

「ノウハウマップ」によるノウハウ保持者の視覚化

八木橋英男[†]

本論文は、大規模基幹システムの開発後に、開発者から保守者へのシステムノウハウ移転を如何にして視覚化し、その移転を確実なものにするか、という試みに関する報告である。ノウハウ保持の状況を視覚化したものを、本論文では「ノウハウマップ」と呼ぶこととする。「ノウハウマップ」は、当初の目的に加え、保守期間におけるノウハウレベルの確保や、プロジェクト毎のノウハウの充足性チェックにも適用できることが分かった。

Visualization of Know-How Holders by “Know-How Map”

Hideo Yagihashi[†]

This paper describes the report of the attempt how the know-how transfer from the system development staff to system maintenance staff is visualized and ensured after the development of the large-scale mission-critical system. The tabular format visualization of system know-how holders is named as “Know-How Map” newly. “Know-How Map” can be applied to not only the original purpose but also keeping the know-how level during system maintenance phase, and checking the know-how adequateness of the new team at the case of the project team building.

1. はじめに

ある証券系大規模基幹オンラインシステムが本番稼働したのは、2006年10月のことである。本システムの開発の概要を示すと、以下の通りとなる。

- システム開発規模： 22,000 KLOC
- ピーク時開発要員数： 1,000人以上
- 開発期間： 5年
- プログラム言語： オンライン (Java)、バッチ (Open COBOL)

通常、大規模システムの開発後は、開発要員数は短期間の内に急激に減少する。本システムにおいても、開発後3年間で約3割まで要員が減少した。(図1, 表1)

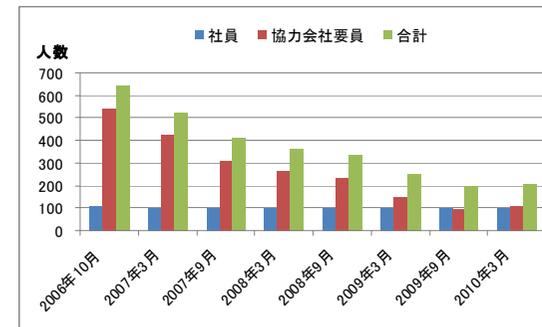


図1 開発要員数の推移

表1 開発要員数の推移

年月	社員	協力会社要員	合計
2006年10月	105	540	645
2007年3月	100	425	525
2007年9月	100	310	410
2008年3月	100	265	365
2008年9月	100	235	335
2009年3月	100	150	250
2009年9月	100	100	200
2010年3月	100	105	205

[†] 日本証券テクノロジー株式会社
Nippon Securities Technology Co., Ltd.

このような急激な要員減少の中で、大規模システムの保守品質を維持する為には、システム資産とシステム開発のノウハウを、システム保守要員に確実に引き継ぎされなければならない [1, 2].

そこで、確実なノウハウ引継の推進施策検討の為に、2007年の5月に検討タスクが組成され、システム資産を持つ部門から14名のメンバーが選出された [a]. 打ち合わせは、4ヶ月で12回に及んだ。検討の結果、システム資産とシステム開発のノウハウ引き継ぎの為に、次に述べる三種類の文書を整備することとなった。

- システム資産管理台帳
- ノウハウマップ (アプリ系)
- ノウハウマップ (システム技術)

2. システム資産管理台帳

システム資産管理台帳の説明に入る前に、大規模基幹システムで採用された技術要素について以下に補足する。

- OS : HP-UX
- プラットフォーム : J2EE MVC
- ミドルウェア : Weblogic, Oracle
- オブジェクト指向設計 : RSU (Route Stage Unit) [b]
- フレームワーク : 独自開発
- 自動運用 : JP1 及び独自開発

上記技術要素を基に、証券系大規模基幹システムのシステム開発文書が作成された。システム開発ノウハウを修得する為には、最初に成果物の構成を一覧できるシステム資産管理台帳が整備されなければならない。

システム資産管理台帳は、次に述べる4種類の文書で構成されている。

- 設計書 (基本設計書, 詳細設計書)
- プログラム
- シェル

a) 検討タスクの主目的は、開発直後の基幹システムのノウハウ引継にあったが、社内で維持している全システムのノウハウ維持も含めて検討することとなり、幅広く検討メンバーが召集された。

b) NECと当社の共同開発により作成された設計方法。

- その他文書

システム資産管理台帳は、図2に示すように、証券業務分類毎に整理されている。又、各文書においては、文書規模も把握されており、[文書]ではファイル数、他は、LOC数で集計されている。

システム資産 [その他定義] 管理台帳												
業務分類	小分類	その他定義 中分類	所有者	担当 部署	管理 手段	管理 方法	本数		ライン数		特記事項	
							物理	論理	有効 ライン数	総 ライン数		
国内株式	00.共通	DB定義	MM証券	AP12	R		nnn	nnn	nnn	nnn		
国内株式	00.共通	画面定義	MM証券	AP12	R		nnn	nnn	nnn	nnn		
国内株式	00.共通	帳票定義	MM証券	AP12	R		nnn	nnn	nnn	nnn		

システム資産 [シェル] 管理台帳													
業務区分	小分類	シェル 中分類	シェル 小分類	シェル 種別	所有者	担当 部署	管理 手段	管理 方法	本数		ライン数		特記事項
									ネット	ジョブ	有効 ライン数	総 ライン数	
国内株式	00.共通	シェル	本番用	Bシェル	MM証券	AP12	R		nnn	nnn	nnn	nnn	
国内株式	01.銘柄	シェル	本番用	Bシェル	MM証券	AP12	R		nnn	nnn	nnn	nnn	
国内株式	02.注文	シェル	本番用	Bシェル	MM証券	AP12	R		nnn	nnn	nnn	nnn	

システム資産 [プログラム] 管理台帳																
業務区分	小分類	プログラム 中分類	所有者	担当 部署	管理 手段	管理 方法	言語	プログラム			コーポ			合計ライン数	特記事項	
								本数	有効 ライン数	総 ライン数	本数	有効 ライン数	総 ライン数			有効 ライン数
国内株式	00.共通	共通M・部品	MM証券	AP12	R		COBOL	nnn	nnn	nnn	nnn	nnn	nnn	nnn	nnn	
国内株式	00.共通	本体プログラ	MM証券	AP12	R		COBOL	nnn	nnn	nnn	nnn	nnn	nnn	nnn	nnn	
国内株式	00.共通	本体プログラ	MM証券	AP12	R		Java	nnn	nnn	nnn	nnn	nnn	nnn	nnn	nnn	

システム資産 [文書] 管理台帳										
【基準時点(YY/MM)】		09/03								
【作成本部】		XXX1開発本部								
業務区分	小分類	工程分類	文書小分類	文書名	所有者	担当部署	管理手段	管理方法	ファイル数	特記事項
国内株式	00.共通	機能設計	(CRUD図)	機能設計書	MM証券	AP12	R		nnn	
国内株式	00.共通	機能設計	(ER図)	機能設計書	MM証券	AP12	R		nnn	
国内株式	00.共通	機能設計	(コード設計書)	機能設計書	MM証券	AP12	R		nnn	
国内株式	00.共通	機能設計	(ジョブフロー)	機能設計書	MM証券	AP12	R		nnn	
国内株式	00.共通	機能設計	(画面設計書)	機能設計書	MM証券	AP12	R		nnn	

図 2 システム資産管理台帳の例

システム資産管理台帳の更新タイミングは、半年毎 (3月, 9月基準) である。シ

システム保守要員は、システム資産管理台帳により、簡単に各文書を参照することができる[c]。又、半年毎に、システム資産の増減もチェック可能となった。実際、大きく変動があった場合は、システム保守要員に対して理由を確認し、登録ミスのないことを再チェックすると共に、規模の変動に注意を喚起している。

3. ノウハウマップ

システムノウハウの保持者を明らかにすると共に、システムノウハウレベルを視覚化する為に、考案されたのがノウハウマップであり、それは以下の二種類で構成されている。

- ノウハウマップ（システム技術）
- ノウハウマップ（アプリ系）

ノウハウマップ（システム技術）は、システム開発の各工程で使用しているシステム技術のノウハウを視覚化している。最近では、システム技術に限らず、幅広く業務を遂行するために必要な業務知識や法務知識なども対象となってきている。

開発工程			
要求定義	システム設計	製造	テスト
株式業務知識 債券業務知識 ： 経理業務知識	RSU 設計 オブジェクト指向設計 構造化設計 ： データベース Oracle、SQL、Sybase、・・・ ：	プログラム言語 JAVA Open COBOL ：	テストツール 計測ツール ：
プロジェクトマネジメント技術			

図 3 ノウハウマップ（システム技術）の対象

ノウハウマップ（アプリ系）は、システムが対象としている各業務分類毎に要件定義ノウハウとシステム設計ノウハウを視覚化している。各開発工程における成果物と

c) システム資産管理台帳には、各文書へのリンクも収録されている。

しては、製造工程以降のものもあるが、それらのノウハウは、ノウハウマップ（システム技術）の把握で十分と思われることから、ノウハウマップ（アプリ系）の対象外とした。

業務区分	開発工程	
	要件定義	システム設計
株式	業務フロー 画面 帳票 ：	データフロー I/O構成 ER図 ：
債券	：	：
顧客計表	：	：
：	：	：

図 4 ノウハウマップ（アプリ系）の対象

3.1 ノウハウマップ（システム技術）

ノウハウマップ（システム技術）は、概ね図 5 で示す構成となっている。

技術大分類	技術中分類	技術小分類	ノウハウ種類	ノウハウレベル上限	ノウハウレベル						
					社員			協力会社要員			
					社員1	社員2	...	要員1	要員2	...	
ソフトウェア											
プログラム言語	Open COBOL	環境技術	3	3	2		3				
		利用技術	5	2	2		3				
C		環境技術	3	3	2		3	4			
		利用技術	5	2	3		4	3			
JAVA		環境技術	3	2	2						
		利用技術	5	2	3						
：											
データベース	Oracle	環境技術	5			4					
		運用技術	5	4	4						
SQL		環境技術	5			3					
		運用技術	5								
：											
：											

図 5 ノウハウマップ（システム技術）

(1) ノウハウマップ（システム技術）の把握単位

ノウハウマップ（システム技術）は、三階層（大・中・小）の技術分類毎に作成されている。技術大分類は、ハードウェア、ソフトウェア、開発方法、等で構成されている。技術中分類は、大分類がソフトウェアの場合、OS、プログラム言語、データベース等で構成されている。技術小分類は、製品や詳細内容で構成されている。

(2) ノウハウ種類

ノウハウマップ（システム技術）のノウハウ種類は、以下の三種類で構成されている。

- 環境技術： システム環境構築技術
- 運用技術： システム運用技術
- 利用技術： システム開発技術

システム技術によって、どのノウハウ種類を把握するかが異なってくる。プログラム言語の場合は、プログラム言語環境を構築する環境技術と、プログラム言語を利用してプログラミングする利用技術が把握対象となる。データベースの場合は、全てのノウハウ種類を把握対象としている。

(3) ノウハウレベル

ノウハウレベルは、5段階で把握される。各レベル定義は、ITSSのスキルレベルのレベル1～5に対応している[d][3]。レベル1は基礎知識保有のレベルであり、レベル3は独力で遂行できるレベル、レベル5は専門家のレベルとなっている。

(4) ノウハウレベル上限

システム技術分類毎に、当社における修得可能なノウハウレベルの上限を設定している。例えば、プログラム言語の環境技術は、ベンダではないのでレベル3を上限としているが、利用技術はレベル5まで可能としている。

(5) ノウハウ保持者

各システム技術のノウハウ保持者は、社員と協力会社要員別に個人毎に把握されている。

(6) ノウハウマップ（システム技術）の更新タイミング

ノウハウマップ（システム技術）の更新タイミングは、半期毎（6月、12月基準）

d) IPA「ITスキル標準 V3 2008」で規定しているシステム技術レベルで7段階の設定がなされている。

である。ノウハウレベルの更新に際しては、各要員本人が設定したレベルをマネージャーが確認した上で設定している。

図5のノウハウマップ（システム技術）の例で補足すると、Open COBOLの環境技術では社員も協力会社要員も同じレベル3であるが、利用技術では、社員のレベルが2に対し、協力会社要員のレベルが3となっており、社員のレベル不足が協力会社要員によってカバーされているのが分かる。

3.2 ノウハウマップ（アプリ系）

ノウハウマップ（アプリ系）は、概ね図6で示す構成となっている。

業務区分	ノウハウレベル									
	小分類	社員				協力会社要員				...
		要件定義	システム設計	要件定義	システム設計	要件1	要件2	要件1	要件2	
株式	共通	3	3	3	3			3	4	
	銘柄			3	3	4	4	3	3	
	注文	3	3	4	4			4	4	
	約定	3	3	4	4			4	4	
	:									
債券	共通	5	4	4	4			4	4	
	銘柄	5	5	3	3	3	4	3	3	
	注文	3	3	2	2			3	4	
	約定	3	3	4	4			4	4	
	:									

図6 ノウハウマップ（アプリ系）

(1) ノウハウマップ（アプリ系）の把握単位

ノウハウマップ（アプリ系）は、システムが対象としている業務の区分とその小分類毎に作成されている[e]。

(2) ノウハウ種類

ノウハウマップ（アプリ系）のノウハウ種類は、以下の二種類で構成されている。

e) 証券系大規模基幹システムの場合、小分類レベルでは約690の分類になっている。
(H22年6月時点)

(3) システム別ノウハウレベル状況グラフ

システム毎にノウハウレベル別の小分類数の当期集計と前期の増減をグラフ化したものが図9の報告書である。本報告書により、システム別のノウハウレベル状況や前期からの育成・維持状況が概観できる。

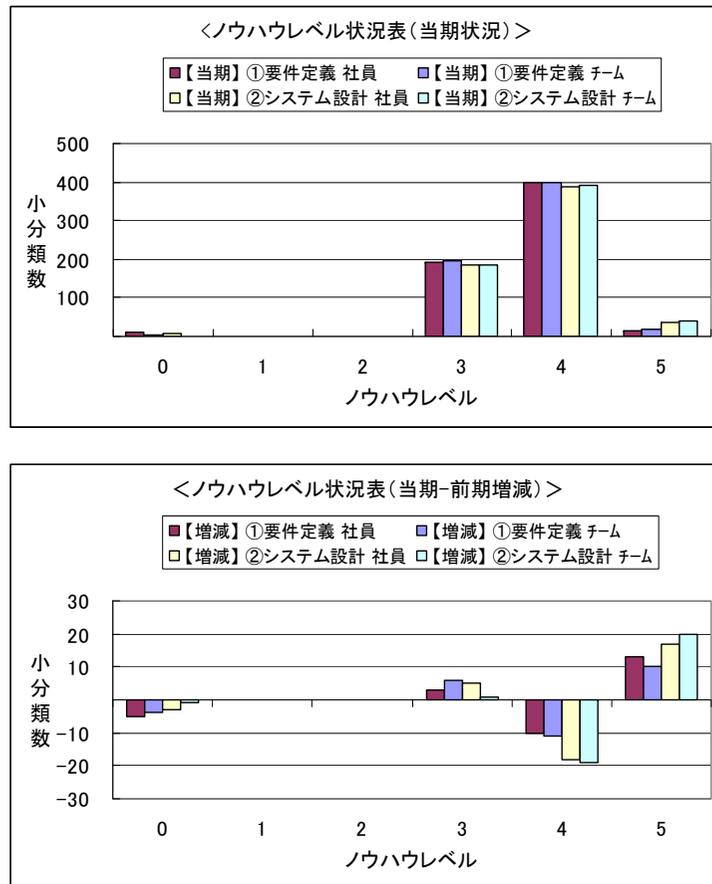


図 9 ノウハウレベル状況表

4. 「ノウハウマップによるノウハウ保持者の視覚化」の効果

(1) ノウハウレベルの維持

証券系大規模システムのチーム全体平均ノウハウレベルの推移を図10に示す。システム保守要員数(半期平均)の推移(図11)からわかるように、システム保守要員数は急激に減少しているが、ノウハウレベルはレベル4を維持していることが分かる。

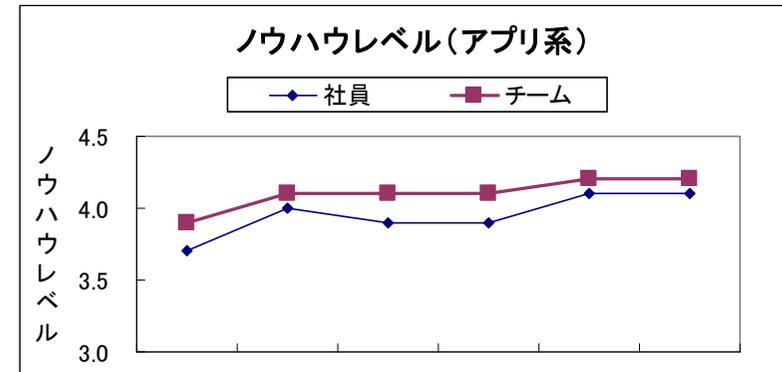


図 10 ノウハウレベル (アプリ系) の推移

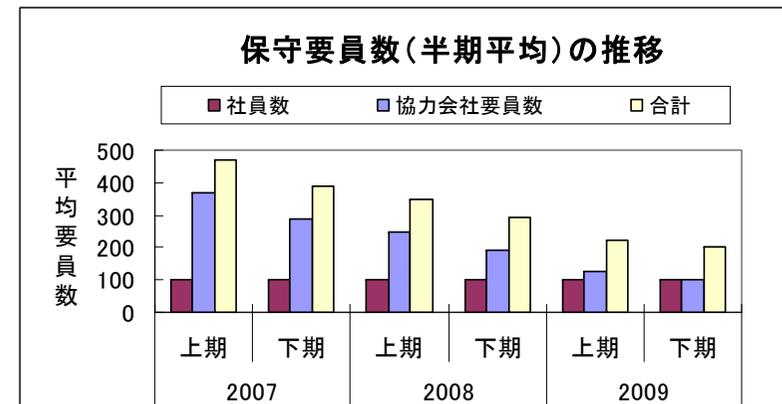


図 11 保守要員数 (半期平均) の推移

(2) プログラムエラー発生件数の低減

証券系大規模システムにおけるリリース規模(100KLOC)当りのプログラムエラー発生件数のグラフが図12である[f]。全体的にプログラムエラー発生比率は減少しつつあることがわかる。

しかし、一般にプログラマー比率は、リリース経過年が過ぎるに応じて減少していくことから、これだけでは、本当にプログラムエラー比率が減少しているとは断定できない。

そこで、リリースに関する修正作業が原因で発生したプログラムエラー件数のみを抽出して、分析してみたのが図13である。プログラムエラー発生期間については、3ヶ月と1年の二種類を対象とした。本図によれば、リリース規模当りのプログラムエラー発生件数は、全体として減少傾向にあることが分かる。

因みに、リリース規模とシステム保守要員数当りのリリース規模をまとめたのが、図14である。本図によれば、半期毎にはバラつきがあるものの、年度ではリリース規模は同程度に推移していることがわかる。

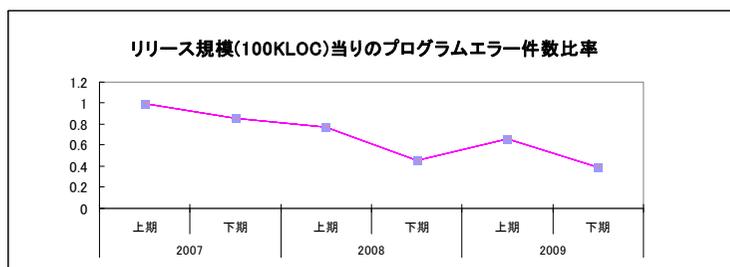


図 12 リリース規模当りのプログラムエラー件数比率

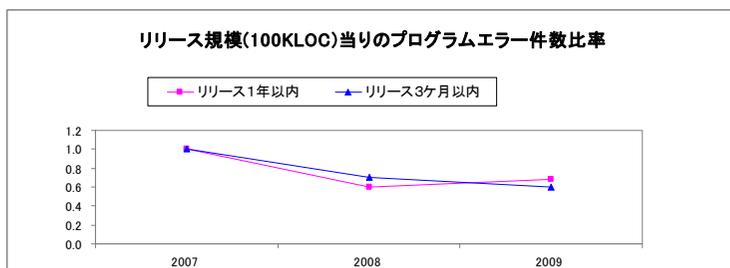


図 13 リリース1年以内の規模当りプログラムエラー件数比率

f) 2007年上期のプログラムエラー発生件数を1として、比率を表示している。

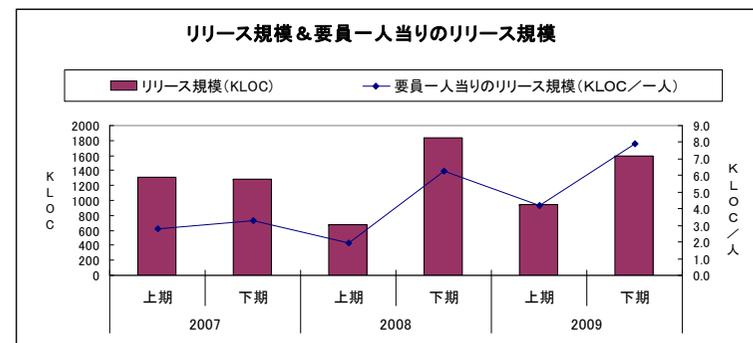


図 14 リリース規模及び保守要員一人当りのリリース規模

(3) ノウハウレベル向上の計画立案

ノウハウマップにより、業務及びシステム技術のノウハウレベルの把握が明確となった。ノウハウレベルが3よりも低い部分があれば、その点を向上するように育成計画を立てる。

又、ノウハウの重層化においてもノウハウマップは有効となる。トップノウハウ保持者を外した次席レベルでのノウハウレベルをチェックすることにより、トップノウハウ保持者が異動でなくなった場合のノウハウ充足程度が分かる。脆弱なところがあれば、その点を育成計画の対象とすることにより、ノウハウの重層化が可能となる。重要な業務区分や、システム技術においては、ノウハウ重層化の実現が大事となる。なんらかの不足事態によりトップノウハウ保持者が不在となったとしても、安定的な実力を維持することが可能となるからである。

(4) チームビルディング時のチームノウハウ充足度チェック

プロジェクト発足時に、プロジェクトが対象とする業務区分・小分類とプロジェクトメンバー情報により、チームノウハウレベルを確認することができる。チームノウハウレベルが平均レベル3以下となる領域がある場合は、その不足部分をカバーするようにチーム体制の再構成を行うか、開発推進の際の強化ポイントとして利用する。

(5) 計画的要員移動

有能なシステム要員を育成するには、定期的な要員異動が重要となる。想定する要員異動を反映したノウハウマップを作成することにより、当該要員異動によりノウハウレベルが脆弱となる領域が明確となる。当該領域についてノウハウの重層化を事前に推進指示することにより、スムーズな要員異動を実施することが可能となる。

(6) 適正保守範囲

JUAS の「2009年ソフトウェア開発統計調査」によれば、金融業種におけるシステム保守要員一人当りの担当ソフトウェア規模は、107KLOCとなっている[4].

ノウハウマップとシステム資産管理台帳の把握単位は、業務区分・小分類毎と同一となっているので、ノウハウ保持要員数一人当りのソフトウェア規模は容易に算出が可能となっている。当該係数が異常値となった場合は、業務の重要性、システムの安定度を勘案し、必要となれば要員配置の見直しを行うこととなる。

5. まとめ

ノウハウマップ（システム技術）の着想は、ITSSでも提起されているので既に一般的なものとなっていると思われる。しかし、ノウハウマップ（アプリ系）の着想は、もしかすると斬新なものではないかと考えている。ノウハウマップ（システム技術）がシステム開発工程に対応したシステム技術のノウハウを整理するという点から、横軸の視点だとすると、ノウハウマップ（アプリ系）はシステムが対象とする業務区分に関するノウハウを整理するという点から、縦軸の視点であると言える。

大規模システムのシステムノウハウを整備・維持するには、当該横軸と縦軸の両方の視点からのアプローチが有効ではないかと考えている。

謝辞 本論文を作成するに際しては、南山大学、青山先生の丁寧なご指導を賜りました。又、IPA-SEC リサーチフェロー、菊島様からは、本論文作成のきっかけと作成への激励を頂きました。茲に、謹んで感謝の意を表させていただきます。

参考文献

- 1) I. Nonaka and H. Takeuchi, The Knowledge—Creating Company, Oxford University Press, 1995.
- 2) I. Rus and M. Lindvall, Knowledge Management in Software Engineering, IEEE Software, Vol. 19, No.3, pp. May/Jun.2002, pp.26-38.
- 3) IPA, ITスキル標準 V3 2008, 2008, http://www.ipa.go.jp/jinzai/itss/download_V3_2008.html
- 4) JUAS,ソフトウェアメトリクス調査 2009,2009, <http://www.juas.or.jp/en/>.