



技術的負債リスク チェックリスト

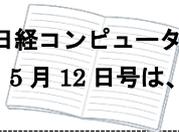
のご案内

2022年5月

システム企画研修株式会社

ご案内の背景

日経コンピュータ誌
5月12日号は、



「技術的負債に向き合う」

という特集記事を掲載しました。
以下がその前置きです。

システムのブラックボックス化が引き起こす「技術的負債」の問題に、日本企業が真剣に向き合い始めた。

システムの改修や新機能追加にかかるコストが膨れ上がり、ITによる事業強化や業務改革といったDXの足かせになる事態を防ぐためだ。

旧来のアーキテクチャの見直しからクラウドをはじめとした最新技術の取り込み、生産性や保守性の見える化と継続的な改善、経営陣と開発陣の密な連携まで課題は山積み。

先進企業の取り組みを通して、長く険しい、技術的負債に立ち向かう道のりを追った。

- ▶ ご承知のようにこのテーマは、経済産業省が2018年の「DXレポート」の中で「2025年の崖」として問題提起したものです。
- ▶ この問題提起にもかかわらず、その後、大半の企業においては、「検討はしているが、何らの手も打っていない」という状況でした。

今回の特集は、「**2025年の崖**」対策に取り組みだした先進企業の紹介記事なのです。

- ▶ そもそも、技術的負債（Technical debt）という言葉の元祖は、1992年にウォード・カニンガム氏で、「ソフトウェアの複雑さがだんだん増していった、機能追加や適切な改修が難しくなっている現象」とされています。
- ▶ まさに「2025年の崖」のことを指しています。

この特集には以下の解説があります。

具体的な対策は、**システム構造の見える化**、特に**システム関連ドキュメントの整備**は、技術的負債問題の悪化を防ぐ重要なポイントとなる。

開発生産性の見える化も有効な対処策である。
何らかの数字や指標を定めて開発効率を測るようになれば、**問題が悪化している兆候をつかんだり改善活動の効果を検証したりするためのよりどころとなる。**

ということで、今回ご案内するチェックリストは、まさにそのような観点で

「技術的負債」が大きくなるリスクを定量的に捉えよう

とするものです。



技術的負債リスク チェックリスト

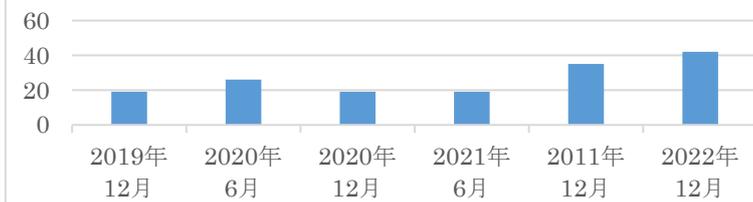
のご案内

1. 当チェックリストご利用の「目的・ねらい」(Why)

(1) 当チェックリストのご利用目的

- ① 各部門において「技術的負債」を発生させる要因を継続的にチェックしていただきます。
- ② そのチェック結果を定量的かつ継続的に把握することによって、「技術的負債」発生リスクの増減を把握します。

保守業務リスクスコア



(2) 当チェックリストのねらい

- ① 「技術的負債」発生リスクの大きな要因が明確になった場合は、当チェックリストで参考提示する方法も参考にしてその対策を検討していただきます。
- ② それにより、技術的負債の発生が低減されます。
- ③ マイグレーション実施の判断基準とすることもできます。

2. 当チェックリストの内容 (What)

❖ 当チェックリストの構成は以下のとおりです。

(1) チェックリスト本体

- 開発業務 (組織単位でチェック)
 - 保守業務 (システム単位でチェック)
 - マイグレーション業務 (システム群単位でチェック)
- の3種 (後掲抜粋をご参照ください)

(2) チェック項目解説

チェック項目の種類により、以下が解説されます。
(後掲サンプルをご参照ください)

- 1) チェック項目の意味解説
- 2) チェック項目の必要性解説
- 3) チェック項目の悪化要因の想定解説
- 4) チェック項目の悪化要因に対する対策案の参考提示

4. 当チェックリストのご利用方法 (How)

- ❖ 内容の追加・変更はご自由です。
- ❖ 随時、弊社からチェック項目および解説内容の追加をご提供します。

3. 当チェックリストの対象 (Where)

❖ 以下の3種のチェックリストがあります。

- (1) 情報システム開発：現在 29 項目
- (2) 情報システム保守：現在 31 項目
- (3) 情報システムマイグレーション：現在 8 項目

5. 当チェックリストご提供時期 (When)

❖ 現在ご提供開始中

6. 当チェックリストの・ご提供者 (Who)

❖ システム企画研修株式会社

7. 当チェックリストご提供料金 (How Much)

❖ ご提供チェックリストの永久使用権料金
システム部門または情報サービス業社員数

- 100人未満 : 50万円 (消費税別)
- 200人未満 : 60万円 (同上)
- 300人未満 : 70万円 (同上)
- 500人未満 : 80万円 (同上)
- 1,000人未満 : 90万円 (同上)
- 1,000人以上 : 100万円 (同上)



技術的負債リスク チェックリスト

のご案内

「技術的負債リスク」チェックリスト（抜粋） 保守業務編

チェック項目の内容を確認したい場合は、解説欄をクリックしてください。

システム名	チェック年月日	チェック者		
中項目	チェック項目	<input type="checkbox"/>	解説	
生産性	案件ごとの保守業務生産性を把握しているか。	<input type="checkbox"/>	M17	
	年間ごとの保守業務生産性を把握しているか。	<input type="checkbox"/>	M18	
	年間ごとの保守業務生産性の低下要因を把握しているか。	<input type="checkbox"/>	M19	
	生産性把握方式は、的確に生産性を把握していると言えるか。	<input type="checkbox"/>	M20	
システムの見える化	当システムのドキュメント体系は設定されているか。	<input type="checkbox"/>	M26	
	必要ドキュメント体系は常時更新されているか。	<input type="checkbox"/>	M27	
	必要ドキュメント体系の内容は信頼できるか。	<input type="checkbox"/>	M28	
	システム構造の見える化できるドキュメントはあるか。	<input type="checkbox"/>	M29	
	システム構造見える化ドキュメントは常時更新されているか。	<input type="checkbox"/>	M30	
	システム構造見える化ドキュメントの内容は信頼できるか。	<input type="checkbox"/>	M31	

「技術的負債リスク」チェックリスト（抜粋） 開発業務編

担当組織名	チェック年月日	チェック者		
中項目	チェック項目	<input type="checkbox"/>	解説	
開発方式	この3年間で開発の生産性を高めるための新開発方式の検討・導入をしているか。	<input type="checkbox"/>	D14	
	この3年間で保守の生産性を高めるための新開発方式の検討・導入をしているか。	<input type="checkbox"/>	D15	
	保守の生産性を高めるための新開発方式導入の効果を把握しているか。	<input type="checkbox"/>	D16	
開発業務実施方法（注）	開発業務の見積り誤差の分析をしているか。	<input type="checkbox"/>	D17	
	この1年間で開発業務の見積り方法の改善をしているか。	<input type="checkbox"/>	D18	
	この1年間で開発業務の要件定義方法の改善をしているか。	<input type="checkbox"/>	D19	
	この1年間で開発業務の外部設計方法の改善をしているか。	<input type="checkbox"/>	D20	
	この1年間で開発業務の内部設計方法の改善をしているか。	<input type="checkbox"/>	D21	
	この1年間で開発業務のテスト方法の改善をしているか。	<input type="checkbox"/>	D22	
	この1年間で開発業務のリリース方法の改善をしているか。	<input type="checkbox"/>	D23	

注：組織内で共有したものを対象とする。

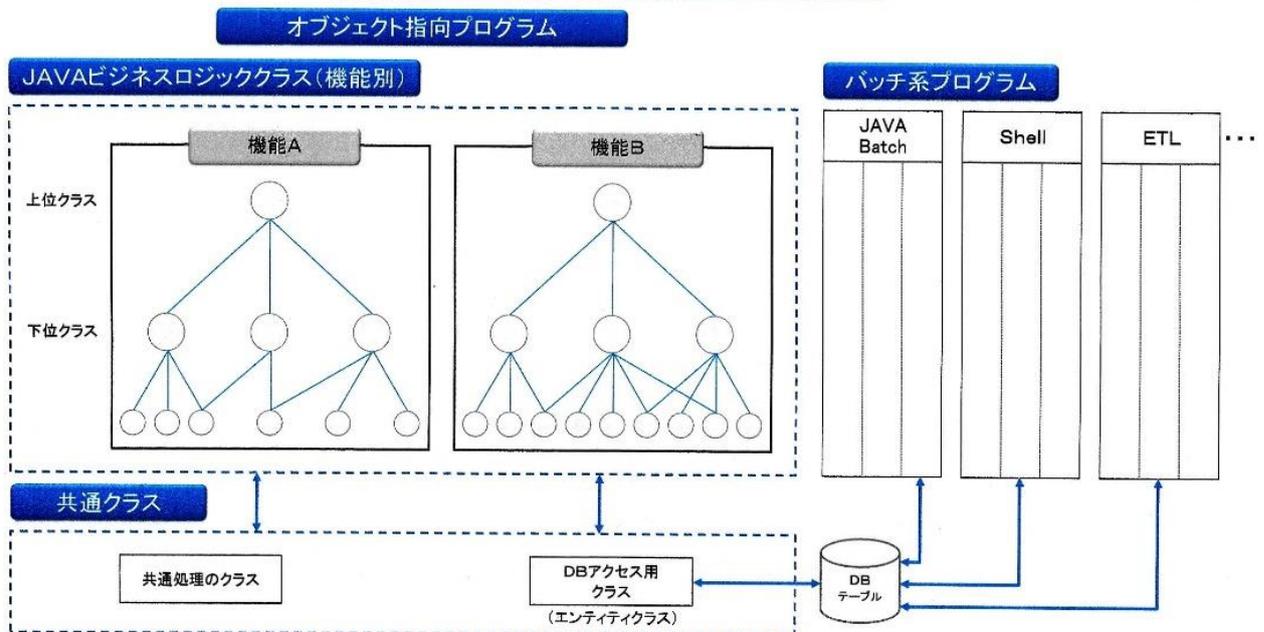


チェック項目解説サンプル

保守	システムの見える化	システム構造の見える化できるドキュメントはあるか
----	-----------	--------------------------

- 保守にとっては、どこかの変更がどこに影響するのかの判断材料が最も必要なものである。
- したがって、まずは全体構造図が必要である。
- [オープン系システムの全体構造図の基本型](#) はこれである。

オープン系システムの構造



- [従来型システムの全体構造図の基本型](#) はこれである。(掲載省略)
- 次いで、変更対象となるシステムコンポーネント（入力、テーブル、等）と処理ロジックとの関係を示す表も必要である。
- この方法の具体化を検討した研究会では、この表をRマトリクスと称した。RはRefer（参照）のRをとっている。
- [オブジェクト指向システムのRマトリクスの例](#)を示す。(掲載省略)
- JAVAプログラムの場合は、データ項目の使用先クラスが抽出できるツールがあるので、Rマトリクスは不要である。



技術的負債リスク チェックリスト

のご案内

- 従来型のシステムのRマトリクスの例を示す。

影響区分記入Rマトリクスの事例

	横が処理系									
	受入力画面処理	受入力に基づくマスタ更新処理	受注データファイル更新処理	出荷依頼処理	出荷処理	売上計上処理	売上実績処理	売上実績表	新商品売上実績マスタ更新	新商品売上実績データ作成
得意先マスタ	R		R		R					
商品マスタ	R		R							
受注画面データ	R C0	R C5								
受注マスタ	R	R C0	R C0	R C1		R C3				
出荷依頼ファイル				R C1	R					
出荷実績ファイル					R	R				
売上ファイル						R C1	R			
新商品売上ファイル						R FC0			R FC4	
新商品売上実績マスタ(新設)									R FC0	R FC4
売上実績ファイル							R	R		
新商品売上実績データファイル										R FC0

縦がファイル系

- このRマトリクスが完全に更新されていれば、データ項目系の変更があった場合に、最大限度どこの処理（クラス）に影響するかを把握できるので、影響調査の範囲を限定することが可能となる。