Centers for Disease Control and Prevention (CDC)
Office of Infectious Diseases (OID)
National Center for Immunization and Respiratory Diseases (NCIRD)
Office of the Director

Contract No: 200-2014-60994

**FORMAL DELIVERABLE 10A**
**IMMUNIZATION-RELATED GUIDANCE**
**ATTACHMENT C:**
**USER-CENTERED DESIGN (UCD) PRIMER**

*September 28, 2015*
VERSION HISTORY

Formal Deliverable 10a, Attachment C, will be updated to reflect changes that incorporate the Centers for Disease Control and Prevention’s review and feedback, as well as any new requirements or changes to the business environment in which the project exists. The following table will track changes and updates to the document.

<table>
<thead>
<tr>
<th>Version #</th>
<th>Implemented by</th>
<th>Revision Date</th>
<th>Description</th>
<th>Approved By</th>
<th>Approval Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Draft</td>
<td>CNIADV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

## VERSION HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Introduction to Usability Principles

## 2 Defining Usability

2.1 Effectiveness
2.2 Efficiency
2.3 Satisfaction

## 3 NIST EHR Usability Protocol (EUP) and User Error

3.1 Ongoing Risk Assessment and Hazard Mitigation

## 4 User-Centered Design

4.1 Understand User Needs, Workflows, and Environments
4.2 Engage Users Early and Often
4.3 Set Performance Objectives
4.4 Design the User Interface from Existing Human Factors Principles
4.5 Test and Evaluate
4.5.1 Formative Testing
4.5.2 Summative Usability Testing
4.6 Adapt and Iterate Design with Users until Performance Objectives are Met

## 5 User-Centered Design Methods

5.1 Methods Used Early in the Process
5.1.1 Field Observation
5.1.2 Focus Group
5.1.3 User Interviews
5.1.4 Online User Surveys
5.1.5 Stakeholder Interviews
5.1.6 Task Analysis and Cognitive Task Analysis
5.1.7 Task Mapping
5.1.8 Card Sorting
5.2 Methods Used in the Middle of the Process
5.2.1 Iterative Design
5.2.2 Heuristic Review
5.2.3 Formative Testing
5.2.4 References
5.3 Methods Used Late in the Process
5.3.1 Summative/Validation Testing
VERSION HISTORY

Attachment C: UCD Primer, will be updated to reflect changes that incorporate the Centers for Disease Control and Prevention’s review and feedback, as well as any new requirements or changes to the business environment in which the project exists. The following table will track changes and updates to the document.

<table>
<thead>
<tr>
<th>Version #</th>
<th>Implemented by</th>
<th>Revision Date</th>
<th>Description</th>
<th>Approved By</th>
<th>Approval Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Draft</td>
<td>CNI ADV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


1 INTRODUCTION

CNI Advantage, LLC (CNIADV) conducted the Immunization-centric Electronic Health Record (EHR) Certification Process Pilot Project (the Project) for the Centers for Disease Control and Prevention (CDC), National Center for Immunization and Respiratory Diseases (NCIRD). The project evaluated whether EHRs and other clinical software could address critical safety or workflow issues related to usability and functions for immunization management. The parent document, Immunization-Related EHR Guidance, describes the overall user experience as a combination of utility, usability, and the ability for a user to complete the goal or task with personal meaningfulness (UX). As illustrated in Exhibit 1, utility addresses what the system does, whereas, usability addresses how the system does it. UX addresses how the overall user experience relates to personal meaningfulness. This document was developed as part of the Project and offers guidance on how to conduct a UCD process in order to improve usability outcomes related to the use of immunization functionality in EHRs. This document does not offer guidance on the topics of utility or UX.

The usability of EHRs and other clinical software may be significantly improved through the implementation of UCD processes. Immunization functionality in EHRs is of particular interest in the area of EHR usability. EHRs represent a high-risk environment where complex and often safety-critical tasks must be executed. Immunization management is part of that high risk environment. Even an occasional error can result in severe consequences, such as health risk or mortality. Immunization management systems should therefore be designed and optimized to reduce errors and minimize negative outcomes. UCD is a systems design methodology that uses scientific research methods to discover and understand human behavior and thought as it relates to the system user (Flach & Dominguez, 1995; Lowry, Ramaiah, Patterson, Latkany, Brick, & Gibbons, 2015). UCD uses a variety of methods, some of which are discussed in this document. The National Institute for Standards and Testing (NIST) published the “Guide to the Processes Approach for Improving the Usability of Electronic Health Records” (NISTIR 7741) that discusses the use of UCD processes in the design of EHRs (NIST, 2010). Some aspects of a sound UCD approach include the following:

- Understanding user needs, workflows, and work environments;
- Engaging actual users early and often in the design and development process;
- Setting user performance objectives;
- Designing the user interface from previously known human factors principles;
- Performing formative and summative usability activities; and,
- Adapting and iterating the design with representative end users.

A sound UCD process is crucial for achieving positive usability outcomes, and usability activities are cornerstones of the UCD process. This document summarizes key usability activities that can be used to develop a sound UCD process.

## 2 DEFINING USABILITY

The International Organization for Standardization (ISO) defines usability as the “extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” (ISO, 2010). The Office of the National Coordinator for Health IT (ONC) and the Institute of Medicine (IOM) highlight the need for clinical software to focus on human factors, safety culture, and usability (ONC, 2012; IOM, 2012).

A UCD process is a framework aimed at optimizing the usability of a system, product, or service, while at the same time including human factors specialists performing activities (e.g., risk analysis, failure and effects analysis, etc.) focused on areas that impact patient safety. Related to usability, some researchers have suggested that healthcare IT teams need to be educated on “two bins of usability” (Ratwani, Fairbanks, Hettinger, & Benda, 2015; Fairbanks & Ratwani, 2014). One bin is the user interface design (e.g., displays and controls, screen design, clicks, drags, and colors & navigation, etc.). The other bin is Cognitive Task Support (e.g., workflow design, data visualization, support of cognitive work, and functionality, etc.).

The following sections describe the concepts of effectiveness, efficiency and satisfaction in greater detail.

### 2.1 Effectiveness

In the context of EHR usability, effectiveness refers to the extent to which a system enables a user to successfully complete tasks (NIST, 2010; NIST, 2012). A typical measure of system effectiveness is the proportion of users who are able to successfully accurately complete a task. Narrowed to the context of immunization functionality, examples of effectiveness measures might include whether users can successfully: (1) locate a patient’s immunization forecast; (2) reconcile immunization history from independent sources; (3) determine which vaccines are due based on the patient’s history and forecast; and (4) document the administration of a vaccine.

NISTIR 7741 (NIST, 2010) identifies the following three primary measures of effectiveness in the context of EHR usability, each of which applies to immunization functionality:

- Percentage of tasks successfully completed (success rate);
- Percentage of tasks achieved per specified unit of time; and,
- Path through which a task is accomplished.

Success is a predetermined benchmark by which a user’s performance on a task is judged. For example, predetermined benchmarks for the completion of a task might be whether the user completed the task, did not complete the task, or completed the task with assistance (e.g., from a moderator, a peer, a manual, or help within the application, etc.). Examples of successful task
criteria for immunization-specific functionality might include opening an immunization forecast for the correct patient, determining the correct vaccines a patient should be given and correctly identifying which ones should not be given, or documenting the administration of a vaccine without errors. If a user’s task performance meets or exceeds the benchmark, then the task is counted a success. If task performance falls short of the benchmark, then the task is not considered a success. Paths of success must be predetermined for each task. Some paths may lead to a “successful” outcome but are none-the-less considered erroneous paths (i.e., when a user completes the task in a way that produces undesired consequences, including documenting with free-text in a field where structured data input is needed to affect quality measures). Evaluators must define acceptable and unacceptable paths to completion in order to assess whether or not a user achieved success on a given task.

2.2 Efficiency

In the context of EHR usability, efficiency is the extent to which a user must expend resources in the process of achieving goals (NIST, 2010; ISO, 2010; and NIST, 2012). Examples of efficiency measures include time spent completing a task, number of clicks to complete a task, and mouse cursor movements used to complete the task. Narrowed to the context of immunization functionality, efficiency might be measured as the time spent completing tasks (NIST, 2010). Hypothetical examples of task times include the following: (a) finding and opening a patient forecast took 45 seconds; (b) determining which vaccines are due took 10 seconds; and (c) documenting administration of a vaccine took 38 seconds. The time it takes to complete a task includes the time the user spends thinking about the software, viewing and thinking about the patient data, and executing the needed interactions with the software (e.g., mouse clicks, scrolling, navigating, entering data into a form, etc.). Time measurements for tasks can be compared against benchmark values. Potential sources for benchmarks include the following:

- Competing software systems with similar tasks;
- Older software systems, prior versions of a product;
- Expert performance using the same product version;
- Target performance standards established by a reputable source; or
- Other systems or design alternatives to which a comparison would be desired.

2.3 Satisfaction

Satisfaction is concerned with the subjective perception of the user rather than objective performance metrics. Such perception includes both negative and positive feelings towards the system (NIST, 2010; ISO, 2010; and NIST, 2012). There are several methods of measuring satisfaction. The System Usability Scale (SUS) is a generally accepted and trustworthy method (NIST, 2010). The SUS is a validated measure of subjective usability perceptions of a system (NIST, 2010; Brooke, 1996). The SUS instrument contains ten statements which the user rates on a 5-point Likert scale from Strongly Disagree to Strongly Agree. The ten statement responses are used to calculate a single overall score from 0 to 100. This score represents the user’s perception of the usability of the system that was rated. More information about the SUS instrument including statement wording, data collection. Scoring can be found in other sources (Brooke, 1996; Sauro, 2013).
3 NIST EHR USABILITY PROTOCOL (EUP) AND USE ERROR

NISTIR 7804, “Technical Evaluation, Testing and Validation of the Usability of Electronic Health Records” (NIST, 2012), established guidelines for an EHR Usability Protocol (EUP) and defined the procedures for design evaluation, and testing of EHR systems to include (a) application analysis; (2) expert review of the user interface by trained human factors experts; and, (3) validation testing of the application user interface. The report highlights that EHR usability is critical in that “usable EHRs have the potential to reduce ‘use errors’ and improve patient care” (p. 7). The report includes errors of omission (e.g., not ordering a vaccine that is due, etc.) and errors of commission (e.g., ordering a vaccine that should not be given, etc.) in the definition of use error. Both types of errors can be consequences of poorly designed user interfaces. The purpose of the EUP is to “encourage user-centered development processes focused on safety by facilitating the design of EHR interfaces with good usability” (p.7). The authors hoped to “encourage system developers to apply human factors best practices and incorporate user-centered design processes into the development and deployment of EHR systems” (p.7). The report includes research findings to illustrate the relationship between usability and patient safety outcomes and a model for understanding this relationship.

3.1 Ongoing Risk Assessment and Hazard Mitigation

This document focuses on the UCD process and the methods used in the process. As mentioned earlier, the UCD process is aimed at optimizing usability. Improved usability should positively impact safety associated with the design and use of EHRs (NIST, 2012). However, the UCD process must include ongoing risk assessment that is aimed at identifying hazards and mitigating them by: (1) designing in such a way that the hazard is removed; (2) guarding against hazards that cannot be designed out; and (3) providing warnings in cases where designing and guarding have not proved sufficient.

Alert fatigue is a known issue across the industry. In some cases, users carefully reviewed the alert. In other cases, users routinely dismissed the alert. A potential risk related to alert fatigue include desensitization to alerts. A common practice is to provide warnings over designing and guarding. Warnings, and more generally feedback, to users about potential patient risks can be provided specific to the level of risk. Types of feedback defined in the context of software usability include (Lowry, 1996):

1. Notification: A notification is a visual clue or displayed message that informs a user but does not require any action (examples include visual clues such as a persistent banner, color change, bolding, etc. (E.g., “This patient is allergic to ___”).

2. Alert: An alert or warning is displayed when user action may result in unintended consequences, for example loss of data, etc. These types of messages shall require the user to acknowledge the message before they can move on (e.g., click OK).

3. Errors: Error message should be displayed in plain language describing specific error condition and instructions what steps a user need to take. This condition requires the user to fix something before they can move on (e.g., fix this date, it is invalid).

Notification in this context indicates that the system provides indication so the user is aware of a potential problem, including the severity of previous reactions. Notification should include information, if available, about prior decisions, such as if a provider documented a reason to
override notification about allergy or diagnosis in the past. The method for fulfilling the requirement is not prescriptive. Examples of notification include visual clues.

## 4 USER-CENTERED DESIGN

NISTIR 7741, “Guide to the Processes Approach for Improving the Usability of Electronic Health Records” (NIST, 2010), describes activities to be carried out during the development of electronic health record (EHR) applications. This standard focuses on methods related to UCD and usability testing.

Exhibit 2 illustrates NISTIR 7741 UCD activities that should be incorporated into a UCD process. These include the following:

- Understand user needs, workflows and work environments;
- Engage users early and often;
- Set user performance objectives;
- Design the user interface applying known human behavior principles and familiar user interface models;
- Conduct usability tests to measure how well the interface meets user needs; and
- Adapt the design and iteratively test with users until performance objectives are met.

The ISO 9241-210 standard, “Human-Centred Design for Interactive Systems,” further illustrates primary activities that should be carried out in iterative fashion until defined usability objectives are obtained (ISO, 2010). The ISO standard includes the following UCD activities that take place between planning the human-centered design process and ending with a design solution that meets user requirements:

---

**Exhibit 2. NISTIR 7741 (NIST, 2010) “Guide to the Processes Approach for Improving the Usability of Electronic Health Records” (Reprinted courtesy of the National Institute of Standards and Technology, U.S. Department of Commerce. Not copyrightable in the United States).**

---
Understand and specify the context of use;
Specify the user (and organizational) requirements;
Produce design solutions to meet user requirements;
Evaluate designs against requirements; and,
Iterate between above activities as needed.

These two standards highlight characteristics of a sound UCD process aimed at achieving the following:

- Involving actual end users early and often;
- Striving to understand the end user’s characteristics, needs, and environment(s);
- Including measurable and observable performance objectives based on user requirements;
- Including activities aimed at formulating design solutions and evaluating those solutions in an iterative fashion; and,
- Including activities aimed at providing objective evidence that performance objectives have been met.

NISTIR 7741 (see Exhibit 2) and ISO 9241-210 provide guidance regarding specific activities suitable at each phase of the UCD process. NISTIR 7804 (NIST, 2012) provides additional guidance for specific activities to help reduce known, predictable, and unnecessary risks to patients. NISTIR 7804 also provides a framework for teams to use to determine whether an EHR, based on its unique performance objectives, can be used in a safe and effective manner as designed.

The basic UCD activities that NISTIR 7741 specifies to include as part of a UCD process are described in the following sections.

4.1 Understand User Needs, Workflows, and Environments

The design of any system cannot be successfully executed without a proper understanding of who will use the system, what it will be used for, how it will be used, and the environment in which it will be used. In the context of EHR immunization functionality, various users and user needs exist. For example, multiple users may need to have access to patient immunization history or look at patient allergies. Providers with prescribing authority need to order vaccine doses. Provider or nurse users need to be able to document historical or administered vaccines. Such user needs should be met by the system in the context of the workflow environment of each user.

Additionally, an understanding of user workflows will aid designers in creating a usable system. Analysis of goals along with steps and sub-steps taken in reaching those goals is essential to understanding how users perform tasks and how EHRs might support those tasks. A workflow analysis should be performed for all relevant tasks, including those which appear simple or mundane.

While systems, including EHRs, are often developed and tested in well-controlled environments, the actual use of such systems takes place in less-controlled environments and has accompanying constraints and characteristics. The nature of the environment has the potential to affect usability of the system and should be considered by designers to maximize the extent to which the system serves users given their specific environmental constraints. Designers should document both common and uncommon environments in which the system is used. Developing an
understanding of the range and type of environments in which the system is used early in the
development cycle is important to inform initial designs and future iterations.

4.2 Engage Users Early and Often

Before creating any initial designs, system designers should work with users to gather information that might support their needs. For immunization workflows, this includes how immunization functionality is structured within the larger EHR system, in addition to those items discussed in Section 4.1. Design teams often begin developing a system prior to observing users in their actual environments and do not obtain input from actual end users of the system. The result is a design that is more likely to mirror a designer’s perceptions of users’ needs, goals, and abilities than those that a user actually experiences. The UCD process involves engaging the user early and iteratively in the product development lifecycle to provide a basis for initial designs, as well as to inform future iterations as the product matures. This process ensures that users’ needs are met from the start. Failure to engage users early in the development process can result in designs that are difficult to change later in the lifecycle in order to make adjustments discovered during later user research.

4.3 Set Performance Objectives

Qualitative data is collected by engaging users in various UCD activities, including formative usability testing. Early research with system users can provide qualitative data that can be transformed into quantitative benchmarks, or performance objectives. These performance objectives can serve as the standard to which the mature system is compared. Performance objectives allow performance data from actual user testing to be compared to the objectives. Effectiveness, efficiency, and satisfaction (discussed in sections 2.1, 2.2, and 2.3) are common quantitative performance objectives. Examples of performance objectives that might be used for EHR immunization functionality are presented in Exhibit 3 (adapted from NISTIR 7741).

<table>
<thead>
<tr>
<th>Task</th>
<th>Effectiveness</th>
<th>Efficiency</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify vaccines that are due</td>
<td><strong>Goal</strong>: 90% correct on first trial irrespective of the path taken by the user</td>
<td><strong>Goal</strong>: Successful completion in 90 seconds by 80% of users</td>
<td><strong>Goal</strong>: All participants give task difficulty ratings of “Easy” or “Very Easy”</td>
</tr>
<tr>
<td>Order inactivated poliovirus vaccine</td>
<td><strong>Goal</strong>: 90% correct on first trial via optimal path</td>
<td><strong>Goal</strong>: Successful completion in 90 seconds by 90% of users</td>
<td><strong>Goal</strong>: All participants give task difficulty ratings of “Easy” or “Very Easy”</td>
</tr>
</tbody>
</table>

1 Note that “performance” here is based on the user’s interaction with the system, and not the system’s performance.
<table>
<thead>
<tr>
<th>Task</th>
<th>Effectiveness</th>
<th>Efficiency</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify any patient allergies</td>
<td><strong>Goal:</strong> 100% correct on first trial irrespective of the path taken by the user</td>
<td><strong>Goal:</strong> Successful completion in 30 seconds by 95% of users</td>
<td><strong>Goal:</strong> All participants give task difficulty ratings of “Easy” or “Very Easy”</td>
</tr>
</tbody>
</table>

**Exhibit 3. Example of User Performance Objectives on Key Immunization Tasks**

### 4.4 Design the User Interface from Existing Human Factors Principles

Along with engaging end user(s) in activities throughout the process, a UCD process involves applying known human factors knowledge. Such knowledge can be gained through proper educational programs and principles included in scientific literature (e.g., Card, Moran, & Newell, 1983; Wickens & Hollands, 2000; Wickens, Liu, & Gordon, 1997). Applying evidenced-based human factors principles to systems design at the start of and during the design and development process will help develop a usable system. Throughout the entire product life cycle, continued application of human factors knowledge provides a method for identifying potential usability issues and solving problems. Problems are more likely to be resolved (e.g., due to feasibility and cost) if they are discovered earlier in the design and development. This encourages organizations to develop safe and usable systems. Human factors knowledge and principles can be applied to inform the design of EHRs, including design of immunization functionality capabilities.

### 4.5 Test and Evaluate

Usability testing serves multiple roles in a UCD process. Two types of usability testing are **Formative** and **Summative/Validation** testing, each of which serves different purposes (NIST, 2010).

#### 4.5.1 Formative Testing

Formative usability testing is an evaluation method that helps to “inform” and “form” the product design. Participants in formative usability test sessions should be representative of target end users on key criteria. For example, an evaluation that tests whether users can interpret an immunization forecast screen should include participants who perform that workflow as part of the job. Similarly, usability tests of tools for entering immunization administrative information should include participants who actually perform this task in their work. Test participants should not be vested in the project team (e.g., advisory clinicians who no longer see patients). Participants should not include prior end users who have moved to other roles where they no longer perform the task (e.g., administrator, informaticist). Test sessions should be moderated by a usability specialist. Formative test sessions may be highly exploratory (e.g., during the design phase) where basic design concepts are presented and the test participants are asked to provide feedback. Formative testing may also be more advanced and task-based where test participants are asked to perform representative tasks on actual and/or prototype products (e.g., during late design phase or development phase). Formative usability testing should be performed iteratively during design and development phases to identify and resolve issues associated with the user interface design that may impact efficiency, effectiveness, and satisfaction.

#### 4.5.2 Summative Usability Testing

Summative usability testing is a validation of a product’s usability in the context of how it is intended to be used in its target environment. Representative users take part in summative test
sessions where they perform representative tasks under realistic conditions on final or near-final products. The results from a summative usability test serve as objective evidence of the product’s usability (i.e., effectiveness, efficiency, and satisfaction) as well as evidence of how the product might support or not support safe use. Usability testing is not equivalent to user acceptance testing, where a separate testing team follows step-by-step scripts to identify system defects. Usability testing focuses on identifying potential usability issues by exposing the product design to end users and asking them to use the system as they would in reality.

4.6 Adapt and Iterate Design with Users until Performance Objectives are Met

The evolutionary, iterative design of the EHR functionality culminates in summative usability testing that is designed to assess whether the system meets the predefined performance objectives set by the team. If the system does not meet the performance objectives, then further adaptations are made and retesting is performed. Once the system meets performance objectives, the UCD process is complete (NIST, 2010). The likelihood of meeting performance objectives increases the earlier UCD is engaged in the design and development process. Addressing UCD early provides greater opportunity for testing processes to identify problems that can be solved by the design/development team and also allows a greater number of iterations. Organizations must make the decisions and take necessary actions (e.g., secure budget and resources) prior to the start of the UCD process so that UCD activities can be carried out from the very beginning of the process (e.g., during discovery and requirements definition).

5 USER-CENTERED DESIGN METHODS

As described earlier, a UCD process aims to optimize usability. Every product and process is different, even in same or similar domains (e.g., EHRs). Prescriptive guidelines cannot address all aspects of design, design details, and development for any or all products. This means that organizations must rely on their own usability maturity (HIMSS, 2011) and the design process itself – a sound UCD process – to achieve usability outcomes. Teams should include a human factors expert who can guide the selection and execution of the right method at the right time in order to optimize for usability.

Exhibit 4 illustrates some of the methods used throughout the different stages of the UCD process. The following sections then describe some of the methods that can be used in a UCD process. These are separated into methods that are typically used early in the UCD process, in the middle of the UCD process, late in the UCD process, or post UCD process.
5.1 Methods Used Early in the Process

As shown in Exhibit 4, the Discover and Define phases take place early in the UCD process. Methods used during this part of the process include Observation of Users, Focus Groups, User Interviews, Stakeholder Interviews, Online Surveys, Task Analysis (including Cognitive Task Analysis), Task Mapping, and Card Sorting. These are described in the following sections.

5.1.1 Field Observation

5.1.1.1 What is Field Observation?

Field Observation involves going directly into the environments where product end users work and observing them as they work. Observers gain information that does not rely on interpretation that has been made by others and translated into written (or other) descriptions or requirements. Observers watch end users as they interact with the product or system to perform their work. Observers also pay attention to other systems and individuals that the end users interact with that may impact the user’s interaction with the system. Observation of users in the field is one of the primary UCD methods for gaining a robust understanding of what users really do, who they interact with, what they use to aid them, and what obstacles they face. What an end user says s/he does to perform a task does not always match what s/he actually does while operating in their actual environment. This does not mean that end users intentionally misrepresent what they do. It is more often the case that end users have become experts at what they do. Therefore, they leave out important steps when describing what they do because they may not think about them (e.g., because they have become so automatic) or to the user they might seem unimportant. Users also may leave out interactions with other artifacts (e.g., cheat sheets, manuals, reference guides, “how to” documents written up by a staff member, etc.) or other user roles (e.g., coworkers, help desk) in the system. However, these are extremely important to note as they can be an indication of shortcomings in the system (e.g., the software is not user friendly so a cheat sheet is taped to the computer to help users of the software understand how to use it).

Observing users in the field requires that observers be very sensitive to several key areas including: (1) keeping confidentiality of organizations (e.g., healthcare practices, hospitals); (2)
maintaining respect and confidentiality for others in the site environment (e.g., patients); (3) being as unobtrusive as possible in order to minimize impact on the ability of users to perform their work; and (4) minimizing the number of visitors during a given visit (especially in physically small spaces like doctors’ offices).

Field observations rely on observation of end users without interacting with them. They may also include properly placed times and situations for supplementing the observation with interview questions. This is known as contextual inquiry. This allows observers to ask questions about “why” a person may have made a certain decision or performed a specific action. Interviewers must insure questions are appropriately timed so that they interfere as little as possible in the work and so that the person responding to the questions can do so without others (e.g., patient, manager) hearing their responses which may present a problem (e.g., complaints about the system that may reflect negatively on the user).

5.1.1.2 When Should Field Observation be Used?

Field Observations are typically performed early in the UCD process during the Discover phase. NISTIR 7741 states that “in the UCD process, design begins in the field by learning about users, their goals, and their environments long before any screens are built” (NIST, 2010, p. 26). Field observations should be performed early in the UCD process and include multiple observations to ensure that representative sites (e.g., family practice, specialist, health department) and end users (e.g., physicians, nurses, administrative staff) are observed. Field observations are also useful during other phases of the UCD process. For example, as the team begins to define the requirements and design the product, field observations are helpful to validate what the team has learned about user goals and context that informed those requirements and design. Field observations are also extremely helpful for onboarding new team members. Team members can gain insight into the product(s), workflows, and use environments very quickly by seeing users in the context of their work and work environments.

5.1.1.3 What are the Benefits of Field Observation?

The primary output of field observations is knowledge that the observers gain by seeing the users in the context of their work. Observers typically take notes during site visits and share important findings with other members on the team. The team may decide to define one or more structured templates for capturing notes or other data (e.g., artifacts, estimates of task time) during the visits.

5.1.1.4 References

References for further information include:

- NISTIR 7741: Guide to the Processes Approach for Improving the Usability of Electronic Health Records. (http://www.nist.gov/manuscript-publication-search.cfm?pub_id=907313);
- http://www.usabilitynet.org/tools/userobservation.htm; and,
5.1.2 Focus Group

5.1.2.1 What is a Focus Group?
A focus group is a guided discussion where a moderator leads an interactive group of representative users through a series of questions and activities focused on specific product or topic.

5.1.2.2 When Should Focus Group be Used?
Focus groups are typically performed early in the product or website planning process to inform requirements definition and design. Focus groups are very useful for gaining a better understanding of user opinions, preferences, attitudes, and reactions to a particular topic, concept, or prototype.

5.1.2.3 What are the Benefits of a Focus Group?
The output of a focus group typically includes findings and recommendations that inform requirements, feature offerings, and/or the design of a new or available product or website. Outputs also include insight into who the users are, their characteristics, environments of use, and how the product or website fits (or does not fit) into their lifestyle.

5.1.2.4 References
References for further information include:

5.1.3 User Interviews

5.1.3.1 What is a User Interview?
A user interview is a semi-structured or structured interview that may be conducted in person or over a phone to collect in-depth information about user needs, goals, experiences, attitudes, and opinions.

5.1.3.2 When Should User Interviews be Used?
User interviews are typically performed at the beginning of the UCD process during the discovery and definition phases. However, interviews may be used during any phase to gather user input on specific questions that the team needs to answer about the product. For example, interviews might be used to obtain input from users on attitudes and preferences for early design concepts, to obtain feedback on prioritization of information that is displayed in prototypes, or to gather post-launch feedback on whether a product is meeting user needs. Qualitative and quantitative data can be gathered with user interviews. User interviews may be conducted with few too many users. Robust, useful data can be gathered from small sample size interviews and used to help define requirements and designs. Results from small sample interviews are not likely to be generalizable to the larger population of users. However, they may be used as a starting point for defining studies to be conducted with larger sample sizes of users. For example, a small sample study might be used to better define the demographics of the user population and to understand the product domain and user terminology to aid in forming interview questions.

5.1.3.3 What are the Benefits of User Interviews?
Results of small or larger sample interviews can be used to inform the requirements and design. The output from a user interview typically includes a report that details:
The primary goals of the interview;
Questions asked to achieve those goals;
Participant demographics of those who were interviewed;
Descriptive statistics for the quantitative data collected for ratings, rankings, or other closed-ended questions;
Graphs of quantitative data (e.g., means, frequencies, rankings, etc.);
Qualitative findings (e.g., common themes or trends from open-ended questions); and
Recommendations based on findings.

If desired, user interviews can be recorded and/or transcribed. If the interview is recorded, then the interviewer should inform the interview and obtain consent.

5.1.3.4 References
References for further information include:

5.1.4 Online User Surveys
5.1.4.1 What is an Online User Survey?
An online user survey is similar to a structured interview with users but it is completed online and without an interviewer. A list of questions is displayed online and each participant’s responses are recorded.

5.1.4.2 When Should Online User Surveys be Used?
Online user surveys can be used anytime during the UCD process or beyond. They offer a fast and affordable way to collect large amounts of data from a large number of participants. Online surveys should have very specific questions that need to be answered. These may include open-ended questions and questions with fixed response options. Online surveys can be used to collect demographics, opinions, preferences, and behaviors as well as feedback on why a participant provided a given response.

5.1.4.3 What are the Benefits of Online User Surveys?
Results can be used to help inform product requirements and design based on the responses that were obtained. Online user surveys can be used to better define the user population. They can also provide great insight on users’ opinions and overall impressions. Online surveys can be used to provide benchmark values for measures of interest. For example, the same online user survey can be used repeatedly to track opinions over time.

5.1.4.4 References
References for further information include:

5.1.5 Stakeholder Interviews
5.1.5.1 What is a Stakeholder Interview?
A stakeholder interview is a semi-structured or structured interview conducted with individuals who are not end users, yet whom have a “stake” in the project. The interviews may be conducted
in person, over the phone, or online. Stakeholders include business owners, product owners, infrastructure representatives, and others who can help inform the project team about business needs, scope, boundaries and constraints. Stakeholders may also include subject matter experts (SMEs) who are assigned to the development team in some capacity. SMEs can provide expert opinions and answer questions related to vocabulary, functionality, experience with the product, and early design concepts as needed by the design team (e.g., to prepare early design concepts for testing).

5.1.5.2 When Should Stakeholder and SME Interviews be Used?
Stakeholder interviews are often performed early in the UCD process during the discovery phase. However, they can be used at any point during the process to assist the project team as needed.

5.1.5.3 What are the Benefits of Stakeholder Interviews?
Results can be used to help inform product design and project boundaries and scope. Results may be similar to those described earlier for User Interviews and Online User Surveys.

5.1.5.4 References
References for further information include:
- see References provided earlier for User Interviews and Online User Surveys.

5.1.6 Task Analysis and Cognitive Task Analysis

5.1.6.1 What are Task Analysis and Cognitive Task Analysis?
A task analysis is a breakdown of the tasks and subtasks required to successfully operate a system. Task analysis is typically performed by one or more usability/human factors experts. The analysis is aimed at determining the tasks and subtasks and allocating them to the system (e.g., computer, machine, other product, etc.) and the human user, based on which is the best part of the system to perform that task. For example, computers are good at repetitive, very well defined mathematical computations while humans are better at complex problem solving tasks.

A cognitive task analysis is appropriate for situations where there are large mental demands as opposed to, or in addition to, physical demands on the user. The cognitive task analysis is aimed at understanding and defining the breakdown of mental processes and demands involved in completing these tasks. These include such things as perception, information processing, decision making, memory, and attention. Cognitive task analysis can be informed through other user research methods, such as user interviews and card sorting, although this is not always the case.

5.1.6.2 When Should Task Analysis and Cognitive Task Analysis be Used?
Task analysis and cognitive task analyses are typically performed at the beginning of the UCD process during the definition phase. The results of task analysis can help inform design (e.g., prioritization and layout of information on a display) and training (e.g., defining step-by-step instructions for completing a task within a system). It can also inform other activities that are carried out during the user-centered design (UCD) process, including structured interviews, formative testing, and summative testing.

5.1.6.3 What are the Benefits of Task Analysis and Cognitive Task Analysis?
The output of a task analysis/cognitive task analysis is typically a report that details:
A list of the tasks and subtasks embedded within the user-system interaction;

- Associated cognitive demands posed on the human (e.g., decision making, memory, etc.); and
- Recommendations for addressing the demands posed by the system on the user.

Additional details that are reported when appropriate include:

- A breakdown of the psychological demands (e.g., perception, attention, memory, decision making, shared or divided attention among multiple tasks, vigilance, etc.) on expert and novice users (or other categories of users).
- Recommendations for which tasks should be performed by the system (e.g., computer) and which should be performed by the human user. This is often based on what each is “good” at doing. For example, computers are typically “good” at doing things like calculations, searching through information, simple repetitive tasks, and auto-populating high probability default values into entry fields) while humans are better at thinking, problem solving, final decision making, personal communication, and so forth. The results might also include considerations related to work load (e.g., which resource is free to do the task vs. which resource is overloaded).
- Graphical task flow diagrams (e.g., that outline the subtasks required to complete a task with visual indications for which subtasks are allocated to the user(s) and which tasks are performed by other components in the system).
- Criticality ratings for each task.
- Potential sources of error or failure (which can be fed into a formal failure modes and effects analysis).
- Estimates of mental workload associated with each task.

Exhibit 5 illustrates a sample artifact from a cognitive task analysis that the team performed to address some of the immunization-related EHR capabilities identified as part of the Project. The artifacts produced from a cognitive task analysis are used to inform requirements and design activities.
5.1.6.4 References

References for further information include:

- [http://www.usability.gov/how-to-and-tools/methods/task-analysis.html](http://www.usability.gov/how-to-and-tools/methods/task-analysis.html); and,

5.1.7 Task Mapping

5.1.7.1 What is Task Mapping?

A task map, also referred to as a user story map, is a visual depiction of the tasks and subtasks users might perform to complete a workflow in a given system (see Exhibit 6). Task maps can be constructed with pen and paper or electronically. Typically, task mapping involves identifying key tasks that users might perform to complete one or more workflows and the steps required to complete those tasks. The task map visually illustrates the temporal order in which tasks are performed. Task maps are often informed by earlier user research methods such as user observation, interviews, and task analysis. Input from qualitative sources such as observation, interviews, and surveys can help guide what tasks a user would or would not perform and the order in which they would be performed. Observations of users performing relevant tasks are particularly informative. Task maps might also focus on mismatches between tasks that are currently being performed, the skills or characteristics that are best suited to perform them, and who/what is assigned to currently perform them.

5.1.7.2 When Should a Task Map be Used?

Task mapping is usually performed in or near the beginning of the UCD process during the definition phase. Task mapping can inform early design wireframes and prototypes as well as the development of tasks for formative and summative usability tests. Task maps are more complete and effective if the task mapper has sufficient knowledge of user workflows and needs.
5.1.7.3  What are the Benefits of Task Mapping?

The output of a task map is a physical or electronic visual representation of how the user will accomplish tasks in the system along a timeline. Task maps can vary, but some aspects of task maps include:

- Graphical diagram of tasks a user would undertake;
- Tasks organized along a horizontal timeline;
- Subtasks listed underneath primary tasks;
- Important details associated with the task;
- Artifacts that may be consumed or produced during a specific task;
- Interactions required with other personas or systems;
- Alternative tasks (when completion of a task could involve multiple independent paths);
- Open-ended notes, comments, and questions to be answered; and,
- Needed or current assignments for completing the task.

5.1.7.4  References

References for further information include:


5.1.8  Card Sorting

5.1.8.1  What is Card Sorting?

Card sorting is a usability evaluation method used to design and/or evaluate the information architecture for a product or website. Card sorting can also be used to identify logical groupings for toolbars and menus. During a card sort, representative users are provided cards containing items to sort into meaningful groups or categories.

There are three types of card sorting methods:

- Open Card Sort – Representative users to sort cards into groups that they label.
- Closed Card Sort – Representative users sort cards into pre-labeled groups.
- Inverse/Reverse Card Sort or Tree Testing – Representative users are asked to find specific information or topics that have already been sorted into labeled groups.

5.1.8.2  When Should Card Sorting be used?

Card sorting is typically performed early in the UCD process during the requirements definition phase when the information architecture is still being developed. It can also be used during the design phase to refine how information is grouped or labeled (e.g., in a navigation menu). It is sometimes used later in the UCD process (e.g., during the development phase) to identify if there are usability issues associated with how the information has been grouped or labeled (e.g., to inform further refinement of the information architecture).

5.1.8.3  What are the Benefits of Card Sorting?

The output of a card sort typically includes findings and recommendations associated with groupings and labels of information. The output can be used to define or validate the information architecture of a product or website and improve discoverability of information.

5.1.8.4  References

References for further information include:
5.2 Methods Used in the Middle of the Process

As shown in Exhibit 4, the Design and Develop phases take place in the middle of the UCD process. Methods used during this part of the process include Iterative Design, Heuristic Evaluation, and Formative Testing. These are described in the following sections.

5.2.1 Iterative Design

5.2.1.1 What is Iterative Design?

Product design is an iterative activity. Design ideas and concepts are formed early in the UCD process “design” phase (see Exhibit 4). Designs are based on the user requirements that were produced during the definition phase of the UCD process. Activities aimed at creating and communicating design include sketching, wireframe development, creation of design comps, and prototyping. Wireframes and prototypes can be used in formative usability testing to gather user feedback and uncover potential usability issues with the design(s). Exhibit 7 illustrates an example of low fidelity wireframes that might be used in a usability test. The design activities result in screen layouts and screen flows that support workflows (e.g., those needed to provide immunization capabilities in EHRs).

Exhibit 7. Example of Wireframe.

5.2.1.2 When Should Iterative Design be Used?

Iterative design is not a single activity. It is a continual activity that typically occurs throughout the middle of the UCD process. Changes to the design are based on results of other UCD activities (e.g., expert heuristic evaluation, formative usability testing with end users, interviews, etc.). Teams should continue to place emphasis on activities that engage end users to gather feedback on the design(s). Qualified usability specialists should lead UCD activities and review the findings from those activities in order to identify potential usability issues and provide mitigation recommendations based on human factors and design principles. The usability
specialist(s) should discuss those findings as needed with others (e.g., designers, developers) on
the team to determine and prioritize the needed design updates. This activity often includes
making decisions that consider a variety of factors including the following: usability, feasibility
(e.g., considering technological or platform limitations), development effort (e.g., hours required
and calendar time needed), and support of other objectives (e.g., product portfolio releases).

5.2.1.3 What are the Benefits of Iterative Design?
The output from iterative design that includes UCD activities and user feedback is a product that
is likely to better meet user needs. The earlier the UCD activities begin, the more likely a team
will be able to implement design changes when usability issues are discovered.

5.2.1.4 References
References for further information include:
- http://www.usability.gov/how-to-and-tools/methods/wireframing.html; and,

5.2.2 Heuristic Review

5.2.2.1 What is a Heuristic Review?
A heuristic review is a usability evaluation method where one or more trained usability/human
factors experts review a product (e.g., hardware or software user interface, website) and compare
it against established human factors criteria and design principles to identify potential usability
issues.

5.2.2.2 When Should a Heuristic Review be Used?
Heuristic reviews can be conducted at any time during the UCD process. However, the earlier a
heuristic review can be performed, the more likely it is that issues identified during the
evaluation can be resolved prior to product release.

5.2.2.3 What are the Benefits of Heuristic Review?
The output of a heuristic review typically includes a list of potential usability issues. These are
often categorized and assigned a severity rating based on perceived usability risk and overall
impact on the user experience. Actionable and prioritized recommendations for resolving or
mitigating potential usability risks are provided for the issues that are identified during the
heuristic review. Results often include visual illustrations of the issues (e.g., annotated screen
captures from the application with callouts of the issues). The results of the heuristic review are
presented to product teams and are used to facilitate discussion, track status, add comments, and
assign priorities.

5.2.2.4 References
References for further information include:
- NISTIR 7804: Technical evaluation, testing, and validation of the usability of electronic
  health records. (http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909701);
  and,
5.2.3 Formative Testing

5.2.3.1 What is Formative Usability Testing?

Formative usability testing is a task-based evaluation that involves end users that helps to “form” the product design. Formative testing is used to identify and resolve issues associated with the usability of the system. Representative end users take part in exploratory test sessions where they are asked to perform representative tasks on actual and/or prototype products. Formative usability testing typically involves sessions where end users carry out critical tasks and frequently executed tasks (e.g., to support immunization workflows) so that potential usability issues can be discovered and addressed in the design. Formative testing allows evaluation of identified usability goals (e.g., measures of efficiency, effectiveness, and satisfaction) of wireframes, prototypes, and applications so that improvements can be made to the wireframes, prototypes, and applications to improve performance on the identified usability goals.

5.2.3.2 When Should Formative Usability Testing be Used?

Formative usability testing is typically initiated during the design phase (e.g., using prototypes) and then performed iteratively during the design and development phases (e.g., with advance prototypes or actual product). Usability issues can be identified as part of an initial formative test, design changes can be made to resolve or mitigate those issues, and the product can then be retested as part of a second formative test to ensure the usability issues have been addressed appropriately. Iterative usability testing provides the opportunity to identify issues, correct issues, and continue to identify issues as the design is evolved.

5.2.3.3 What are the Benefits of Formative Usability Testing?

The output from a formative usability test typically includes documentation of the findings for each task and recommendations for resolving or mitigating usability issues that were identified in the design. These may be provided in a presentation or report format. Objective measures of performance (e.g., task times, number of actions, number of errors, number of mouse clicks) and subjective measures of satisfaction (e.g., System Usability Scale (SUS) scores, satisfaction ratings, task difficulty ratings, preference ratings or rankings) may be collected during formative testing. These can be used to establish target success criteria for summative testing. Additional outputs from formative testing may be produced depending on the needs of the design and development teams. For example, these might include (1) a prioritized spreadsheet file that contains a categorized and prioritized list of the issues that can be used for assigning and tracking actions; (2) entry of issues into a defect tracking database; or (3) updates to the project’s UCD tracking dashboard. Exhibit 8 illustrates an example of findings and recommendations from a usability test.
5.2.4 References

References for further information include:


5.3 Methods Used Late in the Process

As shown in Exhibit 4, the Beta testing and Launch phases take place late (near the end) in the UCD process. The primary UCD method used during this part of the process is Summative (or Validation) Testing. This method is described in the following section.

5.3.1 Summative/Validation Testing

5.3.1.1 What is Summative Usability Testing?

Summative testing is a comprehensive usability evaluation of a product in the context of how it is intended to be used in an actual use environment. Representative users take part in summative test sessions where they are asked to perform representative tasks under realistic conditions on final or near-final products. Summative/validation testing involves planning and carrying out critical tasks and frequently conducted tasks to evaluate the efficiency, effectiveness, and satisfaction of wireframes, prototypes, and applications (e.g., to support immunization workflows) in order to demonstrate how a team provides objective evidence that designs meet identified usability goals.

5.3.1.2 When Should Summative Usability Testing be Used?

Summative testing should be performed near the end of the product development phase on a final or near-final product. The goals of summative testing are to establish benchmarks for efficiency, effectiveness, and satisfaction and to determine if the product has successfully met usability goals or success criteria that were established during formative testing.

5.3.1.3 What are the Benefits of Summative Usability Testing?

The output from a summative test typically includes a report that describes the detailed methods and results of the test. The methods include a description of the test design, location and
environment, tasks, procedures, materials used (e.g., forms), measures collected, participant demographics, and instructions that were provided to participants during the test sessions. The results describe the findings for each task. These include performance summaries, a determination of whether or not usability goals or success criteria have been met for the product, a use error analysis discussing the results organized around a risk analysis of use, mitigation strategies for observed use errors, and recommendations for next generation releases or refreshes based on human factors principles. The report should also include a thorough description of the test methods, participant demographics NISTIR 7742 (NIST, 2010) provides a common industry format report template for documenting a usability test. See Exhibit 9 for the table of contents of what is included in the report.
Exhibit 9. Table of Contents from NISTIR 7742 Common Industry Format Usability Test Report (NIST, 2010)

5.3.1.4 References

References for further information include:

- NISTIR 7804 (NIST, 2012). Technical evaluation, testing, and validation of the usability of electronic health records. ([http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909701](http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909701)); and,
6 REFERENCES


