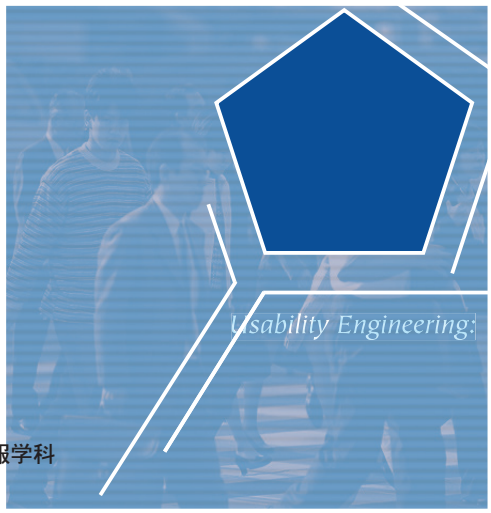
ソフトウェア開発におけるユーザビリティ工学

3



平沢尚毅
小樽商科大学商学部社会情報学科
hirasawa@res.otaru-uc.ac.jp

ソフトウェア開発において，ユーザビリティを改善，向上させることは不可欠のことである．しかしながら，ユーザビリティに関連する活動が，明確に開発プロセスの中に位置づけられているケースは多くはない．継続的にユーザビリティを向上させてゆくためには，開発プロセスに組み込まれることが必要である．そのために，ユーザビリティをソフトウェア品質としてとらえ，品質マネージメントの対象にする考え方を解説する．さらに，ソフトウェアのユーザビリティを向上させるには，テストによる評価活動だけでは効果が少なく，ユーザと利用状況に関する適切な分析が求められる．この点から，要求定義，あるいは，システムコンセプト構想のような開発初期での活動との連携が，ユーザビリティを発展させる上で重要となることを説明する．

■ はじめに

ソフトウェアにおけるユーザビリティはGUI(Graphical User Interface）やWebシステムをいかに使いやすいものにするかという課題から，最近，注目されるに至ったものと思われる．しかしながら，ユーザビリティ工学は，Human Factors Engineering（以下HFE）☆1のような人間科学の応用研究領域であって，システム開発へのかかわりは，必ずしも，最近のものではない．まずは，システムやソフトウェア開発におけるユーザビリティの沿革を簡単に追ってみる．

HFEは，すでに，1960年代から，システム開発に関与している．米国の例を挙げれば，Meister（1965）[1]らは，宇宙開発のような大規模システム開発プロセスの中で，人間要素にかかわる評価を体系化している．欧州に眼を移せば，1970年には，Shackelが先端技術を人間中心にフォーカスした研究をするために，HUSAT（Human Sciences and Advanced Technology）研究所を英国のLoughborough工科大学に設立している．ここでは，設立当初から，コンピュータシステムの使いやすさが研究されていた．また，Boehm（1976）[2]は，ソフトウェア品質としてユーザビリティを取り上げている．彼は，ユーザビリティを「Usability of a software product is the extent to which the product is convenient and practical to use」として定義している．

実際に，研究領域として，大きく取り上げられるようになったのは1980年代初頭からである．前述のMeister（1982）[3]は，システム開発への新たなHFEのかかわり方について論文をまとめている．この頃，欧州では，積極的にHFE研究の中で取り組まれ，Software ErgonomicsあるいはCognitive Ergonomicsと呼ばれる領域が生まれていた．たとえば，コンピュータサイエンスに対するSPRITE（Strategic Programme for Reseach in Information Technology Ergonomics, Shackel（1985）[4]）のようなHFEの戦略的な研究が計画され，実行されてきたのもこのころである．そして，その後のGUIの発展とともに，ユーザビリティのソフトウェアにおける重要性が明確になってきたことは周知のことである．特に，Nielsen（1990）[5]は，ソフトウェア開発プロセスの中で，ユーザビリティを改善，向上させるための一連の活動を体系化した点で高い貢献をしたといえる．このような活動一般をユーザビリティ活動と呼んでいる．

一方，HFEを人間工学と訳すとすれば，国内の人間工学では，ソフトウェア工学に関する領域には，ほとんど関与してこなかった．そのため，この間の欧米とのギャップは非常に大きいものがある．その明確な理由は分析する必要はあるが，日本の人間工学領域の独自性と同時に，ソフトウェア開発においても，使いやすいソフトウ

☆1 日本語では人間工学があてはまるが，実際の対象領域が同一ではないので，以下HFEとする．また欧州では，Ergonomicsとされるが，ここでは，同義として取り扱う．

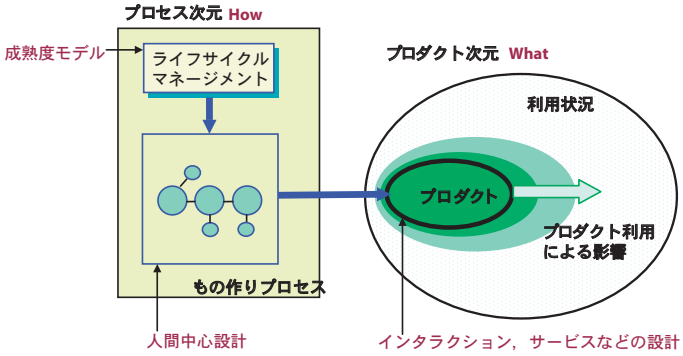


図 -1　ユーザビリティを向上させるアプローチ

ェアは当たり前すぎるほどであったために，専門的にユーザビリティを取り上げて対応するまでの必要性を感じていなかったことも否めない．情報サービス系企業の経営者の中にも，いまだにユーザビリティという言葉も聞いたことがない人がいるほど，認知されていないのが現状である．

欧米の歴史的な経緯から分かるように，ユーザビリティ工学は，ソフトウェアあるいはシステム工学と HFE のような人間科学，社会科学との学際領域に位置づけられるものである．ISO についていえば，HFE にかかわる専門委員会とソフトウェア工学にかかわる専門委員会との 2 つに分かれて取り扱われている．HFE 系では，ISO/TC 159/SC 4 が担当している．TC 159 は，Ergonomics（Human Factors Engineering）の TC（Technical Committee）であり，SC 4 は，Human System Interaction の SC（Sub Committee）である．これに対して，ISO/JTC 1/SC 7 ソフトウェア工学[6]では，主に，WG 6 の Evaluation and metrics，WG 10 の Process assessment の中で取り上げられている．実際には，ISO/TC 159 と ISO/JTC 1/SC 7 の 2 つの委員会を兼任している委員がいるほど，両者は緊密な関係にある．

このように，ソフトウェア開発におけるユーザビリティ活動は，歴史的に見れば不可分のものである．そのため，システムあるいはソフトウェア開発という軸から見て，ユーザビリティ活動を位置づけながら，その本来の役割を再検討して，ソフトウェア開発におけるユーザビリティの意義を再整理してみたい．

■ソフトウェアユーザビリティを向上させるアプローチ

ユーザビリティは，ソフトウェア品質とみなすことができるため，品質として管理することが可能である．この際，2 つの側面から管理することができる．1 つは，プロダクトそのものの品質の側面からであり，もう 1 つは，その品質を確保するためのプロセスの側面からである．この関係を示したものが図 -1 である．

まず，プロダクトの品質からの側面から検討してみる．ISO9216（Software product quality）[7] を引用すれば，ユーザビリティは，ソフトウェアの 6 つの品質の 1 つである（図 -2）．さらに，ユーザビリティは，理解性，学習性，操作性，魅力性，ユーザビリティ標準への適合性という特性から構成されている．これらは，ソフトウェア品質の客観的な特性としてとらえることができる．

一方，ソフトウェア品質は，実際のユーザが利用した結果，初めて最終的な評価が得られるものである．これが，ユーザの視点による品質である．この実際の利用状況における品質が利用品質（Quality in use）といわれるものである．ISO9216 では，「特定のユーザが，特定の利用状況において，効果的，生産的，安全に，かつ満足して，特定の目標を達成するためのソフトウェアの能力」として定義している．要は，ユーザの利用状況で，ソフトウェア利用によるユーザへの影響の程度を指しているものである．ユーザにとっての本来的な使いやすさを考えた場合，この利用品質をユーザビリティ活動の対象に含めることは自然な拡張である．むしろ，ユーザビリティ工学の目標は，この利用品質の改善，向上にあ

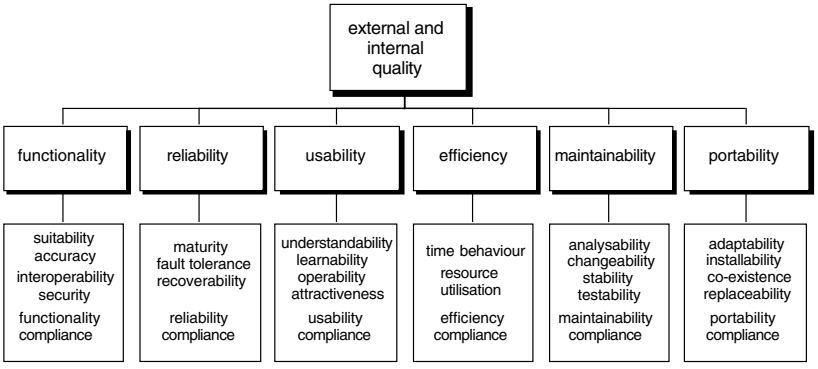


図 -2　ISO9216 におけるソフトウェア品質の構成

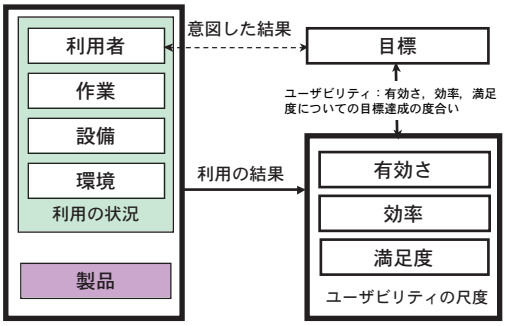


図 -3　ISO9241-11（JIS Z 8521）によるユーザビリティのフレームワーク

る．現在，TC 159/SC 4 が規格化した，ISO9241-11（JIS Z 8521）（Guidance on usability）[8] では，ユーザビリティの定義をこの利用品質とほぼ同義にしている（図 -3）．そのため，TC 159 と JTC 1/SC 7 とでは，ユーザビリティ定義の読み替えが必要になっている．いずれにしても，ソフトウェアのユーザビリティは，利用品質の枠組みまで拡張して考えた方がソフトウェア開発におけるユーザビリティエンジニアの役割がより重要なものになる．

もう一度，図 -1 に戻って考えると，次には，どのようにして，プロダクトのユーザビリティを向上させるかということになる．そのための基本的なプロセスをまとめた規格が ISO13407（Human-centred Design Processes for Interactive Systems）[9] である．そして，利用品質の改善と向上のための設計活動全般を人間中心設計（HCD: Human-centred Design）と呼んでいる．ユーザビリティへの要求は，HCD プロセスを通じて実現することになる．このプロセスをマネージメントすれば，計画的に，ユーザビリティの向上に貢献することができる．HCD プロセスをマネージメントする段階では，プロジェクトごとの改善活動になるが，さらに，組織的な改善活動レベルへと拡張するためには，HCD プロセスをいかに組織的に体系立てながら，ライフサイクル全体に関連させてマネージメントしてゆくかということになる．その上で，参考となるのが能力成熟度モデルの考え方である．

以上の 2 つのアプローチ（プロダクトアプローチおよびプロセスアプローチ）について，次に詳説する．

● プロダクトアプローチ

ソフトウェア品質のフレームワーク

ISO9216 では，ソフトウェア品質のフレームワークを 3 つの側面から定義している．これらの関係を示したのが図 -4 である．

まず，一般的に，理解されている品質は，ソフトウェアが実行可能であり，テスト段階の特性を指している．ここでは，機能的なふるまいが評価される．この段階での品質は外部品質（External quality）と呼ばれる．通常のユーザビリティテストは，この段階で行うことを意図

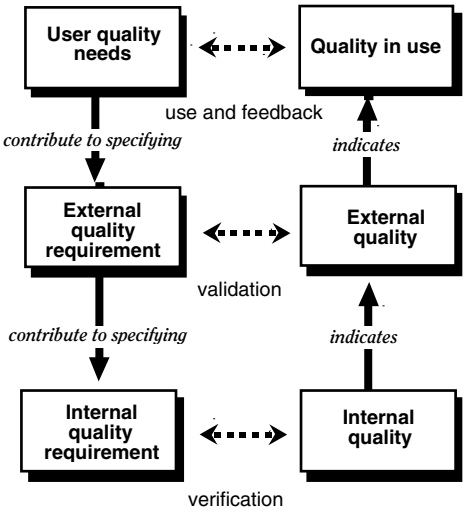


図 -4　ISO9216 によるソフトウェア品質モデル

していることが多い．

次に，ソフトウェアが実行可能になる前の段階の品質であり，仕様書やコードを対象にしたもので，内部品質（Internal quality）と呼ばれる．この段階でのユーザビリティは，要求仕様書をもとに評価される．ユーザビリティ評価活動でいえば，専門家によるインスペクションが行われる．認知的ウォークスルーなどの手法が用いられることになる．

最後に，実際のシステムを利用する際の品質である．これは，前述のように，ユーザビリティ活動の対象とすべきものであり，実際の利用状況をアンケート調査，面接，観察などによって評価する．しかしながら，システムをリリースする前に，利用状況を予測して品質を評価することは，よりユーザの要求にかなった使いやすさを提供することにつながることになる．また，ユーザビリティテストを実施する側からすれば，網羅的にユーザビリティ品質特性を評価するよりは，対象を絞って，効率的に実施することにつながる．

以上のように，ユーザビリティは，内部品質，外部品質，利用品質として評価されることになる．

ユーザビリティメトリックス

ユーザビリティを品質として継続的に改善し続けるには，定量的にとらえることが重要になる．そのためには，適切な尺度と測定法を決めて，基準値を設定する．すなわち，ユーザビリティメトリックスを設定することになる．これまで，ユーザビリティメトリックスについては，Shackel（1991）[10] などが提案してきた．その意図は，絶対的な定量化にあるのではなく，プロジェクト内あるいは開発組織内で使いやすさに対する情報の共有にある．デザイン，品質保証，設計，営業などの複数部門間において，ユーザにとっての使いやすさの具体的なイメージを共有し，ユーザビリティの数値目標を共有す

ユーザビリティ特性	外部品質メトリックス	内部品質メトリックス
Understandability	Completeness of description	Completeness of description
	Demonstration accessibility	Demonstration capability
	Demonstration accessibility in use	Evident functions
	Demonstration effectiveness	Function understandability
	Evident functions	
	Function understandability	
	Understandable Input and Output	
Learnability	Ease of function learning	Completeness of user documentation and/or help facility
	Ease of learning to perform a task in use	
	Effectiveness of the user documentation and/or help system	
	Effectiveness of the user documentation and/or help system in use	
	Help accessibility	
	Help frequency	
Operability	Operational Consistency in use	Input validity checking
	Error correction	User operation cancellability
	Error correction in use	User operation Undoability
	Default value availability in use	Customisability
	Message understandability in use	Physical accessibility
	Self-explanatory error messages	Operation status monitoring capability
	Operational error recoverability in use	Operational consistency
	Time between human error operations in use	Message Clarity
	Undoability (User error correction)	Interface element clarity
	Customisability	Operational error recoverability
	Operation procedure reduction	
	Physical accessibility	
Attractiveness	Attractive interacion	Attractive interaction
	Interface appearance customisability	User Interface appearance customisability
Usability compliance	Usability compliance	Usability compliance

表 -1　ISO9216 による外部／内部品質におけるユーザビリティのメトリックス項目

ることにつなげることができる．**表 -1** には，ISO9216 で提案されている内部，外部品質におけるユーザビリティメトリックス項目を示した．

次に，利用品質のメトリックスを考える際に重要なことは，ユーザはメーカ側の品質分類を理解しているわけではない，という当たり前の事実である．たとえば，ユーザは，機能性，信頼性などの品質特性を認識しているのではなく，利用している状況で，「よく考えているな」，「なかなか便利だ」などと各品質特性の総合的な「良さ」として認識されているものである．したがって，新しく追加した機能性について，外部品質の評価結果が良くても，実際の利用者には認識されず，それとは異なる問題を指摘されることもあり得る．利用品質に対しては，**表 -2** のようなメトリックス項目が提案されている．

ユーザから見た品質を判断する上で重要になるのが，ユーザがシステムを利用する状況である．この利用状況を特定するには，ISO9241-11 が参考になる．利用状況は，システムを利用するユーザとステークホルダー，そのシステムを利用する目的とそのための作業，関連設備，作業環境によって構成される．同じシステムでも利用される状況によって，ユーザがシステムへ要求することが異なるのは当然である．利用品質は，利用状況に依存するので，これらの情報を特定し，適切に管理することが求められることになる．利用状況は，状況を構成する要素を明確にした上で，シナリオなどによって実際の利用時のイメージを作成していると都合がよい．

利用品質特性	利用品質メトリックス
Effectiveness	Task effectiveness
	Task completion
	Error frequency
Productivity	Task time
	Task efficiency
	Economic productivity
	Productive proportion
	Relative user efficiency
Safety	User health and safety
	Safety of people affected by use of the system
	Economic damage
	Software damage
Satisfaction	Satisfaction scale
	Satisfaction questionnaire
	Discretionary usage

表 -2　ISO9216 による利用品質におけるメトリックス項目

HCD 1	HCD 2	HCD 3	HCD 4	HCD 5	HCD 6	HCD 7
Ensure HCD content in systems strategy	Plan and manage the HCD process	Specify stakeholder and organisational requirements	Understand and specify the context of use	Produce design solutions	Evaluate designs against requirements	Introduce and operate the system
Practices	*Practices*	*Practices*	*Practices*	*Practices*	*Practices*	*Practices*
represent stakeholders *collect market intelligence* *define and plan system strategy* *collect market feedback* *analyse user trends*	*consult stakeholders* *plan user involvement* *select human-centred methods* *ensure a human-centred approach* *plan HCD activities* *manage HC activities* *champion HC approach* *support HCD*	*clarify system goals* *analyse stakeholders* *assess H&S risk define system* *generate requirements* *set quality in use objectives*	*identify user's tasks* *identify user attributes* *identify organisational environment* *identify technical environment* *identify physical environment*	*allocate functions* *produce task model* *explore system design* *develop design solutions* *specify system and use* *develop prototypes* *develop user training* *develop user support*	*specify context of evaluation* *evaluate for requirements* *evaluate to improve design* *evaluate against system requirements* *evaluate against required practice* *evaluate in use*	*manage change* *determine impact* *customisation and local design* *deliver user training* *support users* *conformance to ergonomic legislation*

図 -5　ISO18529 による人間中心設計プロセスとプラクティス

● プロセスアプローチ

HCD ライフサイクルプロセスモデル

前述のように，それぞれの 3 つの段階で定義されたユーザビリティへの品質要求は，開発プロセスを通じて実現されることになる．利用品質を含めた，広義でのユーザビリティの向上を実現するための設計活動は HCD であるので，この HCD プロセスを通じて実現されることになる．

HCD プロセスを定義するために，ISO13047 を活用することを考えると，それがプロジェクトマネージメント向けの記述であるために，内容が粗くなっており直接には利用しにくい．そのためには，システムライフサイクルプロセス（SLCP：System Lifecycle Process）に組み込むことができるように詳細化した ISO18529（Human-centred lifecycle process descriptions）[11] が向いている（図 -5）．これには，HCD1 から HCD7 までのプロセスが記述されており，各プロセスの下に基本プラクティスが記述されている．HCD3 から HCD6 は，ISO13407 と重なる部分である．HCD1，HCD2，HCD7 のプロセスは，SLCP と連携するために追加されたものである．

しかしながら，ISO18529 は HCD プロセス側から関連プロセスを集約した形式になっているので，ソフトウェアを開発する側から見れば理解しづらい．そのため，HCD を SLCP に展開したプロセスモデルがあると便利である．これを HCD ライフサイクルプロセスモデルという．このモデルは，HCD プロセスと SLCP を統合することによって構築することができる．SLCP はそれぞれの開発環境によって異なるものであるが，ここでは，ISO18255（System Life Cycle Processes）[12] の Technical Processes を利用した事例を示す．

ISO18255 の Technical Process は，Stakeholder Requirements Definition から，Disposal までの 11 プロセスから構成されている．これらのプロセスから HCD プロセスにかかわりのあるプロセスを 7 つに集約し，ISO18529 にある基本プラク

Process	Basic Practices				
Requirements Definition & Analysis	利用状況の分析	ユーザ要求の定義	ユーザ要求の分析	利用品質計画	長期モニタリングの計画
Design	ヒューマンインタフェース設計	モックアップによる検証			
Implementation	プロトタイプ製作	システム運用方法の開発	保守およびサービスシステムの設計		
Integration	統合化システム設計	統合化システム運用方法の開発			
Verification & Validation	システム構成要素の検証	システムの確証			
Operation & Maintenance	システムの導入	システムの保守			
Disposal	システムの回収				

図 -6　システムライフサイクルにおける HCD プロセスとプラクティス

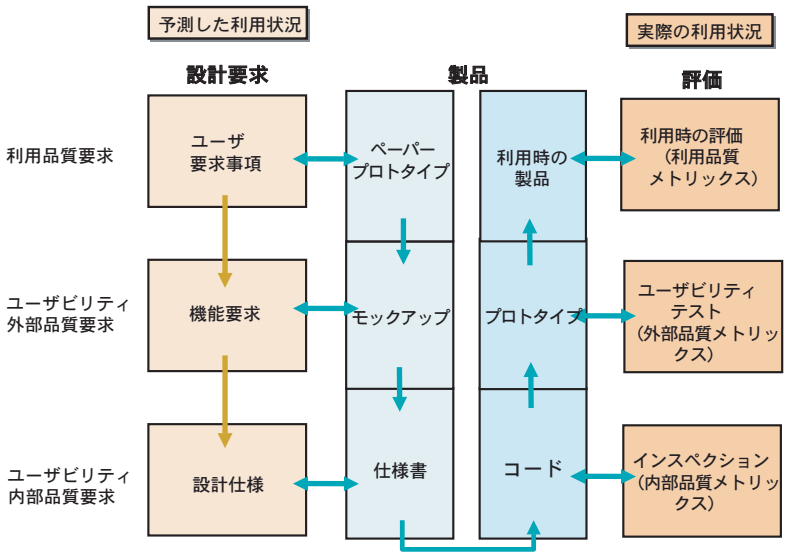


図 -7　ISO9216 の品質モデルに基づいたユーザビリティ評価の位置づけ（Bevan[13] を再編）

ティスをマージする．図 -6 に典型的な SLCP に応じた HCD の基本プラクティスの例を示した．当然ながら，このモデルは開発環境ごとにカスタマイズするものである．このようなプロセスモデルを作成することによって，ユーザビリティ要求を達成するために開発プロセスを通じて，何をすればよいかの指標を得ることができる．

要求定義プロセスの重要性

HCD ライフサイクルプロセスモデルから分かるように，ソフトウェアのユーザビリティを向上させるためには，開発ライフサイクル全体からマネージメントされる必要がある．前述したように 3 つのユーザビリティ品質への要求は，要求定義プロセスを通じて明確にされるものである．図 -7 に，ユーザビリティ品質要求と開発プロセスとの関係を模式的に記述した．これから分かるように，要求定義プロセスを通じてユーザビリティ要求が明確にされてゆくことが理解できる．ユーザビリティ評価を左右するのは，要求定義プロセスでのユーザ要求分析の如何にあるということである．テストフェーズの段階で，ユーザビリティに関するテスト項目を設定したとしても，利用状況との一貫性がとれず，最終的に利用品質を向上させることにはつながらないわけであり，無駄な評価活動を実施する危険性がある．結果的には，使いやすいと評価された機能であっても，実際はユーザが欲しくないものを満載したソフトウェアを提供してしまうことが起こり得るわけである．

また，コストの面から見ても要求定義での評価は重要である．Bias&Mayfew（1996）[14] によれば，要求定義の段階でのユーザビリティに対する変更コストが，開発段階では 6 倍のコストがかかり，さらに，保守段階では 10 倍になるとしている．コストの面からも，初期の段階からの HCD プロセスの導入は重要であることが分かる．

HCD 成熟度モデルの構築

HCD ライフサイクルプロセスモデルを開発することによって，プロセス能力成熟度モデルへと発展させることができる．すなわち，ソフトウェアアセスメントモデルに統合し，組織的なプロセスの評価／改善活動へつないでゆくことができる．適切に HCD プロセスモデルが設定されていれば，それぞれのプロセスの能力水準を設定することができ，HCD ライフサイクルプロセスの成熟度☆2 を設定することができる．これによって，段階的にプロセスの改善を行うことができるようになる．プロセスごとの能力水準設定には，たとえば，表 -3 のような ISO15504（Software process assessment）[15] の能力水準が適用可能である．

このようにして，HCD 成熟度モデルが構築される段階になって，ソフトウェアのユーザビリティの向上を組織的に支援する体制が整備された状態になったといえる．プロジェクトごとに，成果物のユーザビリティのばらつきが大きいようでは，組織的に未整備の状態にあるといえる．仮に，「ユーザに優しい」，「使いやすさ」ということを製品コンセプトに明示していても，組織的な

☆2　一般的に成熟度を設定する場合，2 通りの考え方がある．1 つは，プロジェクト全般の組織的な成熟度を見る場合で，状態モデル（Staged Model）と呼ばれる．一方，開発プロセスごとの成熟度を見る場合は，プロセスに共通な能力水準を設定する．このようなモデルは，継続モデル（Continuous Model）と呼ばれる．前者として有名なモデルが，CMM（Capability Maturity Model）であり，後者が ISO15504（Software process assessment）である．現在，CMM は 2 つのモデルを統合した CMMI（Capability Maturity Model Integration）となっている．

水準	プロセス	状態
0	不完全なプロセス	プロセスを実行していないか，プロセスの成果を出していない．このレベルでは，プロセス属性を実施した証拠がわずかしか存在しないか，またはまったく存在しない．
1	実施されたプロセス	プロセスの成果を出して，プロセスが実施されている．
2	管理されたプロセス	「実施されたプロセス」を，定義した目標値に基づいて計画し，追跡し，検証し，調査するという管理方式に基づいて実行している．
3	確立されたプロセス	エンジニアリングの基本指針に基づき，プロセスの成果を出すことができるプロセス標準を利用しながら，「管理されたプロセス」を実施する．
4	予測可能なプロセス	プロセスの成果を出すために，定義した制約の中で，「確立されたプロセス」を一貫して実施する．
5	最適化しているプロセス	現在および計画している関連したビジネス目標に効果的に適合させるために，ダイナミックに「予測不可能なプロセス」を変更し，適用する．

表 -3　ISO15504 によるプロセス能力標準

実体を伴うことは難しい．この観点から見れば，これまでのいくつかの企業におけるヒアリング（人間生活工学研究センター [16]）などから推測すると，国内の情報サービス産業の多くは，組織的に継続してユーザビリティ改善活動を行っているところはきわめて少ないと考えられる．

また，HCD 成熟度モデルは，ユーザビリティの向上を効率的にマネージメントする上で有効である．常に十分に，コストや時間をかけてユーザビリティ改善を図ることができるならば問題はないが，実際のプロジェクトでは，限られた資源の中で，ユーザビリティの成果を最適に近づけてゆくことが求められる．たとえば，新規システムであれば，利用状況分析とユーザ要求定義に配分するコストを増やし，成熟したシステムであれば，ユーザビリティ評価活動に注力するといったような対応を考えることができる．このように，プロジェクトの条件に応じて，SLCP 全体から HCD プロセスへのコスト配分を検討する上で，HCD 成熟度から指針を得る．

■ HCD 戦略構想プロセス

● HCD 戦略構想とは

図 -5 の ISO18529 を参考にしながら，HCD のライフサイクルプロセスモデルを構築しようとすれば，HCD 戦略構想プロセス（HCD1）と HCD プロセスマネージメント（HCD2）が，直接に SLCP と連携しないものであることに気づく．HCD プロセスマネージメントは，前述の成熟度モデルとして実現できる．しかし，あまり聞き慣れない HCD 戦略とは何をするものなのか．いくつかのポイントを上げるとすれば，次のようなことがいえる．

①新システムを利用するという観点から，ユーザ，ステークホルダー（システムに直接的，間接的にかかわる人々）像を明確にする．
②潜在的なユーザ像をもとに，予見的なニーズを探索する．
③潜在的なユーザ像をもとに，新システムの利用環境や利用文脈を想定する．
④以上の結果，システムが利用される場全体の構想を経営戦略，商品戦略，技術戦略との関係から位置づけてゆく．

これらは，マーケティング活動に類似しているが，マーケティング活動が，商品戦略上の市場を明確にして，誰に売るのかということにフォーカスしているのに対して，HCD 戦略は，誰が，どのように利用するのかという，ユーザが利用する場や状況を明確にする点で異なる．当然，対象となる市場がある程度明確になっていることが前提であるので，両者は補完関係にある．

HCD 戦略が必要になった背景には，情報通信技術の発達によって，先端技術によるさまざまなシステムが構想できるようになった反面，その構想が，実際にユーザが進んで利用するものかどうかを明確にしなければならない局面が増えたためである．既存のシステムと違って，ユーザにとっては，初めてのシステムであったり，初めてのタスクであったり，初めての操作であるために，ユーザがそれを受け入れてくれるかどうかの予測は難しいわけである．技術的な可能性から生まれたシステム新規性や複雑性に対して，いかにしてユーザの受容性を高め，新しい利用の場を創ってゆくかを考えなければならない時代になっている故の自然な要請である．

水準		HCD戦略の狙い	ユーザへの視座
1	AWAKE	・現ユーザが当該商品について抱く疑問点，不満点を解消し，ウォンツを実現する． ・新商品に対するクレームを減少させ，商品イメージを向上させる． ・主に現ユーザのリテンション効果を狙う．	・ユーザが述べた課題を改善する． ・ユーザの問題に気づく （問題に対する個別対応）
2	KNOW	・現ユーザが抱く商品の使用価値に着目し，要求を発掘し，実現する． ・現ユーザに対するCSの向上を狙う．	・モノを用いた活動の支援性を向上させる． ・ユーザ活動を知り，要求に変換する． （タスク分析，文脈分析が必要）
3	UNDERSTAND	・利用状況とその改善について，現ユーザの目線ですべてを捉え，分析，構想・構造化を図り，新たな市場を発掘する． ・類似の新ユーザの取り込みを狙う．	・環境や場の設計を視野に入れる． ・利用状況の構造的な理解をする． （場のモデル化が必要）
4	INFER	・現ユーザの利用状況に囚われない，新しい利用状況を提示する． ・新しい事業領域への展開を狙う．	・現ユーザや現ユーザの業態の制約から自由になった視座を得る． ・類推や推論による拡張的理解をする （拡張的なモデルが必要）
5	CREATE	・人々のライフスタイルや社会システムまでを変えるような革新的な利用状況を提示する． ・新たな企業像・企業シンボルの旗を打ち立て，新ブランドを確立する．	・社会システムの変革を伴う視座を持つ． （技術・複合技術の可能性に対する深い洞察が必要）

表 -4　システム構想の段階

このことを，Web システムを介した商取引の事例で考えてみる．ユーザにとって，Web を介した商取引は，それまでに経験したことのない行動である．ビジネスモデルとしては有効性が明確であっても，ユーザが向き合う Web サイトを介して，どのようなことができ，これまでとどう変わるのかが，ユーザ自身の体験レベルで理解できないとシステムに触れようとしない場合が多い．いわゆる使いやすさは，使ってみようと思い，実際に使い初めてからの段階の問題になる．このように，ユーザの視点から，ユーザがどのような体験を無理なくあるいは意欲的にできるかを構想することが求められることになる．そのため，受け入れられるまでのユーザビリティ（Perceived usability）と実際に使った場合のユーザビリティ（Objective usability）と分ける場合もある．また，ユーザ経験を構想するということから，Experience Designer と呼ばれる職種も出現している．

● HCD 構想の段階

さて，HCD 戦略構想は，ユーザがシステムを利用する状況を構想するものであるが，ユーザを見る視座，ユーザへの理解度によって，構想する利用状況の範囲が異なってくる．システム導入によってユーザへ与える影響の度合によって，必要な技術，スキルも異なり，難易度も異なってくる．この影響の度合を段階的にとらえることができれば，後々の開発プロセスでのユーザビリティ活動をより戦略的にマネージメントすることができる．

現在，筆者は，（株）三菱総合研究所と NTT アドバンステクノロジ（株）と共同で，HCD 戦略構想の成熟度モデルを開発中である．その基本的な考え型を**表 -4** に示した．ここでの成熟度の段階は，製品改善から始まり，タスク，ワークフロー，ライフスタイルへとユーザへの影響の度合いが拡張する．現在，それぞれの段階ごとに必要となる資質などを分析し，それぞれの段階の構想に必要となる能力をアセスメントできるように開発中である．

■まとめーユーザビリティ活動の導入と発展に向けてー

ソフトウェア開発全般を対象としたユーザビリティ活動という大きなテーマをいただいたために，総花的な構成になってしまったが，ソフトウェア開発プロセスの中にユーザビリティをどのように位置づけるかという問題を，ある程度の筋道を立ててみたつもりである．要点は次のようにまとめることができる．

- ユーザビリティは，ソフトウェア品質特性の 1 つである．そのため，ソフトウェア品質マネージメントの管理対象として組み込むことができる．品質マネージメントに組み込む場合，プロダクトアプローチとプロセスアプローチがある．
- プロダクトアプローチの場合，ユーザビリティへの品質要求を明確にし，メトリックスを選定し，基準値を設定した上で評価活動を実施する．品質要求は，開発プロセスに応じて，内部品質，外部品質，利用品質の 3 つの側面から整理する．

- ユーザビリティへの要求を達成するために，システムライフサイクルの中で，人間中心設計プロセスをマネージメントする．
- システムライフサイクルにおける人間中心設計プロセスは，能力水準を設定することによって，プロセスアセスメントモデルを構築することができ，ユーザビリティ活動を組織的に支援することができる．

以上は，ソフトウェア品質マネージメントが整備されていることを前提にした場合に活きる内容である．しかしながら，中小企業の中には，ソフトウェア品質への取り組みが進んでいないところも少なくはない．そのような場合であっても，ユーザビリティへの要求は，時流からも，必然的に求められる．ユーザビリティは，その特性から，利用品質，外部品質，そして，内部品質とソフトウェア品質のすべてにかかわるため，ソフトウェア品質を検討する上での都合のよいテストケースとなり得る．つまり，ユーザビリティ，HCD プロセスは，ソフトウェア品質マネージメントシステムの構築への先導役にもなり得る．また，顧客満足（CS：Customer Satisfaction）経営を標榜する経営層に対しては，HCD プロセスが顧客価値を実現する具体的な手段として提案することも可能である．

このように，ユーザビリティをソフトウェア開発組織に導入する見通しは明確に立っており，また，経営者の賛同を得ることも難しくないように思われる．しかしながら，実際のプロジェクトに実装するには大きな壁がある．最大の問題は，HCD プラクティスがきわめて少なく，プロジェクト活動の現場への導入には，他の設計メンバからの同意が得られないか，優先度は低くなる．開発現場は，従来のやり方を変化させることには，強い抵抗がある．そして，ユーザビリティ活動の導入に対して，いつも同じ質問を浴びせかける－「それをやって，売上に結びつくのか」．答えは，「YES」なのだが，国内のプラクティスが少ないために，胸をはって説明できない．実際の開発プロセスの中にユーザビリティ活動が活きない限りは，HCD プロセス標準を設定しても，絵に描いた餅ならぬ，紙に書いただけの標準化に終わってしまう．

そのためには，月並みの提案にしかならないかもしれないが，早急に人材育成を進めることが最重要テーマであると考える．そして，その資質は，ユーザビリティ評価にとどめずに，要求分析／要求定義の活動を並行して取り組める人を育成すべきである．では，どこで教育してもらえるのか．現状では，私の知る限り，国内では，テュフラインランドジャパン [17] で実施しているセミナー以外はないと思われる．ユーザビリティエンジニア育成カリキュラムに関しては，ヒューマンインタフェース学会のユーザビリティ専門研究会から資格制度とともに提案される可能性がある．将来的には，資格と並行して育成カリキュラムも構想されることと思う．

本稿で解説したように，ソフトウェア開発におけるユーザビリティ工学への使命と期待は，ユーザインタフェースの使いやすさの次元を超えて拡張されつつある．使いやすさだけでなく，いかにしてユーザに受け入れられるのか，という問題は大きな課題となっている．ユーザビリティ工学が，ユーザによるシステムの受容と利用という２つの側面を包含して，ユーザの価値を高めることへ貢献する可能性があることをご理解いただければ幸甚である．

謝辞　本稿を執筆するにあたって，ソフトウェア品質に関する最新の資料を込山俊博氏（NEC ソリューションズ）よりいただきましたことに謝意を表します．

参考文献

1) Meister, D. and Rabideau, G. F. : Human Factors Evaluation in System Development, John Wiley & Sons（1965）.
2) Boehm, B.W. , Brown, J. R. and Lipow, M. : Quantitative Evaluation of Software Quality, Proceedings of the Second International Conference on Software Engineering（1976）.
3) Meister, D.: The Present and Future of Human Factors, Applied Ergonomics, 13, 4, pp.281-287（1982）.
4) Shackel, B. : Ergonomics in Information Technology in Europe -a review, BEHAVIOUR AND INFORMATION TECHNOLOGY, 4, 4, pp.263-287,（1985）.
5) Nielsen, J.: Usability Engineering, AP Professional（1990）.
6) ISO JTC 1/SC 7, http://www.info.uqam.ca/Labo_Recherche/Lrgl/sc7/index_e_frameset.html
7) ISO9216-1 : Software Product Quality — Quality Model（2000）.
8) ISO9241-11（JIS Z 8521）: Ergonomics of Human System Interaction — Guidance on Usability（人間工学－視覚表示装置を用いるオフィス作業－使用性についての手引き）（1999）.
9) ISO13407（JIS Z 8530）: Human-centred Design Processes for Interactive Systems（1999）.
10) Shackel, B. and Richardson, S. : Human Factors for Informatics Usability — Background and Overview. In Shackel, B. and Richardson, S.（ed.）, Human Factors for Informatics Usability, Cambridge University Press（1991）.
11) ISO18529: Ergonomics of Human System Interaction — Human-Centred Lifecycle Process Descriptions（2000）.
12) ISO18255: System Life Cycle Processes（2002）.
13) Bevan, N.: Implementing ISO 13407, Private Paper（1999）.
14) Bias, R. G. and Mayhew, D.J.: Cost-Justifying Usability, Morgan Kaufmann（1996）.
15) ISO15504: Software Process Assessment（1998）.
16) 人間生活工学センター（HQL）: 人間中心設計に係わる国際規格への対応に関する調査研究（2000）.
17) テュフラインランドジャパン, http://www.jpn.tuv.com/jp/company-j/index-j.shtml

（平成 15 年 1 月 6 日受付）