

# システム開発の「働き方を改革」を システムの“見える化”で推進しませんか？

——既存システムの設計書を自動生成し、標準化して一元管理

プログラム更改において、設計書がない、あるいは設計書の内容が変更されていないために、影響調査に時間がかかりプログラミングのための時間が確保できない、チームでの共同作業ができない、担当者の引継ぎがスムーズにいかないなどの課題を抱えている情報システム部門は多い。このような課題を解消するにはどうすればよいのだろうか。

## 仕様書の管理不足や属人化による弊害

システム開発では、要件定義や基本設計、詳細設計の各フェーズにおいて、設計書をはじめさまざまなドキュメントが作成され、これらのドキュメントに基づいてシステム開発担当者がプログラミングを行う。この一連のドキュメントを、いかに一元管理するかが、情報システム部門の大きな課題となっている。

例えば、すでに開発したシステムに対し、現場から「ここを直してほしい」「あの機能を追加してほしい」などの要望が上がってくることは頻繁にあることだ。こうした場合、プログラムの修正が優先され、当初作成したドキュメントの変更は後回しにされる傾向にある。さらに、次のプログラム修正依頼がきて同様のことが繰り返され、プログラムは修正されているのに、ドキュメントは当初のままの状態で整合性がとれなくなりブラックボックス化してしまう。こうなると、後々のプログラム修正や担当者を変更した際に、ドキュメントが役に立たず、変更の影響調査をするのにソースコードを1からトレースしなければならないとなり、ムダな時間を費やしてしまうことになる。

また、ドキュメントは更新されていても、案件ごとや作成者ごとに記述の形式や内容が異なっていることがある。そのため、ノウハウが属人化してしまい、担当者の異動や退職による引継ぎなどで苦勞することになる。

ドキュメントが正しく管理されているのは、全体の数パーセントに過ぎないという調査結果もある。システム開発部門の生産性を向上

させ、人手不足を補うためにも、設計書をはじめとするドキュメントの一元管理や“見える化”が必要とされている。

## リバースエンジンとVSSDで 既存システムを“見える化”

こうした課題を解決するのが、第一コンピュータリソースが提供する、「システムの“見える化”サービス」(図1)だ。

これは、独自開発のリバースエンジンを使って、既存システムのソースコードを、システム設計専用ツール「Verasym System Designer (VSSD)」に取り込み、必要なドキュメントを自動生成するサービスだ。

既存のプログラムのソースコードから、VSSDでドキュメントを自動生成し、その後はVSSDを活用して変更管理ができるため容易にシステムを把握することができ、属人化を防ぐことができる。プログラム修正時の調査工数を削減し、影響範囲の調査漏れをなくすこともできるので、システム開発の生産性と品質を大幅に向上できる。

またチーム開発において、ExcelやWordでドキュメントを作成するのではなく、VSSDで常に最新の設計情報を共有

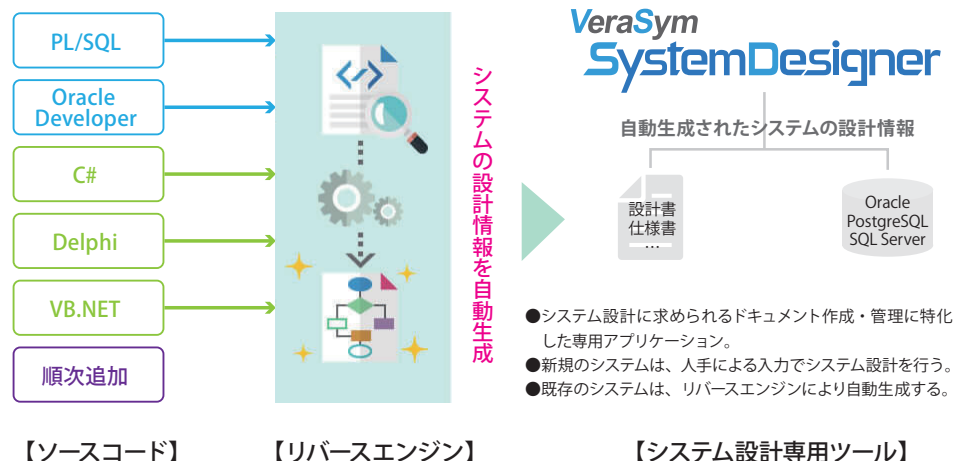


図1：システムの“見える化”サービスの概要

することで、システムの設計変更をリアルタイムに反映することも可能だ。現在、PL/SQLとOracle Developer、C#、Delphi、VB.NETをサポート。今後、要望に応じて対象ソースコードを増やしていく計画である。

## 既存システムの“見える化”による5つの効果

既存システムを“見える化”することにより、システム変更時の影響範囲をスピーディーに把握できるようになるだけでなく、設計書の一元管理、属人化防止などにより、システム開発の生産性向上や人手不足対策にも貢献する。

### ●プログラム間の関連性を容易に把握できる

WordやExcelで作成したドキュメントは、システム間の関連性を把握することが難しい。既存システムの“見える化”により、ドキュメントを変更したときに、その仕様変更がどこまで影響するかを容易に把握することができる。

### ●ドキュメントの一元管理を実現

WordやExcelなどの汎用ツールを使って作成された設計書は、案件ごと、担当者ごとに書式が違っていたり、記述内容の粒度が不ぞろいだったりする。また、担当者のPCに保管されていたり、案件ごとに保管場所が分かれていたりする。こうした課題を、ドキュメントの一元管理により解消できる。

### ●ドキュメントの属人化を防ぐ

ドキュメントが属人化するのを防ぎ、ドキュメントの担当者が部署を異動したり、退職したりした場合でも、迅速かつ的確に引継ぎすることができる。

### ●設計書の再作成工数を大幅に削減

システム開発後、数年が経過すると、一部のシステムにおいてドキュメントが残っていないこともある。こうした場合

においても、仕様変更や機能拡張を行う前に、関連するすべてのソースコードを手作業でトレースすることなく、簡単に最新の設計書を自動生成することができる。

### ●システム変更時の業務プロセスを確立

現状のプログラムの状態を設計書に反映できるため、次のプログラム変更の際から、「ドキュメントを変更してから、プログラムを修正する」という、あるべき作業プロセスを遵守できるようになる。

## システム設計専用ツール「VSSD」とは？

ここで、システムの“見える化”サービスの中核となる、VSSDについて説明しておこう。

VSSDは、システム開発における要件定義や基本設計、詳細設計のフェーズで作成する各種ドキュメントを作成するシステム設計専用ツールだ。WordやExcelで作成されてきたドキュメントにおける一元管理の課題を、一気に解決してくれる。

例えば、WordやExcelなどの汎用ツールでドキュメントを作成すると、ドキュメント内の整合性や関連するドキュメントの変更などを手作業で確認し、修正しなければならない。VSSDでは、設計書データをリポジトリで一元管理できるので、ドキュメント間の関連性や整合性を保ったまま、チームで共有することができる。

VSSDの特長は、(1)ドキュメント作成、(2)項目情報中心の設計、(3)チーム設計、(4)スパイラル設計の4つが挙げられる。

### (1)ドキュメント作成

システム開発の要件定義、基本設計、詳細設計に必要な、20種類以上のドキュメント(図2)を直感的なインターフェースで作成することができる。

システム概要	システムイメージ	業務フロー	ドメイン定義	データモデル
テーブル定義	ビュー設計	画面設計	画面遷移図	帳票設計
バッチ設計	ファイル定義	問合せ定義	ライブラリ定義	関数定義
クラス定義	ラベル定義	メッセージ定義	区分定義	検証ルール定義
CRUD表	オブジェクト関連図			

図2：VSSDで作成可能なドキュメント (Ver1.9時点)



図3：項目中心の設計

## (2) 項目情報中心の設計

各ドキュメントは、ドメイン定義（項目情報）を中心に設計することで、ドキュメント間の関連や整合性を自動的にチェックすることができる（図3）。

例えば、ユーザー管理画面の設計で、ユーザーIDを使用する場合、WordやExcelなどで設計すると、画面設計とテーブル設計、それぞれにユーザーIDが定義される。VSSDではドメイン定義にユーザーIDを定義するだけで、画面設計とテーブル設計の両方でユーザーIDを使うことが可能。ドメイン定義のユーザーIDの桁数を変更されると、画面定義、テーブル設計の両方に変更が自動的に反映されるので、整合性を維持することができる。

## (3) チーム設計

データベースに、すべての情報を保存することで、排他制御ができるため、複数のメンバーで同時にドキュメントを作成、変更することができる（図4）。

例えば、ドキュメントを変更する場合、まずはドキュメントをチェックインすることで、ほかのメンバーが同じドキュメントの変更をできなくし、ドキュメントの一意性を保つことができる。また、管理者による権限設定により、メンバーごとに使える機能を細かく制限することもできる。

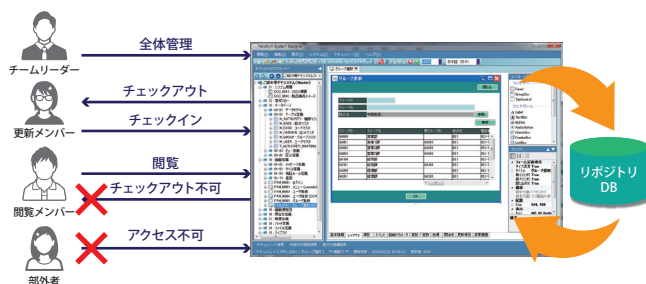


図4：排他制御と制限管理

## (4) スパイラル設計

VSSDでは、「ドキュメントを作成」「モックアップを自動生成」「会議などで検討」「修正点を反映」を繰り返す「スパイラル

ル設計」（図5）を実現することができる。これにより、会議などの場で、ドキュメントを修正し、すぐにモックアップに反映することができる。また、設計書とモックアップを別々に作成するための余分な工数を削減することができるほか、別々に管理することによる設計書との不一致を防ぐことができる。



図5：スパイラル設計

## 代表的な8つの機能でドキュメントを効果的に生成

VSSDの代表的な機能は、(1) 変更通知と反映、(2) システム間の参照、(3) ドキュメントの継承、(4) オブジェクト関連図、(5) ドキュメント出力、(6) データベースとの連携、(7) オブジェクトリンク、(8) ローカルキャッシュの8つである。

### (1) 変更通知と反映

ドキュメントを変更するとき、チェックアウトすることで担当者に権限が与えられ、ほかの担当者は同じドキュメントを変更できなくなる。チェックインをしてドキュメントの変更を確定すると、関連するドキュメントに対し、変更内容がアラート通知される。アラート通知を受け取った担当者は、関連するドキュメントを開き変更を自動反映させることで、整合性が保証される。

## (2) システム間の参照

VSSDでは、「システム」という単位でドキュメントを管理する。このとき、ほかのシステムのドキュメントを参照するという設定をすることで、ほかのシステムで定義されているテーブルやドメインなどを参照することができ、これによりシステム間の設計情報も一元管理が可能となる。

## (3) ドキュメントの継承

VSSDでは、画面や帳票に継承機能があり、設計情報を継承することができる。継承機能により、マスター画面のレイアウトや項目、処理記述などを変更すると、マスター画面を継承して作成した画面の内容も自動的に変更が反映される。

## (4) オブジェクト関連図

テーブルの項目や共通関数などを使用している箇所をビジュアルにツリー表示することができる。ツリー上のオブジェクトをダブルクリックすると、ツリー表示をドリルダウンすることもできる。これにより、仕様変更時の影響確認の精度の向上や工数削減を実現できる。

## (5) ドキュメント出力

VSSDで設計された情報は、ドキュメントとしてExcelなどに出力することが可能。基本的には標準フォーマットで出力されるが、ユーザー指定のフォーマットにあわせて出力することもできる。

## (6) データベースとの連携

VSSDで設計したテーブル定義からDDL文を出力することが可能。既存のテーブル定義を規定のCSV形式で記述することで、テーブル定義を一括で取り込むこともできる。ドメイン・テーブル分析支援による取り込みもできる（有償サービス）。

## (7) オブジェクトリンク

ドキュメント内のオブジェクトや、ほかのドキュメントを一覧から選択することが可能。選択した結果はハイパーリンクとして表示される。リンク先に変更があれば、アラート通知により把握し、反映することが着る。

## (8) ローカルキャッシュ

ローカルディスクにリポジトリをキャッシュすることで、ドキュメントへのアクセス速度が大幅に向上した。ローカルキャッシュにより、ネットワークから切断されたオフライン状態でもドキュメントの編集作業を行うことが可能。オフラインで作業した内容は、サーバに接続したときに同期される。

## VSSDからソースを自動生成する「VSAG」で、さらなる効率化

既存システムの“見える化”だけでなく、VSSDで作成した設計書から、ソースコードを自動生成することもできる。そのため製品が、プログラム自動生成ツール「Verasym Application Generator (VSAG)」だ。

ソースコードを自動生成することで、コーディング量を削減することができ、設計の生産性と品質が向上する。また、常に設計書とソースコードが一致するので、設計書の陳腐化を防止することもできる。

VSAGでは、第1弾として「JavaWeb版」を、第2弾としてウィングアーク1st株式会社の帳票作成・運用ツール「SVF」のソースコードを自動生成する「SVF版」を、第3弾として「.NET版」をリリース。今後、さらにターゲットとなるシステムに合わせ、ラインナップを拡充していく計画。また、顧客の要望にあわせた「Custom版」を開発するサービスも提供している。

### 事例：プログラムの変更業務における影響調査時間を短縮

小林クリエイティブ株式会社では、「ドキュメントの書式が統一されておらず管理場所が分散している」「ドキュメントの属人化によりスキルを完全に継承できない」「仕様変更時にほかのシステムへの影響範囲が把握できない」といったさまざまなドキュメント管理の課題を抱えていた。

そこで、VSSDを導入し、既存システムの“見える化”に着手。プログラムの変更業務における影響調査時間を短縮し、プログラミングにかかる時間の割合を10%向上したほか、ドキュメントの不備に起因するミスや手戻りを大幅に低減。プログラムの修正時に、「まずドキュメントを変更する」という現場の意識改革も実現している。