

設計の枠組みについて

A Study on a Framework for Designing

鈴木 輝 暁
Teruaki SUZUKI

あらまし

設計の枠組み（基本的考え方）は、「①入力－②処理－③出力」である。これらを、設計段階に応じてレベルを決めれば、同じ枠組みで上流から下流まで、一貫した考えで記述することができる。重要点は、この枠組みが「らせん状の繰返し」となることである。また、インタフェース条件を明確化し、文書化すること（ICD（Interface Control Drawing：インタフェース管理図面））が肝腎である。

キーワード：設計、枠組み、基本的考え方、入力－処理－出力、らせん状の繰返し、Interface Control Drawing

Abstract

The framework (fundamental view) for designing is “(1)input - (2)processing - (3)output”. If levels are decided according to design phases, the idea which was consistent from the upper stream to the lower stream by the same framework can describe these. The important point is that this framework serves as “a spiral repetition”. Moreover, it is important to clarify and document an interface control drawing (ICD).

Keyword : design, framework, fundamental view, input-processing-output, spiral repetition, Interface Control Drawing

1. まえがき

企業における開発・技術部門では設計結果の検証として、部門内や部門外の有識者に説明し、次工程に進んでよいかの確認・審査結果などをもらっている。これらの仕組みは品質マネジメントシステム（Quality Management System）であるISO 9001（JIS Q 9001^[1]）でも求められており、多くの企業が認証を受けている。ただし、「仕組み」としては有しているが、具体的な設計結果の検証方法、遡れば設計の枠組みとして「仕組み化」されているとはいいがたいのが現状である。まだ、属人的な部分が多い。「仕組み化」として規程化されていても部分的な設計方法論であり、全

体的設計の枠組み（基本的考え方）が規程化されている企業はまだまだ少ない。

このため、システム全体を理解している人々（通常は少数）が常に確認・審査を強いられることになり、負荷が偏ることになる。説明者である設計者は、設計を枠組み（基本的考え方、フレームワーク）として捉えられず、場面ごとに振れる理解及び説明をしている。このための質疑応答に時間を要し、本来掛けるべき内容の検討に時間を掛けられないでいる。これが確認者・審査者の負荷を増やす一因でもある。更に、年数経過とともにシステム全体を理解している人々が別プロジェクトに異動したりすると、運用・保守に支障をきたすことになる。

本論文では、設計の枠組み（基本的考え方）として、次の四つ：

1. 枠組み（基本的考え方）
2. 顧客要求への対応
3. 「①入力－②処理－③出力」という枠組み
4. インタフェース条件の明確化・文書化

により、設計者及び確認者・審査者の考え方を合わせ、

5. 設計書などの統一化
6. 品質向上

に寄与できる方法論を提案する。

2. 設計の枠組み（基本的考え方）

本章では、「1. まえがき」で説明したとおり、次の順序で論ずる。

- 2.1 枠組み（基本的考え方）
- 2.2 顧客要求への対応
- 2.3 「①入力－②処理－③出力」という枠組み
- 2.4 インタフェース条件の明確化・文書化
- 2.5 設計書などの統一化
- 2.6 品質向上

2.1 枠組み（基本的考え方）

枠組み（基本的考え方）は、「モレなく、ダブリなく」^{注[1]}の概念で学生の頃教わったことのある5W1Hである。更に、ビジネスとしては費用・量を考える必要があり、“How Much”を追加し「5W2H」とする（表1 5W2Hの内容）。

なお、設計とは異なるが、それぞれの分野の枠組みはこの表を基にして、更に簡便化した枠組みが出されている。例えば、マーケティング・ミックスにおける

- ・4P：Product（製品）、Price（価格）、Place（流通）、Promotion（販売促進）

である。それぞれ、次のように対応させることができる。

- ・Product（製品）：What

- ・ Price (価格) : How Much
- ・ Place (流通) : Where
- ・ Promotion (販売促進) : How

表 1 5W2Hの内容

番号	5W2H	内 容	設計時の説明
1	When	(時) いつまでに、いつ	タイミング、駆動条件など
2	Where	(場所) どこで、どこまで	インタフェースの境目はどこかなど
3	Who	(誰) 誰が	どの装置がその対象に信号を送るかなど
4	What	(何) 何を	何の信号を入力し、何を出力するかなど
5	Why	(理由) なぜ	いろいろ考えられる方式の中から、その方式を選んだ理由など
6	How	(方法) どのように	どのように実現するかの方式など
7	How Much	(費用・量) いくら、どのくらい	性能はどれくらいか、また費用はいくらなど

2.2 顧客要求への対応

システムのライフサイクルに関して、次の規格がある。

- ・ 共通フレーム2007 (SLCP-JCF2007
(Software Life Cycle Process – Japan Common Frame 2007))

これに基づいて作られたシステム開発の順序例が、図1^[2]である。

顧客要求には、図1左上部分の「要件定義」が対応する。この「要件定義」の出力／成果物として、「要件定義書」が作成され、通常この段階で顧客側の確認・審査などが実施される。

ここから先のシステム開発は、顧客要求である「要件定義書」を基に実施され、常に設計内容及び試験内容などがモレなく対応し、必要なレベルまでブレークダウンされているかを明示する必要がある。従来、この明示が十分でなく、①顧客及び②受託会社組織内の確認者・審査者にも理解できず、受託会社組織内の設計者のみならず確認者・審査者にも負荷を与えてしまっている。更には、受託会社から顧客側にも「設計結果が顧客要求にどう応えているのか」を明確に示せないことが多く、工程遅れ・設計後戻り・顧客確認時の不具合通知となることが散見される。

要件定義書— 設計結果の対応・適合を明示するために、「コンプライアンス・マトリクス」(Compliance Matrix: 適合状況マトリクス)を作成する。概略例を、表2に示す。表の左欄である要求事項(要件定義書)に対応する設計結果などがあるか確認することにより、モレなく要求事項に対処していることを、表を見ただけで認識できる。つまり、要求事項に対応する設計結果がない場合、モレていることがわかる。

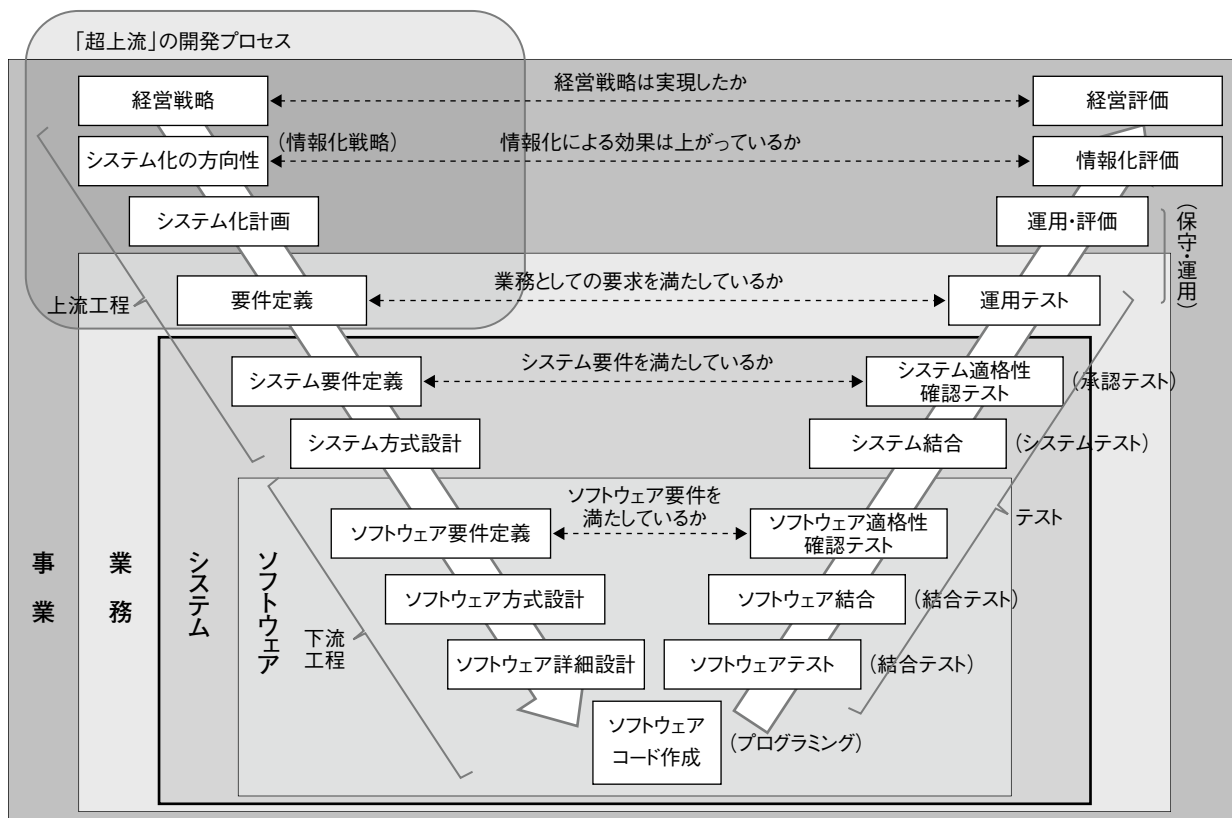


図1 システム開発の順序例^[2]

表2 「コンプライアンス・マトリクス」(適合状況マトリクス)の概要例

要件定義書			△△設計書			備考
項目番号	項目名	内容	項目番号	項目名	設計結果	
1	全般	○○・・・	1	概要	—	
1.1	定義	□□・・・	1.1	用語の定義	—	
		システムとは、・・・の範囲である。			システムは、図1の実線部分で、それ以外は非該当である。	

また、設計結果も単なる結果だけでなく「5W2H」のストーリー概要を書くようにする。通常分厚くなる設計書などをすべて読む前に、「5W2H」のストーリー概要を読めば下流設計者も何を設計すればよいのかを理解できる。更に、確認者・審査者も効率的に確認・審査を行うことができる。

顧客に説明するときも、口頭説明や分厚い設計書群を説明するのではなく、この「コンプライアンス・マトリクス」を使用して説明できる。このような説明により、顧客側も発注先会社側の取り組みに、より安心感を持つと考える。

2.3 「①入力—②処理—③出力」という枠組み

設計の枠組み（基本的考え方）は、「①入力—②処理—③出力」である。通常は、先ず始めに「③出力」を考える。つまり、どういう出力が欲しいのか、または次の下流設計段階に移るには何を決めればよいかである。これが決まれば、「①入力」はある意味芋づる式で決まってくる。「②処理」については、大きく次の二つがある。

- A) データ中心アプローチ (DOA : Data Oriented Approach)
- B) プロセス中心アプローチ (POA : Process Oriented Approach)

本論文は、設計の枠組みを提案するものであり、「②処理」の詳細内容については立入らない。

表3 開発プロセスと「①入力—②処理—③出力」の関係

番号	開発プロセス	—	①入力	②処理	③出力	—
1	要件定義		顧客要求	要件定義	要件定義書	
2	システム要件定義	→	要件定義書	システム要件定義	システム要件定義書	
3	システム方式設計	→	システム要件定義書	システム方式設計	システム方式設計書	
4	ソフトウェア要件定義	→	システム方式設計書	ソフトウェア要件定義	ソフトウェア要件定義書	
5	ソフトウェア方式設計	→	ソフトウェア要件定義書	ソフトウェア方式設計	ソフトウェア方式設計書	
6	ソフトウェア詳細設計	→	ソフトウェア方式設計書	ソフトウェア詳細設計	ソフトウェア詳細設計書	
7	ソフトウェアコード作成	→	ソフトウェア詳細設計書	ソフトウェアコード作成	ソースプログラム	

「①入力—②処理—③出力」は、最初の段階は「顧客要求」を起点として、次の関係になる。

・要件定義—

① 入力：顧客要求 ← 通常は、顧客側から文書（仕様書）として提示される

② 処理：要件定義

③ 出力：要件定義書 ← 文書

次のシステム開発段階は、図1の階層ごとに繰り返される。まとめると、表3となる。このことが、「らせん状の繰り返し」の意味である。

・番号nの開発プロセスの「③出力」=番号(n+1)の開発プロセスの「①入力」

すなわち、「次の下流設計に必要な入力事柄（仕様）は何か」を先ず考え、それを出力とする一つ上流の処理を設計することになる。そして、「次の下流設計に必要な入力事柄（仕様）は何か」が、2.4節で述べる「インタフェース条件の明確化・文書化」である。

2.4 インタフェース条件の明確化・文書化

2.3節で述べたように、「次の下流設計に必要な入力事柄（仕様）は何か」が、「インタフェース条件の明確化・文書化」である。一般的にいえば、他の設計者が読んでも次の下流設計を行える仕様が書かれていることである。設計しようとしている対象に対して、5W2Hが書かれていることが必須である。

5W2Hは、組織の規程・慣習またはプロジェクトの方針によって異なる。または、規程がない・方針もないということも散見される。しかし、責任所在の明確化のためにも重要な事項である。

ICD (Interface Control Drawing: インタフェース管理図面) としてインタフェース条件を明確化・文書化することが必要である。ただし、あくまでも上位概念として「5W2H」がありその詳細文書として位置づける。

2.5 設計書などの統一化

2.4節で述べた「インタフェース条件の明確化・文書化」及び処理設計などが設計書となる。設計者が、一から目次を考える必要もなく、「モレなく、ダブリなく」設計書などを揃えるためには統一化が望ましい。できれば、目次構成を含む雛型を作成し、個々の項目では何を書くということまで規定していることが望ましい。場合により該当しない項目もあると考える。このとき、該当しない項目を削除するのではなくそのまま残し、「なぜ非該当か」の理由を書いておくことが重要と考える。削除すると「モレなく、ダブリなく」を再度誰かが検討する必要があるためである。そのための「枠組み」と捉え、枠組みは崩さない。

雛型や設計書などに改善が必要であれば、PDCAサイクル^{付録[1]}に従って、より良くしていけばよい。また、ISO9001の継続的改善にも対応する。設計者及び確認者・審査者が余計なことに時間

を割くのではなく、より本質的な設計確認・審査を行うべきである。また、第三者から見ても統一性が取れていて分かりやすい。

2.6 品質向上

今までの節で述べたように、設計プロセスを個別に考えて部分最適を行うのではなく、全体最適を行うべきである。このため、最初から満点を狙って初動を遅らせるより、合格最低点でもよいので設計プロセスを動かすことが重要である。後は進めていくうちに、改善すべき事項などが必ず出てくる。このときPDCAサイクルに従って、継続的改善を行うべきである。

ISO9001の考え方は、品質マネジメントシステムである。“management”の語源である“manus = hand (手)”のように「やりくりする」ことが重要である。

3. むすび

設計を個別最適に行うのではなく、設計の枠組み（基本的考え方）を背骨と捉え、全体最適を図る設計の方法論について提案した。

あくまでも基本的考え方は、設計とは一見遠いと思われている「5W2H」である。これに、「①入力—②処理—③出力」を、設計段階に応じてレベルを決めれば、同じ枠組みで上流から下流まで、一貫した考えで記述できることを示した。

また、この枠組みが「らせん状の繰返し」となることを含め、最初から満点を狙うのではなく、ISO9001の品質マネジメントシステムの考えを利用することを述べた。更に、インタフェース条件を明確化し、文書化すること（ICD（Interface Control Drawing：インタフェース管理図面））が肝腎であることを述べた。

参考文献：

- [1] 日本工業標準調査会、『JIS Q 9001（品質マネジメントシステム規格）』
- [2] 木暮 仁、「『経営と情報』に関する教材と意見」, <http://www.kogures.com/hitoshi/webtext/std-kyotu-frame/index.html>,2010

注：

- [1] 「モレなく、ダブリなく」
「モレなく、ダブリなく」自体は、一般的概念である。最近は、「ロジカルシンキング（Logical Thinking:論理的思考）」で、MECE（ミッシーなどと読む：Mutually Exclusive and Collectively Exhaustive、マッキンゼー社が提唱）という用語がある。しかし、ことさら新しい概念ではないと筆者は考えている。

付録：

- [1] PDCA（Plan, Do, Check, Act）サイクルの用語について

品質マネジメントシステム（Quality Management System）の国際規格ISO 9001（日本工業規格：JIS Q 9001（品質マネジメントシステム規格））では、「継続的改善」としてPDCA（Plan, Do, Check, Act）サイクルが用いられている。我が国における多くの本などで、次のAの原語が示されている。

PDCA（Plan（計画）- Do（実施）- Check（評価）- “Action”（調整・改善））

しかし、以下の二つの理由から A=Act と動詞にした方が好ましいと考える。

なお、繰返しとなるが、「誤り」というのではなく「好ましい」の意味である。その点では、併記がよいと考える。

PDCA = Plan - Do - Check - Act/Action

<理由 1>英語は動詞主体

英語は動詞主体であり、名詞より動詞（できれば他動詞）の方が好まれる。もちろん、名詞で統一も誤りとはいえない。英語辞書からそれぞれの品詞を調べると次のとおりである。

plan : 名詞、他動詞
do : 名詞、助動詞、自動詞、他動詞
check : 名詞、形容詞、自動詞、他動詞
act : 名詞、自動詞、他動詞
action : 名詞

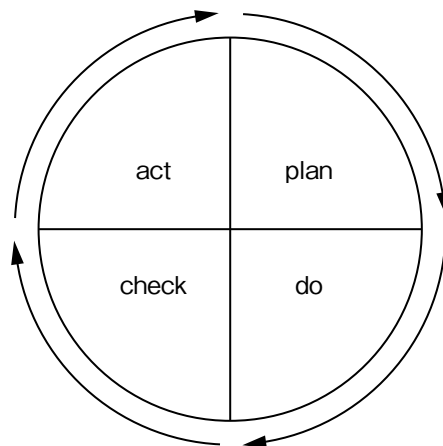
以下のように、まとめられる。

- ① 他動詞で統一とすると、actを選択
- ② 名詞で統一とすると、(1) act、(2) actionを選択

<理由 2>デミング博士（Dr. Deming）の定義

日本語検索サイトだと、2とおりの説明（act, action）がある。ただし、Deming + PDCA で英語サイトを検索すると、act が検索される。

日本では、アクションという用語の方が映画撮影などで多く使われており、Actionが使用されていると考える。



参考：<http://www.atmarkit.co.jp/aig/04biz/pdca.html>