

# 要件定義技法ガイド DFD記述ガイド編

---

第1.10版

2018年08月29日



この作品は [クリエイティブ・コモンズ 表示 - 継承 4.0 国際 ライセンス](https://creativecommons.org/licenses/by-sa/4.0/) の下に提供されています。

要件定義フレームワーク©2018 TIS INC. クリエイティブ・コモンズ・ライセンス(表示-継承 4.0 国際)

---

# 1. DFD概要

# 1. DFD概要

## ■ 本書の目的

本書の目的は、「要件定義フレームワークにおけるDFDの利用目的・用途の理解とDFDの記述方法をガイドする」ことです。

## ■ DFDとは

DFDとは、Data Flow Diagramの略称であり、データが業務・システムをどのように流れるかを視覚的に表現したものです。インプットとしてデータが業務やシステムに取り込まれてから、処理及び格納され、アウトプットとして外に出ていくまでの流れを示します。DFDは、プロセスの流れを描くのではなく、データの流を描くために利用します。

※DFDのイメージ及びDFDを構成する要素については、図1-1、表1-2を参照して下さい。

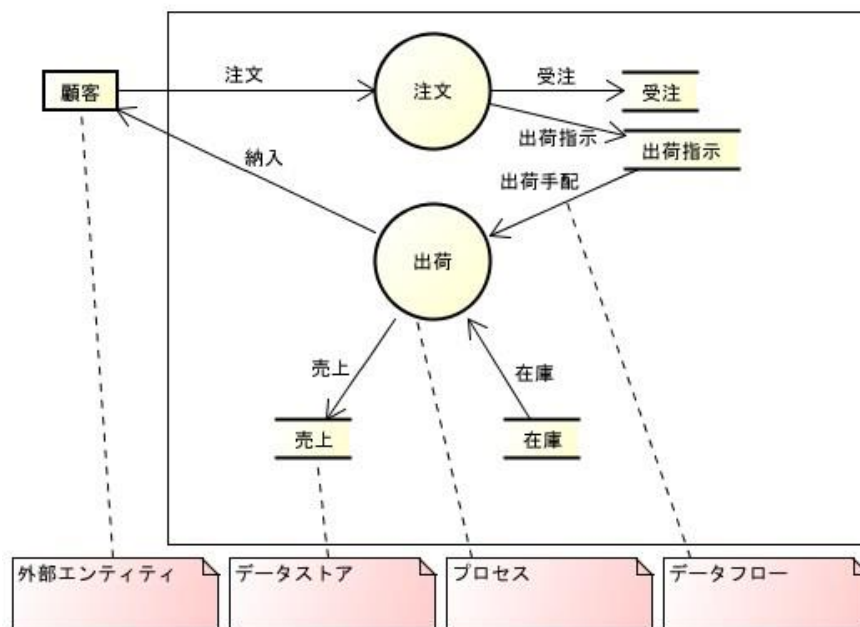


図1-1. DFDのイメージ

# 1. DFD概要

要素名	内容
外部エンティティ	データの情報源または出力先を指す。 モデル化対象となる業務やシステムの外部に存在する人・組織、システムが該当する。
プロセス	何らかの方法で、インプットデータを処理・加工するものを指す。 業務プロセスやシステム処理などが該当する。
データストア	データの保管場所を指す。 帳簿などのファイルや、データベースのエンティティなどが該当する。
データフロー	プロセスと外部エンティティ間やプロセスとデータストア間などのデータの流れを指す。

表1－2. DFDの構成要素

## ■ DFDの特徴

DFDの特徴は、下表の通りです。

組織や場所にとらわれずに、業務データの流れの視点で純粋に業務を分析することにより、業務の目的を達成するために必要となる業務を見出すことができます。

表現可否	内容
○：表現可能	業務や処理の範囲が特定される。
	どんな業務や処理が実施されるかが記述される。
	「データの流れ」が具体的に記述される。
×：表現不可能	業務や処理の主体者は、記述されない。
	業務や処理の「順序」、「タイミング」は、記述されない。
	業務や処理の「実施場所」や「実施理由」は、記述されない。

表1－3. DFDの特徴

[日経 × TECH『図解で極める要件定義 木村さんが指南するDFDの上手な書き方』[5] 図1を参考に作成]

---

## 2. 要件定義フレームワークにおけるDFD利用

## 2. 要件定義フレームワークにおけるDFD利用

本章では、要件定義フレームワークでのDFDの利用目的・用途などについて説明します。

### ■ 利用目的

要件定義フレームワークにおけるDFD作成の目的は、「業務データの流れの視点で、業務を上位から下位に向かい徐々に詳細化・視覚化することで、業務データと業務の関係を階層構造で漏れなく抽出する」こととなります。

### ■ 利用用途

作成したDFDは、以下の成果物作成に利用することを想定しています。  
特に、「業務階層定義」は、各種業務モデルを作成する起点になる重要な成果物です。  
この成果物に、業務を漏れなく定義するために、DFDを利用します。

成果物名	説明
業務階層定義	DFD で抽出された業務プロセスを、業務階層定義の階層化された業務として利用する。
概念データモデル定義	DFD で抽出された業務データを、概念データモデルのエンティティとして利用する。

表2－1. 利用用途

### ■ 作成タイミング

要件定義フレームワークにおいては、業務要件定義でDFDを作成します。  
DFD作成を含めた、プロセス全体の進め方については、業務要件定義プロセスガイドの以下のアクティビティを参照して下さい。

要件定義FW成果物名	アクティビティID	アクティビティ名
業務要件定義プロセスガイド	G1-01-01	現行業務プロセスの調査
	G2-02-02	業務プロセスのモデル化

表2－2. DFD作成に関する要件定義フレームワークのプロセス

## 2. 要件定義フレームワークにおけるDFD利用

### ■ 業務の可視化範囲定義

DFDは、プロセスの粒度をビジネスプロセスレベルからシステム内部の手続きレベルまで設定することができます。そのため、DFDの作成に取り掛かる前に、どの業務粒度まで可視化するかを定義しておく必要があります。各プロジェクトにおいては、プロジェクトや業務の特性、及び他の要件定義成果物の関連性などを考慮した上で、DFDにおける業務の可視化範囲を定義して下さい。

＜可視化範囲定義の検討例＞

例えば、業務階層毎の業務粒度を“表2－3”のように定義した場合、レベル4の業務粒度は、業務フローで可視化するため、その上位のレベル3の業務粒度までをDFDで明らかにするなどの方針をプロジェクト毎に検討します。

階層 レベル	階層領域	説明	業務粒度の例
レベル1	事業機能	事業を推進するための事業活動単位	販売、生産、調達、物流など
レベル2	業務機能	事業機能を構成する、基本的な業務機能単位	事業機能の「販売」を例にした場合、顧客管理、受注管理、出荷管理、請求管理など
レベル3	業務プロセス	業務機能を構成する、業務イベントをトリガーに開始される連続した業務活動の集合	業務機能の「受注管理」を例にした場合、見積業務、受注入力業務、出荷指示業務など
レベル4	業務アクティビティ	業務プロセスを構成する、業務主体者が明確になる業務活動単位	業務プロセスの「受注入力業務」を例にした場合、注文書受領、受注登録、注文請書送付など
レベル5	業務タスク	業務アクティビティを構成する、業務主体者のタスク単位	業務アクティビティの「受注登録」を例にした場合、受注入力、受注入力確認、受注入力完了など。

表2－3. 業務階層レベル毎の業務粒度の例

## 2. 要件定義フレームワークにおけるDFD利用

### ■ 作成の流れ

DFD作成の手順は、以下の通りです。  
対象業務範囲を示す最上位階層のDFDから徐々に業務を詳細化した下位のDFDを作成することで、業務の漏れや不整合を防止します。

No	手順	備考
1	業務の可視化範囲を定義する。(作成準備作業)	※詳細は、前頁参照
2	対象業務の全体像を表現した最上位階層のDFDを作成する。	対象業務範囲を明確にすることが目的
3	上位階層のDFDの各業務プロセス単位に詳細化した下位階層のDFDを作成する。	業務を階層的に分析することが目的
4	“1”で定義した記述粒度になるまで、“3”を繰り返す。	

表2-4. DFD作成手順

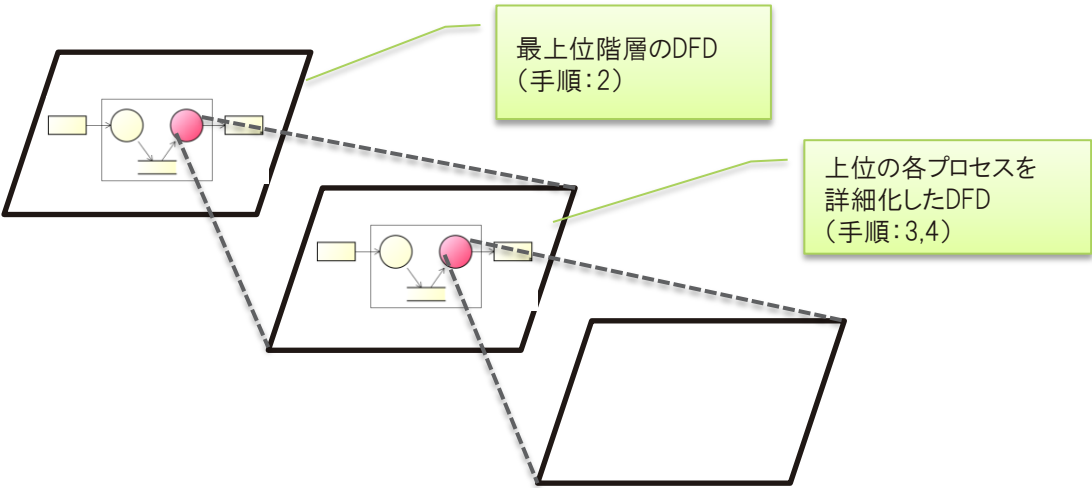


図2-5. 階層化したDFDのイメージ

『～SEのための～ DOA再入門講座』[4]のP34 図3-2から引用、一部改訂]



---

### 3. DFD記述方法

### 3. DFD記述方法

本章では、DFDの表記法や表記ルールを説明します。

#### ■ 表記法

DFDの表記法には、「デマルコ式」と「ゲイン/サーソン式」があります。各要素の表記形式の違いは、下表の通りです。  
本書は、「デマルコ式」で表現しておりますが、各プロジェクトでDFDを作成する場合は、いずれかの表記方法を選択して下さい。

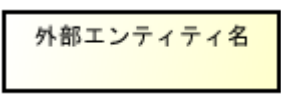
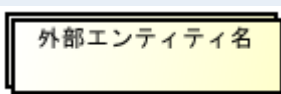


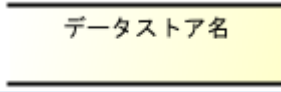
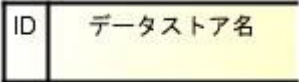
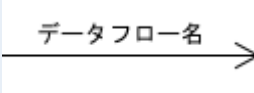
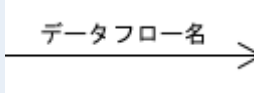
要素	デマルコ式	ゲイン/サーソン式
外部エンティティ		
プロセス		
データストア		
データフロー		

図3-1. DFD要素の表記形式

### 3. DFD記述方法

#### ■ 記述ルール

DFDは、次頁の”表3－3”に記載された記述ルールに従い、作成します。  
※以下の図3－2は、その記述ルールに準拠したDFDの例になります。

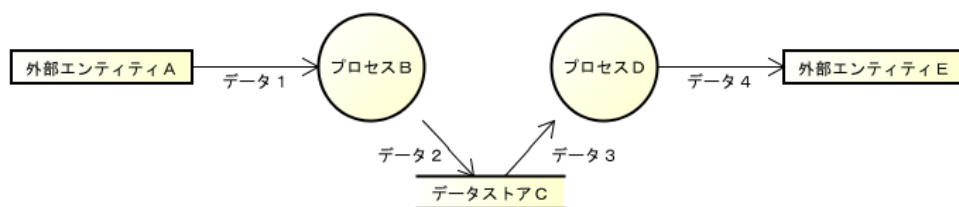


図3－2. 記述ルールに準拠したDFDの例

### 3. DFD記述方法

No	記述ルール	備考
1	<p>• <u>どのプロセスも、少なくとも1つの入力データフローと1つの出力データフローを持つこと。</u></p>	
2	<p>• <u>どのプロセスも、入力データを変換して新しい形の出力データを作成すること。</u></p>	承認などの業務の場合は、データ名(未承認)、データ名(承認済み)などで区別する。
3	<p>• <u>各データストアは、少なくとも1つのデータフローに結合していること。</u></p>	

表3-3. DFDの基本的な記述ルール(1/2)

# 3. DFD記述方法

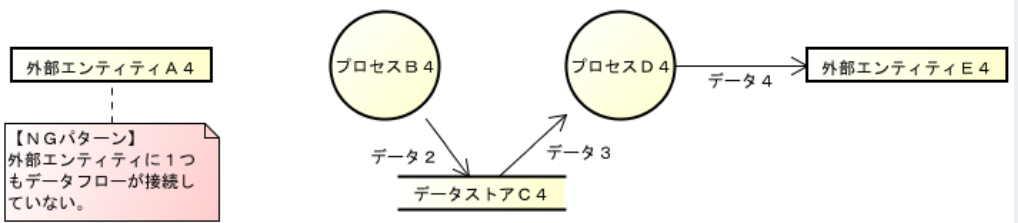
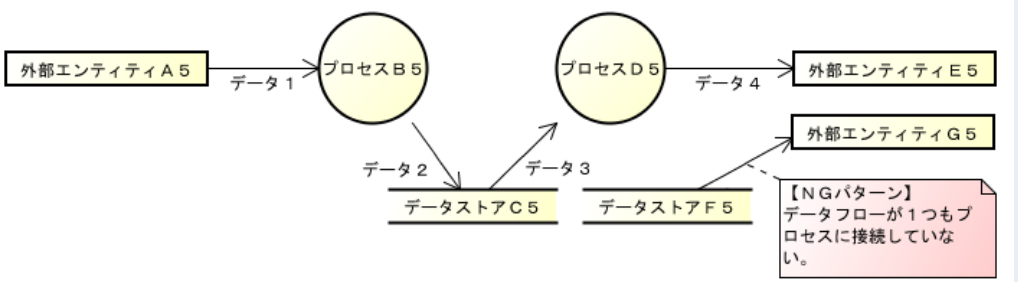
No	記述ルール	備考
4	<ul style="list-style-type: none"> <li>各外部エンティティは、少なくとも1つのデータフローに結合していること。</li> </ul>  <p>The diagram for Rule 4 shows two scenarios. The first, labeled '【NGパターン】' (NG Pattern), shows an external entity '外部エンティティ A 4' with a dashed line indicating it is not connected to any data flow. The second scenario shows a correct DFD with external entities '外部エンティティ A 4' and '外部エンティティ E 4'. '外部エンティティ A 4' is connected to 'プロセス B 4' via 'データ 2'. 'プロセス B 4' is connected to 'データストア C 4' via 'データ 2'. 'データストア C 4' is connected to 'プロセス D 4' via 'データ 3'. 'プロセス D 4' is connected to '外部エンティティ E 4' via 'データ 4'.</p>	
5	<ul style="list-style-type: none"> <li>データフローは、少なくとも1つのプロセスに結合していること。</li> </ul>  <p>The diagram for Rule 5 shows two scenarios. The first, labeled '【NGパターン】' (NG Pattern), shows a data flow 'データ 4' connecting '外部エンティティ E 5' to '外部エンティティ G 5' without passing through any process. The second scenario shows a correct DFD with external entities '外部エンティティ A 5', '外部エンティティ E 5', and '外部エンティティ G 5'. '外部エンティティ A 5' is connected to 'プロセス B 5' via 'データ 1'. 'プロセス B 5' is connected to 'データストア C 5' via 'データ 2'. 'データストア C 5' is connected to 'プロセス D 5' via 'データ 3'. 'プロセス D 5' is connected to '外部エンティティ E 5' via 'データ 4' and to 'データストア F 5' via 'データ 3'. 'データストア F 5' is connected to '外部エンティティ G 5' via 'データ 3'.</p>	

表3ー3. DFDの基本的な記述ルール(2/2)

### 3. DFD記述方法

---

#### ■ 注意事項、他

- 同一階層レベルのDFDのプロセス粒度を合わせる。  
プロセス粒度がばらばらであると、業務の網羅性や全体像が、正しく把握できなくなってしまう。  
また、「業務階層定義」のインプットとしての活用もし難くなる。  
そのため、同一階層レベルのDFD内に記述する業務プロセス粒度が一定になるように注意する必要がある。
- データストア名に実装手段を限定するような、名称をつけない。  
要求分析の段階で作成するDFDであるため、実装手段を限定するような〇〇ファイル、〇〇テーブルといった名称にはせず、データ内容が明確になるレベルで記述した方が良い。  
※ただし、As-IsのDFDを作成する場合は、現行業務を正確に理解するために、実装手段を意識した書き方をした方が  
良いケースもある。