

# 要件定義の5W1H

2023/06/27 垣野内

# 目次

1. 要件定義とはなにか, なぜ要件定義なのか
  - a. 要件定義の定義
  - b. 要件定義の類型
2. いつ要件定義をすべきか
3. 誰とどのような要件定義をすべきか
4. (余談) 技術力と要件定義 あるいはキャリア戦略

# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の定義)

# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の定義)

“作ってほしい人と作る人の間で

「これを実現しましょう」

という合意を定めること”

『はじめよう!要件定義』まえがきより

# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の定義)

この定義を採用すると、  
われわれソフトウェアエンジニアは、  
多かれ少なかれ日常的に  
要件定義をしていることになる。

# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の定義)

しかし、一般的なイメージはおそらくこう。



# 〇〇〇〇サンプルシステム 要件定義書

作成者	
作成日	
最終更新日	
変更番号	

株式会社 サンプルシステム

## 目次

1. 概要	2
1.1. システムの目的	2
1.2. システムの範囲	2
1.3. システムの環境	2
1.4. 期待の機能	2
1.5. システムの非機能要件	2
2. 機能要件	3
2.1. システムの機能	3
2.2. 外部との連携・形式	3
2.3. 外部との連携・形式	3
2.4. 画面一覧	3
2.5. 画面レイアウト	3
2.6. 検索一覧	3
2.7. 検索結果	3
2.8. パソコン画面一覧	3
2.9. データ一覧	3
2.10. データ一覧	3
2.11. データ一覧	3
2.12. データ一覧	3
2.13. データ一覧	3
3. 非機能要件	3
3.1. システム形式	3
3.2. システムの環境	3
3.3. 性能	3
3.4. 拡張性	3
3.5. 保守性	3
3.6. 信頼性	3
3.7. セキュリティ	3
3.8. 可用性	3
3.9. 操作性	3
3.10. 運用	3
3.11. 保守	3

## 1. 概要

1.1. システムの目的	2
1.2. システムの範囲	2
1.3. システムの環境	2
1.4. 期待の機能	2
1.5. システムの非機能要件	2

## 2. 機能要件

2.1. システムの機能	3
2.2. 外部との連携・形式	3
2.3. 外部との連携・形式	3
2.4. 画面一覧	3
2.5. 画面レイアウト	3
2.6. 検索一覧	3
2.7. 検索結果	3
2.8. パソコン画面一覧	3
2.9. データ一覧	3
2.10. データ一覧	3
2.11. データ一覧	3
2.12. データ一覧	3
2.13. データ一覧	3

## 3. 非機能要件

3.1. システム形式	3
3.2. システムの環境	3
3.3. 性能	3
3.4. 拡張性	3
3.5. 保守性	3
3.6. 信頼性	3
3.7. セキュリティ	3
3.8. 可用性	3
3.9. 操作性	3
3.10. 運用	3
3.11. 保守	3



# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の定義)

そこで、要件定義のスタイルにいくつか名前をつけることにする。

(垣野内の独自定義です 🙏 )

# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の種類)

ドキュメントベース要件定義

(いわゆる要件定義)

デザインベース要件定義

(ワイヤー, カンプ, モックなどとも言う)

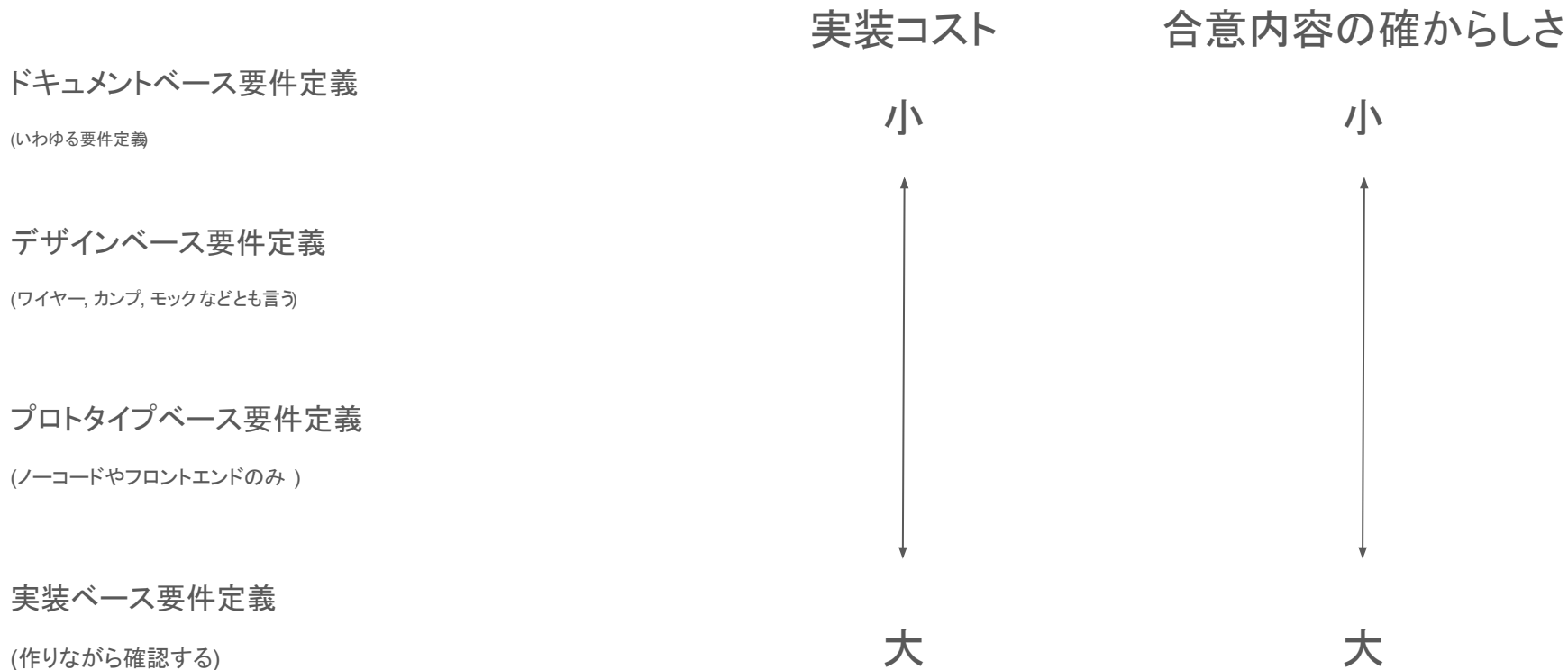
プロトタイプベース要件定義

(ノーコードやフロントエンドのみ)

実装ベース要件定義

(作りながら確認する)

# 1. 要件定義とはなにか, なぜ要件定義なのか (要件定義の種類)



# 目次

1. 要件定義とはなにか, なぜ要件定義なのか
  - a. 要件定義の定義
  - b. 要件定義の類型
2. いつ要件定義をすべきか
3. 誰とどのような要件定義をすべきか
4. (余談) 技術力と要件定義 あるいはキャリア戦略

## 2. いつ要件定義をすべきか

## 2. いつ要件定義をすべきか

以上の議論を踏まえると、問いはこうあるべき。

## 2. ~~いつ要件定義をすべきか~~ いつ、どのスタイルの要件定義が有効か

以上の議論を踏まえると、問いはこうあるべき。

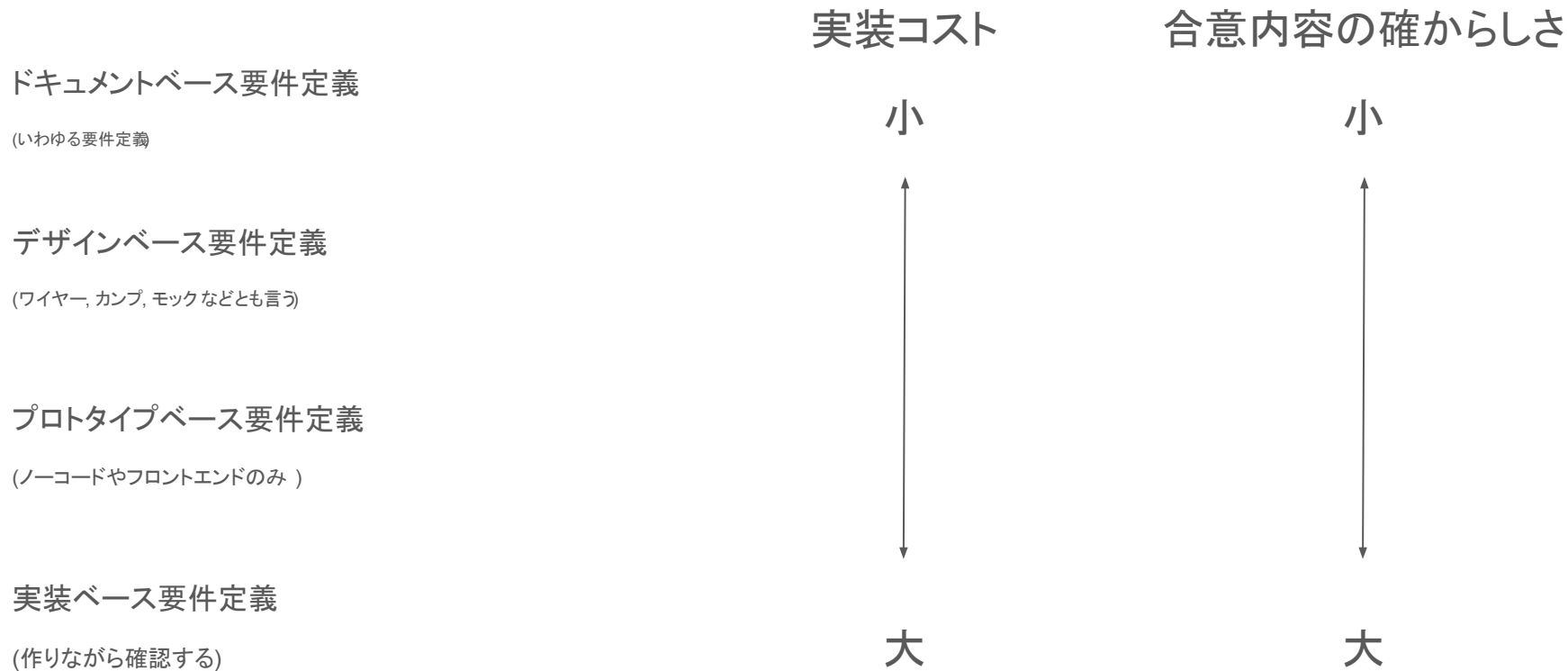
## 2. ~~いつ要件定義をすべきか~~ いつ、どのスタイルの要件定義が有効か

以上の議論を踏まえると、問いはこうあるべき。

いずれのスタイルの要件定義も、手段に過ぎないので、適材適所で使うべき。



## 2. ~~いつ要件定義をすべきか~~ いつ、どのスタイルの要件定義が有効か



## 2. ~~いつ要件定義をすべきか~~ いつ、どのスタイルの要件定義が有効か

垣野内の経験上、ドキュメントベース要件定義が有効なのは次のケースです：

- 作ってほしい人と作る人との間のコミュニケーションコストが高い時
  - 例：開発を外注する場合（暗黙知が伝わりづらいため、実装ベースのスタイルに寄せると手戻りがち
- 作りたいものの実装コストが高い時
  - 例：いわゆる Sler が対象とするようなシステム
  - 例：(今回のように) 大きめのシステムを導入する時
  - 例：(ログインや決済など) 制御が複雑な場合

# 目次

1. 要件定義とはなにか, なぜ要件定義なのか
  - a. 要件定義の定義
  - b. 要件定義の類型
2. いつ要件定義をすべきか
3. 誰とどのような要件定義をすべきか
4. (余談) 技術力と要件定義 あるいはキャリア戦略

### 3. 誰とどのような要件定義をすべきか

### 3. 誰とどのような要件定義をすべきか

以降は、ドキュメントベース要件定義の話にしぼります。

### 3. 誰とどのような要件定義をすべきか

以上の議論を踏まえると、問いはこうあるべき。

### 3. ~~誰とどのような要件定義をすべきか~~

ドキュメントベース要件定義で  
合意内容の確からしさを高めるにはどうすればよいか

以上の議論を踏まえると、問いはこうあるべき。

### 3. ~~誰とどのような要件定義をすべきか~~

ドキュメントベース要件定義で  
合意内容の確からしさを高めるにはどうすればよいか

キーワードは、網羅性。



### 3. ~~誰とどのような要件定義をすべきか~~

ドキュメントベース要件定義で  
合意内容の確からしさを高めるにはどうすればよいか

ここからは垣野内の説明能力を超えるので、

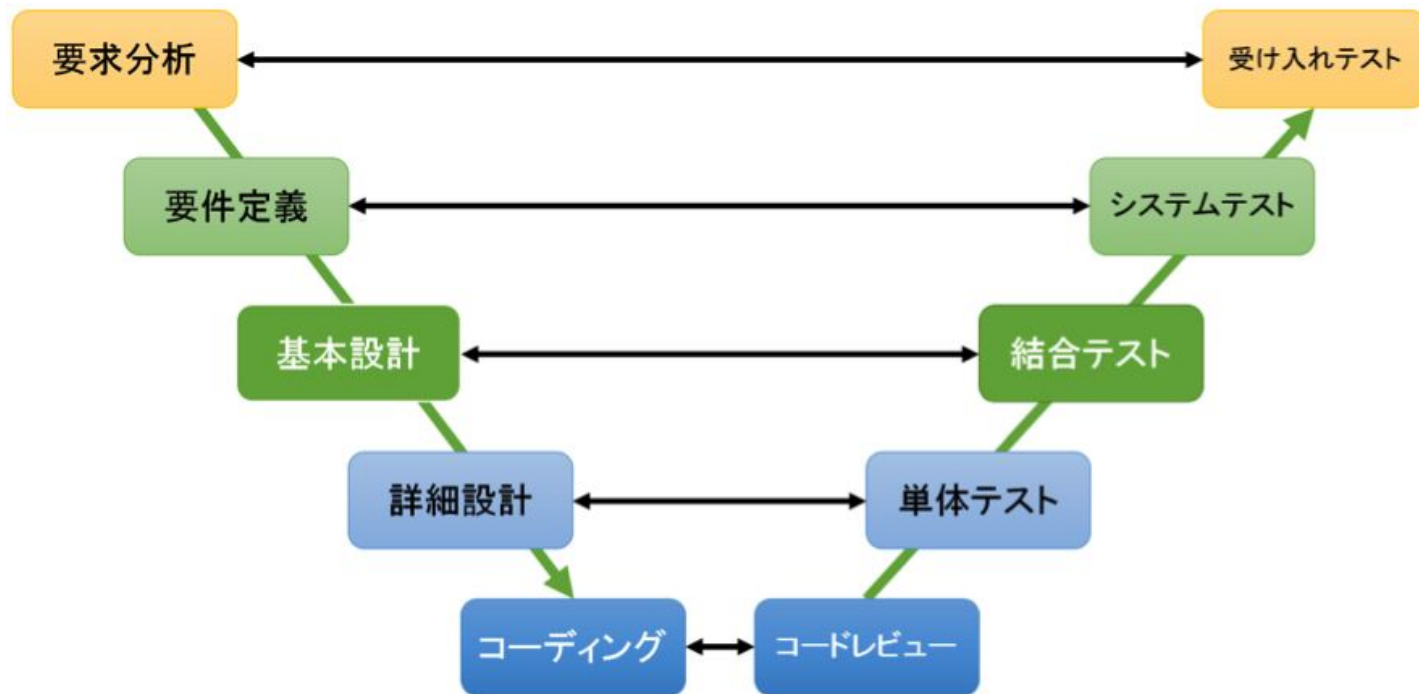
『ユースケース実践ガイド』に譲ります...

# 目次

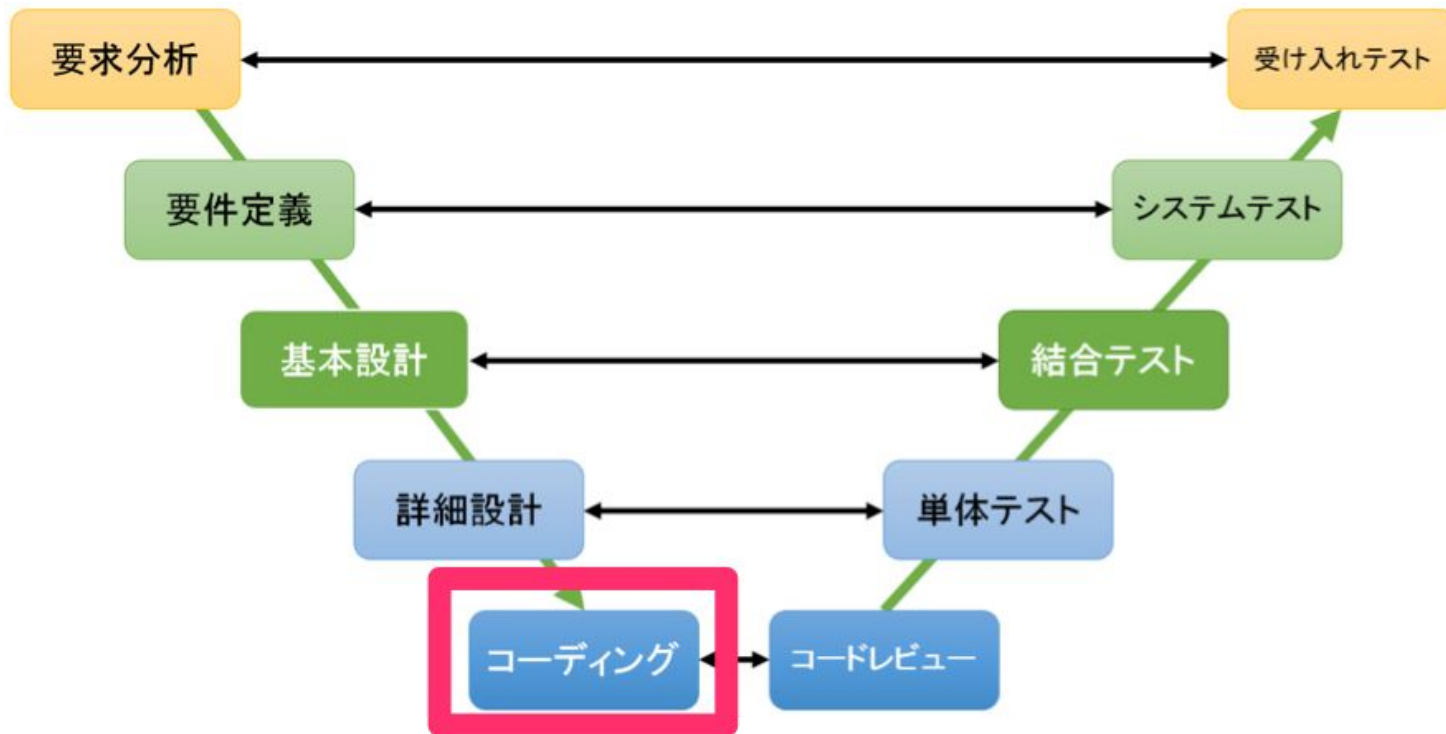
1. 要件定義とはなにか, なぜ要件定義なのか
  - a. 要件定義の定義
  - b. 要件定義の類型
2. いつ要件定義をすべきか
3. 誰とどのような要件定義をすべきか
4. (余談) 技術力と要件定義 あるいはキャリア戦略

## 4. (余談) 技術力と要件定義 あるいはキャリア戦略

## 4. (余談) 技術力と要件定義 あるいはキャリア戦略



## 4. (余談) 技術力と要件定義 あるいはキャリア戦略



## 4. (余談) 技術力と要件定義 あるいはキャリア戦略

