

ソースコード診断

ソースコードを図面化し、設計構造を評価し、良い点と改善点を列挙します。

使える図面

ソースコードから図面を作り
設計構造の改善箇所を列挙します

分かりやすいコードへ

関数／ファイル／フォルダ
がシンプルになります



在庫を資産へ変換します



残業

不具合

ストレス

迅速

高品質

活き活き

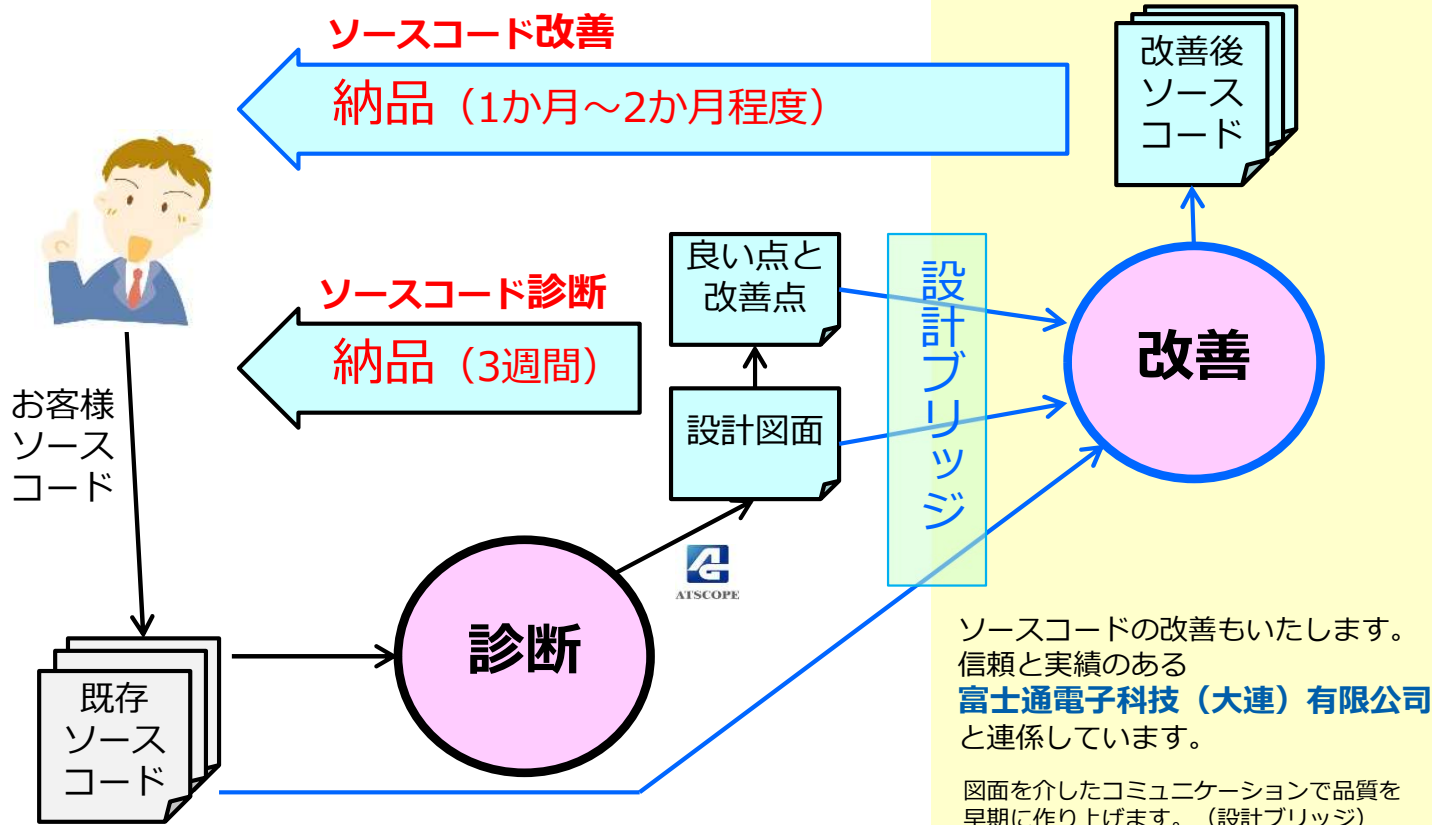
変換

ソフトウェア**在庫**

ソフトウェア**資産**

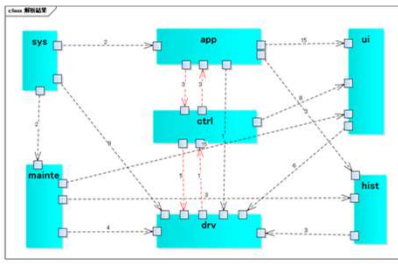
ソースコードのみで構造品質の診断を行います

- ・設計ドキュメントは不要です
- ・コンパイル環境も不要です



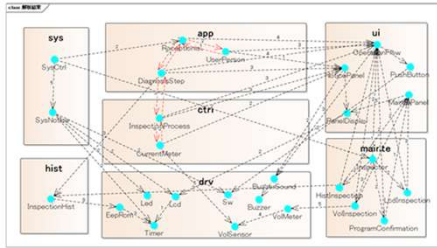
ソースコード診断の例

コンポーネント構造図：フォルダ単位の構造図



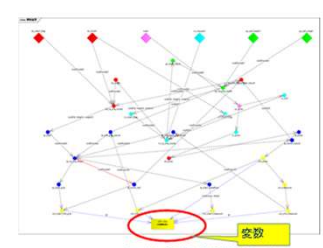
ファイル見取図：

フォルダ内部のファイル同士の呼び出し関係
「●」がファイル、外側の四角はフォルダ、ファイル間の破線は関数呼び出し



関数構造図（変数起点）

変数を起点に、呼び出し構造を選びます
最下層の四角が変数、「●」は関数、◇は親なし関数



ソースコードの特徴（良い点）

No	特徴	説明
1	わかりやすい名前付け	ファイル名、関数名、変数名が問題ドメインの名称になっている。 とても理解しやすい。
2	短い関数	ひとつの関数が、画面スクロールしなくても全体が見えるものが多い。 単体テストもしやすい。
3	部品化	.hと.cがペアになっている。 再利用し易いソフトウェアとなっている。
4	割込み処理の分離	割込み関数がひとつのファイルに集まっている。 イベント順序の違いなどの、際どい動作を追いかけやすい。
5	規則的な処理ロジック	機能ごとにmainがあり、規則性があるため修正箇所を見つけやすい。

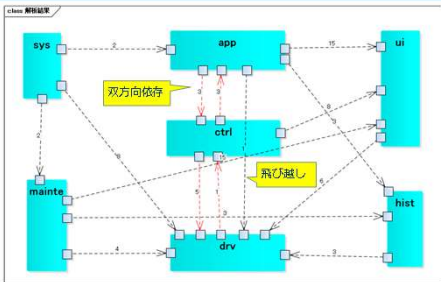
ソースコードの特徴（改善点）

No	特徴	説明
1	大きなファイル	10,000行を超えているファイルがある。
2	ハイブリッド結合の状態で変数	状態変数の取りうる値が7を超えている。
3	割込み内での処理	割込み処理内で長い処理をしている。
4	グローバル変数	変数がファイル外へ公開されている。
5	フラットなフォルダ構成	ひとつのフォルダ内に全てのファイルがある。
6	main処理が一筆書き	mainという関数内部で、複数の処理をしている。ファンアウト数が7を超えている。

設計改善施策のご提案：トップ5

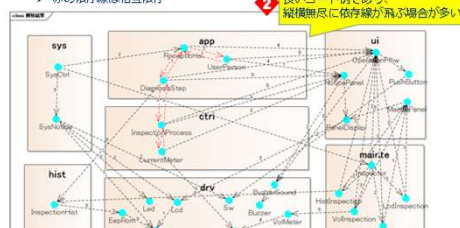
No	項目	予測される問題点	設計改善施策
1	状態変数	状態変数の取りうる値が多く、知っている人しか修正できない。変更の影響範囲の見極めが困難、かつ、テストしきれない。	状態変数を分割する。 データ設計。
2	ファイル分割	内部を追いかけていないと修正できない。少しの変更で影響箇所が多い。	ファイルを分割する。 リファクタリング。
3	データ競合	親なし関数間で共有する変数があり、イベントの発生順序によって再現性の低い障害を誘発している。	制御スレッドを横断する変数をガードする。
4	アーキテクチャ構造	全体構造が見えてこないため、習得のリードタイムがかかる。担当者の引き継ぎが困難。	フォルダ分けして、アーキテクチャ構造を明確にする。
5	カプセル化	グローバルデータが多いため、修正時に副作用を生じやすくなる。不具合の修正がなかなか収束しない。	フラグを構造体へ、変数をファイル内部へカプセル化する。static宣言。

コンポーネント構造図：フォルダ単位の構造図



ファイル見取図：

- フォルダ内部のファイル同士の呼び出し関係
 - がファイルで、ファイル同士の関係を明確できる
 - 赤の依存線は相互依存



関数構造図（変数起点）

- 変数を起点に、呼び出し構造を選びます
 - は関数、◇は親なし関数
 - 制御スレッドが分かるので、変数を排他制御している重要箇所も見えてきます



コード起点アプローチ

リバースモデリング

洗練化

資産化

資産運用

開発戦略

目論見

設計方針

変動点

プロダクトライン
開発

アーキテクチャ
ドキュメント

統合資産

資産活用

資産開発

合わせこみ

部品マスター

機種開発

洗練化

部品表

インフラ

ソース
コード

部品化

設計図

③プロセス構築 ④ツール導入

①教育 ②指標と計測

ソースコード診断は
この部分に相当します
価格：160万円

ソースコード改善は
この部分に相当します
価格：個別お見積り