワイトボックスという用語は、アセットのコード実装がエンドユーザに見えるということ、および開発者がソース コードをコード ベースに組み込むときに変更できるということを示します。Struts などのオープンソース アセットがホワイトボックスと定義されます。再利用プログラムでホワイトボックス アセットが提供する利益は、最低レベルです。これらのアセットは開発の生産性を上げることができますが、これらの使用はアセットの複数のバリエーションの展開につながり、各バリエーションを個別にサポートする必要があるため、メンテナンスの節約の点ではその価値は限られています。ホワイトボックス アセットの中心機能を変更する必要がある場合、展開されている各バリエーションを調査し、変更し、再テストする必要があります。

## 小さいものから始める

莫大な量のアセットが再利用できる可能性がある場合、一覧作成の処理は、主要アセット (最も大きな需要があり、最もよく使用されるアセット) を識別および分類しながら、徐々に進める必要があります。次に、資力の許す限り、その他のアセットに範囲を拡張できます。

## 手順 2: 適切なワークシートへのアセットの入力

 **Measuring\_SOA\_Success\_workbook.xls** をローカル マシンにダウンロードします。

[Services] ワークシートの列 A に、すべてのサービス (または公開されているインタフェースを通じて呼び出されるその他のアセット) を入力します。

[Black Box Code] ワークシートの列 A に、すべてのブラックボックス アセットを入力します。

[White Box Code] ワークシートの列 A に、すべてのホワイトボックス アセットを入力します。

### 手順 3: アセットの生成にかかった合計時間の見積もり

各ワークシートの列 B から G に、再利用可能な各アセットとすべての補助アーティファクトの生成に投入した時間を入力します。この数値は、各アセットの開発努力の合計を反映している必要があります。便宜上、時間の見積もりは個々のライフサイクル ステージに分割されます。

表 1 は、ユーザ ID を認証および確認する再利用可能なセキュリティ サービスの生成に投入された時間の例を示します。

**表 1. 例: 再利用可能なセキュリティ サービスの生成に投入された時間**

ビジネス要件とビジネス モデル作成	7 週間 (280 時間)
分析と設計	6 週間 (240 時間)
コード化と実装	8 週間 (320 時間)
統合とテスト	6 週間 (240 時間)
合計	1,080 時間

表の例に従って、組織のポートフォリオのアセットについて同じ計算を行います。この測定値の累計は自動計算され、列 H に表示されます。なお、入力数値は見積もりです。これらの見積もりは、(時間数、日数、週数、月数、または年数で測定された) 規模の順序については正確である必要があります。たとえば、見積もりが 3 週間から 6 週間の範囲である場合は、3 週間の数値を使用します。ライフサイクルの各段階の見積もり価値が表示されない場合は、合計投入時間の見積もりを列 H に入力します。

再利用価値は、よく管理された再利用環境では特に、すぐに蓄積されます。そのため、アセット評価プロセスは、「手の届く果実」のように小さいものから始める必要があるとい

う考えに沿って、再利用可能なアセットの生成に投入した時間の見積もりが控えめで「必要十分」な程度でも、再利用の価値の証拠としては説得力のあるものになります。

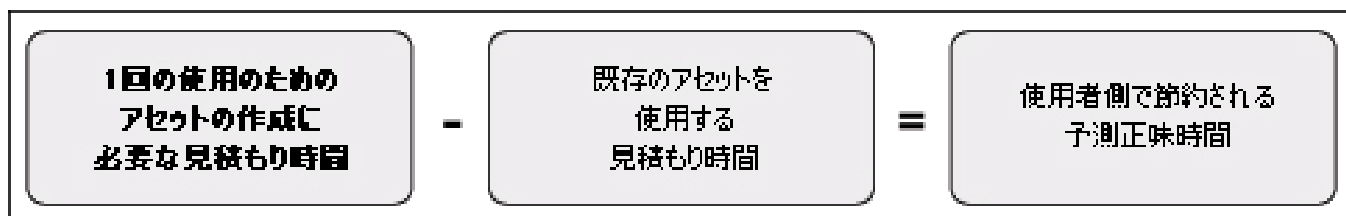
## 手順 4 : 1 回の使用を目的としたアセットの構築に必要な時間の見積もり

表 2. 再利用可能なアセットの生成投資

アセット	生成投資 (P Inv)
サービス	200%
ブラックボックス コード アセット	200%
ホワイトボックス コード アセット	100 ~ 200%

一般に、サービスまたはその他の再利用可能なアセットの構築または収集は、1 回の使用を目的としたアセットの構築よりも、時間と努力が必要です。この追加の時間と努力は、生成投資と呼ばれ (表 21 を参照)、可能な限り広範なユーザに対してアセットの再利用性と可用性を向上させるために必要な開発、パッケージ化、および文書化に関わる時間と努力が含まれます。たとえば、開発者が 1 つのプロジェクトで使用するセキュリティ機能の作成に 540 時間を投入するとします。組織が再利用可能なセキュリティ サービスを構築するには、その約 2 倍 (200%) の時間がかかります。

各アセットの再利用による節約効果の見積もりは、開発者が 1 回の使用を目的とした同等の機能の開発に費やした時間に基づいています。Measuring\_SOA\_Success\_workbook.xls では、合計開発時間と生成投資の値を使用して、その時間と努力を評価します。



たとえば、表 1 は、再利用可能なセキュリティ サービスの生成に合計 1,080 時間が投入されることを示します。再利用可能なサービスの平均的な生成投資が 200% (つまり、1 回使用のアセットの同等の機能の構築にかかる時間の 2 倍) であるとする、開発者は 1 回使用バージョンのセキュリティ サービスの構築に 540 時間を費やしたことになります。この数字 (540 時間) は、このセキュリティ サービスの再利用による節約効果の見積もりを計算するための基盤となります。

表 2 は、さまざまなカテゴリの再利用可能なアセットの生成投資を示します。多くの場合、生成投資は 100% を超えています。これは、1 回使用のアセットの構築に必要な努力に加えてさらに追加の努力が必要であることを反映しています。

ホワイトボックス アセットの再利用が、ある程度の変更を前提としていることを考えると、開発者がアセットに必要な変更を簡単に加えることができることは、そのアセットの再利用性において重要な要素です。ホワイトボックス アセットの変更を容易にするために開発者に提供される追加のドキュメントおよびその他のリソースは、生成投資の一部です。ホワイトボックス アセットの場合、この投資は 100% (アセットの再利用性を高めるために追加の努力が投入されない) から 200% の範囲にわたります。

アセットを再利用可能にするために投入される努力と、アセットを再利用するために必要な努力の間には、明らかに直接的な相関関係があります。この再利用のための努力については、次の節で説明します。

前述の値は、基準として提供されています。管理された再利用プログラムを確立するまで、実際にポートフォリオの再利用可能なアセットに関連する生成投資を測定する組織はほとんどありません。添付のワークブックの目的に応じて、これらの値はそのまま使用されたり、組織の経験を反映するように調整されたりする場合があります。適切なワークシートのセル D4 に適切な生成投資の値を入力します。

次の手順では、開発者がアセットの再利用に費やす時間である消費係数を計算および減算します。

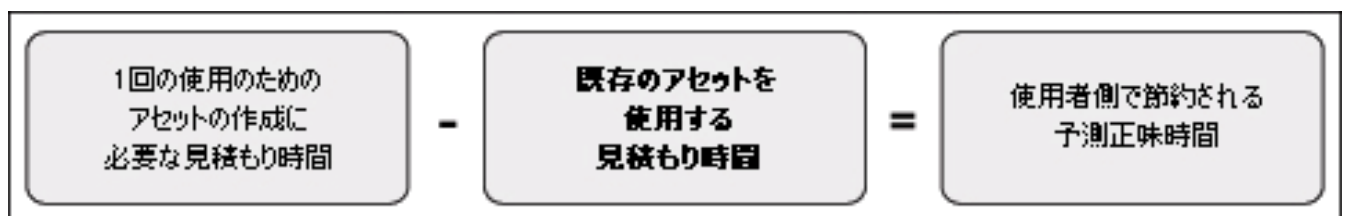
## 手順 5: 既存のアセットを使用するための時間の見積もり

### 表 3. さまざまな種類のアセットの消費係数

アセット	消費係数 (Cfac)
サービス	5%
ブラックボックス コード アセット	20%
ドメイン固有のブラックボックス コード アセット	5%
ホワイトボックス コード アセット	40% (コード変更が 25% 未満)
ホワイトボックス コード アセット	90% (コード変更が 25% を超える)

再利用にコストが発生しないわけではありません。アセットを評価してプロジェクトに再利用するには時間がかかります。この時間は、消費係数を使用して見積もります。

表 3 は、さまざまなアセットのカテゴリに関連する消費係数を示します。前に説明したように、この例で開発者は 1 回使用バージョンのセキュリティ サービスの構築に 540 時間を費やしたことになります。サービス (およびインタフェースとして公開されるその他のアセット) の消費係数は 5% です (ここでは、消費は、アプリケーションの開発において設計時にサービス呼び出しを組み込むことを指します)。前に測定した 540 時間という数字にこの消費係数を適用すると、セキュリティ サービスを再利用するには 27 時間かかるという見積もりが算出されます。



Oracle の経験から、再利用に関連する生成投資と消費係数についての多数の研究結果が裏付けられています。表 3 に、ホワイトボックス アセットのエントリが 2 つあります。最初のエントリは、アセットの再利用時のアセット コードの変更が 25% 未満の場合に適用される消費係数を反映しています。2 番目のエントリは、アセット コードの変更が 25% を超える場合に適用される消費係数を反映しています。大まかに言えば、ホワイトボックス アセットの生成投資と消費係数は反比例します。つまり、アセットを再利用可能にする生成段階で努力が投入されなかった場合 (生成投資 100%)、開発者が再利用するためにアセットの評価および変更には費やす時間は増え、消費係数が高くなります (消費係数 90%)。

表 3 には、ブラックボックス コード アセットと関連アーティファクトのエントリも 2 つあります。

最初のエントリは、ブラックボックス コード アセットの一般的な消費係数を反映しています。一方、2 番目のエントリは、アセットが同じドメイン内で生成および消費される場

合に適用される消費係数を反映しています。ドメイン固有の再利用可能なアセットを使用する場合の開発者にとっての利益が、これらの消費係数に反映されています。開発者がドメインに精通している場合、ドメイン固有のアセットの理解および再利用に必要な時間と努力は少なくなります。

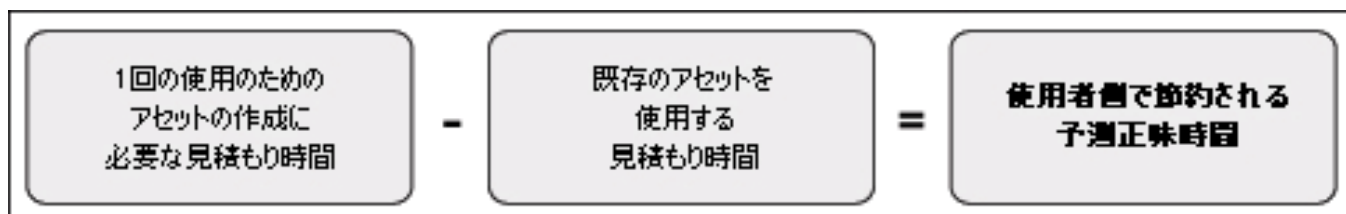
手順 4 で説明した生成投資の値と同様に、消費係数の値は単なる基準として提供され、組織の経験に応じて調整することができます。

適切なワークシートのセル D5 に適切な消費係数を入力します。

## 予測節約正味時間

開発者が 1 回の使用を目的としたアセットの構築に費やす時間と、そのアセットの再利用可能なバージョンを使用するのに必要な時間の差を、予測節約正味時間と呼びます。これが、アセットの再利用により回避される開発コストを表します。

この例のセキュリティ サービスは、1 回の使用を目的とした開発に約 540 時間かかります。前述の計算によると、開発者がこのセキュリティ サービスを再利用するためにかかる時間の見積もりは 27 時間です。この差 (513 時間) が、このセキュリティ サービスの予測節約正味時間の値です。



## 投資の優先順位付け

手順 1 ~ 5 は、アセットの再利用価値の見積もりを計算する方法を示しています。再利用のビジネス ケースの構築や、再利用可能なサービスまたはアセットへの投資の優先順位付けを担当する責任者は、次の手順 6 ~ 8 を使用して、再利用価値を予想 ROI に変換する必要があります。

ポートフォリオ内のアセットの予想 ROI を確定するには、各アセットの年間使用状況を考慮して、予測節約正味時間の合計をドル値に変換する必要があります。この変換は手順 7

で行います。

## 手順 6：各アセットの年間使用状況のレベルの見積もり

ほとんどの組織で、ある年のプロジェクト一覧に多くの開発プロジェクトがあります。しかしながら、組織のポートフォリオにあるアセットを実際に再利用する機会があるのは、これらのプロジェクトの一部だけのようです。たとえば、翌年に組織が資金を投入した 100 件のプロジェクトのうち、25% が設備に関わるもの（建物と資材の更新）であるとしています。このような性質のプロジェクトでは、ソフトウェア アセットの再利用は、現実性がないか、当てはまらない場合があります。

この例では、アセットの再利用が可能かつ適切であると見込まれるプロジェクトは、75 件 (100 件 - 25%) だけです。この数字 (75) を、適切なワークシートのセル D7 に入力する必要があります。次に、各アセットが再利用される回数を見積もります。アセット ポートフォリオにアクセスできるすべてのプロジェクトで、使用可能なすべてのアセットを利用するとは考えられません。たとえば、再利用の機会があるプロジェクトのうち、3 分の 1 のプロジェクトだけが、この例のセキュリティ サービスを再利用できます。このようにして、年間再利用回数 25 回 (75 の 3 分の 1) に変換されます。

[# Annual Reuse Opportunities] という列に、各アセットの使用状況の見積もりを入力します。

## 手順 7：予測節約正味時間の合計のドル値への変換

負荷率という用語は、アセット使用者のさまざまなロールのそれぞれに関連する 1 時間あたりのオーバーヘッド コストを指します。ソフトウェア開発者の平均的な負荷率は、ビジネス アナリストの平均的な負荷率とは異なる場合があります。負荷率は、各組織の標準である必要があります。適切なワークシートのセル D6 に組織の負荷率を入力します。ワークシートで、この負荷率を使用して各アセットの潜在的な再利用価値が計算されます。

## 手順 8：潜在的な再利用価値の計算

[Projected Annual Saving Summary] ワークシートに、すべてのワークシートの節約効果の合計が示されます。この合計は、再利用プログラムの正当性を示すための基礎、および再利用プログラムで目指す目標として使用する必要があります。

## 手順 9：進捗状況の追跡

組織の SOA および再利用の目標へ向けた進捗状況の追跡には、適切な機能を持つエンタープライズ レジストリ リポジトリが不可欠です。レジストリ リポジトリでは、各アセットの価値を追跡する必要があります。次に、アセットが使用されたら、必要なデータを収集して、SOA イニシアチブで使用されるアセットの価値を詳しく報告するレポートを生成する必要があります。

これにより、組織は予測 ROI と実際の利益を比較することができます。

## まとめ

再利用可能なアセットのサイズ、スコープ、および目的は変化することがありますが、アセットの見積もり評価（および実際の価値の正確な報告）は、ビジネス目標に合わせて調整を続けられるように SOA ガバナンスの努力を導いていく上で不可欠です。

SOA ガバナンスの努力を導いていく上でアセット評価の情報が極めて重要であるのと同様に、アーキテクチャ、IT、および企業レベルで確立されたポリシーと標準に合わせた調整、およびこれらのポリシーと標準のサポートを確実に実行してアセットの価値を高めるには、SOA ガバナンスの努力が不可欠です。最終的には、アセットの価値を決定するものは、再利用性の高さや再利用することだけでなく、組織をビジネス目標に近づけるための貢献度でもあります。そのために、アセットの設計、開発、および使用を指揮する上での効果的な SOA ライフサイクル ガバナンスの重要性は、誇張してもしすぎることはありません。

サービスおよびその他のソフトウェア アセットのエンタープライズ ポートフォリオは、SOA 内で極めて重要な役割を果たす一方で、サービス指向アーキテクチャの全体的な価値の 1 つの側面を表します。

SOA 環境のユニークな特性を考えると、運用環境でのサービスの共有や個々のサービスパフォーマンスを含むさまざまな要因が SOA の最終的な ROI に寄与します。このようなその他の要因の ROI は、さまざまな方法で計算できます。

このガイドで概説している方法は、強力なエンタープライズ レジストリ リポジトリと併せて使用すると、エンタープライズ ポートフォリオのサービスおよびその他のソフトウェア アセットの管理と強化、プロジェクトでのこれらのアセットの使用または再利用の指定、

および継続的でプロアクティブな SOA ガバナンスについて十分な情報に基づく決定をするために必要な情報を組織に提供します。

Oracle Registry Repository は、エンタープライズ ポートフォリオのサービスおよびその他のアセットに不可欠な可視性、トレーサビリティ、およびガバナンスを提供します。Oracle Registry Repository の追跡、測定、およびレポート機能は、サービスとアセットの価値を評価および活用する手段を提供します。これによって、組織は SOA への変換および SOA の発展の中で、確実にビジネスとアーキテクチャを調整し、ROI を測定できます。

Oracle Registry Repository は Oracle サービス指向アーキテクチャ ガバナンス ソリューションの 1 つのコンポーネントです。SOA の成功は、人、プロセス、およびテクノロジーを正しく組み合わせることにより実現されます。Oracle Systems は、これらの要素をまとめて、専門技術、経験、およびテクノロジーを提供し、組織が SOA の目標を達成できるよう支援します。

Oracle Registry Repository と Oracle SOA ガバナンス ソリューションの詳細については、[www.oracle.com/soa](http://www.oracle.com/soa) を参照してください。