

Evaluation

James Eagan <james.eagan@telecom-paristech.fr>

Ce cours a été développé en partie par des membres des départements IHM de Georgia Tech et Télécom ParisTech. La liste de contributeurs inclut Gregory Abowd, Al Badre, James Eagan, Jim Foley, Elizabeth Mynatt, Jeff Pierce, Colin Potts, Chris Shaw, John Stasko, et Bruce Walker. Ces matériaux peuvent être utilisés avec attribution pour des buts non-lucratifs.



Dernière mise à jour : février 2015.



Outline

- Overview
- Evaluation Prédictive
- Conception d'une expérience
- Participants, IRB & La Morale
- Collecte de données
- Analyse & interprétation de résultats

Pourquoi Evaluer ?

- Suppose :
 - Utilisateurs et tâches identifiés
 - Besoins connus
 - Interface conçue, prototype construit
- Mais ... est-ce que ça permettra à l'utilisateur d'exécuter ses tâches ? Est-ce mieux que la version précédente ?

THE PROJECT STATUS IS "YELLOW LIGHT."



S. ADAMS E-mail: SCOTTADAMS@aol.com

IN USER TESTS WE FOUND THAT THE PRODUCT LOCKS UP EVERY TWELVE SECONDS. THE INTERFACE IS INCOMPREHENSIBLE AND THE MANUAL IS PURE FICTION.



1/18 © 1996 United Feature Syndicate, Inc. (NYC)

I THINK IT'S CLEAR WHAT WE NEED TO DO...

SHIP IT AND HOPE SOMEBODY WRITES A "DUMMIES" BOOK ABOUT IT?



Genres d'Evaluation

- Interprétive & Prédicative
- Sommative & Formative

Evaluation Prédicative

- Recueillir des données sur l'utilisabilité d'un design pour *un groupe défini* d'utilisateurs *pour une activité définie* dans un *contexte défini*

Buts

- Evaluer la fonctionnalité du système
- Evaluer l'effet de l'interface sur l'utilisateur
- Identifier problèmes particuliers avec le système

Deux Genres

- Evaluation Formative
 - Au début du projet, pendant sa durée de vie
 - Evaluation de la conception
- Evaluation Sommative
 - Après la construction du système. Jugement finale.
 - Evaluation de l'implémentation

Approches

- Expérimental
 - Dans le labo, environnement contrôlé
 - Plutôt quantitatif
- Naturaliste
 - Observation dans un contexte réel
 - Plutôt qualitatif

Compromis

Experimental

- + Reproductibilité
- + Plus “objectif”
- Cher, besoin de vrais utilisateurs & labo
- Réaliste ?

Naturaliste

- + Validité écologique
- + Peu cher, rapide
- Pas reproductible, résultats particuliers aux utilisateurs
- Pas quantitative

Méthodes d'Evaluation

- Expérimentales, Observationnelles
 - Typiquement avec utilisateurs
 - Expériences à partir de spécification d'utilisabilité
- Prédictive
 - (sans utilisateur)

Tout à l'heure

Evaluation Prédicative

- Idée :
 - L'observation d'utilisateurs coûte cher, prend du temps
 - Essayons de prédire l'utilisation plutôt que de l'observer
 - Economiser ressources (rapide, coût réduit)

Approche

- Revues d'experts
 - Un pro de l'IHM (pas de vrais utilisateurs) interagisse avec le système, essaie de trouver de problèmes potentiels
- Idéalement :
 - N'a pas encore utilisé d'autres prototypes
 - Connait bien le domaine ou la tâche
 - Comprend bien perspectives utilisateur

Méthodes de l'Evaluation Prédictive

- Evaluation Heuristique
- « Discount » tests d'utilisabilité
- Cognitive Walkthrough (« passage à travers cognitif »)

Evaluation Heuristique

Développé par Jakob Nielsen
(www.useit.com)
Plusieurs experts d'utilisabilité
évaluent le système selon des
heuristiques simples et généraux



http://www.useit.com/papers/guerrilla_hci.html

Méthode

- Déterminer les inputs
- Evaluer le système
- Collecte d'observations
- Trie de sévérité

Déterminer Inputs

- Qui sont les experts ?
 - Apprendre le domaine, ses pratiques
- Quel est le prototype à évaluer ?
 - Mock-ups, storyboards, ... ou bien un système qui tourne

Méthode d'évaluation

- Reviewers évaluent le système selon les heuristiques haut-niveau (principes d'utilisabilité)

- Visibilité de l'état du système
- Concordance entre le système et le monde extérieur
- Contrôle & liberté pour l'utilisateur
- Standards & cohérence
- Prévention d'erreurs
- Reconnaissance plutôt que rappel
- Flexibilité & efficacité d'usage
- Design esthétique & minimalist
- Aider l'utilisateur à reconnaître, identifier & se rattraper face à une erreur
- Aide & documentation

Méthode d'évaluation

- Reviewers évaluent le système selon les heuristiques haut-niveau (principes d'utilisabilité)

Use simple and natural dialog

Speak user's language

Minimize memory load

Be consistent

Provide feedback

Provide clearly-marked exits

Provide shortcuts

Provide good error messages

Prevent errors

Processus

- Effectuer au moins deux itérations du système :
 - Regarder chaque écran
 - Le flot d'écran à écran
- À chaque étape, évaluer selon les heuristiques
- Chercher de problèmes :
 - Subjectif (si tu pense que ce l'est, ce l'est)
 - T'inquiète pas si c'est un vrai problème

Debriefing

- Récolter tout problème identifié par les reviewers
 - Identifier quels sont ou ne sont pas de vrais problèmes
 - Grouper, structurer
 - Documenter et enregistrer les problèmes

Classer par Sévérité

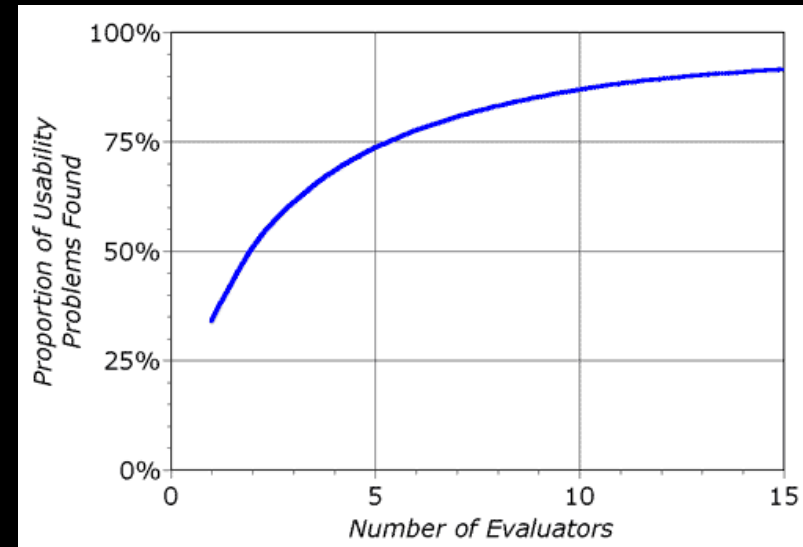
- Echelle de 0 à 4
 - 4 est le plus sévère
- Selon :
 - Fréquence
 - Impact
 - Persistance
 - Impact sur le marché

Avantages

- Peu cher, bon pour des PME qui ne peuvent pas payer plus
- Trouver quelqu'un avec de l'expérience dans la méthode est l'idéale

Application

- Selon Nielsen, 5 évaluations trouvent 75 % de problèmes
- Au dessus, on en trouve plus, mais moins efficace



Problèmes

- Evaluation très subjective
 - Dépend l'expertise des reviewers
- Ces heuristiques, sont-elles les bonnes ?
 - D'autres ont été proposées
- Comment déterminer ce qui est un *vrai* problème d'utilisabilité
 - Quelques "problèmes" identifiés ne le sont pas

Discount Usability Testing

- Hybride de tests d'utilisabilité empirique et de l'évaluation heuristique
- Effectuer 2 ou 3 séances “penser à l'orale” avec un prototype (sur papier ou mock-up)

Mockups

- Pas besoin que les mockups soient parfait
- Plusieurs approches :
 - Doivent être rapide à créer, pas trop cher
 - Souvent des sketches
 - Limitations des papier : emphase sur éléments visuels
 - Parfois difficile à utiliser dans le test

Cognitive Walkthrough

- Mesurer facilité d'apprentissage et d'utilisabilité en simulant comment un utilisateur explore et apprend le système
- Une "expérience de pensée" d'utilisabilité
- Comme "code walkthrough" dans la génie logiciel
- Proposé par Polson, Lewis, et al à UC Boulder



CW Processus

- Construire tâches conçues à partir de la spécification du système ou du mockup
- Walk through (itérer sur) les activités nécessaires pour aller d'écran à écran
- Examiner actions nécessaires pour chaque tâche, essayer de prédire comment un utilisateur fonctionnerait et quels problèmes il trouverait

Besoins

- Description d'utilisateurs et leur contexte
- Description de tâches
- Liste complète d'actions exigées pour effectuer chaque tâche
- Prototype ou description du système

Suppositions

- L'utilisateur a déjà une idée brut quoi faire
- L'utilisateur explore le système, en cherchant d'actions qui peuvent contribuer à effectuer sa tâche
- L'utilisateur choisit l'action qui semble le mieux pour son but
- L'utilisateur interprète les réponses et décide s'il est plus proche à son but

Méthode

- Itérer sur séquence d'actions
 - Action 1
 - Réponse A, B, ...
 - Action 2
 - Réponse A
 - ...
- Pour chaque étape, poser *quatre questions* en essayant de construire une *histoire de croyance*

Questions

- Est-ce que l'utilisateur essaye de produire l'effet associé à cette action ?
- Est-ce qu'il sera capable de remarquer la bonne action ?
- Un fois trouvée, saura-t-il qu'elle est la bonne action pour l'effet souhaité ?
- Comprendra-t-il le feedback après l'avoir effectué ?

Répondre aux Questions

- Est-ce que l'utilisateur essaye de produire l'effet associé à cette action ?
 - Evidence typique
 - Ça fait partie de leur tâche de base
 - Il a de l'expérience avec le système
 - Le système indique qu'il faut le faire
 - Pas d'évidence ?
 - Construire scénario d'échec
 - Expliquer pourquoi

Prochaine Questions

- Est-ce qu'il sera capable de reconnaître la bonne action ?
 - Evidence typique
 - Expérience
 - Dispositif visible (*e.g.* un bouton)
 - Représentation perceptible d'une action tel qu'un item de menu

Prochaine Questions

- Une fois trouvée, saura-t-il qu'elle est la bonne action pour l'effet souhaité ?
 - Evidence typique
 - Experience
 - Item visuel (*e.g.* prompt) pour relier l'action à son effet
 - Toute autre action ne semble pas être la bonne

Prochaine Question

- Comprendra-t-il le feedback après l'avoir effectué ?
 - Evidence typique
 - Expérience
 - Reconnaître un lien entre réponse système et tâche de base

Cette Foix, en Impliquant l'Utilisateur

- Naturaliste vs. Empirique
- Naturaliste
 - Dans un contexte réaliste, typiquement avec un observateur neutre, une observation soigneuse
- Empirique
 - On utilise le système en manipulant de variables dépendante et indépendante

Pourquoi Collecter de Données ?

- Concevoir une expérience pour *récolter de données* afin d'évaluer *d'hypothèses* afin d'évaluer *l'interface* afin d'*améliorer le design*
- L'information produite peut être *objective* ou *subjective*
- L'information peut aussi bien être *qualitative* ou *quantitative*



Quelles sont les plus difficiles à mesurer ?

Piloter une Expérience

- Identifier la tâche
- Identifier les mesures de performance
- Développer l'expérience
- (Obtenir autorisation du bureau d'éthiques)
- Recruter de participants
- Faire passer l'expérience
- Inspecter et analyser les données
- Tirer de conclusions

La Tâche

- Tâches étalonne — collectionnent données quantitatives
- Tâches représentative — rajoutent de profondeur, aident à comprendre le processus
- Quoi, pas Comment
- Questions :
 - Dans le labo ou dans la nature ?
 - Validité — utilisateurs typiques ; tâches typiques ; environnement typique ?
 - Versions pilotes pour trouver de bogues dans l'expé

Tâches Étalonne

- Spécifique, expliquer clairement dans un langage utilisateur
- Exemple : gestion de mél
 - « Cherchez le message de Jean-Louis et répondez, “jeudi matin à 11h.” »
- L'utilisateur fait ces tâches sous plusieurs conditions ; on mesure la *performance*

Performance

- Dépend la tâche
- Mesures spécifiques, objectives
- Exemples :
 - Vitesse (temps de réaction, complétion, ...)
 - Précision (taux d'erreurs, cibles ratées/atteintes, ...)
 - Production (combien de fichiers traités)
 - Score (points gagnés), ...

Variables

- Indépendantes
 - Ce qu'on étudie, ce qu'on veut varié (*e.g.*, fonctionnalité de l'interface, dispositif interactif, technique de sélection, ...)
- Dépendantes
 - Mesures de performance captées (*e.g.*, temps, erreurs, ...)

“Contrôler” une Variable

- Empêcher une variable d’avoir une influence systémique sur les résultats
- Comment le faire :
 - Ne pas la permettre à varier (*e.g.*, que des hommes)
 - Permettre-la à varier aléatoirement (*e.g.*, affecter de participants au groupes au hasard)
 - Contre-balancer la — la varier systématiquement
- Le bon choix dépend les circonstances

Hypothèses

- Ce qu'on pense se passera, la prédiction
- Comment la variable dépendante répondra aux variables indépendantes
- Hypothèse nul (H_0) :
 - Qu'il n'y ait *aucun effet*
 - Données utilisées pour *réfuter* cette hypothèse

Exemple

- L'utilisateur, fait-il cette opération plus vite avec un écran noir-et-blanc ou en avec un couleur ?
 - Indépendante : écran (couleur ou n/b)
 - Dépendante : temps d'exécuter la tâche (secondes)
 - Variables contrôlées : autant d'hommes que de femmes dans chaque groupe
 - Hypothèse : temps de complétion sera plus court avec l'écran couleur
 - $H_0 : \text{Temps}_{\text{couleur}} = \text{Temps}_{\text{nb}}$
- Problèmes ?

Design de l'Expérience

- Within subjects (mesures répétées)
 - Chaque participant fait chaque condition
- Between subjects
 - Chaque participant fait qu'une seule condition

Within vs. Between

- Quels sont les avantages et inconvénients de chaque technique ?

Within Subjects

- Plus efficace :
 - Chaque participant produit plus de données
- Statistiques plus simples :
 - Chaque personne et son propre contrôle
- Moins de participants
- Mais design potentiellement plus compliqué pour éviter d'effets d'ordre

Between Subjects

- Moins d'effet d'ordre
 - Eviter les effets d'apprentissage, de fatigue
- Design, Analyse plus simple
- Plus facile de trouver de participants (une seule condition)
- Moins efficace

Participants

- Ok, on a la tâche, les mesures, le design expérimental, *etc.*
- On a des hypothèses
- Pour avoir les données, il faut donc de participants


COPÉ & CPP

- Comités d'éthiques de la recherche
- Très important dans les pays anglo-saxons et dans d'autres pays européens
- Audit toute recherche impliquant de participants humains
- Assure la sécurité du participant (et donc le chercheur et le labo aussi)
- Pas un audit de la science ; que de sureté et éthiques

Recruter de Participants

- Genres :
 - Volontaires
 - Participants rémunérés
 - Etudiants (parfois rémunérés en crédits de cours)
 - Amis, famille, collègues, connaissances, ...
 - La publique en public (*e.g.*, observer des gens utilisant de bornes vélib)

Participants

- Doivent coller aux utilisateurs (validité)
- Motivation peut avoir une influence (€\$£, importance de la recherche)
- NB : l'Éthiques, COPÉ, principes de consent s'appliquent à tout participant, y compris amis, famille, sujets pilotes

Éthiques

- Une expérience peut être difficile
- Chaque participant doit donner son consentement de passer l'expé
 - Il doit savoir ce que ça implique, à quoi s'attendre, les risques impliqués
- Peut arrêter à n'importe quel moment sans danger ou pénalité
- Chaque participant doit être traité avec de respect

Consentement

- Pourquoi c'est important ?
 - Des gens peuvent être sensible du processus
 - Au cas d'erreurs, le participant peut se sentir pas à la hauteur
 - Peut être mentalement ou physiquement épuisant
- Quels sont les risques potentiels ?

Avant l'expérience

- Soyez prêt afin de ne pas faire perdre le temps du participant
- Assurez que le participant sait que vous testez le logiciel, pas lui
- Respectez la vie privée de l'utilisateur
- Expliquez les procédures sans compromettant les résultats
- Donner une formulaire de consent à signer

Pendant l'expérience

- Assurez le confort du participant
- La séance ne doit pas être trop longue
- Maintenir une ambiance relaxe
- Ne montre pas de mécontentement ou de colère

Après l'expérience

- Décrivez comment la séance aidera à améliorer le système
- Montrez au participant comment faire les tâches ratées
- Assurez l'anonymat du participant (ne jamais identifiez la personne, ne montrez de photos, vidéos qu'avec l'accord explicite du participant)
- Stockez les données anonymement, sûrement ; détruisez-les au bout du projet

Théorie d'attribution

- Une théorie sur pourquoi des gens pensent qu'ils ont réussi ou raté
- Expliquez comment des erreurs ou échecs ne sont pas de problèmes du participant, ce sont des parties de l'interface qu'il faut améliorer

Collecte de Données

Collecte de Données

- Méthodes, techniques
- Données objectives
- Données subjectives
- Données quantitatives
- Données qualitatives

Inspecteur

- Faire de l'évaluation est comme être inspecteur
- But : collectionner de l'évidence qui peut déterminer la validité des hypothèses
- Evidence doit être :
 - Relevant
 - Diagnostique
 - Crédible
 - Corroborée



Données comme Evidence

- Relevant
 - A propos d'hypothèses
 - e.g., mesurer la taux d'erreurs, donne-t-il une meilleur vue de si un système de contrôle aérienne gère les tâches utilisateur ?
- Diagnostique
 - Données montre sans ambiguïté la réponse
 - e.g., demander les préférences de l'utilisateur, montre-t-il si le système marche mieux ? (peut-être)

Données comme Evidence

- Credible
 - Les données sont-elles crédible ?
 - Assez nombreuses, recueillies avec soin
- Corroboré
 - Y a-t-il plus qu'une source de données qui soutient les hypothèses ?
 - e.g., Précision et l'avis des utilisateurs montrent que le nouveau est plus facile à comprendre que l'ancien, mais le temps de complétion est plus lent

Conseils

- Utiliser des données objectives *et* subjectives
- Utiliser plusieurs mesures dans un genre
 - e.g., temps de réaction et précision
- Quand possible, utiliser de mesures quantitatives

Données à recueillir

- Démographique
 - Qui est le participant, pour grouper ou pour corrélation aux autres mesures
 - e.g., gaucher ; âge ; langue maternelle ; jeux vidéos
- Données quantitatives
 - Ce que vous mesurez (temps de réaction ; nombre de fois bâillé)
- Donnée qualitatives
 - Description, observations

Préparer

- Quelles données faut-il ?
 - Dépend la tâche
- Comment collecter les données ?
 - Naturelle, empirique, prédictive ?
- Quels critères sont importants ?
 - Succès à la tâche ? Satisfaction ?
- Quelles ressources sont disponible ?

Collecter les Données

- Capture de la séance
 - Observation, cahier
 - Enregistrement audio/vidéo
 - Interface instrumentée
 - Logs d'utilisation
 - Think-aloud (penser à l'orale)
 - Journal d'incident critique
 - Journal utilisateur

Après la Séance

- Entretiens, debriefing
 - « Qu'est-ce que vous aimiez le mieux/pire ? »
 - « Comment vous changerez ça ... ? »
- Questionnaires, commentaires, ...
- Classification d'événements dans la vidéo

Observer l'Utilisateur

- Pas si facile ...
- ... mais un très bon outil pour obtenir de feedback sur l'interface
- Regarder, écouter, apprendre pendant qu'une personne se sert de votre système
- Vaut mieux le faire par un non-développeur

Observation

Directe

Dans la même pièce

Peut être intrusive

Utilisateur prend conscience de votre présence

Ephémère

Pas cher ; montage facile

Indirecte

Enregistrement vidéo

Moins d'intrusion

Caméras sur l'écran, visage, clavier

Archivée, mais cher à traiter

Environnement

- Observations peuvent être :
 - Dans un laboratoire (d'utilisabilité ?)
 - Plus facile à contrôler
 - Peut demander à l'utilisateur de compléter une série de tâches
 - Sur le terrain
 - Peut regarder les actions quotidiennes
 - Plus réaliste
 - Plus difficile à contrôler

Challenge

- Dans une observation, on peut voir les actions utilisateurs, mais on ne sait pas ce qu'il pense
- Peut utiliser un protocole verbale

Protocole Verbal

- Think-aloud (penser à l'orale)
 - L'utilisateur décrit à l'orale ce qu'il pense en performant les tâches
 - Ce qu'il pense se passe
 - Pourquoi il fait une action
 - Ses buts

Think Aloud

- Technique bien répandu, utile
- Permet à mieux comprendre les pensées de l'utilisateur
- Problèmes potentiels ?
 - Peut être gênant, inconfortable pour l'utilisateur
 - Peut modifier la performance d'une tâche

L'ÉQUIPE

- Une autre technique : co-discovery learning (interaction constructive)
 - Mettre participant en binôme
 - Utiliser think-aloud
 - Peut-être un expert et un débutant
 - Plus naturaliste (comme conversation), enlevant donc un peu d'inconfort de think-aloud traditionnel

Alternatif

- Et si think-aloud est mal-adapté aux circonstances ?
- Protocole post-événement
 - L'utilisateur passe l'expérience, puis regarde une vidéo en décrivant ce qu'il pensait
 - Parfois difficile à s'en souvenir
 - Possibilité d'interprétation post-hoc

Rapport Historique

- En observant l'utilisateur, comment capturer les événements de la séance pour l'analyse subséquente ?

Stylo sur Papier

- Peut être lent
- Possibilité de rater des choses
- Facile et pas cher

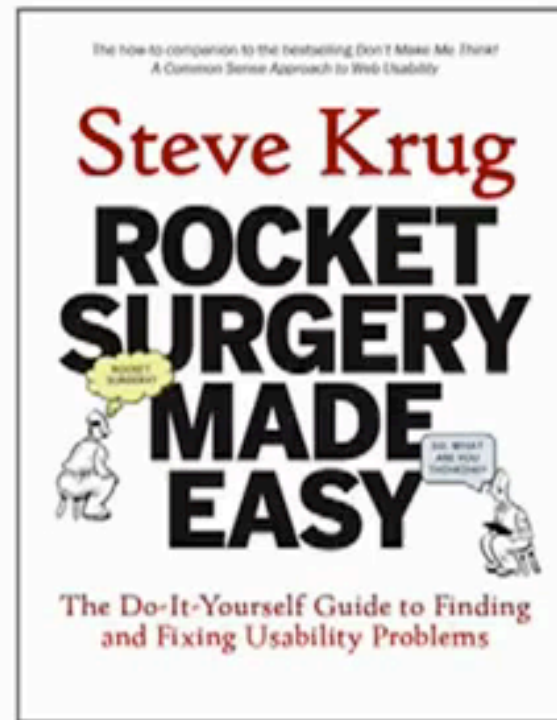


Enregistrement

- Audio et/ou vidéo
 - Marche très bien pour think-aloud
 - Difficile à relier à l'interface
 - Besoin potentiel de plusieurs caméras synchronisées
 - Bon, riche enregistrement de la séance
 - Intrusif
 - Difficile à transcrire et analyser

Demo Usability Test

for readers of



Rev. 1.1 / February 3, 2010
© 2010 Steve Krug
www.rocketurgerymadeeasy.com





Laboratoire d'Utilisabilité

Salle d'Observation

- Grand champ de vision, glace uni-directionnel, isolation de lumière et son
- Portes situées tel que le participant ne peut pas les voir en entrant dans la salle d'observation



Salle d'Observation

- Trois écrans pour voir le participant, l'écran du participant, et les deux à la fois
- Confortable pour plusieurs observateurs
- DVD-R pour enregistrer la séance



Salle du Participant

- Se ressemble à un bureau normal
- Caméra haute-résolution, contrôlée à distance, placée discrètement dans le coin
- Micro



Logs

- Modifier le logiciel pour enregistrer les actions utilisateurs
- t, x, y, genre-d'événement, ...
 - Synchronisé à la vidéo
- Logiciels commercial disponible
- Deux problèmes principaux :
 - Très bas niveau
 - Beaucoup de données ; besoin d'outils d'analyse

Problèmes

- Si l'utilisateur s'enlise sur une tâche ?
- Peut demander :
 - « Qu'est-ce que vous essayez de faire ? »
 - « Qu'est-ce qui vous a fait penser ... ? »
 - « Comment voulez-vous faire ça ? »
 - « Qu'est-ce qui vous le fera plus facile à faire ? »
- Peut-être donner un indice

Données subjectives

- Dans la longue durée, satisfaction est un facteur très important
- Il est donc très important de savoir ce que des gens préfèrent

Méthodes

- Questionnaires
- Entretiens
- Expos
- hot-line
- ...

Questionnaires

- Cher à préparer, mais facile à passer
- Orale ou écrit
 - Orale : peut suivre avec autres questions
 - Orale : mais coûte plus cher, prend du temps
- Formulaire donnent données plutôt *quantitative*

Questionnaires

- Points importants :
 - Limités par les questions posées
 - Faut établir le but du questionnaire
 - Ne pas poser de questions que vous n'utiliserez pas
 - Qui est la publique ?
 - Comment livrer et récupérer le questionnaire ?

Sujet du Questionnaire

- Peut collecter de données démographiques et données sur l'interface impliquée
- Données démographiques :
 - Âge, sexe
 - Expertise sur les tâches impliquées
 - Motivation
 - Fréquence d'utilisation
 - Niveau d'éducation

Données sur l'interface

- Ecran
- Design
- Terminologie
- Capacités
- Apprentissage
- Impression globale
- ...

Format de Questionnaire

- Fermé :
 - Réponses restreint
 - Typiquement facile à quantifier
 - Plusieurs formes possibles

Format Fermé

- Echelle Likert
 - Typiquement 5, 7 ou 9 choix
 - Plus que ça, trop difficile à distinguer
 - Nombre impaire permet un choix neutre

Les caractères sur l'écran était facile à lire :

Difficile à lire

Facile à lire

1 2 3 4 5 6 7

Autres options

- Boîtes à crocher
- Réponses ordonnées

Format Fermé

Avantages

Peut clarifier alternatifs

Facile à quantifier

Peut éliminer réponses inutiles

Désavantages

Obligé de couvrir toute option

Toute réponse également probable

Pas de réponses non-prévues

Format Ouvert

- Solliciter avis non-prévu
- Mieux adapté aux données subjectives, difficile à analyser rigoureusement
- Peut donner des idées sur le design
 - “Pouvez-vous suggérer d’améliorations à cette interface ?”

Points Importants

- Spécificité de questions
 - “Avez-vous d’ordinateur ?”
- Langage
 - Eviter le jargon, la terminologie
- Clarté
 - “Le système, était-il efficace ?”
- Questions tendancieuses

Points Importants

- Biais de prestige
 - Des gens répondent selon comment ils veulent être perçu
- Questions embarrassantes
 - “Où est-ce que vous ne vous pouvez pas débrouiller ?”
- Questions hypothétique
- “Effet halo”
 - Quand une dimension à un effet sur un autre (intelligence/style)

Déploiement

- Délibérer questions internement
- Faire passer une version pilote
- Faire passer la version finale
- Si possible, utiliser une version numérique

