

# CBD 방법론에서의 사용자 중심 설계 프로세스의 적용 방안에 관한 연구

함동한, 김진삼, 박창순  
한국전자통신연구원, 임베디드S/W기술센터

## Incorporating User-Centered Design Process into CBD Methodology

Dong-Han Ham, Jin Sam Kim, and Chang Soon Park

Electronics and Telecommunications Research Institute, Embedded Software Technology Center

E-mail : {dhham, jinsam, cpark}@etri.re.kr

### Abstract

To develop high-quality software systems satisfying the customer's needs, user-centered design (UCD) process has been emphasized in the area of human-computer interaction. And component-based development (CBD) has been extensively studied to realize a systematic software reuse, thereby reducing the time-to-market and the development cost. Balanced integration of user-centered design process and CBD process will be a promising way for successful software construction, which effectively achieves three goals: the quality of software, the cost of software development, and the time-to-market. However, traditional CBD methodology seems to lack the concept of user-centered design process and the relevant methods. This paper proposes a way of incorporating user-centered design process into a CBD methodology. Firstly, this paper gives the brief introduction of UCD. Secondly, we explain typical UCD activities and methods along the phase of systems development lifecycle and propose the use of usability maturity model (UMM) for integrating UCD activities and methods with those of CBD methodology. Finally, as a case study, we describe the results of comparing the UCD activities to the activities of MaRMI-III, which has been developed as a CBD methodology at ETRI.

### 1. 서론 및 연구 배경

소프트웨어 시스템의 품질 중에서 사용성 내지는 사용 품질의 중요성이 여러 문헌에서 보고되고 있다 [6,9]. 요구되는 사용성을 보장하기 위해 시스템의 분석 단계부터 유지보수에 이르기까지

전 생명주기에 걸쳐 사용자 관련 요소가 개발의 중심이 되는 활동을 사용자 중심의 설계 (UCD: User-Centered Design) 라고 한다 [10]. UCD는 인간-컴퓨터 상호작용 (HCI: Human-Computer Interaction) 영역의 핵심 분야로 현재까지 프로

세스, 모형, 기법, 지침 등을 중심으로 연구되어 왔다 [9, 18]. 이 장에서는 사용성 및 UCD 관련 기본 사항을 설명한다.

### 1-1. 소프트웨어 시스템 사용성의 중요성

UCD 활동에 대해 설명하기 전에 소프트웨어 시스템의 사용성<sup>1</sup>이 갖는 중요성에 대해 먼저 언급하고자 한다. 다양한 분야에서 사용성 결함으로 인한 문제점과 사용성 향상을 위한 재설계의 효과에 대해 보고가 되었는데 대표적인 예로 다음의 경우를 들 수 있다 [14].

(1) IBM web site: IBM 웹사이트의 사용성을 개선하기 위해 10주에 걸쳐 100명의 인원을 투입해 몇백만 달러를 사용하였다. 개선의 결과 도움말 버튼의 사용은 84퍼센트 감소했고 판매는 400퍼센트 증가했다 [3].

(2) Online shopping: e-commerce 사이트가 네비게이트 하기 어려워 39퍼센트의 테스트 구매자가 자신이 사고자 하는 물건을 사지 못했다. 또한 56퍼센트의 탐색 노력이 실패했다. 1998년 휴가 시즌에 광고에 1달러를 투자해서 5달러의 수입을 올린 반면에 UCD의 경우에는 1달러의 투자를 통한 사이트 개선이 60달러 이상의 수입을 가져왔다는 보고가 있다 [12].

(3) Software products in general: 300개 이상의 미국 회사들에 의해 수행된 8000개의 소프트웨어 개발 프로젝트를 조사한 결과, Standish 그룹은 단지 프로젝트의 16퍼센트만이 성공적이었음 (초기에 명세된 모든 기능을 계획한 시간과 비용으로 구현)을 발견했다. 프로젝트 실패의 주된 세 가지 원인은 다음과 같다: 사용자 관련 정보의 부족, 불완전한 요구사항 및 명세, 요구사항 및 명세의 변화 [16].

Lederer와 Prasad는 소프트웨어 개발의 63퍼센트가 원래의 개발비용 추정치를 초과하였다고 보고하고 있다 [2]. 이에 대한 네 가지 주된 원인으로 다음의 네 가지가 지적되었다: 사용자로부터 잦은 요구사항 변경, 사용자들이 자신의 요구사항을 제대로 파악 못함, 간과된 사용자 직무, 불충분

한 사용자와 분석자간의 커뮤니케이션. 이를 통해 알 수 있는 것은 개발비용 초과와 주된 원인이 사용자의 요구사항과 모두 밀접한 관련이 있음을 알 수 있다.

Landauer (1995)는 UCD 개념이 없이 설계된 시스템의 사용자 인터페이스는 일반적으로 사용자의 작업 속도를 느리게 하고 오류를 하게 만드는 40여개 정도의 결함을 갖고 있는 것으로 보고하고 있다 [17]. 그에 의하면 또한 UCD 개념을 갖고 설계된 시스템은 25퍼센트 정도의 training 시간을 감소시킬 수 있다고 한다.

지금까지의 예에서 보았듯이 시스템의 기능성이 복잡하고 다양해지면서 사용성의 중요성은 점점 커지고 개발 초기 단계에서부터 사용자 관련 요소와 사용성이 체계적으로 고려되어야 함을 알 수 있다.

### 1-2. UCD 활동 및 프로세스

UCD의 개념이 반영된 설계는 일반적으로 다음과 같은 특징을 갖는다 [6, 10].

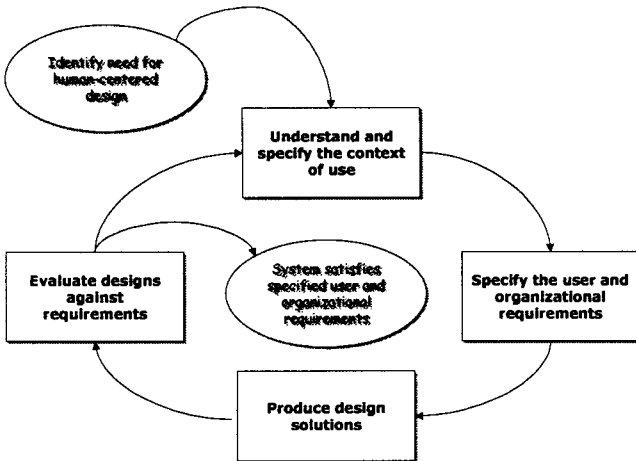
- 사용자의 적극적 참여와 사용자 및 직무요건의 명확한 이해
- 사용자와 기술 사이의 기능의 적절한 배분
- 반복 설계
- 다중학문적 (multi-disciplinary)인 설계

현재까지 UCD 프로세스와 관련해 많은 연구가 이루어져 왔다 [4-8, 11, 15, 19, 20]. 특히 이 중에서 ISO/IEC 국제 표준으로 나와 있는 ISO 13407 [6]과 UMM (Usability Maturity Model)이 주목할만하다 [7].

ISO 13407는 UCD의 표준 프로세스를 규정하고 있는데 UCD가 반영된 시스템 개발은 네 가지 UCD 활동을 포함해야 한다 (그림 1).

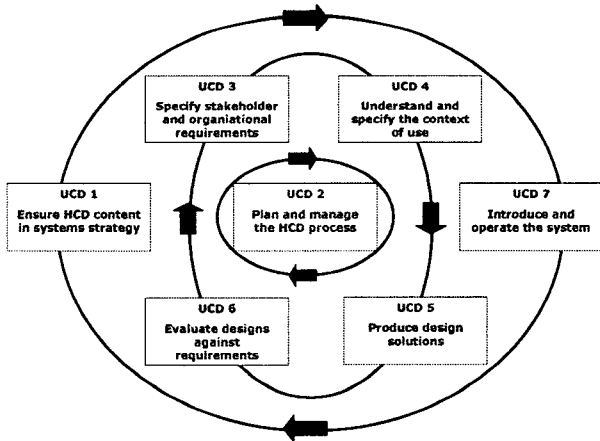
- 사용 상황을 이해하고 명세화하기 (Understand and specify the context of use)
- 사용자 및 직무의 요구사항을 명세화하기 (Specify the user and organizational requirement)
- 설계안을 도출하기 (Produce design solutions)
- 명세화된 요구사항에 대해 설계안을 평가하기 (Evaluate designs against requirements)

<sup>1</sup> UCD가 주요 대상으로 삼는 품질 속성인 사용성이 적지 않은 사람들에게 의해 혐의의 의미로 소프트웨어에서 제공하는 기능을 얼마나 배우기 쉽고 그 기능을 사용하기가 용이한가로 정의되고 있다. 이러한 사용성의 개념을 전체 개발 주기상에서 본다면 주로 설계와 구현과 깊은 관련이 있게 된다. 그러나 UCD에서 다루는 사용성의 개념은 다음의 광의의 의미를 지니고 있다: "특정 사용 상황에서 사용자가 해당 소프트웨어 (제품)를 사용하여 수행해야 하는 일(직무)을 효과적으로, 효율적으로, 안전하게, 주관적 만족감을 느끼면서 할 수 있도록 만드는 품질의 정도". 이런 의미로 사용성을 정의한다면 사용성은 소프트웨어 개발 초기-분석 단계-부터 염두에 두고 전 개발 주기에 걸쳐 다루어져야 하는 품질 속성이다 [6].



(그림 1: ISO 13407의 UCD 프로세스)

UMM은 소프트웨어 프로세스 심사모형인 ISO/IEC 15504와 CMMI의 연구 내용을 기반으로 소프트웨어 개발 프로세스가 얼마나 UCD를 효과적으로 수행하고 있는가를 평가하는데 활용되는 모형이다. 이 모형에서 규정하고 있는 최상위 프로세스는 7가지이며<sup>2</sup> 이 중 네 가지는 ISO/IEC 13407의 것을 그대로 채택하고 있다 (그림 2).



(그림 2: UMM의 최상위 UCD 프로세스)

### 1.3. 연구의 목적

소프트웨어 공학의 목표는 좋은 품질의 소프

<sup>2</sup> 원래 UMM에서는 UCD가 아닌 HCD (Human-Centered Process)라는 용어를 사용함. 그러나 UCD와 HCD는 같은 의미여서 본 논문에서는 용어의 일관성을 위해 HCD를 모두 UCD로 명칭함

트웨어를 가장 저렴하게 적시에 만들어 내는 것이다. 언급한 세 가지의 목표는 그 우선 순위를 정하기 어려울 정도로 중요성을 지닌다. 그런데 시간과 비용은 최종 사용자의 관심 사항이라기 보다는 소프트웨어 개발자 혹은 그 조직의 관심 사항이라 할 수 있다. 최종 사용자 입장에서의 최대 관심 사항은 소프트웨어의 품질이고 그 중에서도 광의의 사용성일 것이다.

그러나 현재 소프트웨어 공학 분야에서 개발된 방법론들은 UCD 관점에서의 활동과 방법을 충분히 다루고 있지 못한 상황이다 [20]. 역으로 UCD 분야도 소프트웨어 공학 분야의 연구 결과들이 갖고 있는 많은 장점들을 제대로 활용 하고 있지 못한 상황이다. 이러한 점을 고려했을 때 소프트웨어 공학과 HCI의 유기적 통합은 소프트웨어를 직접적으로 만드는 개발자의 측면과 그 소프트웨어를 최종적으로 사용하는 사용자의 측면을 동시에 고려하게 만드는 틀을 제공해 줄 수 있을 것이다 [13].

특히 최근의 새로운 개발 방법론으로 부각되고 있는 컴포넌트 기반 개발 방법 (CBD)에 UCD의 개념 및 프로세스를 통합하는 것은 의미 있는 연구일 것이다. 본 논문에서는 이 문제를 다루고자 한다. 이를 위해 우선 일반적인 소프트웨어 시스템 개발에서 UCD 활동 및 방법 (기법)이 어떤 역할과 기여를 하는지 기술한다. 다음으로 CBD 방법론에 UCD 개념과 프로세스를 통합하는 방안에 대해 논의한다. 마지막으로 앞에서 논의된 내용에 기반해서 한국전자통신연구원에서 개발한 CBD 방법론인 마르미-III [1]를 대상으로 한 사례연구를 소개한다.

## 2. 본론

### 2-1. 시스템 개발과 UCD 활동 및 방법

일반적인 시스템 개발 주기의 각 단계별로 UCD에서 강조되는 활동 및 이와 관련된 방법들이 어떻게 대응되는가를 표 1에서 정리하고

(표 1: 시스템 개발 주기에 따른 UCD 활동과 방법)

단계	주요 UCD 활동	주요 UCD 방법
계획 및 타당성 조사	Competitive analysis Cost-justification (Cost-Benefit analysis)	Stakeholder meeting Competitor analysis Content analysis ISO/IEC 13407
요구 분석 및 명세	Know the user/stakeholder Setting usability goals User analysis (User modeling) Task analysis (Task modeling) Domain analysis (Domain modeling) Usability specifications Formative evaluation	User surveys Interviews Contextual inquiry User observation Context analysis Focus groups Brainstorming Card sorting Affinity diagramming Scenario analysis Cognitive task analysis Requirements meeting Function analysis Proactive field study
설계	Develop product concept Prototyping Activity design Information design Interaction design Formative evaluation ° Iterative design	Function allocation Design guidelines Paper prototyping Heuristic evaluation Parallel design Storyboarding Wizard of OZ User interface design pattern User interface architecture
구현	User interface development	Style guides Rapid prototyping
테스팅	Empirical testing Summative evaluation	Diagnostic evaluation Performance testing Subjective evaluation Heuristic evaluation Critical incidence technique Thinking-aloud protocol Interviews Focus group
전환 및 유지보수	Collect feedback from field use	Post release testing Subjective assessment User surveys Remote evaluation

고 있다. CBD 방법의 생명 주기는 각 방법론마다 약간 다른 반면에 대부분의 방법이 표 1에서와 같은 일반적 시스템 개발 주기를 준수하고 있기 때문에 일반적 생명주기를 기준으로 삼았다.

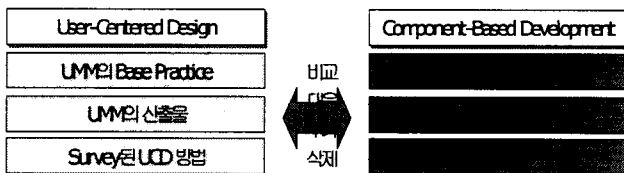
전 단계에 걸쳐 UCD 활동이 이루어지고 관련 방법이 사용되지만 사용성을 높이는 데 큰 기여를 하는 부분은 요구분석 및 명세와 테스트이다.

요구분석과 관련해 UCD의 많은 방법들이 올바른 요구사항을 구조적으로 도출할 수 있도록 도와주는데 이를 통해 사용자, 사용자의 직무, 사용자가 상호작용 하는 도메인 및 그 외 주변환경적 요소에 대한 모델링을 일관성 있고 유기적으로 할 수 있게 된다. 테스트에서는 주로 실험적인 방법을 통해 시스템 사용성에 대해 사용자

가 갖게 되는 객관적 사용성뿐만 아니라 주관적 만족감까지 측정할 수 있도록 도와준다. 설계와 관련해서는 많은 사용자 인터페이스 설계 지침이 제공되고 있고 여러 유용한 사용자 인터페이스 설계 패턴 및 아키텍처 스타일도 연구되어 있다.

### 2-2. CBD 방법론에의 UCD 통합 방안

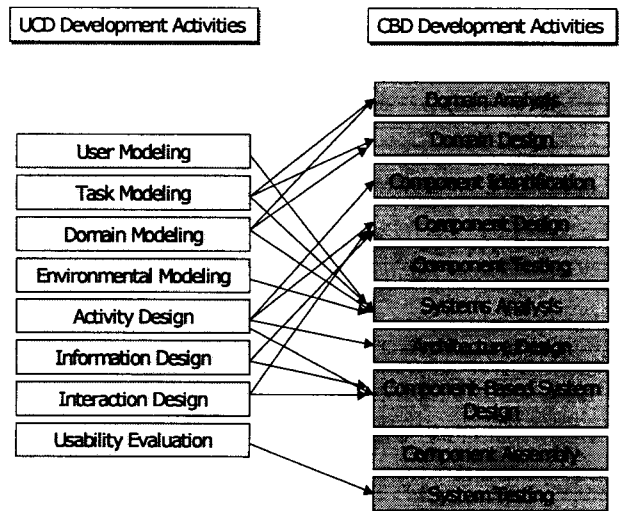
일반적인 개발 방법론이 프로세스, 산출물 양식, 방법(기법) 및 도구로 구성되어 있듯이 CBD 방법론도 이러한 구성을 따른다. 따라서 CBD 방법론에 UCD 프로세스를 통합하기 위해서는 프로세스의 활동뿐만 아니라 UCD에서 연구되어온 방법 및 산출물 등도 함께 고려할 필요가 있다. 본 논문에서는 이러한 목적을 위해 UMM에서 규정하고 있는 프로세스(상위 7개 프로세스 및 하위 44개 base practice) 및 각 프로세스와 관련된 산출물과 표 1에서 정리된 방법을 활용해 CBD 방법론과의 비교, 대응, 보완 및 삭제는 일련의 통합 과정을 제안한다(그림 3). 또한 개발자 관점에서 의사결정과 문제해결을 해야 하는 중요 UCD 개발 활동과 CBD 개발 활동간의 대응을 통해 통합화를 위한 기본 틀로 활용할 수 있다(그림 4). 이 대응 관계가 절대적인 것은 아니나 상대적으로 분석 및 설계 정보의 흐름 관점에서 가장 밀접한 관련이 있는 것들끼리 대응을 시킨 것이다.



(그림 3: UCD 및 CBD 통합화 전략)

### 2-3. 사례 연구

앞 절에서 기술한 내용에 따라 사례 연구로 한국 전자통신연구원에서 개발한 CBD 방법론인



(그림 4: UCD 및 CBD 개발 활동들간의 관계)

마르미-III의 프로세스가 UMM에서 규정하고 있는 base practice를 얼마나 잘 반영하고 있는가를 비교 검토하여 표 2에 정리하였다. 표 2로부터 알 수 있는 사실은 마르미-III가 UCD의 중요한 두 부분인 요구사항 분석 및 테스트를 통한 사용성 검증에 대해 많은 부분 다루고 있다는 것이다. 특히 요구획득 단계 및 아키텍처 단계의 많은 활동들이 정확하고 일관된 요구사항 분석에 많은 중점을 두고 있다는 사실은 마르미-III가 UCD의 가장 기본적인 철학을 잘 다루고 있다는 것을 말해준다.

그러나 시스템 개발 전 단계에 걸쳐 UCD 프로세스를 계획하고 관리하는 활동들에 대해서는 전혀 다루고 있지 않다는 점이다. 마르미-III의 경우 관리 프로세스가 개발 프로세스와 별도로 제공되어 시스템 개발 관리에 도움을 주고는 있지만 주로 시스템 개발의 일정 관리, 위험 관리, 일반적 품질 보증에 중점을 두고 있고 UCD에 대한 고려가 부족하다는 점을 알 수 있었다. 또한 시스템 전환 후에 UCD에서는 지속적인 사용자 지원을 강조하고 있는데 이 부분에 대해서도 마르미-III가 부족하다는 점을 알 수 있다.

(표 2: UMM의 Base Practice와 마르미-III 활동과의 비교 결과)

UMM의 Base Practice	마르미-III의 반영여부	대응되는 마르미-III 활동 (단계명-활동명)
UCD 1: Ensure UCD content in systems strategy		
UCD 1.1: Represent stakeholders	○	요구획득-요구사항이해
UCD 1.2 Collect market intelligence	○	요구획득-요구사항이해
UCD 1.3: Define and plan system strategy	○	요구획득-요구사항이해
UCD 1.4: Collect market feedback	×	
UCD 1.5: Analyze trends in users	×	
UCD 2: Plan and manage the UCD process		
UCD 2.1: Consult stakeholders	×	
UCD 2.2: Identify and plan user involvement	×	
UCD 2.3: Select user-centered methods and techniques	×	
UCD 2.4: Ensure a user-centered approach within the project team	×	
UCD 2.5: Plan user-centered design activities	×	
UCD 2.6: Manage user-centered activities	×	
UCD 2.7: Champion user-centered approach	×	
UCD 2.8: Plan and manage the UCD process	×	
UCD 3: Specify the stakeholder and organizational requirements		
UCD 3.1: Clarify and document system goals	○	요구획득-요구사항이해
UCD 3.2: Analyze stakeholders	○	요구획득-요구사항이해
UCD 3.3: Assess risk to stakeholders	×	
UCD 3.4: Define the use of the system	○	요구획득-요구사항정의
UCD 3.5: Generate the stakeholder and organizational requirements	○	요구획득-요구사항정의
UCD 3.6: Set quality in use objectives	×	
UCD 4: Understand and specify the context of use		
UCD 4.1: Identify and document user's tasks	○	요구획득-요구사항정의
UCD 4.2: Identify and document significant user attributes	○	요구획득-요구사항이해
UCD 4.3: Identify and document organizational environment	○	요구획득-요구사항이해
UCD 4.4: Identify and document technical environment	○	요구획득-요구사항이해
UCD 4.5: Identify and document physical environment	○	요구획득-요구사항이해

(표 2: 계속)

UMM의 Base Practice	마르미-III의 반영여부	대응되는 마르미-III 활동
UCD. 5: Produce design solutions		
UCD 5.1: Allocate functions	×	
UCD 5.2 Produce composite task model	○	요구획득단계-요구사항이해
UCD 5.3: Explore system design	○	요구획득단계-개발전략수립
UCD 5.4: Use existing knowledge to develop design solutions	○	요구획득단계의 전활동
UCD 5.5: Specify system and use of system	○	아키텍처단계-요구사항분석
UCD 5.6: Develop prototypes	○	아키텍처단계-요구사항분석
UCD 5.7: Develop user training	○	인도단계-사용자교육
UCD 5.8: Develop user support	×	
UCD 6: Evaluate designs against requirements		
UCD 6.1: Specify and validate context of evaluation	○	아키텍처단계-테스트및전환계획
UCD 6.2: Evaluate early prototypes in order to define the requirements for the system	○	아키텍처단계-요구사항분석
UCD 6.3: Evaluate prototypes in order to improve the design	○	점진적개발-시스템테스트
UCD 6.4: Evaluate the system in order to check that the stakeholder and organizational requirements have been met	○	점진적개발-시스템테스트
UCD 6.5: Evaluate the system in order to check that the required practice has been followed	×	
UCD 6.6: Evaluate the system in use in order to ensure that it continues to meet organizational and user needs	○	인도단계-인수테스트및시스템관찰
UCD 7: Introduce and operate the system		
UCD 7.1: Management of change	×	
UCD 7.2: Determine impact on organization and stakeholder	×	
UCD 7.3: Customization and local design	×	
UCD 7.4: Deliver user training	○	인도단계-사용자교육
UCD 7.5: Support users in planned activities	×	
UCD 7.6: Ensure conformance to workplace ergonomic legislation	×	

### 3. 결론 및 추후 연구

본 논문에서는 CBD 방법론에 UCD 개념 및 프로세스를 통합하는 문제를 다루었다. UCD

의 개념 및 중요성에 대해 설명한 후 UCD의 활동이 일반적인 CBD 방법론의 각 단계와 어떻게 연관성을 맺는가를 기술하였다. 이에 기반

하여 사례연구로 실제적인 CBD 방법론중의 하나인 마르미-III에 UMM에서 규정하고 있는 44개의 UCD base practice를 활동 수준에서 대응 및 비교해보았다. 이로부터 UCD 관점에서 마르미-III에서 부족한 부분을 파악하고 개선할 수 있는 기회를 가질 수 있었다.

본 논문에서 소개된 연구로부터 보다 구체적인 UCD 프로세스 (세부 작업 수준)의 CBD 방법론으로의 통합화 작업이 현재 진행 중에 있다. 또한 UCD에서 제공하고 있는 유용한 방법 및 산출물 양식과 현재 CBD 방법론에서 제공하는 방법 및 산출물과의 비교/대응/검토 작업을 통해 이 부분에 대한 통합화 역시 필요한 작업이다. 이러한 일련의 작업을 통해 소프트웨어 시스템 개발자와 이를 사용하는 최종 사용자 측면을 모두 고려한 보다 체계적인 방법론의 틀이 마련될 것이다.

#### [참고문헌]

- [1] 마르미-III (version 3.0): 컴포넌트 기반 시스템 개발 방법론, 한국전자통신연구원, 2002.
- [2] A.L. Lederer and J. Prasad, "Nine management guidelines for better cost estimating", *Communications of the ACM*, Vol. 35(2), pp. 50-59, 1992.
- [3] B. Tedeschi, "Good web site design can lead to healthy sales", *New York Times*, August 30, 1999.
- [4] D.J. Mayhew, *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*, Morgan Kaufmann, 1999.
- [5] *Delta Method Handbook*, available at <http://www.deltamethod.net/index.htm>
- [6] ISO/IEC, 13407 Human-Centered Design Processes for Interactive Systems, ISO/IEC 13407, 1999.
- [7] J. Earchy, Usability Maturity Model, available at [http://www.processforusability.co.uk/Usability\\_test/html/downloads.html](http://www.processforusability.co.uk/Usability_test/html/downloads.html)
- [8] J. Nielsen, *Usability Engineering*, Academic Press, 1993.
- [9] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, *Human-Computer Interaction*, Addison-Wesley, 1994.
- [10] K. Vredenburg, S. Isensee, and C. Righi, *User-Centered Design: An Integrated Approach*, Prentice Hall PTR, 2002.
- [11] L.L. Constantine and L.A.D. Lockwood, *Software for Use: A Practical Guide to the Models and Methods for Usage-Centered Design*, ACM Press, 1999.
- [12] L. Enos, "Report: E-holiday glitches could cost \$15B", *E-Commerce Times*, October 17, 2000.
- [13] M.J. Willshire, "Where SE and HCI meet: A position paper", in *Proceedings of ICSE 2003 Workshop on Bridging the Gaps between SW Engineering and HCI*, 2003.
- [14] M. Rauterberg, "User-centered design: what, why, and when", in *Proceedings of TEKOM Conference 2003*, 2003.
- [15] M.V. Harmelen (ed.), *Object Modeling and User Interface Design*, Addison Wesley, 2001.
- [16] Standish Group, *Technical Report*, available at <http://www.scs.carleton.ca/~beau/PM/Standish-Report.html>
- [17] T.K. Landauer, *The Trouble with Computers*, MIT Press, 1995.
- [18] *Usability Tools and Methods*, available at <http://www.usabilitynet.org/tools.htm>
- [19] W.M. Newman and M.G. Lamming, *Interactive System Design*, Addison-Wesley, 1995.
- [20] X. Ferre, "Integration of usability techniques into the software development process", in *Proceedings of ICSE 2003 Workshop on Bridging the Gaps between SW Engineering and HCI*, 2003.