NHS Connecting for Health	ePrescribing System Evaluation – Usability Approach Summary			
	Programme	NPFIT	**DOCUMENT RECORD ID KEY**	
	Sub-Prog / Project	e-Prescribing	***NPFIT-EP-DB-0028.01***	
	Prog. Director	Tim Donohoe	**Status**	**Final**
	Owner	Ann Slee	**Version**	1.0
	Author	Kit Lewis	**Version Date**	11th Sep 08

ePrescribing Benchmarking – Summary of Follow-on Usability Review

Contents

1. Approach

Introduction

This usability review is part of a broader review of existing electronic prescribing IT systems currently being carried out by the ePrescribing team within NHS Connecting For Health. The main body of the review aims to assess the functionality of these systems (ie. what their capabilities are). Further to this, the usability review aims to assess how end-users (ie. healthcare professionals engaged in medicines management activities) perceive and use this functionality, and in particular to see if the design of the systems might increase these users' propensity to make errors in patient care.

Given the limited time available to review each system, and the large size and high complexity of the systems, usability testing was not a feasible method for evaluation. Therefore, a heuristic evaluation was undertaken alone. However, an evaluation using a typical industry-standard set of usability heuristics (see Appendix A & B for examples) would have also taken a very long time, and would not have targeted specifically the most important question – viz. "Does the design of these systems in any way increase users' propensity to make errors?"

Healthcare IT-specific heuristics – the "error trap" approach

So, a new set of heuristics was developed, building from the industry-standard sets but focusing on situations which could reasonably be thought to contribute to user error.

This reduced the list of heuristics to the following:

- Visibility of system status
- Match between system and the real world
- Consistency and standards
- Error prevention
    - NB This heuristic is the most important in the healthcare context, and also the most generalised. When considering clinical safety, the other heuristics become contributing factors to this overall goal. Therefore it was not used as a heuristic in its own right
- Recognition rather than recall
- Aesthetic and minimalist design
- Help users recognise, diagnose and recover from errors

This set was then correlated with specific heuristics relating to known healthcare-related IT system design problems, or "situations likely to cause error". These were derived from the following sources (see Appendix C for full reference details):

1. THEA: A Technique for Human Error Assessment Early in Design
2. Various academic papers on human error in clinical IT systems
3. 4 years of user interface design and patient safety assessments as part of the Common User Interface (CUI) programme
4. Observation of existing electronic prescribing and EPR systems within the NHS
5. Anecdotal evidence from discussion with NHS healthcare professionals

This resulted in the following set of heuristics or "error-prone situations":

A - Dissociating a task from its object
- Incorporating
    - Visibility of system status

B - Disguising the switching of context
- Incorporating
    - Visibility of system status

C - Confusing the sequence or state of a process
- Incorporating
    - Visibility of system status

        - Match between system and the real world
        - Consistency and standards

D - Over-complicating the selection of an item or parameter
- Incorporating
    - Consistency and standards
    - Aesthetic and minimalist design

E - Warning the user inappropriately
- Incorporating
    - Aesthetic and minimalist design
    - Help users recognise, diagnose and recover from errors

F - Creating the illusion of support
- Incorporating
    - Visibility of system status
    - Match between system and the real world

G - Creating the illusion of completeness
- Incorporating
    - Visibility of system status
    - Aesthetic and minimalist design

H - Using confusing or inconsistent words, symbols and layout
- Incorporating
    - Consistency and standards
    - Aesthetic and minimalist design

J - Overloading the user with information
- Incorporating
    - Aesthetic and minimalist design

K - Not providing feedback in response to user input
- Incorporating
    - Visibility of system status

L - Requiring memory or calculation within and between screens
- Incorporating
    - Recognition rather than recall

The PowerPoint presentation which accompanies this document (included below) explains each of these "error-prone situations" in more detail, and provides examples.

Error-traps_GENERIC

Scoring method

In order to produce a numerical score for each system at the end of each evaluation, a scoring system was devised.

1. Weighting of each heuristic - The goal for this system was to weight each of the heuristics according to a subjective assessment of the severity of real-life clinical errors which might result from a user encountering this situation within an ePrescribing system. Please note that in each case, the severity of the situation is in relation to its place in the overall clinical workflow within secondary (ie. hospital-based) care.

20% weighting - Potential to result in severe / fatal errors:
- A - Dissociating a task from its object
- B - Disguising the switching of context

13% or 10% weighting - Potential to result in major errors:
- D - Over-complicating the selection of an iterm or parameter
- E - Warning the user inappropriately
- L - Requiring memory or calculation within and between screens

7% or 4% weighting - Potential to result in minor errors:

- C - Confusing the sequence or state of a process
- F - Creating the illusion of support
- G - Creating the illusion of completeness
- H - Using confusing or inconsistent words, symbols and layout
- J - Overloading the user with information
- L - Not providing feedback in response to user input

2. Grading each problem instance – In addition to the weighting above, each instance of a problem encountered was scored according to its severity. Each instance subtracted from the "perfect score" (ie. no error-prone situations found) of 100%.

Severe problem – subtract 30%
Major problem – subtract 15%
Minor problem – subtract 5%

3. Total scores – for each system, the problems encountered were recorded and scored within a spreadsheet which totalled the score for each heuristic, then weighted and totalled these scores to produce a single final percentage score (100% = no error-prone situations found).

In all cases the actual numerical scores is less important than the differences between the scores for each system, and the prevalence across all systems of the different kinds of error-prone situation. The focus was on applying the criteria as consistently as possible across all of the systems assessed, rather than on the final numbers.

2. Process

The usability review was conducted in conjunction with a broad functionality assessment, part of which was a scripted demonstration during which NHS clinicians were asked to score detailed aspects of each system's performance while following a realistic clinical scenario.

This demonstration played an important part in the usability review process, as it allowed the reviewer to become familiar with the (very complex) systems, and to get an initial sense of where issues might lie. Without this step, the usability review sessions would have been much more difficult.

The overall sequence followed in performing the usability review is as follows:

1. Reviewer attends scripted demonstration
    a. Overall understanding of system & initial review
2. Reviewer visits system supplier and performs detailed review
    a. Detailed review, Q&A, Discussion
3. Review produces first draft of scoring
4. Supplier reviews first draft
5. Discussion & clarification of findings
6. Reviewer finalises report

During the detailed usability review (Step 2, above) the typical process was to return to the demonstration scenario and discuss previously-noted questions and issues, before spending some hands-on time with the system and discussing the most salient initial findings with the supplier. This session typically took 3-4 hours. A key part of the review process was the implicit understanding with the system suppliers that nothing was "off limits" – all the suppliers were 100% co-operative with this approach.

Appendix

A: Jakob Nielsen usability checklist headings (1990-1994)

http://www.useit.com/papers/heuristic/heuristic_list.html (accessed 14-Sep-2008)

1. **Visibility of system status** - The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. **Match between system and the real world** - The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom** - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. **Consistency and standards** - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention** - Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall** - Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use** - Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. **Aesthetic and minimalist design** - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. **Help users recognize, diagnose, and recover from errors** - Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. **Help and documentation** - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

B: Xerox usability checklist headings (1996)

http://www.stcsig.org/usability/topics/articles/he-checklist.html (accessed 14-Sep-2008)

1. **Visibility of system status** - the system should always keep the user informed about what is going on
2. **Match between system and the real world** - the system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control & freedom** - Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.
4. **Consistency & standards** - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Help users recognise, diagnose and recover from errors** - Error messages should be expressed in plain language (NO CODES).

6. **Error prevention** - Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
7. **Recognition rather than recall** - Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
8. **Flexibility and minimalist design** - Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the "average" user (e.g., physical or cognitive ability, culture, language, etc.)
9. **Aesthetic and minimalist design** - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
10. **Help and documentation** - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
11. **Skills** - The system should support, extend, supplement, or enhance the user's skills, background knowledge, and expertise ----not replace them.
12. **Pleasurable and respectful interaction with the user** - The user's interactions with the system should enhance the quality of her or his work-life. The user should be treated with respect. The design should be aesthetically pleasing- with artistic as well as functional value.
13. **Privacy** - The system should help the user to protect personal or private information- belonging to the user or the his/her clients.

C: Specific references

1. **THEA: A Technique for Human Error Assessment Early in Design**
    a. Pocock, S., Harrison, M., Wright, P. & Johnson, P. (2001). THEA: A technique for human error assessment early in design. In Hirose, M., editor, Human-Computer Interaction INTERACT'01 IFIP TC.13 International Conference on human computer interaction, pages 247-254. IOS Press.
    b. http://homepages.cs.ncl.ac.uk/michael.harrison/papers/int01pub4.pdf
2. **THEA – A Reference Guide**
    a. Steven Pocock, Bob Fields, Michael Harrison, Peter Wright; Human-Computer Interaction Group - Department of Computer Science - University of York, 2001
    b. http://homepages.cs.ncl.ac.uk/michael.harrison/papers/thea.pdf
3. **Common User Interface – medications management master hazard log**
    a. 28-Jul-2008; Common User Interface programme team (contact: cuistakeholder.mailbox@nhs.net)

4. **Common User Interface – medications management design guidance**
    a. 2007-8; Common User Interface programme team team (contact: cuistakeholder.mailbox@nhs.net)
    b. Guidance available from www.mscui.net on the following topics:
        i. Medication line (display of a single prescription / medication order)
        ii. Medications views (display of lists of current and past prescriptions)
        iii. Drug chart (display and recording of medicines administration events)
        iv. Search & prescribe (prescribing medications)
5. **Academic references**
    a. Koppel, R. et al (2005) Role of computerized physician order entry systems in facilitating medication errors. JAMA, 293, 1197-1203
    b. Edworthy, J & Hellier, E. (2004) What makes a good alarm? Presentation at the NPSA, London 22nd December 2004.

c. McManus, B. (1999) A move to Electronic Patient Records in the Community: a qualitative case study of a clinical data collection system, the problems caused by the inattention to humans and human error. HECS99.
d. Lin. L et al (1998) Applying human factors to the design of medical equipment: patient-controlled analgesia. J Clin Monit, 14, 253 - 263
e. Sawyer, D. (1996) Do It By Design: An introduction to Human Factors in Medical Devices. US Dept. of Health and Human Services, FDA.
f. Avery, A. J. et al (2005) Identifying and establishing consensus on the most important safety features of GP computer systems: e-Delphi study. Informatics in Primary Care, 13, 3-11.
g. Zhang, J. et al (2003) Using usability heuristics to evaluate patient safety of medical devices. J Biomedical Informatics, 36, 23-30.
h. Nielsen, J. (2005) Medical Usability: How to kill patients through bad design. Alertbox 11th April 2005. (content based on source 1)
i. Ash et al (2004) Unintended consequences of information technology. J American Medical Informatics Association, 11, 104 - 112.
j. Fields et al (1997) THEA: Human error analysis for requirements definition. Technical Report YCS 294. Dept Computer Science, University of York.
k. Garnerin et al (2007) Measuring human-error probabilities in drug preparation: a pilot simulation study. Eur J Clin Pharmacol (2007) 63:769–776
l. … & various ISMP safety alert newsletters available from http://www.ismp.org/newsletters/default.asp