

工程・組織効率化のための設計手法

Design Methodology for Optimizing Product Development Processes and Organizations

森 俊樹

■ MORI Toshiki

設計の大規模化、市場への新商品投入のスピードアップ、コスト競争の厳しさなど、製品開発を取り巻く状況の変化に対応するために、全体最適化の視点から、工程や組織を抜本的に改革する必要がある。設計情報の流れを可視化して、効率的な工程、組織をシステムチックに設計する手法として、DSM (Design Structure Matrix) が知られている。

東芝は、DSM手法をサポートするソフトウェア ツールとしてMTP (Matrix-based Task Planner)を開発し、社内の様々な製品開発プロセスに適用してきた。それらの事例研究から、DSM手法の有効性を確認する一方、対象の明確化、現状の把握、タスク定義の明確化、情報の流れの抽出、タスクの大きさの調整など、DSMを適用するうえでの注意点も明らかになった。

Significant changes have taken place in product development in recent years, including enlargement of the scale of system design, tough time-to-market (TTM) competition, and cost reduction. As a result of these changes, it is necessary to innovate product development processes as well as organizations from the viewpoint of global optimization. The design structure matrix (DSM) is a systematic methodology for the design of efficient processes and organizations.

Toshiba has developed a DSM tool called MTP (Matrix-based Task Planner) to support in-house projects for the innovation of product development processes. From this experience, we learned important lessons for the effective use of DSM such as clarifying its target and task definitions, paying attention to task sizes, and focusing on actual status and information flow.

1 まえがき

市場のグローバル化や顧客ニーズの多様化などに伴い、製品開発を取り巻く状況は大きく変化してきた。

第一に、製品の機能は今後更に複雑化する傾向にあり、それに対応して設計も急速に大規模・複雑化している。それは、ハードウェア設計やソフトウェア設計の各々の枠組みの中で閉じた問題ではなく、その境界もあいまいになりつつある。すなわち、機能の柔軟性を考慮してハードウェア部品をソフトウェアで代用する、あるいは、厳しい性能要求を満たすためにソフトウェアをハードウェア部品で置き換えるなど、全体最適化の視点が必要とされる。

第二に、市場への新商品投入のスピードアップがますます進行している。特に、携帯電話やノートパソコン(PC)など、製品ライフサイクルが非常に短い場合、市場投入のスピードが特に重要である。製品開発の遅れはマーケットシェアに直結し、膨大な機会損失を生む可能性がある。

第三に、コスト競争力の重要性がますます高まってきた。中国などの台頭もあり、グローバル市場におけるコスト競争は非常に厳しい。グローバル化した市場で勝ち残るためには、工程や組織のむだをなくして効率化し、製品開発にかかるコストを極力減らす必要がある。

このような変化に対して、現状の製品開発プロセスや組織は十分対応できておらず、様々な問題が生じている。例えば、工程や組織の効率の悪さに起因して、“必要な情報がタイムリーに提供されない”、“誰から情報をもらってよいかわからない”、“全体の計画や進捗(しんちやく)がよく見えない”などの問題が生じる。

また、作業のやり直しや後戻りも発生している。特に、商品の開発期間短縮のプレッシャーが大きな現場では、必然的に作業の見切り発車が多くなり、後で仕様が変更になる、あるいは、情報の誤りが判明するなど、結果的に後戻り作業となる事態が生じている。

これらの問題の根本的な原因として、以下のことが考えられる。すなわち、製品開発を取り巻く状況の変化に対応するためには、全体最適化の視点から、工程や組織を抜本的に改革する必要がある。しかしながら、現実の設計があまりにも大規模・複雑化してしまったため、全体を見渡して設計することが非常に困難となり、結果として、従来からの工程や組織を引きずって、部分最適に陥っている。

一つの解決策は、工程や組織における設計情報の流れを可視化して、誰がどの工程でどの情報を利用しているのかを正しく把握したうえで、効率的な工程や組織を設計することである。そのためには、工程や組織を場当たりに設計

するのではなく、システムチックな設計手法が必要となる。

2 DSM

DSM(Design Structure Matrix)^{(1),(2)}は、マトリックス表現に基づく、工程・組織設計のための新しい手法である。1990年代初めころから、主に、米国のマサチューセッツ工科大学(MIT)のSteven D. Eppinger教授らを中心として研究され、自動車業界や半導体業界などで徐々に広まってきた。後戻りや反復作業も含めて、複雑なプロセスの構造を簡潔に表現できるという特長がある。

DSMは、目的に応じて、様々なバリエーションがある。代表的なものとして、作業工程の入出力関係を表す“アクティビティベースDSM”，組織内のコミュニケーションを表す“チームベースDSM”，システムの構成要素の関係を表す“コンポーネントベースDSM”，設計パラメータの依存関係を表す“パラメータベースDSM”，などがある。ここでは、特に、アクティビティベースDSMに注目する。以下、単にDSMというとき、主に、アクティビティベースDSMを指すものとする。

DSMの縦軸と横軸にはプロジェクトを構成する一連のタスク(作業)が並べられ、マトリックス上のマーク(記号や数字など)は、タスク間に何らかの依存関係が存在することを示している(図1)。DSMの縦軸において、上から下への並び順が、実際にタスクを実行する順番と完全に一致していると仮定する。そのとき、マトリックス上の対角線の下側(下三角部分)にあるマークは、前工程から後工程への情報の流れを示しており、フィードフォワードと呼ばれる。一方、対角線の上側(上三角部分)にあるマークは、後工程から前工程への逆方向の情報の流れを示しており、フィードバックと呼ばれる。

DSMを用いて工程や組織を設計するには、まず、現状のプロセスをDSMで記述してみて、現状の問題点を明らかに

したうえで、以下の点を考慮しながら、あるべき姿のプロセスを構築する。

- (1) タスクの順序を変更できないか フィードバックが存在する場合、情報の流れに対するタスクの順序に無理やむだがないかを検討し、もし可能ならば、タスクの順序の組み直しを行う。これは、後戻り作業が発生するリスクを減らすために、DSMの行及び列の順序を入れ替えて、上三角部分にあるマークを減らす、又は、なるべく対角線に近づけることを意味する。
- (2) フィードバックの影響を弱められないか フィードバックの影響を弱める方法を検討する。それには、後工程から前工程への情報の流れを減らす、又は、後戻りが発生したときの影響をなるべく小さくする、という二つのアプローチが考えられる。前者の方法としては、例えば、必要な情報をタイミングよく相手に伝えるためのルール作り、有識者によるデザインレビュー、シミュレーションなどによる事前予測、後工程の専門知識を持つ人材の獲得、などがある。後者の方法としては、例えば、情報伝達スピードやコミュニケーションを高めるためのIT(情報技術)インフラの導入、リスク管理プロセスの確立、などがある。
- (3) タスクの構成を見直せないか 複数のタスクの間で情報の流れが循環しているとき、それらはDSM上のブロックとして表現される。ブロック内のタスクは相互依存の関係にあり、特に後戻り作業が発生するリスクが高い。連携作業が早く収束するように、タスクの構成を見直す必要がある。例えば、作業のまとめ方を変えて、ブロック内のタスクがなるべく同時に、かつ、同じ場所で進められないかを検討する。あるいは、プロセス上流に新たなタスクを追加して、インタフェースを事前に合意するようにして、意思決定プロセスを明確化する。DSMに基づくプロセス分析の強力な武器として、パー

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
要求仕様の決定	A	A												
コンセプト生成/選択	B	X	B											
試験計画の作成	C	X	X	C	X									
製造装置の手配・購入	D			X	D			X						X
試作モジュールの設計	E	X	X			E								
試作モジュールの製作	F				X	F								
試作モジュールの試験	G			X	X	X	G							
製品モジュールの設計	H	X	X		X		X	H		X			X	
製品モジュールの検証	I			X					I		X			
型設計	J	X	X				X	X		J				X
型の製作	K									X	K			
型の手直し・修正	L						X	X	X	L				
組立用工具の設計	M						X		X				M	
量産開始	N			X					X		X			N

図1. DSMの例 — DSM上のXは、タスク間に依存関係があることを示す。例えば、タスクEは、AとBの情報に依存し、C、F、G、Hへ情報を提供する。

Example of DSM

	A	B	E	F	C	G	H	J	M	D	K	L	I	N
要求仕様の決定	A	A												
コンセプト生成/選択	B	X	B											
試作モジュールの設計	E	X	X	E										
試作モジュールの製作	F			X	F									
試験計画の作成	C	X	X	X	C									
試作モジュールの試験	G			X	X	X	G							
製品モジュールの設計	H	X	X	X			X	H	X	X				
型設計	J	X	X				X	X	J	X				
組立用工具の設計	M							X	X	M				
製造装置の手配・購入	D				X		X		X	D				
型の製作	K							X			K			
型の手直し・修正	L						X	X			X	L		
製品モジュールの検証	I				X							X	I	
量産開始	N									X	X	X	N	

図2. パーティショニングの実行結果 — 図1に対して、パーティショニングを実行した結果を示す。タスクH、J、Mは、相互依存の関係にあることがわかる。

Result of DSM partitioning

パーティショニングがある。これは、DSMの上三角部分にあるマークを減らす、又は、なるべく対角線に近づけるようにDSMの行及び列の順序を入れ替える処理である(図2)。パーティショニングを実行した結果、ブロックの大きさは常に最小となることが保証されている。パーティショニングのアルゴリズムには、パス探索による方法、隣接行列のべき乗による方法、グラフ理論による方法など、様々な方法がある。

3 ソフトウェアツールMTPによるDSMの適用

東芝では、DSM手法をサポートするソフトウェア ツールとして、MTP(Matrix-based Task Planner)を開発し、社内の様々な業務プロセス改善で活用している(図3)。MTPは、DSMの作成・編集などの基本機能のほかに、依存関係の強さを数値で表した数値的DSMに対応して、パーティショニングの結果を階層化ブロックとして表示することができる。また、大きなブロックを分解するにはどの依存関係を断ち切ればよいかを分析する機能や、クリティカルパスを表示する機能など、DSM手法による工程や組織の設計に有用な機能を備えている。

当社では、これまで、ノートPC、家電製品、半導体設計、宇宙機器(図4)など、社内の様々な製品開発プロセスに対してDSMを適用してきた。

それらの事例研究から、DSMの効果的な適用には、以下のような注意点があることがわかった。

- (1) 対象を明確にする プロセスのどの部分を改善したいのかを明確にして、DSMの適用範囲を絞り込む必要がある。対象が不明確だと、DSMが必要以上に大きくなったり、あるいは、目的と合わない的外れなものになってしまうおそれがある。入力情報の精度や作業量を

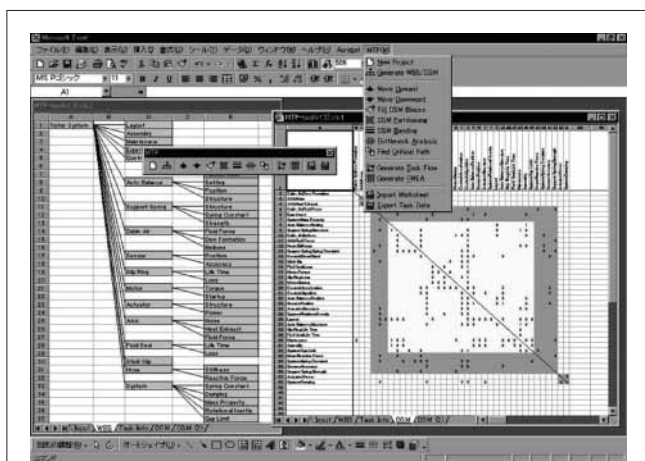


図3. MTPの表示画面 — 依存関係の分析結果を階層化ブロックとして表示することができる。

MTP screen display

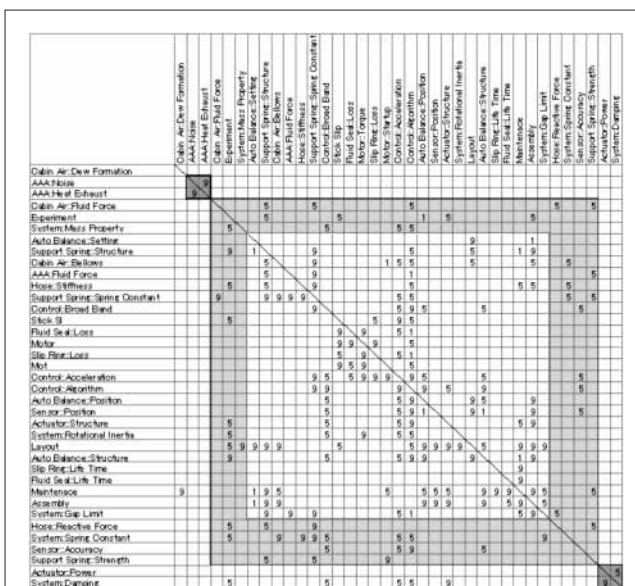


図4. 宇宙ステーション モジュールのDSM — 宇宙ステーションモジュールの開発において、各工程が複雑に関連していることが読み取れる。

DSM of space station module

- 考慮して、タスクの総数は、せいぜい30から50個程度が適当である。それには、あまり問題のなさそうなプロセスは無視して、問題のありそうなプロセスを対象を絞り込むほうがよい。もし、絞り込みが難しいならば、まず、プロセス全体を粗いDSMで分析してみて、問題のありそうな箇所を更に細かく見ていくというやり方もある。
- (2) 理想ではなく、現実の姿をとらえる 現実のプロセスを正しくDSMに入力して、問題点をきちんと把握したうえで、あるべき姿のプロセスを検討する。さもないと、現実と理想のギャップが見えず、方向を誤る可能性がある。関係者からの情報収集では、この現実と理想の姿を混同しがちなので注意が必要である。アンケートなどで情報を集めるよりも、適切な担当者が関係部署を回り、ヒアリングなどで直接、情報収集するほうがよい。
- (3) タスクの定義を明確にする タスクはわかりやすい名前を付けて、その中身を明確に定義する。タスクの定義があいまいだと、人によって解釈がばらつき、DSMの作成過程で矛盾が生じる可能性がある。タスクの定義のほか、各タスクの担当部署や使用する情報システムなども追加情報として記入しておくとうい。
- (4) 作業の流れではなく、情報の流れに注目する DSMの依存関係は、プロセスにおける作業の流れではなく、情報の流れを表す。すなわち、“このタスクを完了させるには、ほかのタスクからどのような情報が必要か”を示している。ここで、作業の流れと情報の流れは必ずしも一致しない。また、依存関係の入力は、DSMの作成でもっとも手間のかかる作業である。正確な情報収集の

こつとしては、なるべくタスクの入力情報に着目して、“どのタスクからの情報を必要としているのか”を聞くようにする。依存関係が多すぎる場合には、直接的に必要な情報に絞るか、数値などで依存関係の強さをレベル分けしてもよい。

- (5) タスクの大きさを調整する タスクが適切な大きさとなるように、グループ化や分割を行って、タスクの大きさを調整する。タスクの大きさが適切でないと、DSMから必要な情報を読み取るのが難しくなる。例えば、連続するタスクを同一の担当者又はチームが実施する場合は、なるべく一つのタスクにまとめたほうがよい。また、タスクへの入力情報は3から7個程度が適当である。大幅に超える場合は、一つのタスクとして大き過ぎるため、複数のタスクに分割したほうがよい。
- (6) 納得するまで、何度も作り直す 関係者を集めてレビューを実施し、タスクの定義や依存関係を検証する。DSMは一度完成したら終わりではなく、関係者全員が納得するまで、何度も作り直すことが重要である。

4 あとがき

設計の大規模化、TTM (Time-To-Market) の短縮、コスト競争の厳しさなど、製品開発を取り巻く状況の変化に対応するために、全体最適化の視点から、工程や組織を抜本的に改革する必要がある。設計情報の流れを可視化して、効率的な工程や組織をシステムチックに設計する手法として、DSMを紹介した。

当社では、DSM手法をサポートするソフトウェア ツールとしてMTPを開発し、社内の様々な製品開発プロセスに適用してきた。それらの事例研究から、DSM手法の有効性を確認する一方、対象の明確化、現状の把握、タスク定義の明確化、情報の流れの抽出、タスクの大ききの調整など、DSMを適用するうえでの注意点も明らかになった。

DSMは、後戻りや反復作業も含む複雑なプロセスの分析に有効であり、特に、設計などの工程を含む製品開発プロセスの分析に適している。これは、すなわち、イノベーションを必要とする製品開発プロセスには、本質的に試行錯誤を伴う複雑な学習工程が含まれていることを意味する。製品開発を取り巻く状況が大きく変化するなか、DSMの適用領域は広がっており、今後も、工程・組織効率化のための設計手法として有効に活用していきたい。

文献

- (1) Massachusetts Institute of Technology. The Design Structure Matrix-DSM. < <http://www.dsmweb.org/>>. (accessed 2004-09-17).
- (2) Eppinger, Steven D. Innovation at the Speed of Information. Harvard Business Review. 79, 1, 2001, p. 149 - 158.



森 俊樹 MORI Toshiki

ソフトウェア技術センター 技術開発担当主務。
ソフトウェア品質に関する技術開発に従事。ASME, 精密工学会会員。
Software Engineering Center