



Usability

Benchmarking

Study Planner Workbook v1.0

Version 1.0
Tyrone Austen
February 26, 2010

Part 1. Table of Contents

Part 1.	Table of Contents	2
Part 2.	Usability Benchmarking Overview.....	4
Part 3.	PLAN	5
A.	Definitions.....	5
B.	System Selection	5
C.	Purpose	5
D.	System User Selection.....	6
E.	Scenario Creation	7
E.1	System Feature Mapping	7
E.2	Scope of System Functionality to be Tested.....	7
E.3	Scenario Selection or Creation	7
E.4	System Automation Task Set Development (optional)	8
F.	Test Environment Selection	9
F.1	Testing environment location	9
F.2	UB Testing Equipment	9
G.	Selection of the study metrics	10
G.1	System Benchmark Metrics	10
G.2	Usability Metrics (Coding scheme).....	10
H.	System User UB Study Training	11
Part 4.	DO	11
A.	UB System Testing	11
A.1	Normal Condition System Tests.....	11
A.2	Think Aloud System Tests	12
A.3	System Automation Testing (optional)	13
Part 5.	STUDY	13
A.	Data Compilation and Analysis	13
A.1	Formatting the Test Recordings.....	13
A.2	Normal Condition Scenario Compilation and Analysis.....	16
A.3	'Think Aloud' Scenario Compilation and Analysis.....	17
A.4	System Automation Compilation and Analysis	18
A.5	Interview Compilation and Analysis	19
A.6	Comparative Analysis of Conditional System Tests	19
Part 6.	ACT	20
A.	Courses of Action for UB Deliverables.....	20
A.1	System Benchmark Content	20
A.2	Usability Testing Content	20
A.3	System Automation Content	21

A.4	Interview Response Content	21
A.5	Comparative Analysis of Conditional System Tests Content	21
Part 7.	Bibliography.....	22
Part 8.	Appendices.....	24
Appendix 1 –	System User Selection Questionnaire	25
Appendix 2 –	UB Test Equipment Costs and Recommendations	26
Appendix 3 –	Remote Laboratory Setup	29
A3.1 -	Setup for a stationary system	29
A3.2 -	Setup for a portable system	29
A3.3 –	Remote Laboratory Setup Figures	30
Appendix 4 –	Coding Scheme	32
A4.1 –	Example Coding Scheme	32
A4.2 –	Annotated Transcript Example	34
Appendix 5 –	Definitions	35
Appendix 6 –	Example Scenario (for an ePrescription System)	36
Appendix 7 –	Excerpt Example of a Scenario Feature Tracking Table	37
Appendix 8 –	After Scenario Questionnaire (ASQ).....	38
A8.1 -	Instructions	38
A8.2 –	ASQ	38
Appendix 9 –	Post Study System Usability Questionnaire (PSSUQ)	39
A9.1 -	Instructions	39
A8.2 –	PSSUQ	39
Appendix 10 –	System Automation Task Set Example	45
Appendix 11 –	UB Study Subject Reference Sheet.....	46

Part 2. Usability Benchmarking

Overview

Usability Benchmarking (UB) studies capture data pertaining to a system's performance and usability (learnability, ease of use, efficiency, safety, and effectiveness) by recording the actions taken by system users during the fulfillment of realistic use case scenarios. These recordings can then be analyzed to produce quantitative system use benchmarks (i.e. the amount of time required to complete a system task) and system usability statistics.

During Usability Benchmarking studies, system users are tested and recorded over two phases of use case scenario testing (normal condition testing and 'Think Aloud' testing). For each tested user, the user is first recorded completing the normal condition scenarios under conditions similar to real practice. After each normal condition scenario test, the user fills out a brief usability questionnaire (the After Scenario Questionnaire or ASQ). Once all normal condition tests are completed, the user fills out a second type of usability questionnaire (the Post-Study System Usability Questionnaire or PSSUQ).

The second phase of user testing involves having the user complete the 'Think Aloud' scenarios, while verbally providing information about their current actions (thinking aloud).

The normal condition and 'Think Aloud' test data is collected for each system user selected for the study. An optional third phase of usability benchmarking, system automation testing, requires an additional round of tests to be conducted by a single system user. During system automation testing, the tested user completes a series of system actions that are designed to elicit automated system responses (i.e. warning messages, calculations, etc.). The system responses are then recorded and checked for appropriateness.

The audio/video recordings and questionnaire results from the normal condition, 'Think Aloud', and (optionally) system automation tests are then coded and analyzed by a research team to produce:

- System benchmarks that can be used for evaluating and comparing system performance (i.e. task completion time, number of actions required to fulfill a task, etc.);
- A list of system usability problems and errors (that were encountered by the users during the tests);
- Comparisons between the normal condition and 'Think Aloud' system test flows.

The following document is intended to walk study designers through the stages of planning and implementing a UB study. The overall UB study design follows the rapid Plan-Do-Study-Act (PDSA) cycle, which is based on the work by Shewhart and Deming (Shewart & Deming, 1939). UB study designs are iterative in nature. The results from each PDSA cycle can be used to improve the system under study, and once these improvements are made, another PDSA cycle can be carried out to further evaluate and improve the modified system.

Part 3. PLAN

A. Definitions

Throughout the UB study, there may be a need to use uncommon words (i.e. industry specific jargon, acronyms, etc.). To help alleviate any confusion brought on by these terms, a running dictionary of uncommon words, phrases, and acronyms is to be kept.

Specify all uncommon, industry specific terms and acronyms here:

B. System Selection

The first task in a Usability Benchmarking study is to select the system that will be evaluated.

Specify the name and version of the system under study, as well as the system's development company:

C. Purpose

UB studies can serve four primary purposes:

- To establish health information system use benchmarks.
- To capture usability issues in the process flows of health information system users.
- To compare the differences between the two tested conditions of system use (normal, and 'Think Aloud')
- To test the system's automated responses.

The choice of which UB study goals are included in an individual study design is dependent on the audience that the study is designed for. There are four classes of UB study audiences that can stand to benefit from Usability Benchmarking results:

- Health Information System Procurers – can utilize UB study results to evaluate systems
- Health Information System Administrators – can utilize UB test data to determine user groups or system components of interest for special training sessions

- Health Information System Researchers – can utilize standardized system benchmark measures: to compare and contrast health information systems of the same type (i.e. for system procurement, etc.) and to learn about how system user process flows change under different testing conditions (normal vs. 'Think Aloud').
- Health Information System Designers – can utilize the various system benchmarks and usability results as indicators regarding how to improve their systems.

Specify which of the four primary UB goals will be achieved in this study and for which stakeholders will these goals be aimed towards:

D. System User Selection

Once the system has been selected and the study's scope has been defined, the next step in a UB study is to select a system user. Past usability studies have indicated that a sample of 8-10 subjects is sufficient (Nielsen, 1993), (Rubin, 1994), unless inferential statistics are to be used, in which case the sample should be increased to 15-20 subjects.

A UB study sample must be representative to the user population from which it's drawn. This is ideally achieved through random sampling from the general user population; however, limiting factors, such as study resources or the availability of user subjects may confine the study to convenience sampling techniques. For convenience sampling, the study subjects are selected based on availability for the study. To increase the validity of such non-random samples, the subjects can be selected based on user profiles (where users are selected from each category of profiled user, i.e. beginner user, novice user, expert user, etc.). The questionnaire found in Appendix 1 provides questions that can be given to a convenience-based user pool to determine relevant categories of users for the study.

The sampling technique used for the study is the following:

Provide the following details about each of the selected system users:

Name:

Age: Gender:

Educational Background:

Occupational History (relevant to the system under study):

How long has the user been using the system under study?

On average, how many hours per day/week/month are spent on the system under study by the user?

E. Scenario Creation

E.1 System Feature Mapping

The scope of what will and will not be evaluated during a UB study needs to be clearly defined. In order to determine which system tasks are to be performed by the test subjects, it is first required that the system's functionality is classified. For this, it is recommended that a previously established system feature checklist is used. An example of a system feature checklist for an EMR system can be found on ACP Online (Carter, 2004): <http://www.acpinternist.org/archives/2004/04/emr/cklist.pdf>.

Has the functionality of the selected system been defined?

Yes

E.2 Scope of System Functionality to be Tested

From this feature list, the analyst can then determine which of the system features will and will not be included in the investigation. For example:

System Features Included in the Study	System Features Not Included in the Study
Prescription ordering	Patient lookup
Medication error checking	Scheduling
...	...

Has the scope of what system features are/are not to be tested been clearly defined?

Yes

E.3 Scenario Selection or Creation

Scenarios are story like descriptions of working cases that can be used to guide and provide working context during simulated tests. Scenarios help to direct the user into utilizing the features selected for investigation in the study (as opposed to having to wait for the user to utilize such features naturally, in an ethnographic-type study).

An example UB scenario for an ePrescription system study can be found in Appendix 6.

Scenario development can be a lengthy process, so whenever possible, previously developed scenarios that test the selected system functionality should be used. If needed, the readymade scenarios can be customized to better meet the needs of the given study. For a complete scenario set for UB testing of ePrescription systems, see Usability_Benchmarking-Scenarios.docx.

If the scenarios must be developed from scratch, there are a number of considerations to keep in mind:

- Use case scenarios should be developed to be as close to the system user's normal working conditions as possible. In most cases, this can be challenging, for the study designers are not always experts in the domain of the system's application. For this reason, scenario development is an iterative and collaborative process between the study's designers and domain experts.
- High quality scenarios are concise, meaning they only test a select few system features at a time, and precise, meaning they test only the system functionality that was selected for the study and no more.
- To help build user confidence and comfort during system testing, the scenarios should be developed to increase in complexity, which will allow the users to start testing with basic scenarios and gradually build up to more complex cases.
- When developing the scenarios, keep record of which system features are tested by each scenario. This will aid with later analysis efforts. An example of a feature tracking table can be found in Appendix 7.

Two different sets of scenarios are written during scenario development: the normal condition and the 'Think Aloud' scenario sets. Both the normal condition and the 'Think Aloud' scenario sets test the same system features (those selected in part 3, section E.2) with the same scenario breakdown (what features are tested in which scenarios). For example, scenario #1 for both the normal condition and 'Think Aloud' scenario sets could test writing a basic prescription, scenario #2 for both sets could test issuing a prescription renewal, and so on. The parallelism in tested functionality between the two testing condition sets allows for comparisons to be made between the two testing conditions. The scenario stories and contexts, however, must be different enough between the two conditional sets to prevent scenario recognition by the tested system users, which has the potential to change users' normal process flows.

Have the scenarios been selected or created?

Yes

Have the scenarios been verified by domain experts for accuracy?

Yes

E.4 System Automation Task Set Development (optional)

An automated system feature is one that does not require user interaction. To test system automation, the system must be provided with the necessary conditions to trigger the automated responses. The system responses can then be tested for correctness. If, in part 3, section E.2, any automated system features were selected, a system automation task set, which creates these automation triggers, is to be developed. An example of a system automation task set from an ePrescription system UB study can be found in Appendix 10.

Are there any system features that need testing that are automated?

Yes

If so, have task sets been developed that create the triggering conditions for these automated features?

Yes

F. Test Environment Selection

F.1 Testing environment location

Usability Benchmarking studies are performed in remote laboratories created in the system users' workplaces. Remote laboratories utilize modern, portable computing and recording equipment to test systems in their natural environment. Using the tested system's natural setting greatly improves the test's ecological validity and it improves the comfort level of the tested system users.

Record the location of the remote laboratory where the UB test will occur:

F.2 UB Testing Equipment

Over the past several years, the equipment required for conducting UB studies has become more and more portable, powerful, and cost effective. This has made the 'Portable Discount Usability' (Kushniruk & Patel, 2004) approach far more feasible to conduct. For a complete list of UB test equipment and software recommendations, see Appendix 2. For details on setting up the remote laboratory for the UB system test, see Appendix 3.

Complete the following UB test equipment checklist to assure that all required equipment is secured for the test:

Equipment/Software	Purpose	Acquired
Computer (desktop or laptop)	Runs the system under study and/or the screen capture and video analysis software.	<input type="checkbox"/>
Extra laptop battery or AC adapter	If a laptop is used for the study, it is recommended that extra power supplies are brought to ensure the life of the computer throughout the study period.	<input type="checkbox"/>
Computer microphone	Captures the audible (i.e. 'Think Aloud') recordings. Preferably a headset (to improve sound quality)	<input type="checkbox"/>
Portable digital video camera	Optional, but highly recommended - Captures the physical reactions of the system users during testing, as well as post test interview responses.	<input type="checkbox"/>
Sufficient video capture memory/media	Sufficient media (tapes, memory cards, DVDs, or computer memory) to ensure all test recordings can be captured and stored.	<input type="checkbox"/>
Power Bar	To assure sufficient electrical outlets during testing for the video camera and laptop, a power bar (and possibly an extension cord) should be brought.	<input type="checkbox"/>
Screen capture audio/video recording software	Records the test subjects' interactions with the interface of the system under study. Make sure that the screen capture application is compatible with the tested system (i.e. if security prohibits the installation of applications in the user's environment, then a portable application, ran from a flash drive can be used, i.e. CamStudio Portable).	<input type="checkbox"/>
Video editing software	Censors confidential video sections, such as patient names (i.e. VirtualDub).	<input type="checkbox"/>
Video conversion software	Converts video files to appropriate formats for analysis or presentation (i.e. Handbrake).	<input type="checkbox"/>
Video analysis software	Helps usability testers transcribe, annotate, and analyze the recorded audio/video data (i.e. Transana).	<input type="checkbox"/>

Portable computing device	OPTIONAL (for portable system tests only) - Runs the system under study	<input type="checkbox"/>
Link Cable	OPTIONAL (for portable system tests only) - Routs the video output from the portable device that is running the system, to the computer so that it can be screen captured.	<input type="checkbox"/>

G. Selection of the study metrics

G.1 System Benchmark Metrics

System user observations are rich sources of system benchmarking data. Measures such as: task completion time, the number of actions (i.e. mouse clicks) required to complete a task, the number of supported features provided by the system, and the number of found usability issues are all strong indicators for system quality that could be utilized by system evaluators (i.e. system procurers, eHealth researchers, etc.) to rate or compare systems across this set of standardized measures.

List all benchmark indicators that are to be captured during the UB test:

G.2 Usability Metrics (Coding scheme)

For UB studies, a coding scheme is a complete collection of the names (codes), definitions, and examples of the metrics used to measure the usability of a system. The coding scheme serves as the reference manual for researchers as they annotate the audio/video data from the experimental sessions. Using this manual, whenever usability events occur in the recordings, the analysts mark the events according to their standardized coding scheme. The codes, within the scheme, can be used to identify: interface problems, content problems, as well as any user-entered system errors that occurred during the data collection phase of the study. The codes can measure both descriptive and interpretive events, and can be divided by severity. The coding scheme can be created from scratch or a ready-made coding scheme can be used or revised to suit the needs of the study. It is recommended that the 'minimalist approach' to generating coding schemes is applied, where a ready-made coding scheme is tailored to meet the hypothesis-driven needs of the unique study (Hofmann, Hummer, & Blanchani, 2003). An example of a coding scheme and its application is provided in Appendix 4.

Define the coding scheme that will be used to capture the usability events during the UB system testing:

H. System User UB Study Training

In the final planning stage of a Usability Benchmarking study, test subjects are informed about how the UB test will unfold, and are asked to consent to being part of the study. In this stage, before the system testing begins, the UB study planners provide the system users with a written description of the UB test plan, to prepare them for the study. An example of a UB system test introduction document is as follows:

“Thank you very much for agreeing to participate in the ACME ePrescription system Usability Benchmarking study! Our research goal for this system evaluation study is to: generate system use benchmarks; expose all current usability issues with your system; and compare and contrast your system use behaviour under different testing conditions.

To accomplish our study goals, we would like to record you as you undertake the tasks described in several use case scenarios, which portray realistic working conditions. First, you will complete a set of scenarios under normal working conditions. Then, you will complete a second set of similar scenarios, except this time, you will be asked to ‘Think Aloud’ (verbally state your thought processes as you complete the scenario tasks). The ‘Think Aloud’ comments of interest to the study include: what you are doing (i.e. “I’m looking for the close button”) and any difficulties you are having (i.e. “How do I get out of this window?”).”

A point form instructional sheet is also to be created for the subjects’ reference during the testing process. This can be given to the subjects on the day of or day before testing. An example subject reference sheet is provided in Appendix 11.

Is the system user properly informed about the UB study methodology?

Yes

Have you acquired written consent from the user, documenting their willingness to participate in the study?

Yes

Part 4. DO

A. UB System Testing

UB system testing involves recording the system users while they complete the use case scenarios. During testing, there are two rounds of traditional UB system tests (normal condition and ‘Think Aloud’), as well as a third optional round of system automation testing. All rounds of system testing are conducted during the data collection (DO) phase of a UB test. For instructions on how to setup the UB system test lab or how to record the subject, see Appendix 3.

A.1 Normal Condition System Tests

A.1.1 Normal Condition Scenario Testing

For normal condition system tests, the system users are recorded as they perform the tasks described in the normal condition use case scenarios. The subjects are to conduct the scenarios one at a time, in an order that gradually increases in complexity. While the users are performing the scenarios, their actions are recorded through screen and voice capture software (and optionally, with a video camera). The system users are to be encouraged not to make any extra effort to explain their actions to the analyst, as the normal condition tests are designed to replicate their normal working environment.

Have all of the normal condition system tasks been completed by each system user and recorded?

Yes

A.1.2 User Questionnaires

At the completion of each normal condition scenario, the users are to complete an “After Scenario Questionnaire” or ASQ (Lewis, 1995) (see Appendix 8 for the ASQ). The ASQ is used to supplement the usability testing results. Once all normal condition scenarios have been completed, the users are to complete a single “Post Study System Usability Questionnaire” or PSSUQ (Lewis, 1995). The PSSUQ is used to supplement the overall usability assessment of the system under study (see Appendix 9 for the PSSUQ).

Has an ASQ been filled out by all users for each completed scenario?

Yes

Has a PSSUQ been filled out by each user upon completion of all their normal condition tests?

Yes

A.1.3 Post-Scenario Interviews

After a scenario is completed by a user, a post-scenario interview is to be conducted by the analyst. Post-scenario interviews follow a semi-structured interview format, where the analyst follows a questioning template, but allows additional areas of interest to be addressed (if mentioned by the subject).

During the post scenario interview, the analyst is to ask the subject questions pertaining to:

- if the subject could clarify any actions or events witnessed during the system tests that were unclear to the analyst;
- if there is any additional input that the users may have about the given scenario under study (i.e. issues concerning the system components utilized in the scenario);
- whether or not there is anything about the test or scenarios that the user thinks can be improved (i.e. the prescription dose used in the scenario was inaccurate, the task order was incorrect, etc.).

It is best practice to video record the post-scenario interviews. This allows for in-depth analysis of the interviewees’ responses, and assures that none of the test subjects’ feedback is missed.

Have all witnessed system testing events been clarified?

Yes

Has the system user provided his/her feedback about the UB test?

Yes

A.2 Think Aloud System Tests

A.2.1 Think Aloud Scenario Completion

After a study subject has completed all of the normal condition scenarios, they are then asked to complete the scenarios listed in the ‘Think Aloud’ set. Thinking aloud is a cognitive science technique where subjects are recorded as they vocalize their thoughts while interacting with a system. By having the users ‘Think Aloud’, they reveal their user experience, while such thoughts are still in short-term memory, which can increase the level of detail and quality of the cognitive process data produced by the subjects (Ericsson & Simon, Protocol analysis: Verbal reports as data (Revised edition)., 1993).

During the 'Think Aloud' testing phase, the system users complete the scenarios defined in the 'Think Aloud' set, while they verbally state their thought processes. Think aloud comments are analogous to the comments made by a sports play-by-play announcer, where all actions are stated aloud, as they happen in real time. Such comments can involve: what the user is doing (i.e. I'm looking for the file menu); points of confusion by the user (i.e. I'm not sure what to do here); aspects of the system or processes that the user likes/dislikes (i.e. "It takes too long to do this"); and aspects that the user would like to change about the system or processes (i.e. "I wish the system had this feature").

Neither the ASQ nor the PSSUQ questionnaires are completed by the users after completing the 'Think Aloud' tests. The ASQ and PSSUQ questionnaires are only to be completed during the normal condition testing round.

Have all of the scenarios been completed (and recorded) as the user 'Thought Aloud'?

Yes

A.2.2 Post-Scenario Interviews

The same interview format used during the normal condition scenario testing round is to be used during the 'Think Aloud' testing round.

Have all witnessed system testing events been clarified?

Yes

Has the system user provided his/her feedback about the UB test?

Yes

A.3 System Automation Testing (optional)

System automation testing is performed by one user at a single point in time. It can be completed before or after the normal condition and 'Think Aloud' user testing. During system automation testing, the user is recorded as they follow the system automation task set, which creates the system conditions necessary to trigger the targeted automated system responses. Before the user completes each task, he/she is to state aloud the identifier of the task to be completed (this will aid later analysis efforts).

Part 5. STUDY

A. Data Compilation and Analysis

A.1 Formatting the Test Recordings

Before the rich scenario data can be analyzed, it must first be compiled into an appropriate form. The normal condition, 'Think Aloud', system automation, and interview test results are all contained within audio/video recordings that may need to be massaged into an appropriate format before analysis. How the audio/video recordings are properly formatted for analysis depends on several factors, including:

- the tool that was used to capture the recordings;
- the formats that the recordings are stored in;
- the quality of the audio/video formats (i.e. is the audio and video synched, is the recording in real time, etc);
- whether or not there's sensitive material present in the recordings

Although it is out of the scope of this paper to describe all possible audio/video formatting issues that can occur in a UB study, an example of the issues encountered during the author's application of the UB methodology is provided to help provide some possible solutions to such issues.

A.1.1 Unsynchronized Audio and Video

During the data collection stage of the author's UB study, the choice of screen capture application was limited by the fact that no software could be installed on the test subject's computer. To get around this, a portable screen capture application (CamStudio Portable) was used to record the scenario results. Although this tool was able to capture the test subject's screen and voice activity, the resulting audio/video recordings were unsynchronized (the audio ran faster than the video). To resolve this problem, VirtualDub (VirtualDub.org), a free video capture/processing tool was used to re-synchronize the audio/video tracks. To correct this problem using VirtualDub:

1. Open the video recording in the VirtualDub player
2. Select Video -> Direct Stream Copy
3. Select Video ->Frame Rate -> Change so video and audio durations match -> OK
4. Re-save the video recording

A.1.2 Incompatible Video Format

Ideally, when recoding the post-scenario interviews, the video camera used will record (via FireWire) directly into the analyst's laptop, in a form that is compatible with the analyst's analysis and presentation tools (i.e. Transana, PowerPoint, etc.). During the author's UB study; however, there was no laptop available with a FireWire port, so the interviews were recorded with a digital video camera into a DVD format (Video_TS). Because the DVD format was not compatible with the video analysis software used by the analyst (Transana), the recordings had to be converted into a different format.

To re-format the interview recordings, the free and open-source video formatting tool HandBrake (HandBrake Project 2009) was used. To convert a DVD formatted audio video recording to a format that is compatible with Transana and PowerPoint in HandBrake:

1. Copy the Video_TS folder from the video camera DVD that contains the interview results
2. Open HandBrake
3. Select Source -> Video_TS -> Locate and select the Video_TS folder
4. Under Destination, Select Browse, and choose the destination folder and file name (make sure that the file name extension is listed as .MP4)
5. Under Output Settings, Select Format -> MP4
6. Select Start

A.1.3 Video Censoring

During the UB scenario, some sensitive material that wasn't relevant to the study was present in the scenario recording at times (i.e. patient identifiers, etc.). Before the recorded scenario material could be analyzed, it was first necessary to remove the sensitive material from the test recordings. To achieve this, VirtualDub (VirtualDub.org)was again used. To censor sections of a video recording using VirtualDub¹:

1. Open the video in the VirtualDub player
2. Select Video -> Filters -> Add -> Fill -> OK
3. Move to the point in the video that requires censoring
4. Set the X1, X2, Y1, and Y2 offsets so that the portion of the video that requires censoring is framed (a shortcut to this is to set all offsets to 1, then to click and drag the offsets on the screen to their appropriate location)

¹ This type of filtering is for relatively still sensitive material. Sensitive material that constantly moves within the video may require a different censoring technique

5. Select OK
6. Highlight the previously created fill filter in the filters menu and select Blend, then OK
7. Repeat steps 2-6 for each portion of the video that requires a uniquely sized and/or positioned filter
8. Select View -> Curve Editor
9. Position the video to the first frame of the point in the recording that requires the application of one of the created filters (this is easiest done through first moving the slider bar, then clicking the next/previous frame buttons)
10. Select the filter that was framed to cover the sensitive material that the recording is currently positioned on
11. Add two filter points on the exact frame where the sensitive material begins (do this by pressing shift and left clicking one point right beside the other on the curve editor display, where the current frame is signified by a vertical line)
12. Click and drag the first point to the far bottom of the current frame line in the curve editor (this will hide the filter up until this point)
13. Click and drag the second point (to the right of the first point) to the far top of the current frame line in the curve editor, directly above the first point (this will turn on the filter to cover the sensitive material)
14. Go to the exact frame of the recording where the sensitive material is no longer present on the screen
15. Insert two more points directly on the current frame line in the curve editor
16. Click and drag the first point to the top of the curve editor current frame line (this will keep the filter present up until this point)
17. Click and drag the second point to the bottom of the curve editor current frame line, directly under the first point (this will now hide the filter after this point)
18. Repeat steps 9-17 for all areas of the recording with sensitive material present
19. Select Video -> Compression -> Microsoft Video1 -> OK
20. Ensure that both the Audio and Video menus are set to Full Processing Mode
21. Select File -> Save as AVI and save the new video in the intended destination folder.

A.1.4 Ensure Quality and Remove Duplicate Recordings

After audio/video formatting efforts have been completed, ensure that the reformatted recordings are acceptable:

- Check the timing of the videos. Use a stopwatch or an accurate clock and compare the timing with the clock shown on the screen captured videos to make sure that the re-formatted video recordings run in real time.
- Ensure that all sensitive materials have been censored
- Test that all recordings are compatible with all analysis and presentation tools that will be used in the study

Are all of the videos accurately synched to real time? Is all sensitive material censored in the video recordings? Are all videos in a format that can be used for analysis and presentation?

Yes

If the formatted recordings pass the above checks, then delete all of the unformatted videos permanently to eliminate any remaining sensitive material.

Are all unformatted, uncensored videos permanently deleted?

Yes

A.2 Normal Condition Scenario Compilation and Analysis

A.2.1 Normal Condition Scenario Compilation

The following steps are to be undertaken to compile the normal condition scenario results of a UB study:

1. Load the normal condition test videos into the video analysis application (i.e. Transana)
2. Within the recording transcript of each scenario, timestamp the start and stop time for all of the scenario tasks
3. Timestamp any pauses that occur within the scenario tasks so that the pauses fall between their respective task start and stop times (i.e. if the subject takes time out from a task to use the washroom or any other action that is not related to the current task, a timestamp is created for when the interruption starts and ends within the task, so that the interruption is removed from the overall task calculations)

Have all normal condition scenario task (start and stop) and pause (start and stop) timestamps been recorded in the video analysis application?

Yes

A.2.2 Normal Condition Benchmark Analysis

The goal of normal condition scenario analysis is to produce the system benchmark measures that were identified in part 3, section G.1. The types of system benchmarks tracked will vary between studies (i.e. completion of task time; the number of required task actions; number of supported system features; etc.); however, the general process of analyzing the benchmarks is standardized.

To perform the benchmark analysis, using the video analysis application, review each task from each scenario, and record all appropriate benchmarks in an analysis template (created using either a spreadsheet or a database). Analysis templates (i.e. Benchmarks-Blank_template.xlsx, a benchmark analysis template for ePrescribe systems) help facilitate benchmark analysis through the provision of such functionality as: structured coding, standardized scenario entries, and automated analysis via formulas or queries).

Once all benchmark data has been reviewed in the scenario recordings and coded in the analysis template, the template can be used to present scenario and overall system summaries of the identified benchmark indicators.

Have all system benchmarks that were defined in part 3, section G.1 been captured and clearly documented in an analysis template?

Yes

A.2.3 ASQ and PSSUQ Compilation and Analysis

Each user will have completed an ASQ for each normal condition scenario as well as a single PSSUQ. Tabulate their questionnaire responses in the normal condition analysis template (as seen in Benchmarks-Blank_template.xlsx). The final ASQ and PSSUQ scores can be used to supplement the usability findings from the 'Think Aloud' scenario analysis.

Have all ASQ and PSSUQ responses been tabulated?

Yes

A.3 'Think Aloud' Scenario Compilation and Analysis

A.3.1 'Think Aloud' Scenario Compilation

To compile the 'Think Aloud' scenario recordings:

1. The 'Think Aloud' test videos are loaded into the video analysis application (i.e. Transana)
2. The start and stop time of each task are timestamped in the scenario transcripts

Once the 'Think Aloud' scenarios have been successfully imported and task coded in the video analysis tool, the recording contents can be broken down further using the coding scheme that was established in part 3, section G.2. This UB codification approach is derived from the 'Scoring' level of protocol analysis (Newell & Simon, 1972), where audio/video recordings are coded and analyzed for issues of study relevance.

The codification of UB system test data takes place over three stages:

1. Validating the coding scheme
2. Annotating the recordings
3. Verifying the annotated transcripts

A.3.1.a Validating the Coding Scheme

It may become evident after watching the user perform the system tests that the coding scheme needs refinement. It is necessary to ensure that the scheme is clear and concisely tailored to meet the hypothesis driven needs of the study before the annotation phase begins. To reduce the potential of wasted time and effort in the form of having to re-annotate all of the recorded material after making a change to the coding scheme, it is recommended that before all recordings are annotated, a single (sample) scenario is separately annotated by multiple analysts, then collaboratively verified. The sample scenario annotation/validation test process is to be repeated until no further changes are made to the coding scheme. Once the coding scheme is finalized, all test recordings are then re-annotated under the final coding scheme (as described in the ensuing sections).

The coding scheme has been verified and defined as follows:



A.3.1.b Annotating the Transcripts

A minimum of two analysts are to be used when annotating the test recordings. This will reduce the subjectivity of the observational codes.

To annotate the recordings, the analysts will individually review each of the 'Think Aloud' scenario recordings, and mark up their transcripts with annotations from the usability coding scheme in the video analysis application. When a usability event (as defined by the coding scheme) occurs within the scenario recording, the event is annotated between the start and stop timestamps of the task in which the event occurs. Each usability event annotation includes (from left to right):

- A start time-stamp, which states exactly when the usability event started in the scenario recording (i.e. 00:00:46);

- (Optional – to be recorded only if an error is associated with the usability event) An error category and type (i.e. slip, medication type), which signifies that an error occurred in the scenario that is directly associated with the usability event;
- The usability code or name (i.e. “Navigation”), which lists the type of usability event that occurred;
- A brief description of the usability event (i.e. “The user wasn’t able to find the correct medication in the list so he entered it manually with incorrect spelling. He later found the medication on the list and selected it, correcting the spelling issue.”);
- The event stop timestamp (i.e. 00:00:52), which signifies when the usability event stops in the scenario recording.

The end result of this stage will be multiple sets of annotated transcripts ($T = A \times S \times U$, where T is the number of annotated transcripts, A is the number of analysts who independently transcribe the transcripts, S is the number of Think Aloud scenarios completed in the study, and U is the number of system users who completed the scenarios).

The transcripts from all tasks under all conditions have been separately annotated by more than one analyst.

Yes

A.3.1.c Verifying the Annotated Transcripts

To ensure that the transcript annotations are reproducible and not subjective (Simon, 1979), inter-coder agreement by a minimum of two evaluators is to be established for all transcript annotations (Todd & Benbasat, 1987), (Vessey, 1984), (Payne, Braunstein, & Carroll, 1978). To accomplish this, the analysts must come together to discuss any differences between their annotated transcripts. All differences are then resolved either through discussion or by re-reviewing the usability events in question. Once all differences are resolved, only a single annotated transcript will exist for each system task condition, for each user ($T = S \times U$).

All disputes between the analysts’ separate annotated transcripts have been reconciled so that only a single annotated transcript exists for each ‘Think Aloud’ scenario from each user.

Yes

A.3.2 ‘Think Aloud’ Usability Analysis

The system test recordings and their annotated transcripts are all rich sources of qualitative data. This qualitative data can also be turned into quantitative data by tabulating the frequencies of the coded annotations (Babour, 1998), (Borycki & Kushniruk, 2005). With this quantitative data, inferential statistics can be applied to generate predictions for future usability events (Patel, Kushniruk, Yang, & Yale, 2000). Again, an analysis template (i.e. Usability_Analysis_Tool-Blank_template.xlsx, a template for ePrescription system usability analysis) is to be used to standardize the analysis results and to automate the statistical calculations. The template will typically format the study results as summary tables (i.e. the frequency and type of user problems occurring within task #1 for system X).

Have the coded usability results from the annotated transcripts been tabulated, and entered into an analysis template/summary table?

Yes

A.4 System Automation Compilation and Analysis

A.4.1 System Automation Compilation

To compile the system automation test results for analysis, load the automation tests into the video analysis tool and timestamp the start and stop of each system automation task. Make sure to place the task stop timestamp after the system provides its automated response (if any response is provided).

All system automation tasks have been loaded into the video analysis software and have been timestamp coded for the start and stop of each automation task test.

Yes

A.4.2 System Automation Analysis

To analyze the compiled system automation results, create an analysis template (i.e. System_Automation-Blank_template.xlsx) that contains fields for: each system automation task; the intended system response (the response that the system is supposed to give); and the actual system response (what the system did during the test, in response to the task). After the template is created, review each task with the video analysis application and enter the system's task responses from the recordings into the template. Any tasks where the intended system response differs from the actual system response should then be flagged for additional review and or reporting to the system vendor.

Have all system automation tasks been reviewed by comparing the intended system response to the actual system response for all system automation tests?

Yes

A.5 Interview Compilation and Analysis

For any post-scenario user interview recordings, load the recordings into the video analysis application. To help facilitate qualitative analysis of the users' responses, it may be necessary to transcribe the subject's responses; however, the choice of qualitative analysis technique used for the post scenario interview analysis is at the discretion of the analysts.

Have the post scenario interview recordings been loaded into the video analysis application?

Yes

A.6 Comparative Analysis of Conditional System Tests

The final source of UB study analysis is a comparison of users' behaviour between the normal condition and 'Think Aloud' tests. Comparisons could be made around not only the benchmark indicators, but also between the system user's general process flows.

The annotated transcripts can be used to map the process flows from each scenario condition. Once the flow of actions for each task condition has been mapped, each of the conditional task flows can then be compared against one another for differences (i.e. scenario #1's process flow in normal condition testing vs. 'Think Aloud' testing, scenario #2's process flow in normal condition vs. 'Think Aloud', etc.).

The system flows and benchmark indicators from each of the task conditions has been mapped out and compared against one another for differences.

Yes

Part 6. ACT

A. Courses of Action for UB Deliverables

The courses of action taken with the UB deliverables are dependent on the goals of the UB study (defined in part 3, section C). The following are examples of how the UB study deliverable content can be utilized within the context of healthcare.

A.1 System Benchmark Content

The system benchmark content can be used by any group or individual who wishes to assess a system based on a set of standardized measures. An example of such use would be an organization that's in the health system procurement process. The benchmark measures captured by the UB system tests can be used to compare one system against another (i.e. one person using EMR #1 can complete the "Order Prescription" scenario in 3 minutes and 45 seconds but second person using EMR #2 can complete the "Order Prescription" scenario in 2 minutes and 11 seconds), and thus, help to determine which system should be purchased by the organization.

Has the UB system test benchmark data been released in a format that will allow members from the healthcare industry to review the standardized measures when procuring health systems?

Yes

Benchmarks can also be used by organizations to establish how much training their employees need on a given system. For example, if some users can complete a given task in 2 minutes, but the majority of the organization's employees take (on average) 8 minutes to complete the same task, then this would be a strong indication that the organization could save time and money by further training their staff in the use of the given system task.

Have the system use measures been compared against the organization's employee system use measures as a means of training justification?

Yes

Finally, if the system benchmark indicators are released to the public, they may serve as motivation towards system improvement for the system's developers. If a system developer is made aware that their system was reported as being slower than a competing system, then the developers would likely aspire to improve their product to prevent a loss in market share.

Have the system benchmark indicators been released to the public to promote competition among system developers for creating systems of the highest ratings for quality and efficiency?

Yes

A.2 Usability Testing Content

The usability testing results can be given to the system developers to provide them with feedback regarding how to improve their product through the elimination of found usability issues. Once these system usability revisions have been made by the developers, UB or other usability tests can be repeated to determine if any usability issues were created in the system revisions or missed by the previous tests. This process creates an iterative feedback cycle of system improvement.

Have the usability results been provided to the system's developer as a request for system improvements?

Yes

A.3 System Automation Content

Inform the system's manufacturer about any automated system responses that were found to differ from ideal behaviour. These results can be prioritized, based on the desirability of the needed change in order to ensure that the critical system automation corrections are made first.

Have all system automation discrepancies been resolved with the system's vendor?

Yes

A.4 Interview Response Content

Qualitative analysis of the interviews may provide relevant information to support the UB study findings (i.e. quotes about certain system usability aspects) or reveal areas of improvement for future UB studies. These results are typically used to supplement the final UB study report write-up.

Has information that supports the study findings and contributes to UB study learning been extracted from the post-scenario interviews and included in the final UB study report write-up?

Yes

A.5 Comparative Analysis of Conditional System Tests Content

The system test condition comparison results can be utilized by academia to strengthen evidence regarding user system testing. If it is found that users drastically change their process flow when asked to 'Think Aloud', then it may be necessary to revise how such evaluations are conducted. It has been determined that the system user's process flow in the 'Think Aloud' conditional tests (*circle one*):

- a) matched the system flow of the normal condition tests; therefore, no action is required.
- b) did not match the system flow of the normal condition tests; therefore, further investigation is required.

Part 7. Bibliography

- Babour, R. (1998). The Case of Combining Quantitative Approaches in Health Services Research. *Journal of Health Services Research Policy* , 4 (1), 39-43.
- Bellotti, V. (1988). Implications of current design practice for the use of HCI techniques. In D. M. Jones, & R. Winder, *People and Computers IV* (pp. 13-34). Cambridge: Cambridge University Press.
- Borycki, E., & Kushniruk, A. (2005). Identifying and Preventing Technology-Induced Error Using Simulations: Application of Usability Engineering Techniques. *Healthcare Quarterly* , 8, 99-105.
- Bouwman, M. J. (1983). The Use of Protocol Analysis in Accounting Research. *Unpublished (working paper)* .
- Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale: Lawrence Erlbaum Associates.
- Carter, J. H. (2004, April). *EMR Feature Checklist 1.0*. Retrieved June 18, 2009, from American College of Physicians - Internal Medicine: <http://www.acpinternist.org/archives/2004/04/emr/cklist.pdf>
- Dykstra, D. J. (1993). A Comparison of Heuristic Evaluation and Usability Testing: The Efficacy of a Domain-Specific Heuristic Checklist. *PH.D. Diss* . Texas, United States of America: Department of Industrial Engineering, Texas A&M University.
- Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data (Revised edition)*. Cambridge: MIT Press.
- HandBrake Project 2009. (n.d.). Retrieved November 27, 2009, from HandBrake: <http://handbrake.fr/>
- Hofmann, B., Hummer, M., & Blanchani, P. (2003). State of the Art: Approaches to Behaviour Coding in Usability Laboratories in German-Speaking Countries. In M. Smith, & C. Stephanidis, *Human - Computer Interaction - Theory and Practice* (pp. 479-483). Sankt Augustin: Lawrence Erlbaum Associates.
- Kushniruk, A. (2001). Analysis of complex decision making processes in health care: cognitive approaches to health informatics. *Journal of Biomedical Informatics* (34), 365-376.
- Kushniruk, A. W., & Patel, V. L. (2004). Cognitive and usability engineering methods for the evaluation of clinical information systems. *Journal of Biomedical Informatics* (37), 56-76.
- Kushniruk, A. W., Patel, V. L., & Cimino, J. J. (1997). Usability testing in medical informatics: cognitive approaches to evaluation of information systems and user interfaces. *Proc AMIA Annu Fall Symp* , 218-222.
- Kushniruk, A. W., Patel, V. L., Cimino, J. J., & Barrows, R. A. (1996). Cognitive evaluation of the user interface and vocabulary of an outpatient information system. *Proc AMIA Annu Fall Symp* , 22-26.
- Kushniruk, A., & Patel, V. (2005). Cognitive Approaches to the Evaluation of Healthcare Information Systems. In J. G. Anderson, & C. E. Aydin, *Evaluating the Organizational Impact of Healthcare Information Systems* (2nd Edition ed., pp. 161-162). New York: Springer.
- Kushniruk, A., & Patel, V. (1995). Cognitive computer-based video analysis: its application in assessing the usability of medical systems. *Medinfo* , 1566-1569.

Kushniruk, A., Triola, M., Borycki, E., Stein, B., & Kannry, J. (2005). Technology induced error and usability: The relationship between usability problems and prescription errors when using a handheld application. *International Journal of Medical Informatics* (74), 519-526.

Lewis, J. R. (1995). *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. Boca Raton: Human Factors Group - IBM.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice-Hal.

Nielsen, J. (1994). Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. In D. Mayhew, *Cost-Justifying Usability* (pp. 245-272). Orlando: Academic Press, Inc.

Nielsen, J. (2001). *How to Conduct a Heuristic Evaluation*. Retrieved March 03, 2009, from useit.com: http://www.useit.com/papers/heuristic/heuristic_evaluation.html

Nielsen, J. (2005). *Ten Usability Heuristics*. Retrieved March 5, 2009, from Useit.com: http://www.useit.com/papers/heuristic/heuristic_list.html

Nielsen, J. (1993). *Usability engineering*. New York: Academic Press.

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. *Proceedings ACM/IFIP INTERCHI'93 Conference*, (pp. 206-213). Amsterdam.

Nielsen, J., & Mack, R. L. (1994). *Usability Inspection Methods*. New York: John Wiley & Sons.

Nielsen, J., & Molich, B. (1990). HEURISTIC EVALUATION OF USER INTERFACES. *Conference on Human Factors in Computing Systems Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people* (pp. 249-256). New York: ACM.

Patel, V. L., Kushniruk, A. W., Yang, S., & Yale, J. F. (2000). Impact of a Computer-Based Patient Record System on Data Collection, Knowledge Organization and Reasoning. *Journal of the American Medical Informatics Association* , 7 (6), 569-585.

Payne, J. W., Braunstein, M. L., & Carroll, J. S. (1978). Exploring Predecisional Behavior: An Alternative Approach to Decision Research. *Organizational Behavior and Human Performance* , 22 (1), 17-44.

Payne, S. J. (1986). Task-Action Grammars: A Model of the Mental Representation of Task Languages. In S. J. Payne, & T. R. Green, *Human-computer interaction* (Vol. 2, pp. 93-133). Lawrence Erlbaum Associates.

Perlman, G. (1994). Practical usability evaluation. *Conference on Human Factors in Computing Systems* (pp. 407-408). Boston: ACM.

Polson, P., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies* , 36 (5), 741-773.

Rosson, M. B., & Carroll, J. M. (1995). Narrowing the specification-implementation gap in scenario-based design. In J. M. Carroll, *Scenario-based design: envisioning work and technology in system development* (pp. 247-278). New York: John Wiley & Sons, Inc.

Rubin, J. (1994). *Handbook of usability testing: how to plan, design and conduct effective tests*. New York: Wiley.

Seffah, A., & Metzker, E. (2009). Usability Engineering Methods Plethora. In A. Seffah, & E. Metzker, *Adoption-centric Usability Engineering* (pp. 15-33). London: Springer-Verlag London Limited.

Shewart, W. A., & Deming, W. E. (1939). *Statistical Method from the Viewpoint of Quality Control*. Washington: The Graduate School, The Dept. of Agriculture.

Shneiderman, B. (2003). *Designing the user interface* (4th Edition ed.). New York: Addison-Wesley.

Simon, H. A. (1979). INFORMATION PROCESSING MODELS OF COGNITION. *Annual Review of Psychology* , 311-396.

Smith, S., & Mosier, J. N. (1986). *Guidelines for Designing User Interface Software - MTR-10090*. Bedford: The MITRE Corp.

Todd, P., & Benbasat, I. (1987). Process Tracing Methods in Decision Support Systems Research: Exploring the Black Box. *MIS Quarterly* , 11 (2), 493-512.

Vessey, I. (1984). An investigation of the Psychological Processes Underlying the Debugging of Computer Programs. *unpublished Ph.D. thesis University of Queensland* .

VirtualDub.org. (n.d.). Retrieved November 27, 2009, from VirtualDub.org: <http://www.virtualdub.org/>

Wallden, S. *The Usability of Medical Web Sites*. Tampere: Hypermedia Laboratory - University of Tampere.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: a practitioner's guide. In J. Nielsen, & R. L. Mack, *Usability inspection methods* (pp. 105-140). New York: John Wiley & Sons, Inc.

Part 8. Appendices

Appendix 1 – System User Selection Questionnaire

Please answer the following user background questions:

1. My full name is:

2. Using the date format YYYY/DD/MM (i.e. 1984/08/23), please state your date of birth:

3. Please state your educational background:

4. Please state your relevant (with respect to *[the chosen system]*) occupational history?

5. On average, I work with [the chosen system] _____ [circle one] _____:

Never (b) Daily (c) Weekly (d) Monthly (e) Yearly

6. Within the time frame circled in question five, I spend an average of _____ [write in the box below]

hours working with *[the chosen system]* :

7. I would personally rate my skills with *[the chosen system]* to be at a(n) _____ [circle one] _____ level:

Beginner (b) Intermediate (c) Expert

8. I would be available to work on the Usability Benchmarking study for _____ [circle one] _____ per week:

a) Less than 1 hour (b) Between 1 and 3 hour
(c) Between 3 and 5 hours (d) More than 5 hours

9. Using the date format YYYY/DD/MM (i.e. 2009/09/13), please state the time frame that you would be able to work on the Usability benchmarking study:

From: to

Appendix 2 – UB Test Equipment Costs and Recommendations

Equipment	Purpose	Recommendations	Examples	Cost
Computer (desktop or laptop)	Runs the system under study and/or the screen capture and video analysis software.	<p>Any computer made within the past 5 years that has the hard drive capacity to store a large amount of recorded testing data should suffice (keeping in mind that 5 minutes of testing creates approx. 200 megabytes of audio and video files).</p> <p>Due to portability, a laptop is the recommended option (as it would allow the evaluators to easily move the system to and from different test sites). If a laptop is chosen, it is heavily recommended to secure a backup power supply (i.e. a second battery or an AC adapter) to ensure that the computer is operational throughout the duration of the test. It is also recommended to select a laptop contains a FireWire port to enable direct video recording to the laptop from a video camera.</p>	<p>See the following computer/electronics store Web sites for examples:</p> <ul style="list-style-type: none"> • Dell Canada - http://www.dell.ca/ • Future Shop – http://www.futureshop.ca • Staples – http://www.staples.ca 	\$400 - \$2000
Computer microphone	Captures the 'Think Aloud' recordings.	<p>Any computer microphone would suffice. For highest audio quality, a headset should be used.</p> <p>To allow for movement by the test subject during the study, either a standard microphone or a wireless headset should be used.</p>	See the above computer/electronics store Web sites for examples.	\$5 - \$100
Digital video camera	Captures post-scenario interviews and (optionally) the test subjects' physical responses during scenario testing.	Any standard digital video camera that can output computer readable files (i.e. .avi, .mpg, etc.) will suffice. A camera that contains a port that can link to a laptop (i.e. via FireWire) is preferable.	See the above computer/electronics store Web sites for examples.	\$100 - \$500
Power Bar	To ensure enough power outlets are secured for the test (i.e. for the laptop, video camera, etc.)	Any computer power bar will be sufficient	See the above computer/electronics store Web sites for examples.	\$5-\$20

Screen capture audio/video recording application	Records the test subjects' interactions with the interface of the system under study.	The open source product CamStudio (http://camstudio.org/) is free and easy to use. CamStudio Portable can be used at sites that don't allow software installations.	<ul style="list-style-type: none"> • CamStudio - \$0 (http://camstudio.org/) • HyperCam - \$40 (http://www.hyperionics.com/hc/) 	\$0 - \$40
Video conversion software	Changes the formatting of video recordings from one format to another (i.e. DVD to .AVI)	The free and open-source video formatting tool HandBrake (HandBrake Project 2009)	<ul style="list-style-type: none"> • Handbrake - \$0 (http://handbrake.fr/) 	\$0
Video editing software	Facilitates creating edits to the video recordings (i.e. censoring sensitive material, creating clips, etc.)	The free video capture/processing tool VirtualDub (VirtualDub.org)	<ul style="list-style-type: none"> • VirtualDub - \$0 (http://www.virtualdub.org/) 	\$0
Video analysis application	Helps usability testers transcribe, annotate, and analyze the recorded audio/video data.	The most popular tool for analyzing usability test recordings is Transana (http://www.transana.org/). Although the tool is open source, the author of the tool is now asking for licence fees to help cover the costs of maintenance. For low-budget usability tests, there are free to use video analysis applications (i.e. ANVIL, Subtitle Workshop, etc.); however, at the time this paper was written, these tools have not been tested by the paper's author. For high-budget usability tests, there are vendor made video analysis applications available (i.e. Observer XT); however, such tools come at a much higher cost than the readily available open source solutions.	<ul style="list-style-type: none"> • Transana - \$50 - \$500 (http://www.transana.org/) • ANVIL - \$0 (http://www.anvil-software.de/index.html) • F4 Video - \$0 (http://www.audiotranskription.de/english/transcription/f4video-pc/videotranscription-with-f4video.html) • Subtitle Workshop - \$0 (http://www.urusoft.net/products.php?cat=sw&lang=1) • ELAN - \$0 (http://www.la-mpi.eu/tools/elan/) • Observer XT - \$1000 - \$8000 (for academics) (http://www.noldus.com/human-behavior-research/products/the-observer-xt) 	\$0 - \$8000
Video Camera Media	OPTIONAL (for cameras that cannot record directly to the computer)	Any media that is both compatible with the selected video camera, in sufficient volume to capture all of the video recordings (estimate the # of recording hours ahead of time and ensure that enough media has been secured to capture this).	See the above computer/electronics store Web sites for examples.	\$10-\$50

Portable computing device	OPTIONAL (for portable system tests only) - Runs the system under study	The portable computing device that naturally runs the system under study should be used	See the above computer/electronics store Web sites for examples.	\$0 - \$2000
Link Cable	OPTIONAL (for portable system tests only) - Routes the video output from the portable device that is running the system, to the computer so that it can be screen captured.	Typically, this cable will be supplied with the portable computing device (i.e. a USB 2.0 cable).	See the above computer/electronics store Web sites for examples.	\$0 - \$30
Portable computing device video routing application	OPTIONAL (for portable system tests only) - Routes the video output from the portable device that is running the tested system to the computer that is running the screen capture application	Any application that will stream the live video from the portable computing device as it is being operated on by a user will suffice.	<ul style="list-style-type: none"> • Tapsmart handshare - \$30 (http://www.mobilityware.com/HandShare/HandShareProduct.htm) • Mdesk - \$20 (http://www.motionapps.com/product/mdesktop/) 	\$20 - \$30
Total:				\$510 - \$12,770

Appendix 3 – Remote Laboratory Setup

A3.1 - Setup for a stationary system

For a stationary system (i.e. a desktop application), there are four essential components needed for collecting data in a UB test: a computer with a monitor, which is used to run the tested system; a microphone, which is used to record the test subject's voice; a video screen capture application, which is used to record the system's display on the monitor as the test subject interacts with the system; and a video camera, which is used to record the user's physical actions as he/she perform the tests.

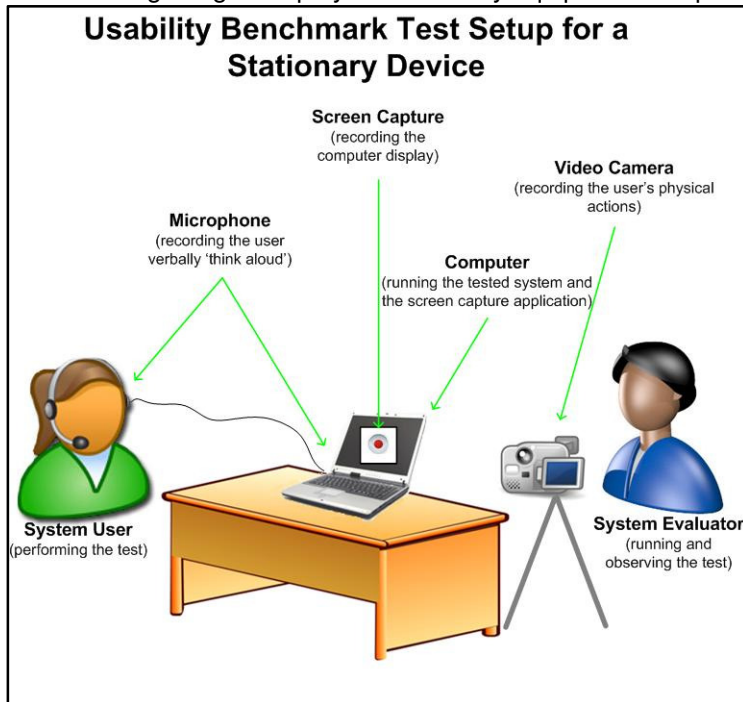
During the test, the application under study will run through the computer and be presented on its monitor. As the test is being conducted, the video screen capture application will record all of the events which are displayed on the monitor, and save them as a video file. The video screen capture application will also utilize the microphone (which is normally a headset worn by the test subject) by recording all of the subject's auditory comments during the test on the screen capture video file. Finally, the system users' physical actions (i.e. facial expressions, body language, etc.) and post scenario interviews are recorded via portable digital video cameras or Web cameras. In a medical setting, there are occasions when the video cameras need to be placed in unobtrusive locations (as opposed to a tripod-mounted video camera). Possible solutions to such cases include: using ceiling mounted cameras, using compact cameras in hidden locations, or often is the case in a hospital setting where there are already rooms that are readily equipped with such unobtrusive monitoring devices (i.e. student interview rooms), which can be utilized for such testing purposes.

A3.2 - Setup for a portable system

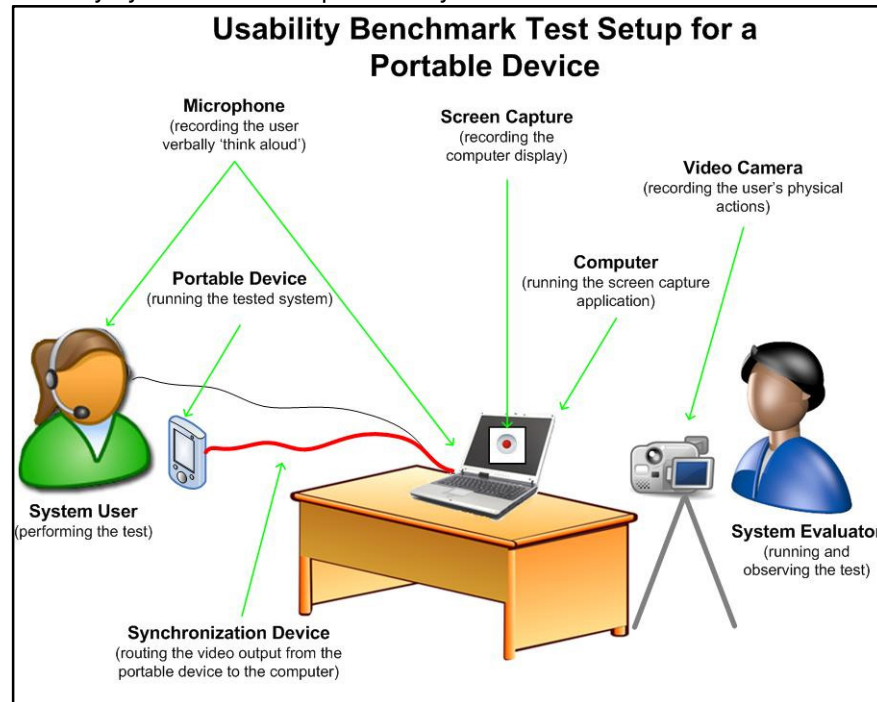
The basic setup for testing a portable computing system (i.e. smartphones, PDAs, etc.) is the same as a stationary computing system, except that an additional device and/or application is needed to synchronize the portable system to the laptop/desktop's monitor. Currently, a lack in memory size, processor speed, and available screen capture software makes recording the screen interactions on a portable device unfeasible; however, it is possible to rout the portable device's video output to a desktop/laptop computer, where that video could then be recorded as normal. In such a setup, while the test subject interacts with the portable system, those interactions must be synchronously displayed on the desktop/laptop monitor (via the synchronization device/application). When the portable device's display has been routed to the desktop/laptop's monitor, then the desktop/laptop's video screen capture application and microphone can be used to capture the screen and voice events (as in the previous setup).

A3.3 – Remote Laboratory Setup Figures

The following images display the UB study equipment setup for stationary system tests and portable system tests:

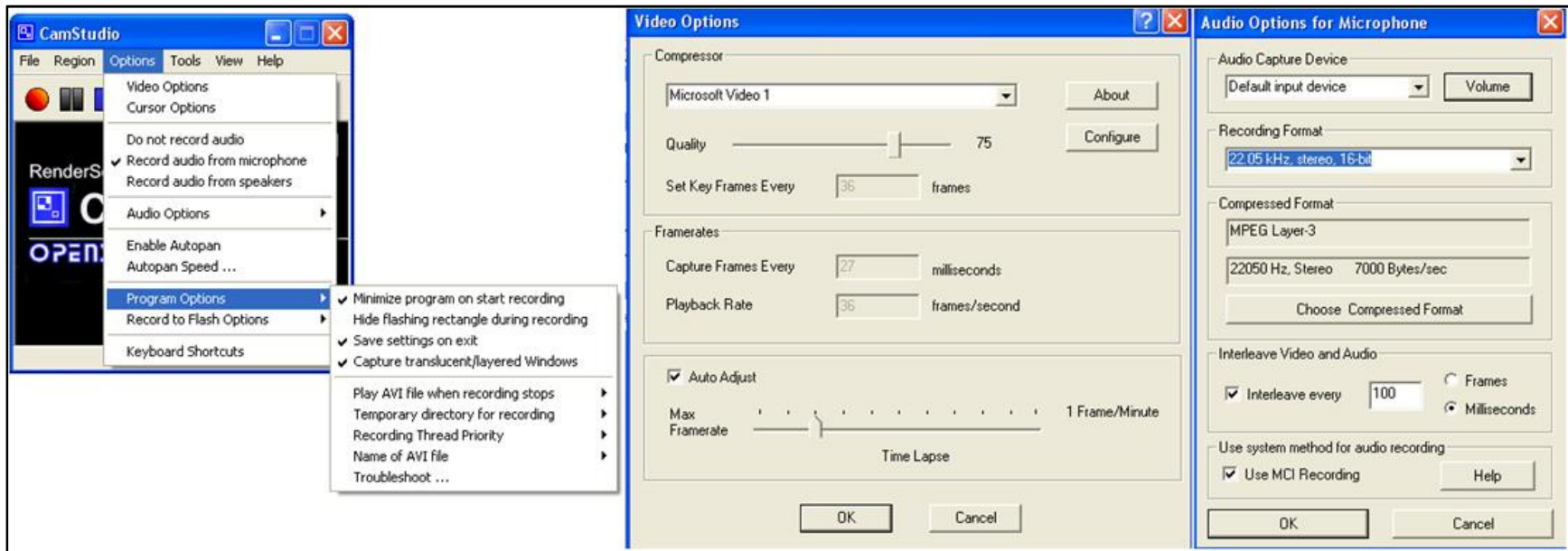


Usability Test Setup for a Stationary Device



Usability Test Setup for a Portable Device

The following image displays the settings required to apply screen capture using the tool CamStudio Portable:



CamStudio Portable settings for screen capture

Appendix 4 – Coding Scheme

A4.1 – Example Coding Scheme

An example usability coding scheme is shown below. For each code in the scheme, the code's definition is provided (which gives a description of when the code should be used by the evaluators during the annotation process), as well as example statements regarding when this coding scheme would be applied. This sample scheme was developed through the analysis of previously used coding schemes defined in past HCI and cognitive literatures (Kushniruk, Triola, Borycki, Stein, & Kannry, 2005) (Kushniruk A. , Analysis of complex decision making processes in health care: cognitive approaches to health informatics, 2001), (Kushniruk & Patel, 1995) (Shneiderman, 2003) (Kushniruk & Patel, 2004) (Kushniruk & Patel, 2005) (Kushniruk, Patel, & Cimino, 1997) (Nielsen J. , 1993) (Nielsen J. , 2005) (Wallden) (Perلمان, 1994).

Usability Problems		
Type	Name	Description
Interface	Aesthetic and Minimalistic Design	Coded if the user specifically mentions the aesthetic look and feel of the system (i.e. the system's graphics, layout, color, etc.) (e.g. "I find this page to be way too cluttered")
	Consistency of Operations	Coded if there are inconsistencies between system operations (i.e. the system uses: inconsistent menu layouts, different control labels for the same control, etc.) (e.g. "Why is the exit button over here now?").
	Data Entry	Coded if the user has difficulty entering data into the system (e.g. if the auto text feature of a data entry field hinders the users typing)
	Feature Not Present	Coded if a feature requested in the scenario is not supported by the tested system (e.g. if the scenario calls for a drug contraindication lookup but the system cannot provide this).
	Flexibility	Coded if the user states that the system is unable to customize to their needs (e.g. "It would be nice if I could rearrange the menu in order of how often I use the features").
	Help and Documentation	Coded if the user expresses a need to learn more about a system aspect but cannot, because the help documentation is either ambiguous or not present (e.g. "I wish I knew how do this").
	Navigation	Coded when a subject shows difficulty navigating through the system or expresses difficulty in finding a system component (e.g. "Where is the main menu link?")
	Overall ease of use	Coded when the user expresses difficulty using the system (e.g. "I find this system very hard to use")
	System Responsiveness	Coded if the user mentions that the response time of the system is slow (e.g. "I have to wait forever for this screen to close")
	Task Speed	Coded if the user states that a system task takes longer than desired (e.g. "Using my old system, I could get this done much faster).

	Understandability	Coded if the system doesn't present data to the user in a language that the user understands (i.e. unclear: error messages, labels, system instructions, etc.) (e.g. "what happens if I press this button?").
	Visibility	Coded if the user is confused about what the system is doing or what the user can do (e.g. "Is the computer loading the program or not?" or "What am I supposed to do with this?").
Content	Database	Coded if the system database doesn't contain the desired data (e.g. A brand of drug that the user wishes to prescribe is not included in the system").
	Default	Coded if the system provides an incorrect default selection (e.g. The default dose for a drug is a dangerous level for most patients).

Errors		
Error	Mistake	Coded if the user makes an error during the system testing and does not correct the actions.
	Slip	Coded if the user makes an error during the system testing, but catches and corrects the error.
Prescription Error Type	Type	The wrong medication is selected
	Dose	The wrong dose is selected
	Frequency	The wrong frequency is selected
	Route	The wrong route is selected
	Period	The wrong period is selected

A4.2 – Annotated Transcript Example

An extraction from a scenario transcript that utilized the above coding scheme is provided below (the patient and case used in this transcript are based on a fictitious scenario):

“(0:00:07.8)<Start>Start the recording and state your name, the current date, the name of the system that is being tested, the name of the scenario that is being tested, and the test condition<End>(0:00:19.4)

(0:00:22.2)<Start>Create a patient record for Frank Glass<End>(0:00:51.5)

(0:00:52.1)<Start>Create and load an encounter/appointment record for Frank<End>(0:01:06.9)

(0:01:33.2)<Start>Update Frank’s medication history to include the Altace, Lipitor, Apo-Acebutolol, and Coumadin prescriptions

(0:01:41.8)Consistency of Operations (or Navigation)- different menu for adding new medications takes extra time for user to remember where to locate it (have to right click to bring up this menu)(0:01:48.3)

(0:02:00.3)Default - Altace default differs from scenario. The fact that it's brought up (though not selected) makes it a default though, so this may not even be an issue (0:02:09.6)

(0:02:43.0)Default - Second 20mg Lipitor not selected for second Lipitor prescription (0:02:45.6)

(0:03:02.1)Mistake - Dose - Default - should be 2x400mg, but the user selected 1x400mg instead (default was 1x100mg) (0:03:08.1)

(0:03:51.1)Overall ease of use - For the Coumadin prescription, the user said "OK, so this is a sliding scale one, which will be hard to put in there (in the system). So I'm just going to put in his tablets of 1mg and 2 mg tablets, for now." Mistake - enters the historic Coumadin tablets as 1mg and 2mg, when the patient was taking 3mg, 3mg, and 1mg(0:04:41.8)

(0:04:45.4)<End> Update Frank’s medication history to include the Altace, Lipitor, Apo-Acebutolol, and Coumadin prescriptions

(0:04:50.8)<Start>Renew the Altace, Lipitor, and Apo-Acebutolol prescriptions for 4 months.

(0:05:51.1)Feature not present - system only allows to enter days for the duration of a prescription (not months) (0:06:14.3)

(0:06:11.8)<End>Renew the Altace, Lipitor, and Apo-Acebutolol prescriptions for 4 months.”

Appendix 5 –Definitions

The following words are defined to provide clarity for the Usability Benchmarking Study Planner Workbook:

Usability – the learnability, ease of use, efficiency, safety, and effectiveness of a system

EMR – Electronic Medical Record – the electronic system used within a clinical practice to collect and store a patient’s health information.

EHR – Electronic Health Record – an electronic system that is integrated across clinical practices, used to collect and store a patient’s longitudinal health information (from cradle to grave).

Heuristic – a rule of thumb that has been established through previous experiences.

Goal – The ultimate aim the end-system user (i.e. order a prescription)

Task/Process – a set of actions that are performed to achieve some goal.

Action – The physical steps required to fulfill a task (i.e. for the task of logging into the system, the user may have to complete the following actions: clicking on the start-up icon, type the login details, click on the OK button, etc.).

Normal Condition User Test– The system user performs the system task as they normally would in an actual case.

Think Aloud Condition User Test – The system user performs the system task as he/she verbally describes their thought processes throughout the process (i.e. what they’re doing, why they’re doing it, what they’re looking for, what is bothering them, what makes them happy, etc.).

Appendix 6 – Example Scenario (for an ePrescription System)

Scenario X – Anthony Rodin – Severe back pain –Controlled Prescriptions

- ❖ **Start the recording and state your name, the current date, the name of the system that is being tested, the name of the scenario that is being tested, and the test condition**

Anthony Rodin was born August 18th, 1973. He is a life-time smoker and has been attending your clinic for the past year. Anthony has been suffering from degenerative disc disease since his mid twenties. Anthony has lived in severe pain since the onset of his condition.

- ❖ **Create a patient record for Anthony Rodin**
- ❖ **Create and load an encounter/appointment record for Anthony**

Two years ago, due to failed rehabilitation attempts and under guidance from his spinal specialist, Anthony had his L-4 and L-5 vertebrae surgically fused. Anthony enjoyed a three month period of reduced pain after recovering from the surgery; however, shortly thereafter, Anthony developed Pseudarthrosis on the fuse site, and is now in more pain than ever. Anthony must wait nine months for his surgery to repair the unhealed fusion. For the past month, Anthony has been taking MS-IR, (morphine sulphate), as needed to help minimize his pain.

- ❖ **Update Anthony’s medication history to include:**
 - **1x10mg MS-IR (morphine sulphate), PO, Q6H, PRN**

After learning that Anthony was able to successfully manage his pain using a consistent daily dose of 40mg, you decide to switch Anthony to an extended release morphine sulphate to help improve his round-the-clock pain management and to reduce the risk of dependency.

- ❖ **Cancel Anthony’s MS-IR prescription, listing the reason: “Changing prescription to an extended-release morphine sulphate”**
- ❖ **Create a prescription for 1x20mg MS-Contin (morphine sulphate), Q12H, for 2 weeks (both in the ePrescribe system and on the required controlled drug prescription form)**
- ❖ **Record the reason for the MS-Contin (morphine sulphate) prescription as being for “back pain”**
- ❖ **Electronically authorize (sign off on) Anthony’s prescription and save it to the test folder (if possible).**

Anthony thanks you for your time as he leaves your office.

Appendix 7 – Excerpt Example of a Scenario Feature Tracking Table

Req. ID	Category	Sub-category	Requirement Description	Example (e.g.)	Tested In Scenario
E-289	Medication Renewals	Manage Renewals	Support the process of managing renewal requests.	e.g. Office Staff receive medication renewal requests from patients, pharmacists and nurses by phone, fax, electronic and generate renewal requests in EMR Application for physician to review and sign off. Once the physician signs off, the office staff are alerted and can call pharmacy.	1,9
E-290	Medication Renewals	Manage Renewals	Allow providers to renew prescriptions easily from a list of active renewal requests.	e.g. Provides a list of renewal requests. e.g. Can cluster requests by patient for better workflow.	1,5,9
E-291	Medication Renewals	Manage Renewals	Allow direct access to the patient summary and chart from the renewal request to review recent medical history, prescription patterns, prescription usage, allergies, alerts, etc.	e.g. Will show allergies, will show if previous prescription has not expired during renewal process.	1,5,9
E-292	Medication Renewals	Manage Renewals	Ability to document that patient or pharmacy has been notified of follow-up associated with medication renewals.	e.g. Document that the pharmacy was faxed or phoned.	9

Appendix 8 – After Scenario Questionnaire (ASQ)

A8.1 - Instructions

The ASQ, developed by (Lewis, 1995), is to be given to a study subject after he/she has completed a normal condition scenario. The user is to circle their answers using the provided 7 point scale (the lower the selected score, the higher the subject's usability satisfaction with their system). After the user has completed the ASQ, the ASQ score can be calculated by taking the average (arithmetic mean) of the 3 questions. If a question is skipped by the subject, the ASQ can be calculated by averaging the remaining scores.

A8.2 – ASQ

The following was developed by (Lewis, 1995):

1. Overall, I am satisfied with the ease of completing this task.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

2. Overall, I am satisfied with the amount of time it took to complete this task.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

3. Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing this task.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

(Lewis, 1995)

Three supplementary ASQ questions were developed for usability benchmarking ePrescription studies. These questions can be added to the ASQ question set if deemed necessary for a given study.

Supplementary Questions

1. How would you rate the difficulty of completing this scenario with your ePrescribe system?

VERY EASY 1 2 3 4 5 6 7 **VERY DIFFICULT**

2. How would you rate the difficulty of completing this scenario using a paper based system?

VERY EASY 1 2 3 4 5 6 7 **VERY DIFFICULT**

3. Overall, when completing this task, I feel that my ePrescribe system improves patient care.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

Appendix 9 – Post Study System Usability Questionnaire (PSSUQ)

A9.1 - Instructions

The PSSUQ is provided to the subject after they have completed all normal condition scenarios. Like the ASQ, the PSSUQ requires that the user circle their response to each question based on a 7-point scale (where the lower the response, the higher the subject's usability satisfaction with their system). The subject can also clarify their answers on the PSSUQ by adding comments in the provided spaces. After the subject has completed filling out the PSSUQ, it is good practice for the analyst to quickly go over the subject's answers in order to make sure the subject hasn't missed anything and that all comments are understood.

The PSSUQ can be used to produce the following measures:

- OVERALL – Overall user satisfaction with their system – calculated by taking the average of questions 1-19
- SYSUSE – System usefulness – calculated by taking the average of questions 1-8
- INFOQUAL – Information quality – calculated by taking the average of questions 9-15
- INTERQUAL – Interface quality – calculated by taking the average of questions 16-18

A8.2 – PSSUQ

The following was developed by (Lewis, 1995):

1. Overall, I am satisfied with how easy it is to use this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

2. It was simple to use this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

3. I could effectively complete the tasks and scenarios using this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

4. I was able to complete the tasks and scenarios quickly using this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

5. I was able to efficiently complete the tasks and scenarios using this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

6. I felt comfortable using this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

7. It was easy to learn to use this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

8. I believe I could become productive quickly using this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

9. The system gave error messages that clearly told me how to fix problems.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

10. Whenever I made a mistake using the system, I could recover easily and quickly.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

11. The information (such as on-line help, on-screen messages and other documentation) provided with this system was clear.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

12. It was easy to find the information I needed.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

13. The information provided for the system was easy to understand.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

14. The information was effective in helping me complete the tasks and scenarios.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

15. The organization of information on the system screens was clear.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

Note: *The interface includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the screens (including their use of graphics and language).*

16. The interface of this system was pleasant.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

17. I liked using the interface of this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

18. This system has all the functions and capabilities I expect it to have.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:

19. Overall, I am satisfied with this system.

STRONGLY AGREE 1 2 3 4 5 6 7 **STRONGLY DISAGREE**

COMMENTS:"

- (Lewis, 1995)

Appendix 10 – System Automation

Task Set Example

The following system automation task set was drawn from the automation tests of an ePrescription system UB study.

- ❖ **Create and load a patient record and encounter for James Druken, born February 23, 1971**

Automated Feature	Test	Target Response
Tests drug : drug interactions	<ul style="list-style-type: none"> ❖ Create a prescription for 1x3mg Coumadin (warfarin), QD, expires in two weeks ❖ Create a prescription for 1x500mg Biaxin (clarithromycin, Q12H, expires in 2 weeks) 	<p>A warning should be issued to alert the physician that a possible drug-drug interaction would occur between the Coumadin (warfarin) and the Biaxin (clarithromycin)</p> <p>A prompt should also inform the physician about the monitoring parameters of warfarin (T-9)</p>
Tests drug : disease interactions	<ul style="list-style-type: none"> ❖ Enter Congestive Heart Failure into James' record ❖ Create a prescription for 2x500 Glucophage (metformin), BID, expires in 3 weeks 	<p>A warning should be issued to alert the physician that Glucophage should not be given to a patient with Congestive Heart failure</p>
Tests drug : allergy interactions	<ul style="list-style-type: none"> ❖ Enter Penicillin as an allergy into James' record ❖ Create a prescription for 2x500mg Amoxil (amoxicillin), Q8H, for 10 days 	<p>An allergy warning should be thrown by the ePrescribe system, as Amoxil (amoxicillin) is in the same drug class as Penicillin</p>
Calculates total number of days duration for a prescription if requested in # of doses where appropriate / possible	<ul style="list-style-type: none"> ❖ Create a prescription for 1x250mg of E-Mycin (erythromycin), Q8H, for 27 doses 	<p>The system should calculate the expiry date to be 9 days away</p>
Generates paper prescription with provider / clinic details as per CPSBC requirements	<ul style="list-style-type: none"> ❖ Print the above E-Mycin (erythromycin) prescription 	<p>The printed prescription meets with CPSBC requirements</p>

Appendix 11 – UB Study Subject Reference Sheet

The following study subject reference sheet was utilized for an ePrescription system UB study:

- UB studies capture data pertaining to a system’s performance and usability (learnability, ease of use, efficiency, safety, and effectiveness) by recording the actions taken by system users during the fulfillment of realistic use case scenarios.
- You (the user) will complete two sets of ePrescribe scenarios:
 - **Normal condition scenarios** – where you complete the scenarios under normal working conditions
 - At the end of each normal condition test, you will be asked to fill out a brief After Scenario Questionnaire (duration: app. 2 minutes)
 - Once you have completed all of the normal condition tests, you will be asked to fill out Post-Study System Usability Questionnaire (duration: app. 5-10 minutes)
 - **Think aloud scenarios** – where you complete scenarios while ‘Thinking Aloud’ (verbally stating your thought processes as you complete the scenario tasks)
 - Think aloud comments are analogous to the comments made by a sports play-by-play announcer (all actions are stated aloud as they happen in real time). Such comments will involve: what you are doing (i.e. I’m looking for the file menu); anything that you find confusing (i.e. I’m not sure what to do here); anything you dislike about the system or process (i.e. “It takes too long to do this”); and anything you would like to change about the system or process (i.e. “I wish the system had this feature”).
- Both the normal condition and ‘Think Aloud’ scenarios are described with three forms of notation:
 - **Contextual Information** (written in paragraph form) – describes the scenario background information (i.e. who the patient is, what their medical background is, etc.)
 - **Action Points** (written with green bullet points) – describe tasks for you to perform, using your ePrescribe system
 - ❖ **This is an example of an action statement.**
 - If you are not able to perform the action point with your ePrescribe system, state aloud: “Due to system limitations, I cannot perform this task” or “I don’t know how to perform this task”, and move onto the next task.
 - You may not agree with the action or actually need to perform the given action (i.e. you may be asked to lookup information that you already know); however, for the purpose of standardizing the tests, it is asked that you follow through with all indicated actions.
 - **Footnotes** (written in the footnotes section of the scenario) – contain additional pieces of information about some of the action points (i.e. information that you are meant to retrieve) or they may contain explanations regarding why a certain action is given.
- At the start of each scenario, state: your name; the current date; the name of the system being tested; and the name of the scenario and condition being tested
- To make the encounters more realistic, you may find it helpful to document patient encounter notes during the scenarios; however, such notes are not essential to the completion of the scenarios
- Create a test folder in which you can save all of the prescriptions that are created in the scenarios. These prescriptions can then be used for future analysis.
- After each scenario, you will be recorded and asked to comment on any additional thoughts you may have about the scenario. This time can be used to: clarify your actions; add any relevant information about the system component that was tested; state what you liked/didn’t like about the scenario, etc.