How to Integrate Usability into the Software Development Process

Tutorial H10

ICSE 2006 - Shanghai (China)
28 May, 2006

Natalia Juristo
Universidad Politecnica de Madrid (Spain)
natalia@fi.upm.es

Xavier Ferre
Universidad Politecnica de Madrid (Spain)
xavier@fi.upm.es
Framework for Usability Integration into the Software Development Process

Xavier Ferré Grau
Natalia Juristo

Overview

- Usability Basics
- Framework Generation
  - Study of HCI Activities and Techniques
  - Characterization and Selection of HCI Techniques
  - Relationship to Development Process Activities
  - Application Times
  - Views of the Integration Framework
- Framework Use
- Overview of HCI Techniques
- Exercise on Framework Use

Tutorial Goals

- Usability: the major omission of developers
- The importance of usability
- Basic usability concepts
- Some techniques for improving usability
- Framework for integrating usability into the development process
Usability is a pending issue

- There are so many software products with immature usability that we all acknowledge the low level of use of usability methods

- Usability is not properly addressed in most developments, on spite of its importance

Quality includes usability

- Usability is a **basic** software system feature

- Several software quality attribute classifications agree on considering usability as a **quality attribute**
Quality Criteria

- Functionality
- Reliability
- Performance
- Usability
- Maintainability
- Portability

(According to ISO)

Usability definition

- Usability can be seen as **Quality in Use**
  - Usability reflects
    - Learnability
    - Efficiency of use
    - Reliability in use
    - Satisfaction

Usability and UI are not synonymous

- A system's usability relates closely to the software's overall functionality
- UI vs. Interaction
  - UI = The visible part of the system
  - Interaction = The coordination of information exchange user-system
Misunderstanding

- Usability problems can be fixed in the later development stages
- This misunderstanding prevents the proper incorporation of usability features into software development

Can usability really wait?

- If usability is relegated to the end of the development process, then there is no time left to make a difference
- Interaction design can have major impact on the overall application

Usability integration into development: SE + HCI

- Both the HCI and SE communities play a crucial role in the development of usable software
- The HCI community has the knowledge about which features a software system must provide to be usable
- The SE community has the knowledge about software systems development
Usability integration into development: Difficulties

- Different viewpoints between usability people and software developers
- Specialist skills required

Integration of usability into the software development process is not easy and obvious at all

SE-HCI Integration Framework

SE vs. HCI

- Software Engineering (SE)
  - Systematic, disciplined, quantifiable approach to software development
  - Special emphasis on software development process definition

- Human-Computer Interaction (HCI)
  - User-centered approach
  - Techniques for managing usability throughout development
Obstacles to HCI-SE Integration

- Strong differences between HCI and SE
  - HCI processes sphere
    - Example: Requirements
  - Terminology
    - Example: User Interface (UI) Design
  - When to consider usability during development

- HCI activities and techniques are not presented properly from a SE viewpoint

Development Process Conditions

- Characteristics of a user-centred process
  - Active user involvement
  - A proper understanding of user and task requirements
  - Multidisciplinary knowledge
  - Iterative development

- Only condition to be met
  - Iterative development

Steps to obtain the Framework
**Study of Activities and Techniques in a User-Centered Process**

1. Study of Acts and Techniques
2. Characterization and Selection
3. Relationship to Process Acts
4. Application Times
5. Integration Framework

**Study of Usability Activities by Source**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Nielsen94</th>
<th>Hix95</th>
<th>Mayhew99</th>
<th>Constantin99</th>
<th>Wixon97</th>
<th>Shneiderman98</th>
<th>Preece94</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities</td>
<td>Specification of the Context of Use</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>Usability Specifications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>Design Activities</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>Evaluation Activities</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>Task Analysis</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>User Analysis</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>Usability Specifications</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Representative Activities in a User-Centered Process**

<table>
<thead>
<tr>
<th>Analysis Activities</th>
<th>Design Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification of the Context of Use</td>
<td>Develop Product Concept</td>
</tr>
<tr>
<td>User Analysis</td>
<td>Prototyping</td>
</tr>
<tr>
<td>Task Analysis</td>
<td>Interaction Design</td>
</tr>
<tr>
<td>Usability Specifications</td>
<td></td>
</tr>
</tbody>
</table>

| Evaluation Activities | |
|-----------------------| |
| Usability Evaluation | |
Difficulties in the Study of HCI techniques from a SE viewpoint

- Diversity of HCI techniques
- Differences between authors
- Terminological diversity
- Lack of assignment of techniques to activities

Identification and Classification Process

1. Compilation of the techniques mentioned in the considered sources
2. Grouping of techniques that refer to the same basic idea
3. Assignment to the HCI activity that they best fit
4. Choice of the most representative name for each basic technique

Technique Classification: Techniques related to the Specification of the Context of Use
Characterization and Selection of HCI Techniques

1. Study of Acts and Techniques
2. Characterization and Selection
3. Relationship to Process Acts
4. Application Times
5. Integration Framework

Technique Characterization

- Source: 94 compiled techniques
- Objective: To provide information about how useful each technique is
- Criteria
  - User Participation
  - Training Needs
  - General Applicability
  - Proximity to SE
  - Usability Improvement/Effort Ratio
  - Representativeness
- Total Rating

Summary Criterion: Total Rating

- All the considered criteria are summarized in a criterion called “Overall Rating”
  - Very useful
  - Useful
  - Not very useful
Characterization of Analysis-Related Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Usefulness</th>
<th>Applicability</th>
<th>Representativeness</th>
<th>Overall Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Modeling</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Focus Group</td>
<td>very high</td>
<td>no</td>
<td>high</td>
<td>very useful</td>
</tr>
<tr>
<td>Pairwise Delphi</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
</tr>
<tr>
<td>Usability Engineering Interviews</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Usability Testing</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>medium</td>
</tr>
<tr>
<td>Lab Studies</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>User Profiling</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Task Scenarios</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>high</td>
</tr>
<tr>
<td>HTA</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Affinity Diagrams</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Usability Specifications</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Personas</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Card Sorting</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>User Role Map</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>Usability Specifications</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Ethnographic Observation</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Contextual Inquiry</td>
<td>high</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Competitive Analysis</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
</tbody>
</table>

**Technique Selection**

- Keep complexity at a reasonable level
- There are too many techniques available
- Select only “very useful” and “useful” techniques
- 35 selected techniques
- Choose a basic reference for each technique
- A pointer the developer may resort to for additional information

**Example of an HCI Technique: Paper Prototyping**

**Description:**

Prototypes allow designers to communicate more effectively with users and they reduce the need and cost of reworking that can occur when products need to be revised later in the life cycle. We need to build prototypes because abstract technical specifications and models are not a good way of communicating when we want to involve users in the design process. Paper prototyping, and especially rapid prototyping, is closely related to iterative design. For prototypes to be effective, they should be built at a minimum cost in terms of resources and time. The difference from traditional software engineering system prototypes is again a difference of focus. Prototypes are useful for usability purposes when they depict mostly system-user interaction, so that they convey how the system will work from the user point of view. So, prototypes can be used to test design ideas with users and to gather their feedback.

[...]

**Basic Reference:**

Relationship Between HCI and SE Process Terminology

- Employ SE process terminology
  - So developers may understand where they should apply usability techniques
- Take the following steps
  1. Map HCI activities to SE activities
  2. Assign HCI techniques to the SE activity group in which their application is more useful

Analysis, Design & Evaluation (HCI vs. SE)
Requirements Elicitation and Analysis
- User Analysis
- Task Analysis
- Develop Product Concept
- Prototyping

Requirements Specification

Usability Evaluation
- Expert Evaluation
- Usability Testing
- Follow-Up Studies of Installed Systems

Interaction Design

Mapping of HCI Activities to SE Activities

Development Activities with a Usability Impact

Assignment of HCI Techniques to SE Activities

- Technique assignment to activity types
  - Activity type whose objectives the technique fulfills
  - First candidate is the activity type to which its HCI activity maps
- Requirements Elicitation and Analysis have been grouped together
  - The HCI makes no clear distinction between Elicitation and Analysis
  - 15 HCI techniques assigned to the combined activity group
### Assignment of HCI Techniques to SE Activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Identification and Analysis</td>
<td>Customer Focus, Use Case Engineering, Scenarios, Analysis, Conceptual Design, Information Architecture, etc.</td>
</tr>
<tr>
<td>User Analysis</td>
<td>User Profiles, Use Case Maps, Usability Specifications, etc.</td>
</tr>
<tr>
<td>Task Analysis</td>
<td>Task Analysis, Task Scenarios, Develop Product Concept, Task Scenarios, etc.</td>
</tr>
<tr>
<td>Requirements Specification</td>
<td>Usability Specifications, Requirements Specifications, etc.</td>
</tr>
<tr>
<td>Requirements Validation</td>
<td>Usability Specifications, Requirements Specifications, etc.</td>
</tr>
</tbody>
</table>

### Study of Application Times

1. Study of Acts and Techniques
2. Characterization and Selection
3. Relationship to Process Acts
4. Application Times
5. Integration Framework

### Usability-Significant Stages in an Iterative Process

- Not all cycles are the same
- Relevant milestones regarding HCI technique application
  - Product concept established
  - Part of the system installed and working
Division of Iterative Stages into Cycles

- Initial Cycles
- Central Cycles
- Evolution Cycles

Beginning of Development

- Time

Milestone: Product Concept Established
Milestone: Part of the System Installed and Working

Assignment of HCI Techniques to Development Stages

- Aim
  - To identify the best application time for each technique
- Technique classification according to their fitness values
  - Especially well-matched
  - Neutral
  - Not usual

Suitability of Selected HCI Techniques for each Development Stage
Framework

1. Study of Acts and Techniques
2. Characterization and Selection
3. Relationship to Process Acts
4. Application Times
5. Integration Framework

Framework views

- Views
  - By HCI techniques
  - By development activities
  - By application times

Integration Framework Overview

- All the knowledge generated for each technique is included in each view
  - Technique characterization
  - Activity type where it applies
  - Suitability for each moment in development time
  - Basic reference
- Each view organizes techniques according to their particular criteria
  - "very useful" vs. "useful" techniques
Usage of the Framework

- Views are complementary ways of accessing the knowledge
- They can be flexibly combined in any situation
  - Preliminary selection of what techniques to include
  - Planning of next cycle
  - ...

Exercise: Development Process

- Definition of Functional Requirements
- Elicitation and Definition of Technical Specifications
- Design
- Suggestions for Changes
- Technical Specifications
- Suggestions for Changes
- Alpha Version
- Final Program Release

Web Tool
http://is.ls.fi.upm.es/udis/miembros/xavier/usabilityframework/
Overview of Selected HCI Techniques

- Requirements Elicitation and Analysis
  - Personas
  - Essential Use Cases
  - Card Sorting
  - Scenarios & Storyboards
  - Paper Prototypes
- Requirements Specification
- Usability Specifications
- Requirements Validation
- Inspections
- Interaction Design
- Menu Trees
- Usability Evaluation
- Inspections
- Thinking Aloud
- User Feedback

Personas

- A precise descriptive model of the user, what he wishes to accomplish, and why
  - Composite archetypes based on behavioral data gathered from many actual users through ethnographic interviews
- Not real people
  - But based on the behaviors and motivations of real people
  - Represent real people throughout the design process
Why Using Personas

- The best way to successfully accommodate a variety of users is to design for specific types of individuals with specific needs.
  - Choose the right individuals to design for, ones whose needs represent the needs of a larger set of key constituents.
  - Prioritize the design elements to address the needs of the most important users without significantly inconveniencing secondary users.
- Personas provide a tool for
  - Understanding user needs.
  - Differentiating between different types of users.
  - Prioritizing which users are the most important target in the design of function and behavior.

Personas: Example

- Costas
  - He is 35 years old, he is married with 2 children. He’s got a diploma on Mathematics and a Masters degree on Economics. He uses a laptop for his work, and the main applications he uses are: MS Excel, Word, etc. He usually looks after information on the stock market in the web (he has some stocks of his company). He owns a cutting-edge mobile phone, that holds pictures of his children he may show to his friends. He manages a team of 3-10 people and he is accounted responsible for the productivity of his team. His main concerns are managing the tasks the people on his team are assigned to.

Use Cases in SE

- A use case is a case of use, or one kind of use to which a system can be put. It is:
  - Supplied functionality
  - An external, ‘black-box’ view
  - A narrative description
  - Interaction between a user and a system
  - A use of the system that is completely meaningful to the user
- Extensively used in object-oriented analysis and design, but with a wrong focus:
  - Implicit design assumptions on the form of the UI
  - Too close to implementation
  - Away from the user sphere
Essential Use Cases

- Two levels of abstraction for describing use cases:
  - Essential use cases
  - Detailed or concrete use cases
- An essential use case is based on the purpose or intentions of a user rather than on the concrete steps by which such intentions are carried out.
- It has three components:
  - A statement of the overall user purpose or intention expressed within the use case.
  - A two-part narrative that structures the interaction between user intentions and system responsibilities.

Example of Essential Use Case

<table>
<thead>
<tr>
<th>ESSENTIAL USE CASE</th>
<th>DETAILED USE CASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>gettingCash</td>
<td>gettingCash</td>
</tr>
<tr>
<td>USER INTENTION</td>
<td>USER ACTION</td>
</tr>
<tr>
<td>identify self</td>
<td>insert card</td>
</tr>
<tr>
<td>choose</td>
<td>read magnetic stripe</td>
</tr>
<tr>
<td>take cash</td>
<td>request PIN</td>
</tr>
<tr>
<td></td>
<td>verify PIN</td>
</tr>
</tbody>
</table>

Card Sorting

- A technique used to understand how users envision the organization of information.
- It consists on asking users to categorize a list of terms:
  - Useful when a list of terms is already available.
- Good for defining information architectures (for websites).
- Advantages:
  - Simple, easy to understand and apply.
  - Inexpensive.
  - Quick.
Card Sorting: The Sorting Session

- Explain the card sorting process to the participants.
- Encourage them to organize the cards in a way that makes sense to them.
- After the participants are done grouping the topics, they are asked to label each one of the resulting groups.

Scenarios

- Personalized, fictional story with characters, events, products and environments.
- Scenarios help the development team to explore ideas and the ramifications of design solutions in particular situations.
- It is an encapsulated description of:
  - An individual user
  - Using a specific set of computer facilities
  - To achieve a specific outcome
  - Under specified circumstances
  - Over a certain time interval

Example of Scenario (I)

Path Smith has just arrived at Geneva International Airport en route to a large conference on Human-Computer Interaction. Pat is carrying a laptop and a large, heavy suitcase and needs to get to the conference centre quickly. Looking around for a bank in order to get some local currency, Pat sees the Eurochange machine with its blue flag style logo showing a circle of twelve stars.

Pat goes up the machine. It seems similar to the automatic teller machine that Pat uses regularly. Pat puts down the suitcase, takes out a credit card and inserts it into the slot. A message is displayed on the screen:

`Enter your PIN`

Pat thinks for a few moments and then types a four-digit number on the numerical pad, listening to the reassuring beep that follows each number pressed. The machine pauses for a few seconds and then displays:

`Select currency required`
Example of Scenario (II)

Pat pauses again. What is the currency in Switzerland? Pat
browses the currencies available, sees "Swiss Franc (CHF)"
and presses the key. The machine displays the message:

```
Exchange rate is 1.47 CHF to 1 EUR
Enter amount required in Swiss Francs in units of [10]
```

Press <Proceed>

Pat types 253 and presses <Proceed>. A message is displayed:

```
Machine deals in bank notes only
Smallest bank note is [10] CHF
Enter new amount to obtain CHF or press <Cancel>
```

Pat enters 260 and presses <Proceed>. There is a whirring
noise and a few other indeterminate clunks and clicks. The
credit card is returned from the card entry slot and the
money deposited in the delivery slot, with a printout of
the transaction.

Storyboards

- Technique from the movie making industry
- A storyboard captures the procedure for doing a task
  pictorially
- Each frame in the storyboard captures a single scene
  - An interaction between two people
  - A person and the system
  - A person and an artifact
  - A system step
- It requires more drawing skills
  than paper prototypes

Example of Storyboard
Prototypes

- A prototype is a representation of all or part of a product or system that, although limited in some way, can be used for evaluation
- Good communication tool with users and other non-technical stakeholders
- Iterative development relies on prototyping techniques to a great extent
- HCI offers to software development a kind of cheap and quick prototypes, which are the less elaborate ones:
  - Paper and chauffeured prototypes
  - Wizard of Oz technique

Paper Prototypes

- Rough prototypes in paper
- They serve to start the co-designing with users
- Paper prototypes are NOT demos
- They convey clearly to stakeholders that there is a lot of work to be done yet
  - And changes are easier to make because not a big amount of effort has been spent in their creation
- They allow for easy and quick changes

Paper Prototypes: Example
Chauffeured Prototypes

- Animation of the intended system behavior using paper prototypes
- Used for evaluation of prototypes with users, mimicking the usage of the system
- The user watches while a member of the development team “drives” the system
  - The user points out which actions he or she would take
  - The developer changes from one screen to another, and explains the intended system reaction to each user action

Menu Trees

- They represent
  - The menu navigational structure, or
  - The dialog boxes navigation
- The menu structure may be very big for medium-big sized projects
  - It can be mounted on a wall
- Card sorting may be used to decide on the menu structure
- For web development: Website maps

Website Map: Example
Usability Inspections

- The goal of any inspection is to find defects
  - Usability inspections are aimed at identifying usability defects
- It is a systematic process (vs. heuristic evaluations which are less formal)
- Two types of inspections according to the focus:
  - Consistency inspections
  - Conformance inspections (with the style guide, or a standard)
- Collaborative usability inspection: A specific kind of inspection with participation of different stakeholders

Thinking Aloud Protocol (I)

- The user is asked to talk out loud while trying to perform the tasks in a usability test session
  - What they are trying to do
  - Which problems they encounter and which strategy they devise in order to overcome them
  - Their (not fulfilled) expectations on the system behavior
  - How they interpret what the UI shows to them
- Not compatible with performance measures
  - Users are slowed down by the effort of verbalizing their thoughts
Thinking Aloud Protocol (II)

- People tend to rationalize on their actions and the system responses
  - They may blame themselves for errors
- During the test introduction to participants, they must be encouraged to think out loud
- Variants of the thinking aloud protocol:
  - Constructive Interaction
  - Retrospective Testing
  - Critical Incident Taking
  - Coaching Method

User Feedback

- Users are the best feedback source for usability problems or weaknesses
- It allows gauging user attitudes to the system and eliciting useful suggestions for improvement
- Different channels may be employed:
  - Online or telephone consultants
  - Online suggestion box or trouble reporting
  - Online bulletin board or newsgroup
Table 1 - Context of Use Specification-Related Techniques

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitive Analysis</td>
<td>Competitive Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financial Impact Analysis</td>
<td>Financial Impact Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contextual Inquiry</td>
<td>Contextual Inquiry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affinity Diagrams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Affinity Diagrams</td>
<td></td>
</tr>
<tr>
<td>Ethnographic Observation</td>
<td>Ethnography</td>
<td></td>
<td></td>
<td></td>
<td>Ethnographic Observation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JEM¹</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Profiles</td>
<td>User Profiles</td>
<td></td>
<td></td>
<td></td>
<td>Usage Profiles</td>
<td>Structured User Role Model</td>
<td>User Profiles Questionnaires</td>
</tr>
<tr>
<td>User Role Map</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>User Role Map</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational Modeling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operational Modeling</td>
<td>Platform Capabilities and Constraints</td>
<td></td>
</tr>
<tr>
<td>Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

User Analysis

Essential Use Cases
HTA²
GOMS³
NGOMSL
Object-Action Interface Model
Task Scenarios
Task Sorting⁴

Task Analysis

Performance Goals
Satisfaction Goals
Usability Goal Line
Preference Goals

This recent technique is not mentioned in any of the sources, but is included for the reasons detailed above in section 3.2.1

Table 2 - Usability Specifications-Related Techniques

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability Specifications</td>
<td>Usability Specifications</td>
<td></td>
<td></td>
<td></td>
<td>Usability Goals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Goals</td>
<td>Objective Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Performance Goals</td>
<td></td>
</tr>
<tr>
<td>Satisfaction Goals</td>
<td>Subjective Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Satisfaction Goals</td>
<td></td>
</tr>
<tr>
<td>Usability Goal Line</td>
<td>Usability Goal Line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Preference Goals</td>
<td></td>
</tr>
<tr>
<td>Preference Goals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

¹ JEM: Joint Essential Modeling
² HTA: Hierarchical Task Analysis
³ GOMS: Goals, Operations, Methods and Selection Rules
⁴ Task Sorting is a variation on the Card Sorting technique
### Table 3 - Analysis-Related Techniques not Specific to any Activity

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis generally</td>
<td>Card Sorting</td>
<td>Card Sorting</td>
<td>Card Sorting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis generally</td>
<td>Affinity Clustering</td>
<td>Affinity Clustering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis generally</td>
<td>Criteria Prioritization</td>
<td>Criteria Prioritization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis generally</td>
<td>Threshold Voting</td>
<td>Threshold Voting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis generally</td>
<td>Task Sorting</td>
<td>Task Sorting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4 - Design-Related Techniques

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop Product Concept</td>
<td>Scenarios and Storyboards</td>
<td>Scenarios, Storyboards and Snapshots</td>
<td>Scenarios and Storyboards</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Brainstorming</td>
<td>Prototyping</td>
<td>Prototyping</td>
<td>Visual Brainstorming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototyping</td>
<td>Scenarios</td>
<td>Prototyping</td>
<td>Prototyping</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototyping</td>
<td>Requirements Animation</td>
<td>Active Prototypes</td>
<td>“High-Fidelity” Mock-ups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototyping</td>
<td>Low-Fidelity Mock-ups</td>
<td>Passive Prototypes</td>
<td>“Low-Fidelity” Mock-ups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototyping</td>
<td>Chauffeured Prototyping</td>
<td>Chauffeured Prototyping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototyping</td>
<td>Wizard of Oz Prototypes</td>
<td>Wizard of Oz Prototypes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Screen Pictures</td>
<td>Scenarios and Screen Pictures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Product Style Guide</td>
<td>Product Style Guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Grammars</td>
<td>Grammars</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>UAN⁵</td>
<td>UAN</td>
<td>UAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>TAG⁶</td>
<td>TAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Menu-Selection and Dialog Box Trees</td>
<td>Menu-Selection and Dialog Box Trees</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Interface State Transition Diagrams</td>
<td>Interface State Transition Diagrams</td>
<td>Transition Diagrams</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Statecharts</td>
<td>Statecharts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Content Model of Interface</td>
<td>Content Model of Interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

⁵ UAN: User Action Notation
⁶ TAG: Task-Action Grammars
### Table 5 - Design-Related Techniques not Specific to any Activity

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both-And Design</td>
<td>Hix, Nielsen, Preece, Shneiderman</td>
<td>Both-And Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel Design</td>
<td>Parallel Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>Cost / Importance Analysis</td>
<td>Impact Analysis</td>
<td>Impact Analysis</td>
<td>Organizing Help by Use Cases</td>
<td>IBIS and PHI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizing Help by Use Cases</td>
<td>IBIS and PHI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design Space Analysis</td>
<td>Design Space Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Claims Analysis</td>
<td>Claims Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 6 - Usability Evaluation Techniques for Expert Reviews

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspections</td>
<td>Conformance Inspections</td>
<td>Heuristic Evaluation</td>
<td>Heuristic Evaluation</td>
<td>Heuristic Evaluation</td>
<td>Heuristic Evaluation</td>
<td>Heuristic Evaluation</td>
<td></td>
</tr>
<tr>
<td>Guideline Reviews</td>
<td>Consistency Inspection</td>
<td>Consistency Inspection</td>
<td>Consistency Inspection</td>
<td>Consistency Inspection</td>
<td>Consistency Inspection</td>
<td>Consistency Inspection</td>
<td></td>
</tr>
<tr>
<td>Collaborative Usability Inspections</td>
<td>Cognitive Walkthrough</td>
<td>Cognitive Walkthrough</td>
<td>Cognitive Walkthrough</td>
<td>Cognitive Walkthrough</td>
<td>Cognitive Walkthrough</td>
<td>Cognitive Walkthrough</td>
<td></td>
</tr>
<tr>
<td>Walkthroughs</td>
<td>Pluralistic</td>
<td>Pluralistic Walkthrough</td>
<td>Pluralistic Walkthrough</td>
<td>Pluralistic Walkthrough</td>
<td>Pluralistic Walkthrough</td>
<td>Pluralistic Walkthrough</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7 - Usability Evaluation Techniques for Usability Testing

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability Testing</td>
<td>Thinking Aloud</td>
<td>Thinking Aloud</td>
<td>Thinking Aloud Protocol</td>
<td>Thinking Aloud</td>
<td>Talk to me (thinking aloud)</td>
<td>Formal usability inspections (in early stages)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constructive Interaction</td>
<td>Constructive Interaction</td>
<td>Constructive Interaction</td>
<td>Constructive Interaction</td>
<td>Constructive Interaction</td>
<td>Constructive Interaction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retrospective Test</td>
<td>Retrospective Testing</td>
<td>Post-Event Protocol</td>
<td>Deferred Reflection</td>
<td>Retrospective Test</td>
<td>Retrospective Test</td>
<td></td>
</tr>
</tbody>
</table>

---

7 IBIS: Issue-Based Information Systems
8 PHI: Procedural Hierarchy of Issues
<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Incident Taking</td>
<td>Critical Incident Taking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coaching Method</td>
<td>Coaching Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Measured Performance</td>
<td>Benchmark Tasks</td>
<td></td>
<td></td>
<td></td>
<td>Measured Performance</td>
<td>Formal usability tests (in later stages)</td>
<td></td>
</tr>
<tr>
<td>Post-Test Feedback</td>
<td>Post-Test Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laboratory Usability Testing</td>
<td>Laboratory Testing</td>
<td></td>
<td></td>
<td></td>
<td>Usability Testing and Laboratories</td>
<td>Laboratory Usability Testing</td>
<td></td>
</tr>
<tr>
<td>Field Usability Testing</td>
<td>Field Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video Recording</td>
<td>Video-taping</td>
<td></td>
<td></td>
<td></td>
<td>Video Recording</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verbal Protocol</td>
<td>Audio-taping</td>
<td></td>
<td></td>
<td></td>
<td>Verbal Protocol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logging Actual Use</td>
<td>Internal Instrumentation of the Interface</td>
<td></td>
<td></td>
<td></td>
<td>Logging Actual Use</td>
<td>Software Logging</td>
<td>Continuous User-Performance Data Logging</td>
</tr>
<tr>
<td>Time-Stamped Keypresses</td>
<td>Time-Stamped Keypresses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Logging</td>
<td>Interaction Logging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote Control Evaluation</td>
<td>Remote Control Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote Videoconferencing Evaluation</td>
<td>Remote Videoconferencing Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8 - Usability Evaluation Techniques for Follow-up Studies of Installed Systems

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Follow-up Studies of Installed Systems</td>
<td>Direct Observation</td>
<td>Observation</td>
<td>Direct Observation</td>
<td></td>
<td></td>
<td>Usage Studies - Random Observation</td>
<td></td>
</tr>
<tr>
<td>Random Observation</td>
<td>Structured Interviews</td>
<td>Structured Interviews</td>
<td>Structured Interviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Questionnaires and Surveys</td>
<td>Questionnaires</td>
<td>Questionnaires and Surveys</td>
<td>Surveys</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interviews</td>
<td>Interviews</td>
<td>Interviews</td>
<td>Interviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structured Interviews</td>
<td>Flexible Interviews</td>
<td>Flexible Interviews</td>
<td>Flexible Interviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Focus Groups</td>
<td>Focus Groups</td>
<td>Focus Groups</td>
<td>Focus Group Discussions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logging Actual Use</td>
<td>Internal Instrumentation of the Interface</td>
<td>Logging Actual Use</td>
<td>Software Logging</td>
<td>Continuous User-Performance Data Logging</td>
<td>Instrumented Remote Evaluation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-Stamped Keypresses</td>
<td>Time-Stamped Keypresses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Logging</td>
<td>Interaction Logging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Activation Software Monitors</td>
<td>Usage Studies – Software Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Feedback</td>
<td>User Feedback</td>
<td></td>
<td></td>
<td></td>
<td>Online Suggestion Box or Trouble Reporting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>TECHNIQUE</td>
<td>Hix, 93</td>
<td>Nielsen, 93</td>
<td>Preece, 94</td>
<td>Shneiderman, 98</td>
<td>Constantine, 99</td>
<td>Mayhew, 99</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------</td>
<td>---------</td>
<td>-------------</td>
<td>------------</td>
<td>----------------</td>
<td>------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Online or Telephone Consultants</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Online Bulletin Board or Newsgroups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Newsletters and Conferences</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semi-Instrumented Remote Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Table 9 - Other Usability Evaluation Techniques</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>TECHNIQUE</th>
<th>Hix, 93</th>
<th>Nielsen, 93</th>
<th>Preece, 94</th>
<th>Shneiderman, 98</th>
<th>Constantine, 99</th>
<th>Mayhew, 99</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Usability evaluation generally</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predictive Metrics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedural</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytic Evaluation Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperative Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technique</td>
<td>U. P.</td>
<td>Training Needs</td>
<td>Applicability</td>
<td>Proximity to SE</td>
<td>Improvement/Effort</td>
<td>Representativeness</td>
<td>Overall Rating</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Card Sorting</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>Very useful</td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>Very useful</td>
</tr>
<tr>
<td>Personas</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>Very useful</td>
</tr>
<tr>
<td>Usability Specifications</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>Very useful</td>
</tr>
<tr>
<td>Affinity Diagrams</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Competitive Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Contextual Inquiry</td>
<td>yes</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>Useful</td>
</tr>
<tr>
<td>Ethnographic Observation</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>2</td>
<td>Useful</td>
</tr>
<tr>
<td>HTA</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>JEM</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Task Scenarios</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>User Profiles</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>5</td>
<td>Useful</td>
</tr>
<tr>
<td>User Role Model</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Financial Impact Analysis</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>GOMS</td>
<td>no</td>
<td>very high</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>3</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Object-Action Interface Model</td>
<td>no</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Operational Modeling</td>
<td>no</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>2</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Technique</td>
<td>U. P.</td>
<td>Training Needs</td>
<td>Applicability</td>
<td>Proximity to SE</td>
<td>Improvement/Effort</td>
<td>Representativeness</td>
<td>Overall Rating</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Menu-Selection and Dialog Box Trees</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>Very useful</td>
</tr>
<tr>
<td>Paper Prototypes</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>Very useful</td>
</tr>
<tr>
<td>Scenarios and Storyboards</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>3</td>
<td>Very useful</td>
</tr>
<tr>
<td>Content Model of Interface</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Context Navigation Map</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>3</td>
<td>Useful</td>
</tr>
<tr>
<td>Interface State Transition Diagrams</td>
<td>no</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>2</td>
<td>Useful</td>
</tr>
<tr>
<td>Organizing Help by Use Cases</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Product Style Guide</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Visual Brainstorming</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>1</td>
<td>Useful</td>
</tr>
<tr>
<td>Both-And Design</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Grammars</td>
<td>no</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Parallel Design</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Screen Pictures</td>
<td>no</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Statecharts</td>
<td>no</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>TAG</td>
<td>no</td>
<td>very high</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>UAN</td>
<td>no</td>
<td>very high</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>2</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Wizard of Oz Prototypes</td>
<td>yes</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>Not very useful</td>
</tr>
<tr>
<td>Technique</td>
<td>U. P.</td>
<td>Training Needs</td>
<td>Applicability</td>
<td>Proximity to SE</td>
<td>Improvement/Effort</td>
<td>Representativeness</td>
<td>Overall Rating</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>4</td>
</tr>
<tr>
<td>Thinking Aloud</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>5</td>
</tr>
<tr>
<td>User Feedback</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
</tr>
<tr>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>6</td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>1</td>
</tr>
<tr>
<td>Laboratory Usability Testing</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>4</td>
</tr>
<tr>
<td>Logging Actual Use</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>5</td>
</tr>
<tr>
<td>Measured Performance</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3</td>
</tr>
<tr>
<td>Pluralistic Walkthrough</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
</tr>
<tr>
<td>Post-Test Feedback</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
</tr>
<tr>
<td>Questionnaires, Interviews &amp; Surveys</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>3</td>
</tr>
<tr>
<td>Audio-/Video-Taping</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>3</td>
</tr>
<tr>
<td>Cooperative Evaluation</td>
<td>yes</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
</tr>
<tr>
<td>Direct Observation</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>3</td>
</tr>
<tr>
<td>Experimental Tests</td>
<td>yes</td>
<td>very high</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>2</td>
</tr>
<tr>
<td>Field Usability Testing</td>
<td>yes</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>2</td>
</tr>
<tr>
<td>Focus Groups</td>
<td>yes</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>2</td>
</tr>
<tr>
<td>Predictive Metrics</td>
<td>no</td>
<td>very high</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>2</td>
</tr>
</tbody>
</table>
Assignment of HCI Techniques to SE Activity Groups
Table 1- Usability Techniques Grouped according to the SE Activity Group in which their Application is most Useful

<table>
<thead>
<tr>
<th>SE Activity Type</th>
<th>Usability Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Elicitation and Analysis</td>
<td>Card Sorting</td>
</tr>
<tr>
<td></td>
<td>Affinity Diagrams</td>
</tr>
<tr>
<td></td>
<td>Competitive Analysis</td>
</tr>
<tr>
<td></td>
<td>Contextual Inquiry</td>
</tr>
<tr>
<td></td>
<td>Ethnographic Observation</td>
</tr>
<tr>
<td></td>
<td>JEM</td>
</tr>
<tr>
<td>User Analysis *</td>
<td>Personas</td>
</tr>
<tr>
<td></td>
<td>User Profiles</td>
</tr>
<tr>
<td></td>
<td>User Role Map</td>
</tr>
<tr>
<td>Task Analysis *</td>
<td>Essential Use Cases</td>
</tr>
<tr>
<td></td>
<td>HTA</td>
</tr>
<tr>
<td></td>
<td>Task Scenarios</td>
</tr>
<tr>
<td>Develop Product Concept *</td>
<td>Scenarios and Storyboards</td>
</tr>
<tr>
<td>Prototyping *</td>
<td>Visual Brainstorming</td>
</tr>
<tr>
<td></td>
<td>Paper Prototypes</td>
</tr>
<tr>
<td>Requirements Specification</td>
<td>Usability Specifications</td>
</tr>
<tr>
<td>Requirements Validation</td>
<td>Inspections</td>
</tr>
<tr>
<td></td>
<td>Collaborative Usability Inspections</td>
</tr>
<tr>
<td></td>
<td>Cognitive Walkthroughs</td>
</tr>
<tr>
<td></td>
<td>Heuristic Evaluation</td>
</tr>
<tr>
<td></td>
<td>Pluralistic Walkthroughs</td>
</tr>
<tr>
<td>Interaction Design</td>
<td>Menu-Selection and Dialog Box Trees</td>
</tr>
<tr>
<td></td>
<td>Context Navigation Map</td>
</tr>
<tr>
<td></td>
<td>Content Model of Interface</td>
</tr>
<tr>
<td></td>
<td>Interface State Transition Diagrams</td>
</tr>
<tr>
<td></td>
<td>Product Style Guide</td>
</tr>
<tr>
<td></td>
<td>Impact Analysis</td>
</tr>
<tr>
<td></td>
<td>Organizing Help by Use Cases</td>
</tr>
<tr>
<td>Expert Reviews</td>
<td>Inspections</td>
</tr>
<tr>
<td></td>
<td>Collaborative Usability Inspections</td>
</tr>
<tr>
<td></td>
<td>Cognitive Walkthroughs</td>
</tr>
<tr>
<td></td>
<td>Heuristic Evaluation</td>
</tr>
<tr>
<td></td>
<td>Pluralistic Walkthroughs</td>
</tr>
<tr>
<td>Usability Testing</td>
<td>Thinking Aloud</td>
</tr>
<tr>
<td></td>
<td>Measured Performance</td>
</tr>
<tr>
<td></td>
<td>Laboratory Usability Testing</td>
</tr>
<tr>
<td></td>
<td>Post-Test Feedback</td>
</tr>
<tr>
<td>Follow-up Studies of Installed Systems</td>
<td>User Feedback</td>
</tr>
<tr>
<td></td>
<td>Logging Actual Use</td>
</tr>
<tr>
<td></td>
<td>Questionnaires, Interviews and Surveys</td>
</tr>
</tbody>
</table>

* These are not SE activity types, but usability activities included to offer developers a structured view of the 15 usability techniques that can be applied for requirements elicitation and analysis.
Suitability of HCI Techniques to Development Stages
Table 1 – Technique Fit with Moment in Development Time

<table>
<thead>
<tr>
<th>Content</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Especially Well-matched</td>
</tr>
<tr>
<td></td>
<td>Neutral</td>
</tr>
<tr>
<td></td>
<td>Not Usual</td>
</tr>
</tbody>
</table>
# Table 2 – How Selected Usability Techniques Fit the Iterative Development Stages

<table>
<thead>
<tr>
<th>SE Activity Type</th>
<th>Usability Technique</th>
<th>Iterative Development Stages</th>
<th>Initial Cycles</th>
<th>Central Cycles</th>
<th>Evolution Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Elicitation and Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis (Requirements Engineering)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card Sorting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Competitive Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affinity Diagrams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contextual Inquiry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethnographic Observation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Analysis *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Role Map</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Profiles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Analysis *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scenarios</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenarios and Storyboards</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Brainstorming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Concept Prototyping *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper Prototypes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Requirements Specification</strong></td>
<td>Usability Specifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Requirements Validation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Menu-Selection and Dialog Box Trees</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface State Transition Diagrams</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Style Guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context Navigation Map</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content Model of Interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizing Help by Use Cases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Expert Reviews</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thinking Aloud</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-Test Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Measured Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laboratory Usability Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Follow-up Studies of Installed Systems</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Questionnaires, Interviews and Surveys</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logging Actual Use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* These are not SE activity types but usability activities included to offer developers a structured view of the 15 usability techniques that can be applied for requirements elicitation and analysis.
### Table 1 – HCI Technique View: Proposed Usability Techniques in Alphabetical Order

<table>
<thead>
<tr>
<th>Technique</th>
<th>U. P.</th>
<th>Training Needs</th>
<th>Applicability</th>
<th>Proximity to SE</th>
<th>Improve-ment/Effort</th>
<th>Represen-tativeness</th>
<th>Overall Rating</th>
<th>Activity Type</th>
<th>Application Time</th>
<th>Basic Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affinity Diagrams</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis</td>
<td>well-matched</td>
<td>[Beyer, 98]</td>
</tr>
<tr>
<td>Card Sorting</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Requirements Elicitation and Analysis</td>
<td>neutral</td>
<td>[Robertson, 01]</td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>neutral</td>
<td>[Lewis, 97]</td>
</tr>
<tr>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>well-matched</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Competitive Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis</td>
<td>well-matched</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td>Context Model of Interface</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Interaction Design</td>
<td>neutral</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Contextual Inquiry</td>
<td>yes</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis</td>
<td>well-matched</td>
<td>[Beyer, 98]</td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>Requirements Elicitation and Analysis (Task Analysis)</td>
<td>neutral</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Ethnographic Observation</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>2</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis</td>
<td>well-matched</td>
<td>[Wixon, 96]</td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>6</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>neutral</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td>HTA</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis (Task Analysis)</td>
<td>well-matched</td>
<td>[Annett, 04]</td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Design</td>
<td>not usual</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>neutral</td>
<td>[Nielsen, 94]</td>
</tr>
<tr>
<td>Interface State Transition Diagrams</td>
<td>no</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>2</td>
<td>useful</td>
<td>Interaction Design</td>
<td>neutral</td>
<td>[Wasserman, 85]</td>
</tr>
<tr>
<td>JEM</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis</td>
<td>neutral</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Laboratory Usability Testing</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>not usual</td>
<td>[Dumas, 99]</td>
</tr>
<tr>
<td>Logging Actual Use</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>5</td>
<td>useful</td>
<td>Evaluation (Follow-Up Studies of Installed Systems)</td>
<td>not usual</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td>Measured Performance</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>not usual</td>
<td>[Dumas, 99]</td>
</tr>
<tr>
<td>Technique</td>
<td>U. P.</td>
<td>Training Needs</td>
<td>Applicability</td>
<td>Proximity to SE</td>
<td>Improvement/Effort</td>
<td>Representativeness</td>
<td>Overall Rating</td>
<td>Activity Type</td>
<td>Application Time</td>
<td>Basic Reference</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Menu-Selection and Dialog Box Trees</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>Interaction Design</td>
<td>neutral</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td>Organizing Help by Use Cases</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Design</td>
<td>neutral</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Paper Prototypes</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Requirements Elicitation and Analysis (Prototyping)</td>
<td>well-matched</td>
<td>[Snyder, 03]</td>
</tr>
<tr>
<td>Personas</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Requirements Elicitation and Analysis (User Analysis)</td>
<td>well-matched</td>
<td>[Cooper, 03a]</td>
</tr>
<tr>
<td>Pluralistic Walkthrough</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>well-matched</td>
<td>[Bias, 94]</td>
</tr>
<tr>
<td>Post-Test Feedback</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>not usual</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Product Style Guide</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Interaction Design</td>
<td>not usual</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td>Questionnaires, Interviews and Surveys</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Requirements Elicitation and Analysis (Product Concept)</td>
<td>well-matched</td>
<td>[Carroll, 97]</td>
</tr>
<tr>
<td>Scenarios and Storyboards</td>
<td>yes</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Requirements Elicitation and Analysis (Task Analysis)</td>
<td>neutral</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td>Task Scenarios</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis (Task Analysis)</td>
<td>neutral</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td>Thinking Aloud</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>5</td>
<td>very useful</td>
<td>Evaluation (Usability Testing)</td>
<td>not usual</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td>Usability Specifications</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>Requirements Specification</td>
<td>neutral</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td>User Feedback</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Evaluation (Follow-Up Studies of Installed Systems)</td>
<td>not usual</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td>User Profiles</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>5</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis (User Analysis)</td>
<td>well-matched</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td>User Role Map</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis (User Analysis)</td>
<td>well-matched</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td>Visual Brainstorming</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Requirements Elicitation and Analysis (Develop Product Concept)</td>
<td>well-matched</td>
<td>[Preece, 94]</td>
</tr>
</tbody>
</table>
Table 2 – Activity Type View: Proposed Usability Techniques Ranked by Activity Type

<table>
<thead>
<tr>
<th>Activity Type</th>
<th>Technique</th>
<th>U. P.</th>
<th>Training Needs</th>
<th>Applicability</th>
<th>Proximity to SE</th>
<th>Improvement/Effort</th>
<th>Representativeness</th>
<th>Overall Rating</th>
<th>Application Time</th>
<th>Basic Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis (Requirements Engineering)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements Elicitation and Analysis</td>
<td>Card Sorting</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Competitive Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Affinity Diagrams</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>not usual</td>
</tr>
<tr>
<td></td>
<td>Contextual Inquiry JEM</td>
<td>yes</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>useful</td>
<td>well-matched</td>
<td>not usual</td>
</tr>
<tr>
<td></td>
<td>Ethnographic Observation</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td>User Analysis *</td>
<td>Personas</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>3</td>
<td>very useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>User Role Map</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td>Task Analysis *</td>
<td>Essential Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>HTA</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>not usual</td>
</tr>
<tr>
<td></td>
<td>Task Scenarios</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td>Develop Product Concept *</td>
<td>Scenarios and Storyboards</td>
<td>yes</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>well-matched</td>
<td>not usual</td>
</tr>
<tr>
<td></td>
<td>Visual Brainstorming</td>
<td>yes</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>not usual</td>
</tr>
<tr>
<td>Prototyping *</td>
<td>Paper Prototypes</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td>Requirements Specification</td>
<td>Usability Specification</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>neutral</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td>Requirements Validation</td>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>6</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
</tbody>
</table>

* These are not SE activity types, but usability activities included to offer developers a structured view of the 15 usability techniques that can be applied for requirements elicitation and analysis.
<table>
<thead>
<tr>
<th>Activity Type</th>
<th>Technique</th>
<th>U. P.</th>
<th>Training Needs</th>
<th>Applicability</th>
<th>Proximity to SE</th>
<th>Improvement/Effort</th>
<th>Representativeness</th>
<th>Overall Rating</th>
<th>Application Time</th>
<th>Basic Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Cognitive Walkthroughs</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Pluralistic Walkthroughs</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td>Design</td>
<td>Menu-Selection and Dialog Box Trees</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Interface State Transition Diagrams</td>
<td>no</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>2</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Product Style Guide</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>not usual</td>
<td>well-matched</td>
</tr>
<tr>
<td></td>
<td>Context Navigation Map</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Content Model of Interface</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>not usual</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Impact Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>5</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Organizing Help by Use Cases</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>6</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Cognitive Walkthroughs</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>neutral</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Pluralistic Walkthroughs</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>well-matched</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Thinking Aloud</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>5</td>
<td>very useful</td>
<td>not usual</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Post-Test Feedback</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>not usual</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Measured Performance</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>not usual</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>Laboratory Usability Testing</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>not usual</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>User Feedback</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>not usual</td>
<td>well-matched</td>
</tr>
<tr>
<td></td>
<td>Questionnaires, Interviews and Surveys</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>not usual</td>
<td>well-matched</td>
</tr>
<tr>
<td></td>
<td>Logging Actual Use</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>5</td>
<td>useful</td>
<td>not usual</td>
<td>well-matched</td>
</tr>
<tr>
<td>Technique</td>
<td>U. P.</td>
<td>Training Needs</td>
<td>Applicability</td>
<td>Proximity to SE</td>
<td>Improvement/Effort</td>
<td>Represen- tative- ness</td>
<td>Overall Rating</td>
<td>Activity Type</td>
<td>Basic Reference</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>----------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>---------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>Paper Prototypes</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3 very useful</td>
<td>Req. Elicitation and Analysis (Prototyping)</td>
<td>[Snyder, 03]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personas</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3 very useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Cooper, 03a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenarios and Storyboards</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3 very useful</td>
<td>Req. Elicitation and Analysis (Develop Product Concept)</td>
<td>[Nielsen, 93]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Competitive Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1 useful</td>
<td>Req. Elicitation and Analysis</td>
<td>[Beyer, 98]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affinity Diagrams</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Beyer, 98]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contextual Inquiry</td>
<td>yes</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Wixon, 96]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethnographic Observation</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>2 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Profiles</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>5 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Mayhew, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Role Map</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTA</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Amnett, 04]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Brainstorming</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>1 useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Preece, 94]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1 useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pluralistic Walkthrough</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4 useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Bias, 94]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card Sorting</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>3 very useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>1 very useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Robertson, 01]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability Specifications</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>4 very useful</td>
<td>Requirements Specification</td>
<td>[Nielsen, 94]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>4 very useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Hix, 93]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Menu-Selection and Dialog Box Trees</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1 very useful</td>
<td>Interaction Design</td>
<td>[Shneiderman, 98]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JEM</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1 useful</td>
<td>Interaction Design</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scenarios</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>3 useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Mayhew, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>6 useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Nielsen, 93]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cognitive Walkthrough</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4 useful</td>
<td>Interaction Design</td>
<td>[Lewis, 97]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface State Transition Diagrams</td>
<td>no</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>2 useful</td>
<td>Interaction Design</td>
<td>[Wasserman, 85]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context Navigation Map</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>1 useful</td>
<td>Design</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizing Help by Use Cases</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1 useful</td>
<td>Design</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Style Guide</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1 useful</td>
<td>Interaction Design</td>
<td>[Mayhew, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content Model of Interface</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1 useful</td>
<td>Interaction Design</td>
<td>[Mayhew, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card Sorting</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>3 very useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>1 very useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Constantine, 99]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stage in Iterative Cycle</td>
<td>Technique</td>
<td>U. P.</td>
<td>Training Needs</td>
<td>Applicability</td>
<td>Proximity to SE</td>
<td>Improvement/Effort</td>
<td>Representativeness</td>
<td>Overall Rating</td>
<td>Activity Type</td>
<td>Basic Reference</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------</td>
<td>------</td>
<td>----------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>Paper Prototypes</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Req. Elicitation and Analysis (Prototyping)</td>
<td>[Snyder, 03]</td>
</tr>
<tr>
<td></td>
<td>Usability Specifications</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>Requirements Specification</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td></td>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Nielsen, 94]</td>
</tr>
<tr>
<td></td>
<td>Menu-Selection and Dialog Box Trees</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>Interaction Design</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td></td>
<td>Thinking Aloud</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>5</td>
<td>very useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td></td>
<td>Competitive Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td></td>
<td>JEM</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>User Role Map</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>User Profiles</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>5</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td></td>
<td>Task Scenarios</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td></td>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>6</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td></td>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Cognitive Walkthrough</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Lewis, 97]</td>
</tr>
<tr>
<td></td>
<td>Pluralistic Walkthrough</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Bias, 94]</td>
</tr>
<tr>
<td></td>
<td>Interface State Transition Diagrams</td>
<td>no</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>2</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Wasserman, 85]</td>
</tr>
<tr>
<td></td>
<td>Context Navigation Map</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Impact Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td></td>
<td>Organizing Help by Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Design</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Measured Performance</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Dumas, 99]</td>
</tr>
<tr>
<td></td>
<td>Post-Test Feedback</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Laboratory Usability Testing</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Evaluation (Follower Studies of Installed Systems)</td>
<td>[Dumas, 99]</td>
</tr>
<tr>
<td></td>
<td>Questionnaires, Interviews and Surveys</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Evaluation (Follower Studies of Installed Systems)</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td></td>
<td>Logging Actual Use</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>5</td>
<td>useful</td>
<td>Evaluation (Follower Studies of Installed Systems)</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td></td>
<td>User Feedback</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Req. Elicitation and Analysis (Follow-up Studies of Installed Systems)</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td></td>
<td>Questionnaires, Interviews and Surveys</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Evaluation (Follower Studies of Installed Systems)</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td></td>
<td>Logging Actual Use</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>5</td>
<td>useful</td>
<td>Evaluation (Follower Studies of Installed Systems)</td>
<td>[Shneiderman, 98]</td>
</tr>
<tr>
<td></td>
<td>Card Sorting</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Robertson, 01]</td>
</tr>
<tr>
<td></td>
<td>Essential Use Cases</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Paper Prototypes</td>
<td>yes</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>3</td>
<td>very useful</td>
<td>Req. Elicitation and Analysis (Prototyping)</td>
<td>[Snyder, 03]</td>
</tr>
<tr>
<td></td>
<td>Usability Specifications</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>Requirements Specification</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td></td>
<td>Inspections</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>4</td>
<td>very useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Nielsen, 94]</td>
</tr>
<tr>
<td></td>
<td>Menu-Selection and Dialog Box Trees</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>1</td>
<td>very useful</td>
<td>Interaction Design</td>
<td>[Shneiderman, 98]</td>
</tr>
</tbody>
</table>

Annex 5
<table>
<thead>
<tr>
<th>Stage in Iterative Cycle</th>
<th>Technique</th>
<th>U. P.</th>
<th>Training Needs</th>
<th>Applicability</th>
<th>Proximity to SE</th>
<th>Improvement/Effort</th>
<th>Represen-tativeness</th>
<th>Overall Rating</th>
<th>Activity Type</th>
<th>Basic Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Thinking Aloud</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>5</td>
<td>very useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td></td>
<td>Competitive Analysis</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td></td>
<td>JEM</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>User Role Map</td>
<td>no</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (User Analysis)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>User Profiles</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>5</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td></td>
<td>Task Scenarios</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Req. Elicitation and Analysis (Task Analysis)</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td></td>
<td>Heuristic Evaluation</td>
<td>no</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>6</td>
<td>useful</td>
<td>Requirements Validation or Evaluation (Expert Reviews)</td>
<td>[Nielsen, 93]</td>
</tr>
<tr>
<td></td>
<td>Collaborative Usability Inspections</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Cognitive Walkthrough</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Lewis, 97]</td>
</tr>
<tr>
<td></td>
<td>Pluralistic Walkthrough</td>
<td>yes</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Interaction Design</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td></td>
<td>Interface State Transition Diagrams</td>
<td>no</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>2</td>
<td>useful</td>
<td>Design</td>
<td>[Wasserman, 85]</td>
</tr>
<tr>
<td></td>
<td>Product Style Guide</td>
<td>no</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Design</td>
<td>[Mayhew, 99]</td>
</tr>
<tr>
<td></td>
<td>Content Model of Interface</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Design</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Context Navigation Map</td>
<td>no</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Design</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Impact Analysis</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Design</td>
<td>[Hix, 93]</td>
</tr>
<tr>
<td></td>
<td>Organizing Help by Use Cases</td>
<td>no</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>1</td>
<td>useful</td>
<td>Design</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Measured Performance</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>3</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Dumas, 99]</td>
</tr>
<tr>
<td></td>
<td>Post-Test Feedback</td>
<td>yes</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>1</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Constantine, 99]</td>
</tr>
<tr>
<td></td>
<td>Laboratory Usability Testing</td>
<td>yes</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>4</td>
<td>useful</td>
<td>Evaluation (Usability Testing)</td>
<td>[Dumas, 99]</td>
</tr>
</tbody>
</table>
Description and Basic Reference for HCI Techniques
Designing Usable Software

How to Enrich Software Development with Usability Activities and Techniques

Xavier Ferré
xavier@fi.upm.es
Universidad Politécnica de Madrid (Spain)
# Table of Contents

INTRODUCTION............................................................................................................................... 1

I - REQUIREMENTS ELICITATION AND ANALYSIS............................................................................ 2
  I.1 AFFINITY DIAGRAMS .................................................................................................................... 2
  I.2 CARD SORTING .............................................................................................................................. 3
  I.3 COMPETITIVE ANALYSIS .............................................................................................................. 4
  I.4 CONTEXTUAL INQUIRY .................................................................................................................. 4
  I.5 ETHNOGRAPHIC OBSERVATION ..................................................................................................... 5
  I.6 JEM (JOINT ESSENTIAL MODELING) ............................................................................................. 5
  I.7 STRUCTURED USER ROLE MODEL ............................................................................................... 6
  I.8 OPERATIONAL MODELING ........................................................................................................... 7
  I.9 PERSONAS ...................................................................................................................................... 8
  I.10 ESSENTIAL USE CASES .............................................................................................................. 8
  I.11 GOMS (GOALS, OPERATIONS, METHODS AND SELECTION RULES) ........................................ 10
  I.12 SCENARIOS & STORYBOARDS .................................................................................................. 11
  I.13 VISUAL BRAINSTORMING ......................................................................................................... 12
  I.14 PAPER PROTOTYPES ................................................................................................................ 13
  I.15 WIZARD OF OZ TECHNIQUE ..................................................................................................... 14

II - REQUIREMENTS SPECIFICATION ............................................................................................ 15
  II.1 USABILITY SPECIFICATIONS ..................................................................................................... 15

III - REQUIREMENTS VALIDATION .................................................................................................. 18

IV - DESIGN........................................................................................................................................... 19
  IV.1 INTERACTION DESIGN GUIDELINES .......................................................................................... 19
  IV.2 CONTEXT NAVIGATION MAPS .................................................................................................. 20
  IV.3 IMPACT ANALYSIS .................................................................................................................... 21
  IV.4 MENU-SELECTION AND DIALOG BOX TREES ........................................................................... 22
  IV.5 ORGANIZING HELP BY USE CASES .......................................................................................... 23
  IV.6 PRODUCT STYLE GUIDE ........................................................................................................... 24

V - EVALUATION.................................................................................................................................. 27
  V.1 COGNITIVE WALKTHROUGH ...................................................................................................... 27
  V.2 HEURISTIC EVALUATION .......................................................................................................... 28
  V.3 PLURALISTIC WALKTHROUGH .................................................................................................. 28
  V.4 USABILITY INSPECTIONS .......................................................................................................... 29
V.5 THINKING ALOUD PROTOCOL ........................................................................................................... 30
V.6 PERFORMANCE MEASUREMENT / LABORATORY USABILITY TESTING .............................................. 31
V.7 QUESTIONNAIRES AND SURVEYS ...................................................................................................... 32
V.8 INTERVIEWS ....................................................................................................................................... 34
V.9 DIRECT OBSERVATION AND VIDEO/AUDIO RECORDING ............................................................. 35
V.10 FOCUS GROUPS .................................................................................................................................. 35
V.11 LOGGING ACTUAL USE .................................................................................................................. 36
V.12 USER FEEDBACK FACILITIES ......................................................................................................... 37
BIBLIOGRAPHY ........................................................................................................................................... 38
Introduction

This document presents usability techniques that may be applied in any iterative development process, organized according to the kind of activity where they fit. For each usability technique a brief description is provided and a basic reference where additional information may be obtained.
Requirements elicitation and analysis include the specification of the context of use. Its aim is to understand and record the implications of the context of use so that they can be considered during system design. There is no single best interaction style or approach suitable for just any user. Specific design alternatives that optimize the performance of some types of users may actually degrade performance of other types of users. Therefore, it is very important to know the particular users that the system is built for, and the kinds of tasks they want to perform using the system, and this is the aim of the specification of the context of use. Context of use is a broad term that comprises different interrelated aspects:

- The characteristics of the intended users. The identification of these characteristics is known as user analysis.
- The tasks the users are to perform. Task analysis deals with this issue.
- The environment in which the users are to use the system, including the hardware, software and materials to be used.

The development of the product concept is also part of requirements elicitation and analysis. The idea of a product concept is based on mental models: When the product concept is vague, ambiguous, inconsistent or obscure, there will be a divergence between the user mental model of the system and the design model that developers work with. Specifying a clear product concept ensures a proper communication between the members of the development team, and the production of a design that clearly conveys the model to the user.

Finally, prototypes allow designers to communicate more effectively with users and they reduce the need and cost of reworking that can occur when products need to be revised later in the life cycle. They are employed as part of requirements elicitation and analysis activities as well.

## I.1 Affinity Diagrams

Affinity diagrams serve for grouping and understanding information.

In order to conduct an affinity diagram session, begin by handing out Post-It notes. Then, ask participants to write one issue on each note.

You can give participants some minutes for this activity, but ask them to stop when a large majority of participants have stopped. Get all participants to gather at a vertical surface suitable for Post-It notes. Windows are appropriate. Encourage participants to place notes, one at a time, on the surface. As each note is placed, other participants may add similar notes in close proximity. Depending on the amount of time, the information being analyzed, and on group dynamics (and patience) it may be worth spending some additional time considering and rearranging the groups. When all notes have been placed and grouped, you can optionally name each group.

If there are more than 8 people, gathering around a common area may not be convenient. In this case, you can handle all the note-placing yourself-get one note from each person in turn; all participants can then
pass you any similar or related notes. This is not as satisfactory as having the group work together, since it is difficult to keep everyone focused on the task.

Affinity diagramming is best used if the work can be followed up quickly. For example, affinity diagramming of issues can lead into discussion of methods to address the issues. Nevertheless, the resultant groupings are arbitrary, so they should be used with a flexible approach (i.e. they are not the absolute truth).

Advice on conducting an affinity diagram session:
- You should allow all participants to contribute. There may be an individual who wants to take control of positioning and moving the notes. Do not allow this to happen. Do not move someone's note without their agreement. Discussion will often indicate that the participant wanted to articulate a different issue.
- Encourage participants to read their notes aloud while placing them on the surface.
- Do not allow the activity to continue past the point of tiredness or boredom, because it is a tiring effort. Avoid having more than two consecutive affinity diagramming sessions during a workshop.


1.2 Card Sorting

At the beginning of any exercise of information architecture design, it is normal to be confronted by a very long list of potential subjects to include. The challenge is to organize this information in a way that is useful and meaningful for the users of the system. While careful investigation and analysis of the information may reveal some clues, making relevant users organize the subjects may be more cost-effective. Therefore, card sorting consists on asking users to categorize a list of terms. It is useful when such a list already exists.

Card sorting is very similar to affinity diagrams, but it is aimed to the design of an information structure (e.g., a website, or a menu tree).

When creating the list of topics some issues should be considered:

- The length of the list should be manageable
- Existing structures should not be reflected
- Do not put “clues” that will lead users to arrange the topics in a particular way
- Each topic should not be too general nor too specific
- The topics chosen must be meaningful to the participants in the session

First you should create the cards, and then select the participants. They should be the actual end-users of the system you are building. If your participant is the manager of the user instead of the end user himself or herself, the results will not reflect a mental model which is natural to the end user. Each session must be dedicated to a homogeneous group of users. If several different groups of users must be tackled, hold a different card sorting session for each one.

In the sorting session you should carefully explain the mechanics and the expected outcome of the session to participants. Then, you should encourage participants to organize the cards in a way that makes sense to them. When they are finished with the grouping, ask them to label each category with a card. You may allow a “to be determined” or “not sure” pile to be created, but do not tell them so at the beginning of the exercise.
Use the notebook to take notes about any important event arising during the session:

- User questions,
- user comments or suggestions, and
- user non-verbal behaviors.

Bear in mind that a card sorting exercise does not produce a finished information design, it is just an elaborate guidance, which is very useful because it has been built according to the user expectations.


**I.3 Competitive Analysis**

Competitive analysis basically involves studying existing products to find out their strengths and weaknesses. These analyzed products may be competing products, but they may also be products from different fields that address issues that are similar to the ones the system will have to deal with. Commercial products that are widely known serve as good references for establishing the product concept, and a competitive analysis of their benefits from a usability point of view can help to focus the discussion and the decision-making process.

If several competing products are available for analysis, we can perform a comparative analysis of their differing approaches to support the user goals. This will provide ideas for the system we are developing, especially for developing the product concept. It can also provide a list of ad hoc guidelines for approaches to specific issues that seem to work, and things that should be avoided.


**I.4 Contextual Inquiry**

In a contextual inquiry interview, the developer sits beside the user and observes how he or she performs the work, interrupting him or her every now and then to ask why a particular action has been taken or what its purpose is. The interview should be performed in an environment that is as close as possible to the usual working environment of the user, because the developer is looking for first-hand knowledge, the kind of understanding about the work structure that the user cannot formulate, unless he or she is performing the work at that very time.

The interview should be a combination of watching and probing.

A sense of partnership should be formed between observer and user, in the sense that they are both looking to explain the internal logic behind the user’s actions, as there can be a lot of tasks that the user does routinely and he or she cannot completely explain. Therefore, the observer must try to make out the work structure and find patterns and distinctions in the way people organize work. Not only the observer gains a better understanding of the user’s work, the user himself or herself also acquires increased insight into his or her work by being forced to look at it from an external perspective. Users themselves are sometimes surprised about some of the actions they perform routinely when they look at them from an analytical point of view.

A good approach to contextual inquiry is the master / apprentice model, where the user is the master, and the observer is the apprentice that wants to learn how the work is done. This model of action for the observer aims to make the user focus on his or her work. Nevertheless, the master / apprentice model
should not be taken literally, since the observer must play an active role in probing the user every time he or she needs an explanation.

Note that competitive analysis is a useful technique for vendor organizations, while observational techniques are more suited for bespoke systems (nevertheless they may be useful as well for vendor organizations when a particular kind of user is targeted).


### I.5 Ethnographic Observation

Observational techniques are based on performing elicitation at the customer organization, at the users’ workplace, observing how they work. The work may be performed either manually, using a competitor product or using a previous version of the software product under development.

Ethnography is a method belonging to Anthropology for studying a particular tribe or culture. The ethnographer participates, overtly or covertly, in people’s daily lives for an extended period of time. When this approach is used for software development purposes, it provides developers information about the context where the user performs his or her tasks, which would be very difficult to apprehend otherwise (for example, by means of interviews). The difference between developers acting as ethnographers and anthropologists is that, apart from trying to understand the user, developers observe the usage of existing software products for the purpose of changing and improving those products. Additionally, the available time for ethnographic observation in software development is a lot less than the time anthropologists spend immersed in a culture.


### I.6 JEM (Joint Essential Modeling)

JEM is a structured, facilitated, collaborative process for concurrent modeling of users and use cases. It is based on the JAD (Joint Application Development) technique.

In JEM, users and developers join in a collaborative effort to define the essential models and reach agreement on core requirements. The objective is to reach consensus on the tasks to be supported by the system under development. Although other models play some part, the principal medium of exchange are use cases, whether essential or detailed.

The preparation and consolidation process prepares materials and an agenda for the subsequent sessions and also generates a candidate list of user roles used as a guide for participation in later joint modeling. Following the development of role and task models, these models are audited for completeness, correctness, and consistency. To complete the process, use cases are prioritized and allocated to project iterations. These activities are carried out in a series of sessions as follows:

- **Framing session**: The purpose of this session is to establish the framework within which the joint modeling sessions will operate.
- **Modeling sessions**: User role and use case models are developed collaboratively during these sessions.
• Review session: First the group reviews the models to ensure that they are complete, correct, and consistent. Second, use cases are sorted to identify which capabilities are to be supported and when.

The main roles in JEM are the users, the lead analyst (who ensures appropriate technique leadership and expertise in modeling), the facilitator, and the scribe. Other potential participants include a sponsor, and other members of the development team.


I.7 Structured User Role Model

The structured user role model is formed by profiles. The main profiles are as follows:

• Incumbents: Common characteristics that users who play a given role share. There are three categories into which the elements in this profile may fall: domain knowledge, system knowledge and other background knowledge.

• Proficiency: How usage proficiency is distributed over time and among users in a given role.

• Interaction: Patterns of usage associated with a given role. The kind of information in this profile falls in one or more of the following categories: frequency, regularity, continuity, concentration, intensity, complexity, predictability or locus of control.

• Information: Nature of the information manipulated by users in a role or exchanged between users and the system. The information in this profile may offer details on the input origins, the flow direction, the information volume and/or the information complexity.

• Usability criteria: Relative importance of specific usability attributes with respect to a given role.

• Functional support: Specific functions, features, or facilities needed to support users in a given role.

Some roles are highlighted as focal roles, which are the ones that are judged to be the most common or typical or that are deemed particularly important from a business perspective or from the standpoint of risk.

Any elicitation technique can be used to acquire the information about users to complete the profiles in the structured user role model. The more complex the user population for a system under development is, the more complete the profiles for a user role must be. For simple systems just some data on a few of the profiles may be enough to describe each user role, and then the user role model is not said to be structured.
This example of a user role map shows the user roles identified for a statistical analysis package. We say that there is affinity between two roles, which is represented by a dashed line, if we identify some similarity or resemblance between them (InformalResearcher and CasualDataMiner have an affinity relationship). When a user role is a subtype of another, we say that the former specializes the latter, and this is represented by a double-lined arrow that goes from the more specific role to the more abstract one (RegularSalesStaff and TempSalesStaff are both subtypes of the general SalesStaff). Finally, there is a composition relationship when one role combines the characteristics or features of two or more other roles and is composed of these other roles, which we represent by a single-lined arrow (WarehouseManager includes both the OrderExpediter and the SystemMaintainer).


### I.8 Operational Modeling

Operational modeling is a collection of various operational and contextual influences that can play a role in usability:

- **Operational Risk Profile**: Operational risk refers to what is at stake if the user and the system fail to correctly complete tasks. For example, what are the consequences of an input error, a failure to complete a transaction, a system lockup, or a delay in processing. Where operational risk is higher in connection with particular roles or use cases, special attention needs to be paid to mechanisms that assure input accuracy and accurate interpretation of output.

- **Device Constraints Profile**: The device constraints profile identifies equipment characteristics associated with specific roles, use cases, or the system as a whole. There may be limitations on the input side, the output side, or both. These constraints include screen size, resolution, and color depth; keyboard or keypad size and layout; and special controls such as sliders, toggle switches, rotary knobs, or similar items. It is especially interesting to describe the device constraints in projects where the devices are fixed by economics or the user community.

- **Environment Profile**: The environment profile is formed of physical factors, such as the type of user location (office, home, factory, etc.), the level of ambient noise, the lighting conditions, temperature, humidity, or the presence of vibration. The information gathered may reflect any
kind of physical condition of the environment that may affect system usage. A key issue here is the level of distraction due to the physical environment. Distractions may be physical (like a noisy fan or repeated phone calls) or mental (like trying to remember to do something that must be temporarily postponed).


I.9 Personas

A persona is a fictional person we create to represent a particular class of real users or system participants. A persona is an archetypal user, who resembles real people (the people interviewed when user research is performed, for example), but does not exactly match any one of them. This technique helps to determine what the product should do and how it should behave, to communicate with non-technical stakeholders because the natural language description of each persona is understandable by anyone. As it helps to build a common language, it helps reaching consensus in the team. Finally, personas may help in marketing and sales plans efforts.

There are three potential problems in interaction design that strongly affect usability: The elastic user, self-referential design and the design for edge cases. The three of them are tackled when personas is used. Each persona has particular goals and behavior patterns.


I.10 Essential Use Cases

A use case is a case of use, or one kind of use to which a system can be put. It is:

- Supplied functionality
- An external, “black-box” view
- A narrative description
- Interaction between a user and a system
- A use of the system that is completely meaningful to the user

In object-oriented analysis and design practice, the use-case model is very important in cycle planning, but once the cycle starts, use cases are regarded as a preliminary version of elements of the internal functionality design. When design elements are labeled as use-case realizations, we are shifting use cases to the design world and, therefore, away from the user realm, losing most user-centered advantages with that shift.

Each use case describes, in narrative form, an interaction that is complete, well defined, and meaningful to some users. The narrative of the use case is divided into two parts: the user action model, which shows the actions the user takes; and the system response model, which shows what the system does in response. Depending on the level of abstraction at which the use case is described, there are two forms for use cases: essential and detailed or concrete. Essential use cases describe a generalized, abstract, technology-free and implementation-independent interaction, in the language of the application domain and of users. On the other hand, detailed use cases reflect the actual interaction as it happens between the user and the system, so they include restrictions imposed by internal design decisions and according to a particular
user interface design. Detailed use cases can also be called concrete, since they reflect the concrete instantiation of the abstract description in its essential form. Detailed use cases are the ones usually employed in object-oriented software development, even if some recent software engineering reference books acknowledge the advantages of essential use case modeling.

Detailed use cases are useful as a technique for improving the usability of the software product, but not as the first approach to describing the interaction between the users and the system. Below we will describe essential use cases as an appropriate initial approach to interaction modeling, which will serve as basis for designing the best scheme to support interaction (using some of the other usability techniques described in this catalogue).

An example of a use case in both forms follows:

<table>
<thead>
<tr>
<th>ESSENTIAL USE CASE</th>
<th>DETAILED USE CASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>gettingCash</td>
<td>gettingCash</td>
</tr>
<tr>
<td>USER INTENTION</td>
<td>SYSTEM RESPONSIBILITY</td>
</tr>
<tr>
<td>identify self</td>
<td>insert card</td>
</tr>
<tr>
<td></td>
<td>verify identity</td>
</tr>
<tr>
<td></td>
<td>offer choices</td>
</tr>
<tr>
<td>choose</td>
<td>enter PIN</td>
</tr>
<tr>
<td>dispense cash</td>
<td>verify PIN</td>
</tr>
<tr>
<td>take cash</td>
<td>display transaction option menu</td>
</tr>
<tr>
<td></td>
<td>display account menu</td>
</tr>
<tr>
<td></td>
<td>prompt for amount</td>
</tr>
<tr>
<td></td>
<td>display amount</td>
</tr>
<tr>
<td></td>
<td>return card</td>
</tr>
<tr>
<td></td>
<td>dispense cash</td>
</tr>
</tbody>
</table>

An essential use case is based on the purpose or intentions of a user rather than on the concrete steps or mechanisms by which that purpose or intention might be carried out.

The first step for the creation of the use case model is to identify the use cases that the system must support. The structured user role model is a starting point for use case identification. For each user role we can ask ourselves what these kinds of users are trying to accomplish, what they need to do in order to fulfill the role, or what capabilities are required to support whatever these users need to accomplish. Once
the use cases have been identified, narratives are written for each use case, and the relationships between them are defined.

To write a use case narrative, we must identify the essential purpose or user intent embodied in the interaction. The name of the essential use case should be simple, it should imply purposeful, goal-directed action. Transitive gerunds, verbs of continuing action with a direct object, make good names for essential use cases. Examples of essential use case names are: findingCustomer, verifyingOrder or insertingMathSymbol. If the user purpose is not well expressed or fully implied by the name of the use case, then an explicit purpose clause should be added to the head of the narrative, describing and detailing the purpose or goal from the user perspective.

The use case diagram or use case map represents the use cases supported by the whole system, and the interrelationships among them and with the users. The use case map along with the narratives of use cases form the use case model.


1.11 GOMS (Goals, Operations, Methods and Selection Rules)

Cognitive modeling aims to produce a computational model for how people perform tasks and solve problems, based on psychological principles. These models may be outlines of tasks written on paper or computer programs which enable us to predict the time it takes for people to perform tasks, the kinds of errors they make, the decisions they make, or what buttons and menu items they choose. Such models can be used to determine ways of improving the user interface so that a person's task has fewer errors or takes less time. They may be also used for deciding between two alternative designs, or between two systems for a purchase.

There are basically four different GOMS models: The Keystroke-Level Model (KLM), CMN-GOMS, NGOMSL, and CPM-GOMS.

The GOMS model is based on goals (edit document) and subgoals (change a word) that the user formulates; the operators available to users, like motor, perceptual or cognitive primitives (click the mouse, look at the menubar); the methods users compose out of sequences of these operators to achieve the goals or subgoals (selection is done by moving the cursor to point to the word and double-clicking the mouse); and the selection rules needed to decide what to do next if the user has several goals pending or if there are several methods that will accomplish a given goal (the word can be removed by selecting it and issuing a “cut” command or by backspacing over it).

GOMS can be used both quantitatively and qualitatively. Quantitatively, it gives good predictions of performance time and learning. Qualitatively, GOMS can be used to design training programs and help systems. The GOMS model is a careful description of the knowledge needed to perform a given task and thus it describes the content of task-oriented documentation. You only need to tell the new user what the goals are, what different methods could be used to achieve them, and when to use each method (selection rules). This approach has been shown to be an efficient way to organize help systems, tutorials, and training programs as well as user documentation.
Its main weakness is that GOMS assumes that the users are expert users (errorless performance is assumed). If the system is being developed for novice or intermediate users GOMS cannot be used.

In NGOMSL, learning time and execution time are predicted based on a program-like representation of the procedures that the user must learn and execute to perform tasks with the system. NGOMSL stands for Natural GOMS Language, because the notation used is a natural language structure to represent the user methods and selection rules.

NGOMSL starts after an initial task analysis has been performed, that is, after the user goals have been identified. The methods must be defined for each goal, by asking the question “how do you do it on this system?” Each method is described as a series of steps. If all the operators in a method are primitive, this is the final level of analysis. However, if some operators are high-level, they must be examined to decide whether a method of analysis is needed. Then we can calculate a time estimate for each goal.

The NGOMSL model below describes how to move an object in the Macintosh Finder tool:

```
Method for goal: move an object.
   Step 1. Accomplish goal: drag object to destination.
   Step 2. Return with goal accomplished.
```

There is a submethod for describing the dragging operation:

```
Method for goal: drag item to destination.
   Step 1. Locate icon for item on screen.
   Step 2. Move cursor to item icon location.
   Step 3. Hold mouse button down.
   Step 4. Locate destination icon on screen.
   Step 5. Move cursor to destination icon.
   Step 6. Verify that destination icon is reverse-video.
   Step 7. Release mouse button.
   Step 8. Return with goal accomplished.
```


**I.12 Scenarios & Storyboards**

A scenario details an interaction example illustrating the flow of specific user actions needed to get some result, concentrating on what the user will see, what the user must know, and what the user can do.

Scenarios are useful where there is no available data about the range and distribution of user task frequencies and sequences, especially for highly innovative systems. In these less well-defined projects, developers find day-in-the-life scenarios helpful to characterize what happens when users perform typical tasks. Scenarios can represent common or emergency situations, with both novice and expert users, and they are especially suited when multiple users must cooperate or multiple physical devices are used.

Storyboards (pictorial representations of scenarios, like the ones used by film directors) may provide additional support to the situations described in scenarios.
Scenarios may serve to convey a shared understanding of the product concept and the kind of users and tasks for which it is intended. It may be used as well to show to the customer what could be offered or provided if the system is actually developed.

An example of scenario for the Eurochange system (a system for currency exchange) follows:

Path Smith has just arrived at Geneva International Airport en route to a large conference on Human-Computer Interaction. Pat is carrying a laptop and a large, heavy suitcase and needs to get to the conference center quickly. Looking around for a bank in order to get some local currency, Pat sees the Eurochange machine with its blue flag style logo showing a circle of twelve stars.

Pat goes up the machine. It seems similar to the automatic teller machine that Pat uses regularly. Pat puts down the suitcase, takes out a credit card and inserts it into the slot. A message is displayed on the screen:

Enter your PIN

Pat thinks for a few moments and then types a four-digit number on the numerical pad, listening to the reassuring beep that follows each number pressed. The machine pauses for a few seconds and then displays:

Select currency required

Pat pauses again. What is the currency in Switzerland? Pat browses the currencies available, sees “Swiss Franc (CHF)” and presses the key. The machine displays the message:

Exchange rate is 1.47 CHF to 1 EUR

Enter amount required in Swiss Francs in units of [10]

Press <Proceed>

Pat types 253 and presses <Proceed>. A message is displayed:

Machine deals in bank notes only

Smallest bank note is [10] CHF

Enter new amount to obtain CHF or press <Cancel>

Pat enters 260 and presses <Proceed>. There is a whirring noise and a few other indeterminate clunks and clicks. The credit card is returned from the card entry slot and the money deposited in the delivery slot, with a printout of the transaction.

Storyboards (pictorial representations of scenarios, like the ones used by film directors) may provide additional support to the situations described in scenarios.


I.13 Visual brainstorming

Brainstorming brings together a set of design and task experts to inspire each other in the creative, idea generation phase of the problem-solving process. It is used to generate new ideas by freeing the mind to accept any idea that is suggested, thus allowing freedom for creativity. The results of a brainstorming session are, it is hoped, a set of good ideas and a general feel for the solution area.
With visual brainstorming we try to generate visual ideas of abstract concepts. Visual ideas are a basis for discussion in a brainstorming session.

Paper prototypes may be used for exploring all kinds of design ideas, and they can help the development team to think about the organizing metaphor for a system. Visual brainstorming involves using exploratory paper prototypes as a means for facilitating communication in brainstorming sessions.

One of the first things you learn in design is to put forward a number of alternatives so that you can then compare them. Having a lot of display space is important for doing this because you can then make design ideas visual. One of the things you can do with visual things is superimpose them, or put them side by side and quite often when you start doing that you like one better than another. Until you have made a comparison, you have no idea why you prefer one to another. The criteria emerge from the comparison. It is not just picking the right idea, but recognizing the right idea in all the mess of different alternatives produced.

Evaluation also comes into brainstorming: when you stop generating ideas you have to start evaluating them. The best exploratory designs produced in the visual brainstorming can then be further developed by constructing more elaborated prototypes, which can be evaluated with users.


### 1.14 Paper Prototypes

Prototypes allow designers to communicate more effectively with users and they reduce the need and cost of reworking that can occur when products need to be revised later in the life cycle. We need to build prototypes because abstract technical specifications and models are not a good way of communicating when we want to involve users in the design process.

Prototyping, and especially rapid prototyping, is closely related to iterative design. For prototypes to be effective, they should be built at a minimal cost in terms of resources and time. The difference from traditional software engineering system prototypes is again a difference of focus. Prototypes are useful for usability purposes when they depict mostly system-user interaction, so that they convey how the system will work from the user point of view. So, prototypes can be used to try out design ideas with users and to gather their feedback.

From the product concept to full-detail design, prototypes of varying fidelity to the final system can be produced for use in usability evaluation activities. Therefore, the prototyping approach to interactive system development involves the production of at least one early version of the system that illustrates essential features of the later operational system. When used early in the development process, a prototype encourages user participation and involvement and allows developers to observe user behavior and reaction to the prototype. They may also serve for eliciting information from users: About the necessary functionality of the system, operation sequences, user support needs, required representations, and about the look and feel of the user interface.

HCI offers to software development a kind of cheap and quick prototypes, which are the less elaborate ones: Paper and chauffeured prototypes, and the Wizard of Oz technique.

Paper prototypes can be presented to users and members of the development team for comment and improvement. At the early stages of development, it may be undesirable to express design ideas by means
of working software, since this may take a significant effort to build. Once there is a glossy version of an idea, it is too easy to get carried away thinking that this must be the only or the best solution to the problem.

Chauffeured prototyping involves the user watching while a member of the development team “drives” the system. It is a way to test whether the interface meets the user needs without the user actually having to carry out low-level actions with the system.


1.15 Wizard of Oz Technique

Wizard of Oz prototyping involves having some kind of behind-the-scenes manipulation to produce the responses of a working system, usually by means of a person providing the responses.

In a typical Wizard of Oz prototyping setting, the user interacts with a screen, but instead of a piece of software responding to the user requests, a developer is sitting at another computer (which is connected to the user computer through the network), answering the queries and responding to the real user. The user is unaware of the trick, so the perception of using a real working system is not spoiled. This kind of prototyping is widely used to prototype and test out user interface designs of many kinds, but especially for exotic or unusual configurations.

For example, the development team may want to try out a telephone-based interface that mixes limited voice recognition with telephone keypad responses. The behavior of the system is simulated by a person at the other end of the telephone line.

It is specially suited for systems including subsystems with some intelligence, like agents or advisors, or that include natural-language processing. That is, processing that can be easily done by a human, but requiring extensive programming efforts for it to be done by a computer.

II.1 Usability Specifications

Usability specifications are quantitative usability goals, which are used as a guide for ascertaining when a system has the proper usability level. They can be compared to non-functional requirements. Their concerns are user satisfaction and performance. Performance is interpreted as establishing the required performance of the new system formed by both the user and the software system, working together towards the achievement of certain goals. In this sense, usability specifications can also be called usability benchmarks.

The knowledge gathered in the specification of the context of use activity is the input for this activity. The usability specifications are defined according to the characteristics of the target user population and the goals and tasks identified in task analysis.

The set of usability specifications represent the system acceptance criteria from a usability point of view. Usability specifications are monitored at the end of each development cycle, establishing how much progress has been made towards the usability objective. They serve as criteria for determining when to stop iterating. While usability attributes are not directly measurable, usability specifications need to be.

Several authors prefer the term usability goal, as these specifications are established as a goal to be achieved in the system design. Usability goals drive design as information shared by the whole development team, which can provide decision criteria when different design alternatives are considered, not just as a test case that will be checked in the evaluation phase.

By establishing usability specifications early in the development process, and monitoring them at each iteration, you can determine whether your system is, indeed, moving towards an improved, more usable, result.

<table>
<thead>
<tr>
<th>Usability Attribute</th>
<th>Measuring Instrument</th>
<th>Value to be measured</th>
<th>Current Level</th>
<th>Worst Acceptable Level</th>
<th>Planned Target Level</th>
<th>Best Possible Level</th>
<th>Observed Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance in normal use</td>
<td>“Answer Request” task</td>
<td>Length of time taken to successfully perform the task (minutes and seconds)</td>
<td>2’ 53”</td>
<td>2’ 53”</td>
<td>1’ 30”</td>
<td>50”</td>
<td></td>
</tr>
<tr>
<td>Performance in normal use</td>
<td>“Answer Request” task</td>
<td>Number of errors during task performance</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>First Impression</td>
<td>Questionnaire</td>
<td>Average score (range –2 to 2)</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
The first step is to identify the usability attributes or sub-attributes that we want to cater for. Depending on the kind of product developed, some attributes might be irrelevant, for example, efficiency could be a secondary goal for a walk-and-use kiosk, while learnability would have top priority.

Attributes like satisfaction or first impression may be evaluated by means of subjective measures, normally in the form of questionnaires.

For performance-related attributes, a set of benchmark tasks (use cases) must be selected and associated with each attribute. A benchmark task is a typical, representative use case a user will perform. Measuring a user’s performance on a benchmark task provides an objective usability metric for the related usability attribute. Benchmark tasks should be as specific as possible, so that there is little variability in their enactment by different users. The benchmark task in the first row could be described as follows: “Suppose you are at the Help Desk counter, and you receive a request. You decide to answer the request, and you look for the answer in your knowledge base ...”.

The value to be measured must be then decided. For a benchmark the main values we can collect are the time to complete a task, or the number of errors during task performance. It is usually sensible to measure both of them, and we would have two rows in the Usability Specification table that are based on the same task.

The last step in defining usability specifications involves establishing the range of levels:
- The current level may refer to the value for the usability attribute in question with the current version of the system or with a competitor we want to challenge with our more usable product. When we are automating a manual procedure, it may refer to the time required to manually perform the task. If the system is very innovative, this field could be left blank.
- The worst acceptable level is the lowest acceptable level of user performance. It means that if the system does not reach this minimum level for any of the attributes in the specifications, the system is unacceptable from a usability point of view. The value for the current version of the product is usually taken as a reference to establish this level, and the level will be higher if the current version is unsatisfactory.
- The best possible level is a realistic upper limit. It should be an attainable level, not a wild dream. A hypothetical expert user should be able to attain this level. You can use developers as users to establish the best possible level, since they are the ones who are better acquainted with the subtleties of the interaction design. Another possibility is to use GOMS to provide theoretical estimates of expert error-free task performance.
- The planned target level is the attainment of unquestioned usability success. It is the most important value, because it is the actual requirement equivalent to traditional requirements. The other values must be filled in beforehand to help to set the planned target level in a sensible range of values. If there is a competitive system with a high usability level, it may serve as a reference for setting the planned target level.

When a system prototype exists, usability testing may be used to establish these levels at reasonable values.

Much expertise in the issue is required to establish good levels for the usability specification table, and this is why a usability specialist might be needed to apply this technique the first times it is used.
The observed results column will be filled in when the specifications are tested by means of usability testing and questionnaires.

As usability practice involves a combination of user participation and iterative development, it includes techniques for validating preliminary models. Any kind of expert review could be employed to validate early prototypes, or models of user-system interaction.

Expert review techniques (Cognitive Walkthrough, Heuristic Evaluation, Pluralistic Walkthrough and Usability Inspections) are detailed in part V: Usability Evaluation.
Interaction design tackles the definition of the interaction environments and their behavior, including the design of the visual elements that form the graphical UI (User Interface).

Interaction design should be made to accommodate the definition of the interaction between the system and the environment that has been produced as part of requirements analysis activities. And the internal structure of the system should be designed to provide a good implementation of this interaction with the environment.

Before detailing each design technique some interaction design guidelines are presented.

IV.1 Interaction Design Guidelines

Each HCI expert has his or her own set of guidelines. In the table below are detailed Jakob Nielsen’s heuristics [Nielsen, 93].

<table>
<thead>
<tr>
<th>Nielsen’s Usability Heuristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Use simple and natural dialogue</td>
</tr>
<tr>
<td>• Speak the users' language</td>
</tr>
<tr>
<td>• Minimize user memory load</td>
</tr>
<tr>
<td>• Be consistent</td>
</tr>
<tr>
<td>• Provide feedback</td>
</tr>
<tr>
<td>• Provide clearly marked exits</td>
</tr>
<tr>
<td>• Provide shortcuts</td>
</tr>
<tr>
<td>• Provide good error messages</td>
</tr>
<tr>
<td>• Prevent errors</td>
</tr>
</tbody>
</table>

Another well respected HCI expert, Ben Shneiderman, presents three main principles [Shneiderman, 98], of which one of them relates to eight rules for interface design, as follows.

- Principle 1: **Recognize diversity**. Before beginning a design we must make the characterization of the users and the situation as precise and complete as possible.

  
  • **Strive for consistency**. Consistent sequence of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent colour, layout, capitalization, fonts and so should be employed throughout.
  
  • **Enable frequent users to use shortcuts**. Abbreviations, special keys, hidden commands and macro facilities
- Offer informative feedback. For every user action, there should be system feedback.
- Design dialogs to yield closure. Sequences of actions should be organized into groups with a beginning, middle, and end. The aim is to provide the user with a sense of accomplishment.
- Offer error prevention and simple error handling. As much as possible, design the system such that users cannot make a serious error. Erroneous actions should leave the system unchanged, or the system should give instructions about restoring the state.
- Permit easy reversal of actions. The units of reversibility may be a single action, a data-entry mask, or a complete group of actions.
- Support internal locus of control. Users should feel they are in control of the system.
- Reduce short-term memory load. Displays should be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequence of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided.

**Principle 3: Prevent Errors.**
- Correct matching pairs. This rule is directed to avoid syntactical errors when writing in a programming language or similar. It can be also applicable to word processors.
- Complete sequences. Group different steps which are often used jointly. Example: The sequence of dialling up, setting communication parameters, logging on, and loading files. Another example: In a word processor, setting the alignment, the font, the font size and the letters in uppercase for section titles; it should not be necessary to set those characteristics every time a section title is entered.
- Correct commands. Automatic command completion for command languages, and limiting the possible choices to permissible commands in a more developed interface.

**IV.2 Context Navigation Maps**

A context navigation map models the interconnections among the various interaction spaces of the user interface. It gives more dynamic expressiveness than tree menus, by specifying the transition between the different interaction contexts that occurs when a use case is enacted. A context navigation map (or navigation map for short) represents the structure of the user interface by modeling the relationships among interaction contexts.

The navigation map models the way users can navigate through the various interaction contexts within the user interface in the course of enacting use cases. When a single use case is represented, the navigation map models the behavioral view, and this is the most usual application of navigation maps. When the map combines all the behavioral views for the various use cases of the system in a single diagram, the result is called an architectural view. For big systems with a lot of interaction spaces the architectural view may get unwieldy, and too many transitions can lead to spaghetti-like diagrams.

Next figure details the notation for navigation maps.
The navigation map models the way users can navigate through the various interaction contexts within the user interface in the course of enacting use cases. When a single use case is represented, the navigation map models the behavioural view, and this is the most usual application of navigation maps. An example of a behavioural view is shown in the next figure:

When the map combines all the behavioural views for the various use cases of the system in a single diagram, the result is called an architectural view. For big systems with a lot of interaction spaces the architectural view may get unwieldy, and too many transitions can lead to spaghetti-like diagrams.


IV.3 Impact Analysis
Impact or cost/importance analysis is a technique for deciding between design options, by relating the options to the usability problems that are affected and choosing the ones that address the most important usability problems. It is performed once we have a set of usability problems identified in any kind of
usability evaluation activity. For each usability problem, we can propose a design decision, and use impact analysis to prioritize these decisions in order to undertake the redesign effort. It is a tool for making the trade-offs necessary in any design process, which are based, in this case, on the usability issues that are addressed.

In an impact analysis, the development team considers the relative importance of the usability problems found, and the cost of the solutions as listed in a table like the following one:

<table>
<thead>
<tr>
<th>Usability Problem</th>
<th>Effect on User Performance</th>
<th>Importance</th>
<th>Solution(s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Too much window manipulation</td>
<td>10 of 35 minutes</td>
<td>High</td>
<td>Fix window placement automatically, but allow user to reposition it</td>
<td>6 hours</td>
</tr>
<tr>
<td>Black arrow on black background</td>
<td>N/A</td>
<td>Low</td>
<td>Reverse arrow to white on black</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

Actually, impact analysis begins once all columns except the Resolution column have been completed for all observed problems. Depending on the number of design decisions to be considered, tools may be used for decision making, such as a graphical representation of problem distribution on the importance vs. cost scale. In principle, highly important problems should be tackled first, but the development team must also consider the resources allocated for the design activity and act accordingly. The process of deciding between design improvements that need to be made is not easy, and all this technique does is provide information in a structured manner so that the development team can make an informed decision.


**IV.4 Menu-selection and Dialog Box Trees**

In a menu-based system, menu trees represent the structure of menu navigation. Menu trees are powerful as a specification tool since they show users and other stakeholders the complete and detailed coverage of the system. Like any map, a menu tree shows high-level relationships and low-level details.

Similar comments apply to dialog-boxes. Printing out the dialog boxes and showing their relationships by mounting them on a wall is very helpful for gaining an overview of the entire system to check for consistency and completeness.

When we cannot create an interaction scheme based on direct manipulation (as in the desktop metaphor in the Mac and Windows operating systems), we can use menu selection. If the menu items are written using familiar terminology and are organized in a convenient structure and sequence, users can select an item easily.

When a collection of items grows and becomes difficult to maintain under intellectual control, designers can form categories of similar items, creating a tree structure. Menu trees represent this structure. With large systems, the menu tree may have to be laid out on a large wall or floor, but it is important to be able to see the whole structure in one go to check for consistency, completeness, and lack of ambiguity or redundancy.
It is difficult to group menu items in a tree so that they are comprehensible to users and match the task structure. Problems include overlapping categories, extraneous items, conflicting classifications in the same menu, unfamiliar jargon, and generic terms. The members of the development team may discuss all these issues while they all share a view of the complete tree structure represented in a menu-selection tree.

Websites that are organized in a highly hierarchical structure can be easily represented by means of tree menus. The following figure shows the menu tree of the STATUS project website:

---

**Legend:**

- **Public Access**
- **Access restricted by password to Project Partners**
- **Access restricted by password to the Commission PO**
- **Access restricted by password to members of the IST Prog.**

---


**IV.5 Organizing Help by Use Cases**

The design of the help subsystem is important for the overall usability of the software product. Good help will not turn an unusable product into a highly usable one, but well-written, well-organized and accessible help can compensate to some extent for limitations in software.

Essential use cases express what the users may want to accomplish with the system, so they are an excellent basis for organizing help contents. This kind of structure is useful for providing procedural help, that is, help with how to perform a task.

Additionally, the design of the help subsystem may be undertaken for a set of general help use cases (or help cases), which give a response to common help requests made by users of all kind of systems: seekingIdentification, seekingInstruction, seekingClarification, seekingElaboration, seekingReminder, seekingLocation, and exploringFeatures.

Procedural help is most helpful when it is organized by use cases that are titled and written in the ordinary language of the users and the application domain and are well indexed. Use cases are a natural way of organizing and providing access to help because they represent the basic intents of users. Each essential use case is a complete and well-defined task based on something a user might try to accomplish. If the
essential use case model has been well constructed, it will reflect how users think about and conduct their work. Each use case then becomes an entry in the help file.

To design support for other kinds of help, we can focus on the common help requests described as help cases (use cases for help seeking). Note that procedural help is expressed by means of the help case seekingInstruction. The following table details common user questions and the corresponding help case:

<table>
<thead>
<tr>
<th>User Question</th>
<th>Help case</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is this?</td>
<td>seekingIdentification</td>
</tr>
<tr>
<td></td>
<td>indicate object brief description</td>
</tr>
<tr>
<td>How do I...?</td>
<td>seekingInstruction</td>
</tr>
<tr>
<td></td>
<td>identify task operational sequence</td>
</tr>
<tr>
<td>What should I do?</td>
<td>seekingSuggestion</td>
</tr>
<tr>
<td></td>
<td>request hint</td>
</tr>
<tr>
<td>What do you mean?</td>
<td>seekingClarification</td>
</tr>
<tr>
<td></td>
<td>request different explanation</td>
</tr>
<tr>
<td>Tell me more</td>
<td>seekingElaboration</td>
</tr>
<tr>
<td></td>
<td>request details or advanced</td>
</tr>
<tr>
<td>Remind me about...</td>
<td>seekingReminder</td>
</tr>
<tr>
<td></td>
<td>identify feature task brief synopsis</td>
</tr>
<tr>
<td>Where is...?</td>
<td>seekingLocation</td>
</tr>
<tr>
<td></td>
<td>identify feature give place, routing</td>
</tr>
<tr>
<td>What can I do?</td>
<td>exploringFeatures</td>
</tr>
<tr>
<td></td>
<td>request overview, topic map</td>
</tr>
</tbody>
</table>


**IV.6 Product Style Guide**

Consistency is a very important asset of any UI from a usability point of view. The Product Style Guide document is a way to assure that all UI elements are built according to a consistent pattern. The style guide specifies the standard for the creation of UI elements that will be followed in any screen of the product. It contains the basic templates, controls, and rules of design for a product, or family of products.
The following table shows the main benefits provided by a Product Style Guide to different stakeholders:

<table>
<thead>
<tr>
<th>End Users</th>
<th>Developers</th>
<th>Business Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Reduced errors</td>
<td>• Maintain control over look and feel</td>
<td>• Produce usable systems that reduce support costs and increase user satisfaction</td>
</tr>
<tr>
<td>• Less frustration</td>
<td>• Minimize re-invention</td>
<td>• Increase market awareness</td>
</tr>
<tr>
<td>• Increased morale</td>
<td>• Capitalize on learning</td>
<td>• Increase product awareness</td>
</tr>
<tr>
<td>• Improved efficiency</td>
<td>• Enable production of reusable software</td>
<td>• Reduce training costs</td>
</tr>
<tr>
<td>• Increased confidence</td>
<td>• Reduce development time</td>
<td>• Improve staff retention</td>
</tr>
<tr>
<td>• Reduced resistance to new technology</td>
<td>• Reduce arbitrary design decisions</td>
<td>• Increase user acceptance of new systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Typical elements of a style guide include:

- General presentation rules for products/processes
- Characteristics of standard screen types (browser, form, search,...)
- Rules for consistent usage of colors, dimming, bold, reverse video, fonts, sound, etc.
- Rules for controls and organization and layout of dialog box fields. Details may be included about the ordering of elements, group boxes, alignment, and use of white space.
- Keyboard shortcut schema
- Rules for message boxes (errors, warnings)
- Rules for the display of status information

The writing of a style guide is not an easy task, as it requires some expertise in this kind of issues. Nevertheless, it usually pays off when undertaken along with other user-centered activities, since the products of usability techniques give the basis for the decisions to take in the production of a style guide. Start with high-level architectural guidelines and standards that will have the most impact (templates for Web pages or major features like search). Have small groups with the relevant expertise write draft sections and have the entire group review them and provide feedback. After defining the high-level pages or screens, work on lower level issues and general issues like color, use of controls, and text guidelines. Base these more detailed issues on the high-level components so the design is internally consistent.

Decide what processes will be implemented to publicize, distribute, update, and enforce the guidelines and standards in the style guide. Convince the product team to implement consistency inspections and make consistency part of everyone’s objectives, bonuses, and job descriptions. Conduct style guide training with managers, developers, and other key groups.

For a style guide to be useful it must be created in a way that ensures its usability. Prepare drafts and test them with UI implementers, refining it until it is fit for purpose. It is a waste of time and money producing a style guide that remains on the shelf and is never used.

Also, for the success of a style guide, it is very important to establish a process of consistency inspections that ensure that the style guide specifications are actually followed in the resulting software product, or family of products.

It is sometimes tempting to think that when having a style guide all usability problems go away. A UI style guide is a foundation for design but does not guarantee the success of a design. A style guide is only one link in the chain of user-centered design activities that are important for product success.
Requirement analysis, user profiles, task analysis, usability evaluation, and iterative design must accompany a style guide.

Usability is a very complex concept, due to the complex nature of humans. Without doing some form of evaluation, it is impossible to know whether or not the design or system fulfills the needs of the users and how well it fits the physical, social and organizational context in which it will be used. No matter how much we stress the performance of user-centered activities in the development process, we will not be able to exactly predict the usability level of the system in advance. For this reason, we need to perform usability evaluation at the end of every iterative design cycle to find out where the product is in usability terms and how much improvement is needed in order to reach the previously specified usability goals.

**V.1 Cognitive Walkthrough**

In a cognitive walkthrough, correct sequences of actions are analyzed, asking if they will actually be followed by users. The cognitive walkthrough analyst identifies problems by tracing the likely mental processes of a hypothetical user. The analysis considers matters like user background knowledge that influence mental processes but are not part of the user interface. The technique aims to identify likely usability problems in the user interface and to suggest reasons for these problems.

Cognitive walkthroughs were developed for systems that can be learned by exploratory browsing, but they are useful even for systems that require substantial learning.

The cognitive walkthrough analyst must begin by defining the assumed user background. The user structure role model should provide this information. Then the analyst must choose a representative task and devise a realistic usage of this task. If usage scenarios have been created, they can be a good source for realistic usage of tasks.

The analyst determines one or more correct sequences of actions for the chosen task. A correct sequence of actions is one that developers would be happy to see users use. Often there will be more than one acceptable way of performing a task. If these variations are important a cognitive walkthrough can be done on more than one, but often it will be sensible to choose the most common, or perhaps the most problematic.

The final step in the preparation of the cognitive walkthrough is to work out as fully as possible what the user will see at each step of the sequence or sequences to be examined. This may sometimes force the developers to create a partial design that is detailed enough to indicate the key interface features along the path. Screen sketches and/or dialogue flow (use cases) are usually enough to perform a cognitive walkthrough.

In the walkthrough itself, the analyst works through the sequence of correct actions, considering the state of the interface before and after each action, trying to determine how likely it is that users will follow that path. The kind of questions the analyst must try to answer are:

- Will the user be trying to achieve the right effect?
- Will the user notice that the correct action is available?
- Will the user associate the correct action with the desired effect?
- If the correct action is performed, will the user see that progress is being made?
For each correct action, the analyst must construct a success or failure story. If all the answers to all the questions are “yes”, including an explanation, then it is a success story. If the answer to one or more of the questions is “no” or “not always”, the analyst has a failure story. The explanation of this answer will tell the development team why the analyst expects that some users will have trouble at this point.


**V.2 Heuristic Evaluation**

Heuristic evaluation is performed to identify the usability problems of a system, so that they can be attended to as part of an iterative design process. It involves having a small set of evaluators examine the interaction design and judge its compliance with recognized usability principles (the heuristics). It can be used as a complement to usability testing with users, since it usually reveals different kinds of usability problems than usability testing.

Each evaluator inspects the interaction design alone. After the evaluations have been completed, the evaluators may gather to report their findings. This procedure is important in order to ensure independent and unbiased evaluations from each evaluator.

During the evaluation session, the evaluator goes through the interaction scheme (screen sketches and/or use case description) several times and inspects the various dialogue elements and compares them with the list of recognized usability principles. These heuristics are general rules that seem to describe common properties of usable interfaces, like the ones described in section IV.1. In addition to the checklist of general heuristics to be considered for all dialogue elements, the evaluator is also allowed to consider any additional usability principles or results that come to mind that may be relevant for any specific dialogue element.

The number of evaluators to be employed depends on the criticality of system usability, but it is clearly better to combine evaluations by several evaluators than have a single evaluator. Experts recommend using about five evaluators and certainly at least three.

Unlike other evaluation methods, such as walkthroughs, the evaluators decide on their own how they want to proceed with evaluating the interface, instead of following the predefined paths given by use cases.

The output from heuristic evaluation is a list of usability problems in the interaction design, annotated with references to the usability principles that were, in the opinion of the evaluator, violated by the design in each case.


**V.3 Pluralistic Walkthrough**

A pluralistic usability walkthrough is a collaborative process involving users, developers and other stakeholders, where all participants are expected to play the role of users. The participants evaluate the interaction design by trying to perform a given task, and they stop at each step to have a group discussion about its usability. The goal of the technique is coordinated empathies to help developers to put themselves in the shoes of users.
A pluralistic walkthrough is driven by a task scenario chosen in advance and for which a storyboard, a series of screen sketches or paper prototypes representing the various contexts or a working prototype have been prepared. For each step in the task, all participants independently decide on what action or actions they would take next and note these on their own copies of the storyboard. No discussion takes place until all participants have completed a given step. When the discussion of the step begins end users speak first to prevent the developers dominating the discussion.

This technique is relatively slow, since all participants have to be at the same step at the same time. But this technique has some appealing advantages, cited by participants in this kind of evaluations, such as:

- They feel that their viewpoints have been heard,
- their expertise was valued, and
- their design concerns were remedied to satisfaction.


V.4 Usability Inspections

Inspections have a long history in software development. The goal of all inspections is to find defects. Usability inspections are aimed at identifying usability defects. The object of inspection may be a finished product, a design or a prototype. Usability inspections refer to systematic processes for inspection, as opposed to heuristic evaluation, which is a less formal usability assessment technique.

When different stakeholders perform the inspection in a collaborative effort, it is called collaborative usability inspection. In this case, the review process is a team effort that includes software developers, end users, application or domain experts and usability specialists, collaborating to perform a thorough and efficient inspection.

There are two variants of inspection, which have a specific focus: consistency inspections and conformance inspections.

In consistency inspections, the goal is to identify inconsistencies across interaction contexts and their contents. The evaluators check for consistency of terminology, color, layout, input and output formats, and so on. When the product belongs to a family of products, teams of designers, at least one from each project, meet to inspect the usability of the different products of the family.

In conformance inspections, the participants inspect the system interaction for compliance with specified standards or with style guidelines. All participants must be familiar with the applicable standards and/or style guidelines.

Collaborative usability inspections, if well conducted, can be more productive than expert inspections. The focus needs to be kept on the user perspective, in order to identify the usability problems that might arise. Developers need to adopt the mindset of an impatient and intolerant user. The presence of actual users in inspections helps to catalyze taking the user perspective. Two special roles in the inspection team are the lead reviewer, who organizes the inspection meetings and moderates the process; and the inspection recorder, who maintains a complete log of identified defects. Another special role in the team may be the continuity reviewer, who has special responsibility for identifying inconsistencies. Apart from members of the development team, it may be useful to have some developers who have not participated in the development effort, because they bring a fresh perspective into the inspection. Members of the
development team are not allowed to defend, explain, excuse or rationalize any aspect of their design or the decisions leading to it. Developers should also avoid making implied promises to the users. The comments and inputs from users should be given special weight in the inspection process, without allowing these to dictate interaction design decisions. The lead reviewer should encourage user participation and protect users from criticism or antagonistic questioning. Users and domain experts should be regarded as authorities, but not as arbiters. Finally, usability experts may also contribute to collaborative usability inspections.


V.5 Thinking Aloud Protocol

Thinking aloud is a technique for performing usability tests with users. The evaluator asks participants (users) to talk out loud while working during a usability testing session, indicating what they are trying to do, or why they are having a problem, what they expected to happen that did not, what they wished had happened, and so on. By verbalizing their thoughts, test participants enable the developer to understand how they view the system, and this helps to identify the major user misconceptions.

The strength of thinking out loud is on qualitative data and not on performance measures. The idea is to get the user’s impression while using the system to avoid later rationalizations. The aim of this kind of testing is to detect the parts of the dialogue that are more problematic from a usability point of view, along with the real causes of the problems.

There are some variants of this technique: constructive interaction, retrospective testing, critical incident taking and coaching method.

Thinking aloud is can be employed in any usability test. There is no difference in the test preparation with performance measurement usability testing. But, before starting with the test, the evaluator must encourage the test participant to think out loud, maintaining a running monologue about what he or she is doing as it is being done.

The evaluator may find that some participants are not good at thinking aloud while they work. They will not talk much, and the evaluator will have to prod them constantly to find out what they are thinking or trying to do. This has an impact on performance measures, so we do not advise combining thinking aloud with performance measurement.

User comments are sometimes indicators of personal user likes or dislikes, so developers should take care not to change part of the system just because of a comment by a single user. It is the responsibility of the evaluator to interpret the user comments and not just accept them indiscriminately. For example, users using a mouse for the first time will often direct a large proportion of their comments toward aspects of moving the mouse and pointing and clicking. In this case, the evaluator should try to abstract from the mouse problems in the dialogue and focus on other system issues.

The following techniques are variants of the basic think-aloud protocol:

- **Constructive interaction**: It involves having two test users use a system together. It is also called codiscovery learning. It aims to overcome the problem of shy test participants, who do not verbalize easily. This is based on the fact that people are used to verbalizing when they are trying to solve a problem in a collaborative effort.
• **Retrospective testing**: The usability testing session is recorded on a videotape and the participant is requested to review the recording. Participant comments while reviewing the tape are sometimes more extensive than comments while performing the task in the test. The evaluator can stop the tape and ask the participant questions at any time, without fear of interfering with the test, which has essentially been completed already. This variant may be useful when the usability testing involves some kind of performance measurement that could be distorted by the dialogue with the reviewer.

• **Critical incident taking**: This variant implies recording both negative incidents (signs of frustration, either with remarks or actions), and positive incidents (satisfaction or closure expressions). Negative incidents help to identify the more important usability problems, while positive incidents help to identify metaphors or details to be used more thoroughly in the user interface because of their success.

• **Coaching method**: The evaluator (or “coach”) steers the participant in the right direction while using the system. The participant can ask the evaluator questions, and the questions may show up usability problems that would remain uncovered otherwise. The evaluator will answer to the best of his or her ability.


**V.6 Performance Measurement / Laboratory Usability Testing**

Performance measurement through usability testing is used for assessing whether usability goals set in usability specifications have been met. It can be used as well for comparisons with competing products. Performance is measured by having a group of users perform a predefined set of test tasks while collecting time and error data. When the test is performed in a special room prepared for usability testing, it is called laboratory usability testing. A laboratory is usually composed of two rooms separated by a one-way mirror: the evaluation room where participants carry out the tests and the main evaluator gives instructions; and the control room, where additional evaluators and other members of the development team can observe the test, without disturbing the test participant. Usual equipment for a usability laboratory includes a video camera to record the screen, another one for recording the participant, tools for software logging and monitors to show in the control room what is happening in the evaluation room.

The opposite to laboratory testing is field testing, where the system is taken to the user environment instead of taking the participant to the system, and the usability test is performed in the user organization. Before performing any test, the first step is to develop the experiment. The participants must be selected, trying to get a representative sample of the total user population. Information in the structured user role model should serve to select an adequate distribution of test participants. The tasks to be employed in the tests must be defined as well, the benchmark tasks that appear in the usability specifications must be tested, but additional representative tasks may be tested as well, not to get performance measurements but to identify usability problems. The evaluator should write down the tasks in the order that the participant will be asked to perform them. This list of tasks may be either given to the participant or read out loud by the evaluator one task at a time. Finally, the evaluator must define the protocol and procedures for the test. This includes the preparation of introductory instructional remarks for participants, which should briefly explain the purpose of the experiment, the system to be tested and what the participant will be
expected to do. It is important to make clear to all participants that the purpose of the test session is to evaluate the system, not the participant. An informed consent form should also be prepared for participants to sign, stating that the participant is volunteering for the experiment, that the data may be used if the participant’s name or identity is not associated with the data, that the participant understands that the experiment is in no way harmful and that the participant may discontinue the experiment at any time.

It is advisable to perform a few pilot tests with three or four participants to ensure that all parts of the experiment are ready. Pilot testing may show up inadequate wording of the tasks that the participants are being asked to perform, or that some part of the procedure needs to be changed. After pilot testing the test plan is refined in order to proceed with the greater part of the testing effort using an improved test plan.

For the test session, the evaluator will usually be sitting beside the participant, especially when qualitative data needs to be collected. The test participant is asked to perform the tasks defined for the test, and both the number of errors and the performance time are measured for each task. It may be necessary to prompt the participant during the session, primarily during qualitative data collection, to get the desired information. The think-aloud technique and its variants may be applied for this purpose in any usability test.

The data collected during the test sessions must then be analyzed. Quantitative data will be formed by performance times, error rates, and also by the user preference expressed in questionnaires. Qualitative data will come from the user comments that the evaluator has taken down or extracted from an audio or video recording of the session. The information gathered in usability tests can tell the development team whether or not the development is going in the right direction (that is, whether we are coming closer to the goals in the usability specifications or not), and it can point out the issues in the interaction dialogue that are a source of usability problems. After an impact analysis, decisions are taken about which usability problems will be tackled first in the next cycle redesign effort.


**V.7 Questionnaires and Surveys**

Questionnaires are used to determine a user’s subjective satisfaction with the system. Measuring user satisfaction provides a subjective (but, nevertheless, quantitative) usability metric for the related usability attribute. Some usability specifications will be related to user satisfaction, and questionnaires are the way to check whether the level specified for this attribute has been reached. Questionnaires are usually administered to usability test participants after the test has taken place, so they can give their opinion about specific parts of the user interface and about the overall system.

When questionnaires are distributed to a lot of users, they are called surveys. While questionnaires issued to usability test participants may contain questions about specific parts that have been used in the test, surveys usually gather opinions on more generic issues. Additional information that is usually collected has to do with individual user characteristics, such as background (age, gender, education), experience with computers, familiarity with specific features (virtual reality, macros, shortcuts), and so on.

It is advisable to do a pilot study before sending questionnaires to a large number of users in order to ensure that it is well designed. Care must be taken to make the questions unambiguous, and the
questionnaire in general should be as simple as possible to increase the chance of respondents completing and returning the questionnaire.

Questions may be open, where the respondent is free to provide his or her own answer, or closed, where the respondent is asked to select an answer from a choice or alternative replies. Closed questions usually have some form of associated rating scale. The most commonly used scale for HCI studies is the semantic differential scale. This scale is based on bipolar adjectives (such as easy-difficult, clear-confusing) at the end points of the scale and respondents rate on a scale between these paired adjectives. This is the scale used in the questions in the table below.

Once the questionnaires have been given to the selected population, the responses obtained on the different rating scales are converted into numerical values and statistical analysis is performed. The main statistics used in the analysis of surveys data are means and standard deviations.

The table below shows sample questions belonging to a questionnaire to be administered to test participants after a usability testing session. The tool to be tested provided facilities for managing problem resolution tasks in a Help Desk. A differential semantic scale with five choices was used for each question, centering the scale around zero. As a mid-scale reading, zero is an appropriately neutral value. Negative scale readings correspond to negative user opinions and positive readings to positive opinions.

The final category of questions is focused on overall user reactions.

<table>
<thead>
<tr>
<th>Question</th>
<th>Unsatisfactory</th>
<th>Satisfactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>General satisfaction</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Suitability for problem solving tasks</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>General comparison with existing system</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Feedback provided to user actions</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Overall opinion about the system:</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Terrible</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Frustrating</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Dull</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Difficult</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Rigid</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>Wonderful</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Satisfying</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Stimulating</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Easy</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Flexible</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Beta-testing is a survey-based form of evaluation. In beta-testing, a working but not completely finished version is supplied to a big pool of customers who are willing to test the product using it to perform their work (or to fulfill their goals). In addition to questions on possible system failures, beta-testers may be asked to answer preference questions after their usage of the system.


**V.8 Interviews**

Interviews involve having an interviewer read questions to a respondent and writing down the responses. For the creation of the questionnaire, refer to section V.7 Questionnaires and Surveys.

After usability testing the evaluator may interview the participant to get the user’s subjective opinion, instead of letting the participant fill in a written questionnaire. Interviews are more flexible, since the evaluator may ask follow-up questions that not were in the script.

Interviews need to be planned for them to yield useful results, much in the same way questionnaires must be carefully planned before being administered to users.

There are two main kinds of interviews: structured, where the questions are predetermined and flexible interviews, where the interviewer is free to follow the interviewee’s replies and to find out personal attitudes. Flexible interviews are less formal, and they are adequate for requirements elicitation and for gauging users’ opinions about a particular idea. No matter how flexible the interview is going to be, a rough plan of the topics to be discussed is still needed.

The interviewer should make the interviewee feel comfortable, establishing interviewer-interviewee rapport. For example, some people feel embarrassed when they criticize a system, particularly when they have to describe their own difficulties in using it.

When the interviewer has a set of questions prepared in case the interviewee digresses or does not say much, it is called a semi-structured interview. A variant for drawing out more information from the interviewee is prompted interviewing, where the interviewer stimulates the interviewee by saying things like “... and can you tell me a bit more about that” or “...and what do you mean by...”. Alternatively, prompting may take the form of showing the interviewee an alternative item such as a screen design, in order to promote further discussion or generate new ideas for discussion.

The trade-off to be considered in structured vs. flexible interviewing is that the less structured the interview is, the more scope there is for picking up relevant issues but the harder it is for the interviewer. Flexible interviews on usability issues have been predominantly used to determine the user’s understanding of the interaction scheme. An issue to consider is that the interviewer should avoid asking leading questions that beg a particular response.

As for questionnaires and surveys, when preparing an interview with domain experts (who are usually a scarce resource), it is better to do a small pilot study to be able to refine the interview script.

V.9 Direct Observation and Video/Audio Recording

Individual users may be directly observed doing specially devised tasks or doing their normal work, with the observer making notes about interesting behavior or recording their performance in some way, such as timing sequences of actions. This is called direct observation. When the observation takes place in the user organization, it is called field usability evaluation.

Video recording can be either an alternative to direct observation or a backup for what happens in a usability evaluation session. For field usability evaluation, audio recording can be useful as well to record the user comments.

The evaluator should be prepared to take copious notes as activities proceed during a usability evaluation session. It may be useful to have a second evaluator also observing the session in order to help take notes. Especially for usability testing sessions, the first evaluator may be in charge of conducting the session (giving instructions, prompting the user) and timing tasks where necessary, while the second evaluator may be in charge of just taking notes.

Even if the evaluator (or evaluators) is fast at note taking, the record of the observation will usually be incomplete. Direct observation only allows for one go at data collection, so the evaluator rarely gets a full record of user activity for detailed analysis. The evaluator has to make decisions about what is important to record and has no chance to revise that decision and look at alternative data later on. For these reasons, if a permanent record is needed, video recording equipment may be used to record usability evaluation sessions. Usability laboratories are usually equipped with video cameras and perhaps some video editing equipment as well. The main advantage of videotaping is to capture every detail that occurs during the session. If multiple cameras are available, one can be aimed at the participant’s hands and the screen, and another at a broader view including the participant’s face. Audiotaping may be done when videotaping is not available, as, for example, in field testing. Just having a record of all the user’s comments may prove invaluable for later data analysis.

The main disadvantage of videotaping is the time it takes to edit the taped material. A ratio of 5:1 (analysis time to recording time) is often cited, that is, it usually takes five hours to analyze one hour of videotape. When more than one camera is used, the editing is also very time consuming, since synchronization problems may arise.

Short video clips of users experiencing problems with a given software product can have a big influence on a development team, especially, if the development team is reluctant to make changes to what they consider to be their already perfect design. These same video clips can also be useful for convincing management that there is a usability problem in the first place.


V.10 Focus Groups

Focus groups are a somewhat informal technique that can be used to assess user needs and feelings after the system has been in use for some time. Focus groups often bring out spontaneous reactions and ideas from users through the interaction between the participants and have the major advantage of allowing some group dynamics and organizational issues. Focus groups are especially appropriate for limited user communities.
In a focus group, a group of users are brought together to discuss new concepts and identify issues over a period of about two hours. Each group is run by a moderator who is responsible for maintaining the focus of the group on whatever issues are of interest. From the user perspective, a focus group session should feel free flowing and relatively unstructured, but, in reality, the moderator has to follow a preplanned script for what issues to bring up.

To prepare a focus group, the moderator needs to prepare a list of the issues to be discussed and set goals for the kinds of information that are to be gathered. During the group session the moderator has the difficult job of keeping the discussion on track without inhibiting the free flow of ideas and comments. Also, the moderator needs to ensure that all members of the group get to contribute to the discussion and guard against having the opinions of any single participant dominate unduly. After the session, data analysis can be as simple as having the moderator write a short report summing up the prevailing mood in the group, illustrated with a few colorful quotes.

Focus group discussions may be held after a set of individual user interviews have been conducted. Then, focus-group discussions may be valuable to ascertain the universality of comments. Individual interviews are costly and time consuming, so usually only a small fraction of the user community is involved. On the other hand, group discussions offer more representative results.


**V.11 Logging Actual Use**

Logging involves having the computer automatically collect statistics about the detailed use of the system. It is mainly used to collect information about the field use of a system after release, but it can also be used as a supplementary method during usability testing to collect more detailed data. It is unobtrusive, so it does not interfere with the user’s normal usage of the system.

When the actual use of the system is logged, this information is particularly useful because it shows how users perform their actual work and because it is relatively easy to automatically collect data from a large number of users working under different circumstances. Typically, an interface log will contain statistics about the frequency with which each user has used each feature in the system, and the frequency with which various events of interest (like, for example, error messages) have occurred.

When undertaking a major redesign for a system that has been in use, it is very helpful to rely on interaction log information to guide the redesign effort.

For this technique to be applied, the software architecture should make it easy for system managers to collect data about the patterns of system usage, speed of user performance, rate of errors or requests for online assistance.

There are different software logging tools that can be employed for logging actual use, but there are two main categories: time-stamped keypresses and real-time interaction logging. Logging time-stamped keypresses simply provides a record of each key that the user presses along with the exact time of the event. Interaction logging is similar, except that the recording includes real-time information, which means that it can be replayed in real time so the observer can see the interaction between the user and the computer exactly as it happened.
Logging may be well intentioned, but user rights to privacy should be respected. Links to specific user names should not be collected, unless necessary. When logging aggregate performance crosses over to monitoring individual activity, managers must inform users of what is being monitored and how the information will be used.

It is usual to combine video, audio and keypresses or interaction logging. The advantage of using combinations of data capture techniques is that evaluators can relate revealing data about body language and comments with records of the actual human-computer interaction. The main disadvantage of this approach is the cost of setting up this kind of synchronized equipment.


**V.12 User Feedback Facilities**

Once the system is in use, the user community is the best source for information on the usability weaknesses of the system. Feedback from the users can be collected by giving them access to special electronic mail addresses, network newsgroups, or bulletin boards. Users can send their complaints and requests for change or improvement.

Offering a help line or a communication channel with users can be implemented in different forms. These are the main ways of gathering user-initiated feedback:

- **Online or Telephone Consultants**: They can provide extremely effective and personal assistance to users who are experiencing difficulties. Many users feel reassured if they know that there is a human being whom they can address if problems arise. These consultants are an excellent source of information about problems users are having and can suggest improvements and potential extensions. Some organizations offer a toll-free number for users, while others charge for consultation by the minute.

- **Online Suggestion Box or Trouble Reporting**: Email can be employed to allow users to send messages to the maintainers or designers. Such an "online suggestion box" encourages some users to make productive comments, since writing a letter may be seen as requiring too much effort.

- **Online Bulletin Board or Newsgroup**: Users may have questions about the suitability of a software package for their application, or may be seeking someone who has had experience using an interface feature. They do not have any individual in mind, so email does not serve their needs. Then bulletin boards and newsgroups can be helpful. Electronic bulletin boards or newsgroups allow the posting of open messages and questions. Mailing lists may be used as well for this purpose.

By soliciting user feedback by any of these ways, the development team can gauge user attitudes and elicit useful suggestions. Furthermore, users may have more positive attitudes towards the system if they see that the software development organization genuinely desires comments and suggestions on the piece of software they are using.

Bibliography


