WHITEPAPER

The Performance-Centered Design and Development Methodology

Dr. Karen L. McGraw

Cognitive Technologies, Inc.

Web: www.cognitive-technologies.com

Ph: 512.380.1204

Email: info@cognitive-technologies.com

COGNITIVE TECHNOLOGIES

Introduction

The “hot term” a few years ago in the development of user interfaces for software systems was user-centered design. In this white paper we argue that the key to successful software applications or systems is really performance-centered design.

We present an overview of an approach we have used with small software applications (e.g., a time tracking tool), large software systems (e.g., a call center application), and special software components (e.g., performance support modules) developed to improve software usability.

What is Performance-Centered Design and Development?

Performance-Centered Design and Development

Performance-centered design and development reflects the user’s:

- Work process
- Information needs
- Workforce characteristics
- Usability requirements

Performance-centered design and development of technology systems and applications is characterized by the following activities:

- Capture/documentation and analysis of the user’s work processes and information needs, including capture and analysis of typical and atypical scenarios that reflect work to be done and the conditions in which it is accomplished
- Analysis of the range of user characteristics embodied in the workforce most likely to use the new system
- Documentation of usability requirements, based on user characteristics and the performance requirements of the job
- Rapid prototyping to develop a proof of concept or visual prototype to test early understandings and determine/refine requirements
- Short, rapid prototyping cycles to develop the technical prototype
- Multi-phase usability analysis and assessment using both formal and informal methods during and after the design/development cycle.

Benefits

A performance-centered approach can help achieve numerous benefits for the final product or application. Operation of an application that is performance centered can have the following affect:

- Lower support costs
- Enable the organization to hire lower-level staff
- Lower turnover
- Lower maintenance cost
- Improve employee morale
- Improve return on technology investment
- Decrease development time and cost.For the people who *use* the performance-centered system the benefits include:
- Improved job satisfaction

- Decreased training time
- Less documentation
- Less fatigue and injury
- Fewer errors
- Better productivity and work quality

Cognitive Technologies' Performance-Centered Methodology

Figure 1 illustrates the iterative methodology we use for performance-centered design and development.

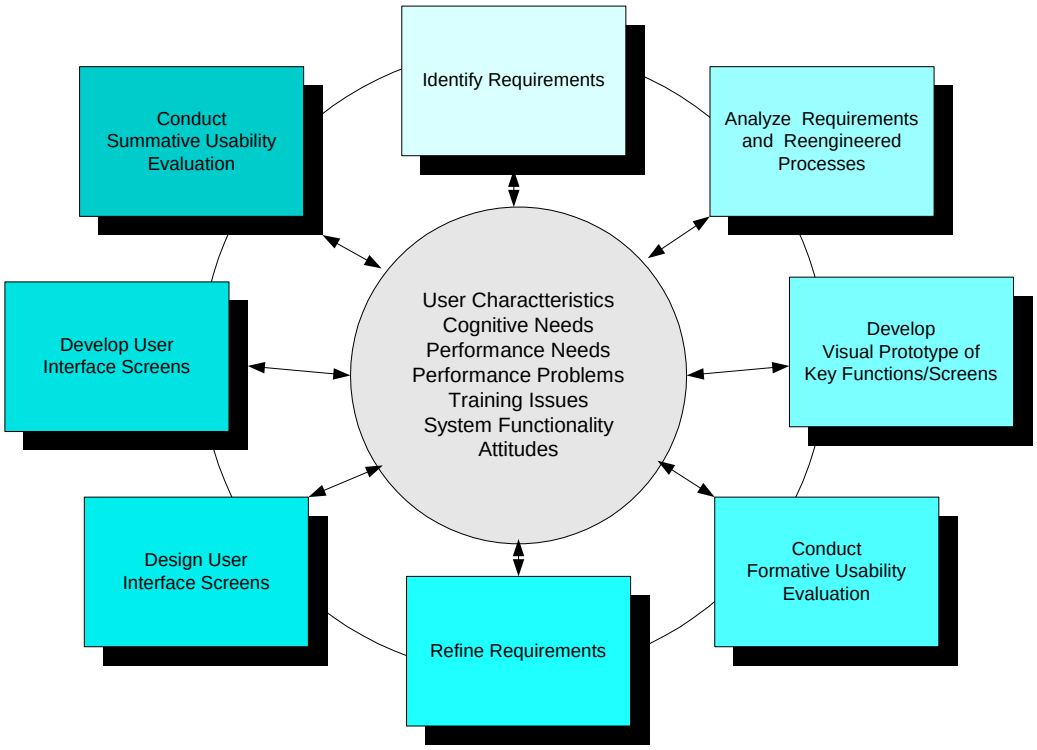

Figure 1. Performance-centered design and development methodology.
© 2000, Cognitive Technologies Group, LLC.

Both the outer and inner "circles" of this methodology are critical to the design and development of effective user interfaces. Factors in the inner circle drive performance-centered design and development. In turn, activities in the outer circle and the products created by these activities must reflect the inner circle issues.

Requirements Identification

This process begins with the explication of initial customer requirements for the new application or system. For example, the customer might say*, "we want a system that will enable us to reduce training time 50%, increase productivity 25%, and decrease turnover in the user community by 20%."* These goals are tied directly to the user's performance on the system.

Thus, the first step in our methodology is to begin a two-pronged requirements identification phase:

1. Capturing user characteristics, stated needs, problems, attitudes, and known requirements.
2. Understanding the functional, environmental, technical (e.g., hardware, software) and other requirements (e.g., security, reliability, etc.) for the system.

- Work process
- Information needs
- Workforce characteristics
- Usability requirements

Knowledge Acquisition

Knowledge acquisition is the process of extracting, transforming, and transferring information and expertise from a knowledge source into another form. It is critical to performance-centered design and development.

To accomplish these tasks we conduct knowledge acquisition (KA).[1] Knowledge acquisition is the process of extracting, transforming, and transferring information and expertise from a knowledge source into another form.

Typical targets for KA activities during the requirements phase include:

- Identifying the information needs, including required input and output, sources and destinations for data, and information manipulation
- Defining responsibilities or assignments, including understanding the job functions and goals, work processes, and critical success factors for the user community
- Documenting standards and criteria for the user's job performance that may be impacted by the new system
- Capturing decision making factors and heuristics (i.e., rules of thumb) users apply in performing job functions affected by the system
- Capturing problem solving patterns and preferences within the user community
- Documenting difficulties and problem areas (in the way the job is performed today) that technology will or can improve
- Documenting current training and support, including time spent in each portion of the curriculum and the most common requests for help or support.

To gather this information we use a mix of established human factors techniques (e.g., task analysis), specialized techniques (such as decision process tracing), and human performance analysis.[2] During KA activities we work with potential or current system users, trainers, supervisors, and system developers or administrators.

[1] McGraw, K. and K. Harbison-Briggs (1989). Knowledge Acquisition: Principles and Guidelines. New York: Prentice Hall.
[2] We use ASTD's *Performance DNA* (2004) methodology developed by Holloway, Mankin, and McGraw.

Analyze Requirements

Analysis activities include analyzing information acquired from the individual knowledge acquisition sessions in order to achieve the following:

- Develop a user profile of characteristics, age, typical education, and abilities
- Produce a system functional model that can be related back to the user's task model
- Produce a preliminary task analysis or a cognitive task analysis
- Develop process flows and identify opportunities to reengineer key work processes, rather than simply create a system that perpetuates less-than-optimal processes
- Understand current training, its costs, and the varying effectiveness of different modules
- Identify performance discrepancies among the user community and their impact on job performance and business results (both financial and non-financial impact, such as customer satisfaction).

While human factors engineers and designers are conducting their analyses, systems analysts focus on systems issues. For example, they must identify the potential impact of the system on architecture and address issues such as system performance requirements, data transfer rates, development tools, integration requirements, and support requirements.

Develop Visual Prototype

During analysis, we worked very closely with target system users, trainers, supervisors, and systems support personnel. After the analysis is complete and the preliminary requirements written, we begin prototyping. To develop a visual prototype, we use analysis output to create a sample scenario(s) or operational sequence diagram for at least one major functional area. Next, we work with a user interface management tool or other software to quickly develop sample screens. This can even be achieved using tools such as PowerPoint and VisualBasic. The resulting visual prototype enables us to test concepts, icons, and layout ideas and to anticipate problems that might surface during development.

Conduct Formative Usability Evaluation

Formative evaluation is conducted with selected members of the target user community to enable developers to ensure that a system is (1) designed to meet the stated purpose and objectives, (2) organized and sequenced to support the job, and (3) uses appropriate presentation and support strategies.

- Work process
- Information needs
- Workforce characteristics
- Usability requirements

Formative Evaluation

Formative evaluation occurs during the development of a system or application's user interface, and summative evaluation, while it is easy to make modifications to improve usability.

Major goals of the formative usability evaluation include:

- Investigate the "intuitiveness" of selected functions and screens, and the effectiveness of buttons, icons, and menu items
- Determine the degree to which a system meets project objectives, including user-centered design and overall usability criteria
- Reveal opportunities for refinement prior to development and implementation.

Formative evaluation activities depend heavily on the type of system being evaluated, how graphical the user interface is, and how the user will be expected to interact with the system. Activities we typically apply during formative evaluation sessions include card sorting of menus and functions (to test menu and function congruency with the user's mental models), icon recognition tests, and a simulation activity that requires users to name the button or menu item they would select to complete a stated a task. We may present a series of storyboards or screens and conduct an interview to determine expected ease of use and potential problem areas. Sessions usually conclude with the completion of a questionnaire.

Refine Requirements

The visual prototype was demonstrated and tested with selected users as a part of the formative usability evaluation. Often, as users view the visual prototype, they more fully understand what is possible and what can be done. We obtain clarification, new ideas, and additional requests during these user evaluation sessions.

Additionally, the visual prototype is demonstrated to other stakeholders (e.g., system support or help desk personnel, trainers, technical personnel, etc.) to elicit feedback that can be used to refine and gain agreement on requirements for performance, reliability, and integration. If potential technical difficulties are noted that impede the design demonstrated, a technical prototype or test bed may be required to test critical issues such as data transfer rates.

Design User Interface Screens

Although some requirements refinement continues throughout system design, the primary tasks that we undertake during this step include:

1. Gain final agreement on functions supported and the level of performance support (i.e., help system, coaching, or "smart tags") that will be provided through the user interface.
2. Develop design guidelines to be used for all screens in the system. This often requires us to work from, modify, and/or extend existing user interface guidelines that the customer may have adopted.
3. Create a top-level design and data flow diagrams which illustrate how information is passed and processed between system components.
4. Identify the contexts (i.e., tasks and screens) in which performance support components will be provided.
5. Outline the general content for each screen, including task-based layouts that will aid job performance.
6. Outline the content of each identified performance support component.

As in all previous steps, the deliverables and working documents produced in this step are presented to users or other stakeholders as required for feedback and refinement.

Develop User Interface Screens

This step involves development of the actual user interface screens and any performance support components that will be available in the system. This includes creating icons and other graphics, developing the individual screens, creating content for the help system or other performance support components, and testing individual screens to ensure the components work (functionality, flow, and performance). As each of these is complete, it is presented to selected users for review and refinement.

Conduct Summative Usability Evaluation

Summative Evaluation

Summative evaluation occurs near the end of the development cycle to identify performance problems that need to be addressed before system integration.

Summative evaluation is conducted near the end of a development cycle. It enables users to identify performance problems that need to be addressed before system integration, and to reveal usability issues in a simulated work context. Primary objectives include:

- Determine the appropriateness of the system to the job context
- Ensure that the system meets stated objectives
- Estimate expected impacts on human performance.

Summative evaluation requires that the system is at least partially operational because most of the activity requires that the participant interact with the system. During a “typical” summative evaluation session we start with an introduction and statement of purpose, and elicit demographic data from the participant. Then we introduce a series of test tasks, one at a time, to the participant. The participant is asked to use and interact with the system (either alone, or with a paired participant) to complete a specific task they’ve been given. Then we monitor task completion, noting factors such as completion time, errors, accuracy ratings, miscues, participant comments, etc. After the task has been completed we may ask a brief set of questions, or conduct a modified protocol analysis session to examine issues in more depth. At the end of the summative evaluation session, we ask the participant to complete a questionnaire intended to measure subjective factors such as user satisfaction with the system, expected ‘learnability’, ease of use, etc. Other techniques, such as focus groups, may also be used during summative evaluation. Data captured during summative usability evaluation includes:

- Types of errors
- Completion times for test tasks
- Accuracy/percent correct for test tasks
- User opinions and preferences.

The activities and output of the summative evaluation phase enables organizations to make final improvements based on hard user data before implementation to improve overall rollout and acceptance of the new system.

Closing

Performance-centered design and development of technology systems focuses not just on creating a system that is “usable” but in fact, on the creation of systems that improve the *performance* of its human users. The performance-centered methodology used by Cognitive Technologies is a systematic approach that yields both quantitative and qualitative benefits to organizations.

About Cognitive Technologies:

Cognitive Technologies, Inc. is a consulting firm specializing in project management, collaborative processes, and organizational effectiveness. We deliver the right solutions to improve the performance of business units through the application of our expertise in projects, people and technology. We are headquartered in Austin TX and have offices in Atlanta GA and the metro DC area.

Sales: 888.PMGURU1 (764.8781)
DC Office: 202.742.5291
TX Office: 512.380.1204
GA Office: 770.977.5555
Email: info@cogtechinc.com