

埼玉学園大学・川口短期大学 機関リポジトリ

Software Development Methodologies (4) (Social Science)

メタデータ	言語: jpn 出版者: 公開日: 2017-07-28 キーワード (Ja): キーワード (En): 作成者: 会田, 邦夫 メールアドレス: 所属:
URL	https://saigaku.repo.nii.ac.jp/records/779

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License.



ソフトウェア開発方法論 (4)

会 田 邦 夫

目 次

1. はじめに
2. エンタープライズ・アーキテクチャー
3. ユーザビリティ・エンジニアリング
4. おわりに

1. はじめに

ソフトウェア開発方法論について、ウォーターフォールモデル、プロトタイプング技法、インクリメンタルモデル、スパイラルモデル、データ中心アプローチ、インフォメーション・エンジニアリング、ラピッド・アプリケーション・デベロプメント、ドメイン分析・モデリング等を既に取り上げ、利用者（ユーザ）および供給者（SE等）の視点から、ソフトウェア開発方法論を実際に適用する際の長所、短所、課題等を考察した⁽¹⁾⁻⁽⁴⁾。

各種ソフトウェア開発方法論は、ソフトウェア開発の実務において、その適用が厳格に行われているわけではないが、準拠していたりして、影響を受けている。各種ソフトウェア開発方法論は、企業においてソフトウェア開発のガイドラインのベースの役割をしていることが多い。

ソフトウェア開発方法論のこのような役割を考慮して、今回はビジネスとITの整合性を図るエンタープライズ・アーキテクチャーを最初に取り上げる。次にユーザの視点から製品の使いやすさを高めていく技術であるユーザビリティ・エンジニアリングを取り上げ、ソフトウェア開発方法に

組み込むことが可能であることを指摘する。これらのソフトウェア開発方法論を実際に適用する際、利用者および供給者の視点から、長所、短所、課題等についてそれぞれ検討する。

2. エンタープライズ・アーキテクチャー (Enterprise Architecture)

エンタープライズ・アーキテクチャーとは、企業や政府の持つべきIT全体を表現する検討のフレームワーク⁽⁵⁾といわれている。エンタープライズ・アーキテクチャーは、たとえば都市計画の全体像であり、都市を構成する個々の建築物が実際のアプリケーション・システムに対応している。

2.1 エンタープライズ・アーキテクチャーの概要

エンタープライズ・アーキテクチャーという概念は、ビジネスとITの整合性を図るため、J. Zachmanが1987年に始めて提唱した⁽⁶⁾。

その後、米国連邦政府、日本の経済産業省等でエンタープライズ・アーキテクチャーのフレームワークが構築されている。これらは、組織体全体におけるITに関するガイドラインが従来欠如していたため、部門毎にシステムが構築され、組織体全体としての整合性に欠ける傾向があったためである。また、その結果、組織体全体としてのシステム構築費用も上昇傾向となり、システム構築費用の抑制も課題となった背景が指摘できる。

J. Zachmanのエンタープライズ・アーキテクチャーは、横軸がデータ(What)、ファンクション(How)、ネットワーク(Where)、ピープル(Who)、タイム(When)、モチベーション(Why)より構成され、縦軸はプランナー視点、オーナー視点、設計者視点、開発者視点、サブコントラクター視点より構成されている。つまり、横軸が6つ、縦軸が5つより構成されている。

IBM社のエンタープライズ・アーキテクチャーは、次の4層のアーキ

テクチャーとガバナンス、移行計画より構成されている⁽⁷⁾。

- (1) ビジネス・アーキテクチャー
 - ・ビジネスプロセスと情報の関係
 - ・ビジネスのルール
 - ・ビジネスが実施される組織、ロケーション
 - ・ビジネス構造
- (2) アプリケーション・アーキテクチャー
 - ・アプリケーション候補
 - ・アプリケーション定義
- (3) データ・アーキテクチャー
 - ・データを扱うユーザ・グループ
 - ・データ・ストア
 - ・データ配置のガイドライン
- (4) テクニカル・アーキテクチャー
 - ・テクノロジー・フレームワーク
 - ・オペレーショナル・モデリング
 - ・リファレンス・アーキテクチャー

ガバナンスは、上記アーキテクチャーの定義と管理のプロセス、役割、責任等を規定したり、戦略を具体化するためにIT、ビジネスの資源をどのように活用するかについて定めたり、IT全般の選択基準を定めたりしている。

2.2 エンタープライズ・アーキテクチャーの目的と重要成功要因

エンタープライズ・アーキテクチャーの目的は次のとおりである。

- (1) 組織体の経営全般とIT全般の長期的整合性を図ることである。これによって、組織体全体におけるIT全体の費用を抑制して、組織体全体における便益の向上に寄与することが可能となる。もしエンタープライズ・アーキテクチャーが欠如しているとすれば、個別のアプリ

ケーションはそれぞれ費用対便益を最適化したとしても、システム間のインターフェイスが過度に複雑となったりして、組織体全体としては費用対便益を最適化できない。

- (2) 組織体の経営の観点から IT 全般について管理可能となる。従来は IT については専門の情報技術者に依存する部分が多く、経営全般と融合することが極めて難しく、その解消する手段が求められていた。
- (3) 組織体全体におけるハードウェア、ソフトウェア、ネットワーク等のリソースを効率的に活用することが可能となる。
- (4) 組織体の経営戦略を支援する IT 全般に関するフレームワークが明確となる。このことによって、組織体全体における IT の諸問題の把握が容易となり、その解決に寄与する。

このようなエンタープライズ・アーキテクチャーを組織体に導入する際の重要成功要因は、次のとおり列挙されている⁽⁸⁾。

- (1) トップマネジメントの積極的な支援
 - ・ CIO のコミットメント
 - ・ 事業部門マネジメントによる理解
 - ・ CIO 自身によるマネジメントへのエンタープライズ・アーキテクチャーの浸透
- (2) ビジネスと IT の確実な関連づけ
 - ・ ユーザの十分な関与
 - ・ エンタープライズ・アーキテクチャーモデル相互の整合性、プロジェクトとの一貫性
 - ・ 継続的改善を前提とした管理プロセス
- (3) 戦略視点・全社視点へのこだわり
 - ・ プロジェクト管理との明確な差別化
 - ・ ビジネス戦略反映の仕組み
 - ・ エンタープライズ・アーキテクチャーの理念の明確化
- (4) エンタープライズ・アーキテクチャー管理運営チームの強いリーダー

シップ

- 社内における信頼感と影響力
 - IT に偏らない広範なスキル
 - 経験豊かなリーダーと次世代を担う若手の混成
- (5) エンタープライズ・アーキテクチャーのプロセス定義および役割と責任の明確化
- 明確なエンタープライズ・アーキテクチャーの管理プロセス
 - エンタープライズ・アーキテクチャーの管理運用に関わる役割と責任の明確化
 - 成果物のオーナーシップの明確化
- (6) 徹底したコミュニケーション
- ユーザのメリット強調
 - 複数手段の活用
 - 関連情報への簡易なアクセス

このようにエンタープライズ・アーキテクチャーを組織体に導入する際の重要成功要因は、経営の視点に立ち、経営の支持を得る⁽⁹⁾ことが前提となっている。ビジネスとITを融合するためには、組織全体における意思疎通を密にする組織運営が求められている。

エンタープライズ・アーキテクチャーの長所は、次のように列挙できる。

- (1) 組織体の経営全般とIT全般の長期的整合性を維持する有力な手段である。組織体の経営全般とIT全般は、従来から水と油の関係で融合しにくい特質を備えていた。これまでそれらを融合する試みやその必要性は認識されていたが、実現手段が乏しかった。
- (2) エンタープライズ・アーキテクチャーの縦軸は、プランナー視点、オーナー視点、設計者視点、開発者視点、サブコントラクター視点より構成されているので、システムに関する認識の齟齬が回避され、IT全般の推進のガイドラインとしてエンタープライズ・アーキテクチャーの役割を期待できる。

- (3) エンタープライズ・アーキテクチャーの横軸は、データ、ファンクション、ネットワーク、ピープル、タイム、モチベーションより構成されているので、データ資源、コンピュータネットワーク、人材等の経営資源の有効な活用、効率的な活用が期待できる。
- (4) エンタープライズ・アーキテクチャーによって、個々のシステム開発の際、データ資源の共同使用、システム間のインターフェイスの簡素化、ソフトウェア技術の共有等が可能となることが期待できる。

2.3 エンタープライズ・アーキテクチャーの課題

最初に利用者の視点から、エンタープライズ・アーキテクチャーの課題について、次のように指摘できる。

- (1) 組織体の将来に関するビジョンの明確化、経営戦略の明示、ビジネスプロセス全体の現状把握とそのあるべき姿を描くことなどのビジネス・アーキテクチャーを明確にすることは、容易ではない。特に組織体が大規模になるに従い、それらは組織全体に関わる重要事項であるため、そのコンセンサスの形成は難しい。また、ビジネスプロセスは細部にわたるが、その一部分は、既にシステム化によりブラックボックスとなっており、その把握さえ困難なケースも多々ある。
- (2) ビジネス・アーキテクチャーは、環境の変化によって変貌するし、時間の推移と共に変わり得る。そのような特質を考慮して、ビジネス・アーキテクチャーを体系化することは難しい。ビジネス・アーキテクチャー自体にダイナミズムを組み込む仕掛けが必要である。
- (3) エンタープライズ・アーキテクチャーを組織体に導入する際の重要成功要因としてのトップマネジメントの積極的な支援、強いリーダーシップ、徹底したコミュニケーション等は、大規模な組織体において、実際に実行することは容易ではない。どの組織体も風通しのよい組織運営を目指しているが、実現することの難しさを痛感している。また、組織体のビジネスの専門性とIT全般の専門性はその特質が異なり、

それらの意思疎通を図るには両者の歩み寄りがなければ不可能である。両者の相互理解から着手しなければならない。

- (4) エンタープライズ・アーキテクチャーは、組織体の社内標準のひとつといえるが、その運用が課題となる。社内標準は、環境の変化とともに見直す必要があるが、それを絶えず行い管理することは容易ではない。また、社内標準を厳格に適用するかどうか、それともガイドラインの役割とするかどうかなどの運用基準も設定しなければならない。
- (5) 大規模な組織体では、システムが業務系と情報系に分かれ、その特質も異なる。データの集中処理と分散処理も混在している。組織全体の基幹システムと部門ごとの個別システムが連携していることも多いが、基幹システムはシステム部門で管理しているが、個別システムは各部門がそれぞれ管理していることも少なくない。このようなケースでは、エンタープライズ・アーキテクチャーでは、どのように調整していくかが課題となる。

次に供給者の視点から、エンタープライズ・アーキテクチャーの課題ついて指摘する。

- (1) 供給者が対象となる組織体のビジネス・アーキテクチャーを理解することは容易ではない。単なる業界毎の標準的な業務知識を超えた組織体毎のきめ細かな把握が必要となる。また、ある組織体の固有なビジネス・アーキテクチャーは、別の組織体では通用しないことが多い。このため、ある組織体のビジネス・アーキテクチャーに精通した供給者を育成したとしても、別の組織体では生かすことができず、人事管理が硬直化する。
- (2) エンタープライズ・アーキテクチャーは、環境の変化によって変貌するし、時間の推移と共に変貌するが、ある期間は新旧のエンタープライズ・アーキテクチャーが混在する。この移行期間は、システム間のインターフェイスが煩雑となり、システム運用・保守に予想以上の負荷がかかりやすい。

- (3) エンタープライズ・アーキテクチャーの縦軸において、設計者、開発者の一部分とサブコントラクターは、組織体の外部に委託する場合も多い。この場合、情報の共有はどの範囲までとするかが難しい。情報の共有を高めないと、エンタープライズ・アーキテクチャーの効果は期待できない。情報の共有を高めた結果、形式的に機密保持契約を締結しても、情報の外部遺漏は起こり得るし、ノウハウは供給者の頭脳に残る。

3. ユーザビリティ・エンジニアリング (Usability Engineering)

ユーザビリティ・エンジニアリングとは、ユーザの視点から、製品の使いやすさを高めていく技術の総称である⁽¹⁰⁾。ユーザビリティ・エンジニアリングは、情報機器、家電電器、公共機器等のテストに適用されている。

3.1 ユーザビリティ・エンジニアリングの概要

ユーザビリティ (usability) とは、使いやすさに関する概念であるが、取り扱いのしやすさ、認知性、分かりやすさ、快適性、心地よさということも含まれている⁽¹⁰⁾。

対象とするシステムのユーザビリティは、図1のように位置づけられている。

図1のシステムの受容性は、システムのユーザとその関連者のニーズを満たしているかどうかに関わる。社会的受容性は、社会的倫理に照合して受け入れられるかどうかを示している。実務的受容性は、有用性、コスト、互換性、信頼性等から構成され、ユーザに現実的に受け入れられるかどうかを示している。実用性 (utility) は、原則としてシステムの機能性がニーズを満たしているか、有用性 (usefulness) は、そのシステムで一定の目標が達成できるかどうか⁽¹¹⁾を意味している。

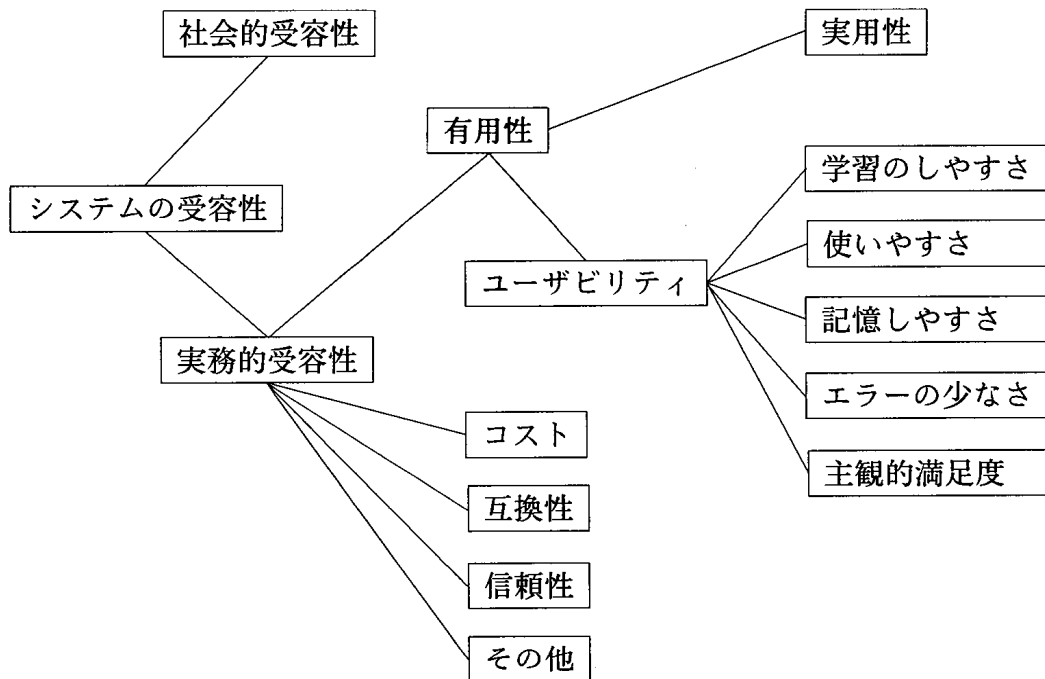


図1 ユーザビリティ

出典：ヤコブ・ニールセン著，篠原稔和監訳，『ユーザビリティエンジニアリング原論』，東京電機大学出版局，2002，p. 21

図1のユーザビリティは，学習しやすさ，使いやすさ（効率性），記憶しやすさ，エラーの少なさ，主観的満足度から構成されている。使いやすさ（効率性）は，一度ユーザがそれについて学習すれば，後は高い生産性を上げられるよう，効率的な使用を可能にすること⁽¹¹⁾を指している。エラーの少なさは，システムのエラー発生率を低くし，もしエラーが発生しても簡単に回復できるようにすること⁽¹¹⁾である。

ユーザビリティ・エンジニアリングのライフサイクルの工程は，次のとおりである⁽¹²⁾。

(1) ユーザを知る。

- 個人ユーザの性質
- タスク分析
- 機能分析
- ユーザの進化

(2) 比較分析

- (3) ユーザビリティ目標の設定
- (4) パラレルデザイン
- (5) ユーザ参加型デザイン
- (6) トータルインターフェイスのコーディネートデザイン
- (7) ガイドラインとヒューリスティクス評価法
- (8) プロトタイピング
- (9) インターフェイス評価
- (10) 反復デザイン
- (11) インストールしたシステムのフォローアップ調査

上記のユーザビリティ・エンジニアリングのライフサイクルの工程について、ユーザを知るということは、対象となるユーザとその使用状況を調査することである。ユーザの進化とは、対象となる製品になれるに従い新しい使用方法を発見したりすることを指している。比較分析は、もし既存の製品が存在すれば、開発対象となっている製品と比較して分析することである。ユーザビリティ目標の設定は、例えば製品使用中のユーザの許容可能なエラー回数を設定することである。

パラレルデザインは、数人の設計者が並行して設計を進めることである。パラレルデザインは、複数の設計案を検討できるので、比較可能でありひとつの長所である。ユーザ参加型デザインは、設計のプロセスにユーザが参加して、ユーザの意向を設計に取り込むことである。トータルインターフェイスのコーディネートデザインは、開発対象としている製品と関連した製品とのインターフェイスの一貫性を考えて、その調和を図ることである。ガイドラインは、製品のインターフェイスの設計の原則を示している。プロトタイピングは、製品の雛形である試作品を設計することである。プロトタイピングによって、ユーザのニーズ、受け止め方等が把握可能となる。

インターフェイス評価は、製品のユーザテストを行い評価する段階である。反復デザインは、インターフェイス評価の結果に基づき、問題点を解

消するために再設計を行うことである。インストールしたシステムのフォローアップ調査は、製品発売後のユーザビリティの調査であり、次の製品開発のために実施する。

3.2 ユーザビリティのテスト

次にユーザビリティ・エンジニアリングの中枢に位置づけられるユーザビリティのテストの目的は、下記のように列挙できる⁽¹³⁾。

- (1) ユーザビリティ上の問題がないかどうか確認する。
- (2) ユーザインターフェイスの代替案の内、最も優れている案を判断する。
- (3) 発売後の製品にユーザから問合せやクレームが集中した場合、その原因を究明するために行う。
- (4) 製品のユーザビリティ水準がどの程度であるかを測定する。

ユーザビリティのテストの主な手法は、次のとおりである⁽¹³⁾。

(1) パフォーマンス（効率）評価

使いやすさ（作業速度、エラー率）を評価する。作業時間を計測したり、エラー数を計測したりして、定量的分析が可能である。

(2) 主観評価

印象（安心して、気分よく、好感をもってできたか）を評価する。インタビュー、アンケート等で調査し、多数のユーザの傾向が把握できる。

(3) インタラクション評価

わかりやすさ（つまづきが少なく、スムーズにできるか）を評価する。作業データをモニタリングで取得することにより、問題点の把握が容易である。

3.3 ウォータフォールモデル、プロトタイピング技法への ユーザビリティ・テスト組み込み

大規模ソフトウェア開発の多くは、従来からウォータフォールモデルに

準拠して開発されている。ウォータフォールモデルの長所、短所、課題については、既に言及した⁽²⁾。ウォータフォールモデルのテスト工程は、更に単体テスト、結合テスト、統合テスト、ユーザテストに細分化できる。ユーザテストでは、利用者が業務で発生するデータを実際に入力して、開発対象システムの検証をする。ユーザテストは、利用者による開発対象システムの検収を含んでいる。このユーザテストの工程に、パフォーマンス（効率）評価、主観評価、インタラクション評価等のユーザビリティ・テストを適時組み込むことを提案する。このことによって、開発対象システムのユーザビリティ向上が期待できる。

大規模ソフトウェア開発の工程に予めユーザビリティ・テストを組み込み、その期間、コストを含めて開発計画を立案して、プロジェクト・マネジメントを遂行することが肝要である。

プロトタイピング技法は、更に「進化型プロトタイピング技法」、「使い捨て型プロトタイピング技法」、「技術検証型プロトタイピング技法」に分類できる⁽¹⁾。「進化型プロトタイピング技法」、「使い捨て型プロトタイピング技法」のユーザテスト工程に、パフォーマンス（効率）評価、主観評価、インタラクション評価等のユーザビリティ・テストを適時組み込むことが可能である。このことによって、利用者参加型のプロトタイピング技法は、更にそのユーザビリティ向上が期待できる。「進化型プロトタイピング技法」は、この結果、利用者の参画意識がより強化され、開発対象システムに対する利用者の受容性の向上に寄与する。

ウォータフォールモデル、プロトタイピング技法以外のソフトウェア開発においても、そのユーザテスト工程にユーザビリティ・テストを組み込むことは可能である。

3.4 利用者および供給者の課題

利用者および供給者の視点から、個別のソフトウェア開発工程にユーザビリティ・テストを組み込むことに関する課題は次のとおりである。

- (1) 組織の構成員である利用者は、通常業務を遂行しながら、開発対象システムへの参画が求められ、開発終了までその負荷がかかる。ユーザテスト工程におけるデータの作成、実行結果の妥当性の確認等は、予想外の時間を要する。利用者が更にソフトウェア開発工程に参加したり、ユーザビリティ・テストを実施したりすると通常業務の遂行に支障が生じるケースもあり得るので、事前の計画において十分検討しておくことが必要である。
- (2) 利用者は、ユーザビリティ・テストの費用対便益を考慮して、ソフトウェア開発工程にユーザビリティ・テストを組み込むかどうか決定しなければならない。費用をかければ通常その便益は向上するが、その兼ね合いが実務では求められている。ユーザビリティ・テストをどの程度組み込むかについても、利用者自ら選択する必要がある。
- (3) 供給者は、ソフトウェア開発計画の工程でユーザビリティ・テストに関する説明を利用者に行い、必要に応じて組み込むことが肝要である。ユーザビリティ・テストの費用対便益について、利用者が承認後にソフトウェア開発計画の工程に組み込むことが肝要である。
- (4) ソフトウェア開発は、通常、開発計画どおり進めることが難しいといわれている。ソフトウェア開発の期間、コストは計画よりも超過することが多い。プロジェクト・マネジメントが正常に機能していない場合である。このような場合では、ユーザテスト自体十分に行われず、さらにユーザビリティ・テストまで実施することは困難なことが予想される。

このような課題は、個別のソフトウェア開発工程の場合に指摘できる。業務用パッケージ・ソフトウェア、統合業務パッケージ・ソフトウェア等の開発の際は、ユーザビリティ・テストは不可欠である。不特定多数の利用者のユーザビリティの向上を図るユーザビリティ・テストのコストは、その販売増加効果で十分吸収できる。

4. おわりに

本論文では、エンタープライズ・アーキテクチャーおよびユーザビリティ・エンジニアリングを取り上げ、利用者および供給者の視点から、長所、短所、課題等について論じた。エンタープライズ・アーキテクチャーにおいて、実際にビジネスとITの整合性を図ることの難しさを利用者および供給者の視点から指摘した。ユーザビリティ・エンジニアリングにおけるユーザビリティ・テストは、ウォータフォールモデル、プロトタイプング技法等のソフトウェア開発方法論のユーザ・テスト工程に適時取り入れられることを指摘した。

ソフトウェア開発が必ず成功する魔法の杖のような唯一絶対のソフトウェア開発方法論が現在存在しないし、今後も存在しないであろうと推測できる。このような状況下では、各種ソフトウェア開発方法論を実際に適用する際の利用者および供給者の視点から、長所、短所、課題等について検討することが不可欠である。

《注》

- (1) 会田邦夫, 「ビジネス情報システムにおけるプロトタイプング技法の一考察」, オフィスオートメーション学会誌, 1990, Vol. 10, No. 4, pp. 51-56
- (2) 会田邦夫, 「ソフトウェア開発方法論」, 川口短大紀要第9号, 1995, pp. 121-133
- (3) 会田邦夫, 「ソフトウェア開発方法論(2)」, 川口短大紀要第10号, 1996, pp. 115-129
- (4) 会田邦夫, 「ソフトウェア開発方法論(3)」, 川口短大紀要第14号, 2000, pp. 1-16
- (5) 山下真澄, 「解説エンタープライズ・アーキテクチャ」, オフィスオートメーション学会誌, 2004, Vol. 25, No. 2, pp. 12-19
- (6) Zachman, J., "A Framework for Information Systems Architecture", *IBM System Journal*, Vol. 26, No. 3, 1987, pp. 454-470
- (7) IBM ビジネスコンサルティングサービス IT 戦略グループ, 『エンタープ

ライズ・アーキテクチャ』, 日経BP社, 2003, p. 6

- (8) 同上, pp. 45-46
- (9) 同上, pp. 197-201
- (10) 黒須正明他, 『ユーザ工学入門』, 共立出版, 1999, pp. 24-25
- (11) ヤコブ・ニールセン著, 篠原稔和監訳, 『ユーザビリティエンジニアリング原論』, 東京電機大学出版局, 2002, p. 20
- (12) 同上, p. 58
- (13) 黒須正明編, 『ユーザビリティテスト』, 共立出版, 2003, pp. 27-28