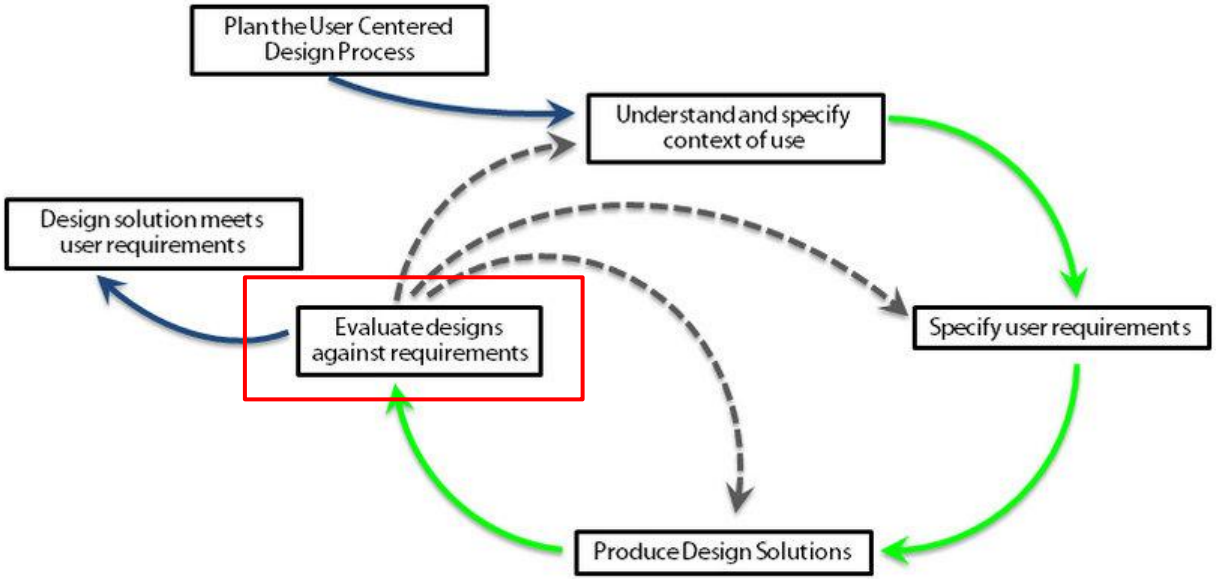
SENG 310
Lecture 13 - June 19th, 2023

Slide credits: Prof. Sowmya Somanath and Prof. Charles Perin

RECAP - HEURISTIC EVALUATION

HUMAN-CENTERED DESIGN PROCESS



https://www.researchgate.net/publication/283963125/figure/fig1/AS:671513787248651@1537112771357/The-User-Centered-Design-Process-according-to-DIN-EN-ISO-9241-210-8.jpg

HEURISTIC EVALUATION

Systematic inspection of an interface design to see if an interface complies with a set of usability heuristics, or usability guidelines.

HEURISTIC

A heuristic is a rule of thumb—a principle for solving a problem or making decisions

- Never chase after a bus, another one is coming…
- Stuck in traffic: car in the lane next to me passed me that lane must be moving faster

Not always right/true, but cognitive shortcuts

DESIGN HEURISTICS

Broad usability statements that can guide a developer's design efforts

Derived from common design problems across many systems and several researchers and practitioners have developed different sets of heuristics (e.g. domain specific)

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

NIELSEN'S 10 HEURISTICS

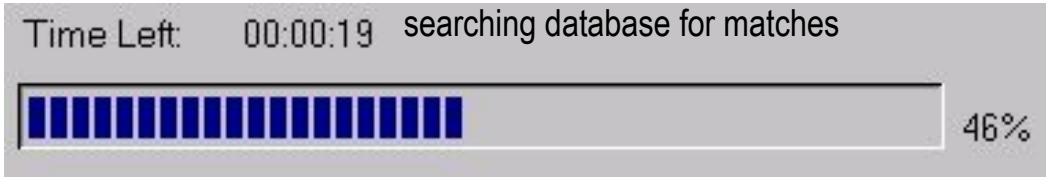
1. **VISIBILITY OF SYSTEM STATUS**
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

1. VISIBILITY OF SYSTEM STATUS

Keep users informed about what is going on

Feedback **should be appropriate** (from the perspective of the user)

Feedback **should be timely**



NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. **MATCH BETWEEN SYSTEM AND REAL-WORLD**
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

2. MATCH BETWEEN SYSTEM AND REAL-WORLD

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: information should appear in natural and logical order based on user's expectations.

Remove modes.

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. **USER CONTROL AND FREEDOM**
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

3. USER CONTROL AND FREEDOM

Users should be provided with clearly marked “emergency exits” to leave unwanted states.

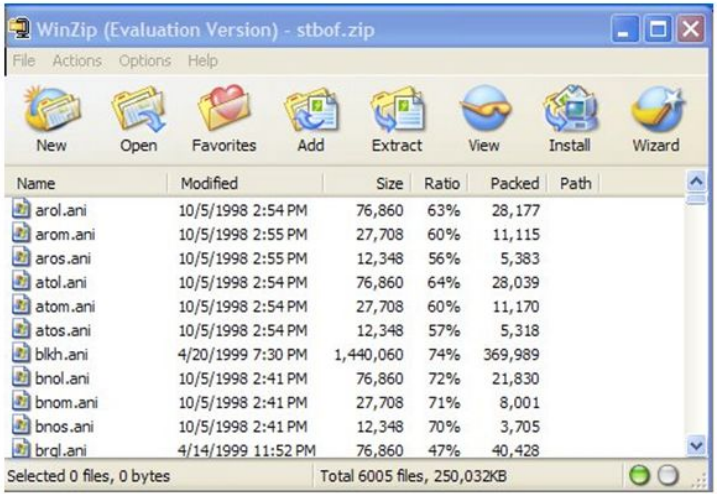
Support undo and redo.

Cancel long operations

3. USER CONTROL AND FREEDOM

Wizard

Center Stage



NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. **CONSISTENCY AND STANDARDS**
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

4. CONSISTENCY AND STANDARDS

Principle of least surprise
- similar things should act similarly
- different things should look different

Adhere to platform guidelines

Consistent language, colour, wording, ordering

Consistent use of input syntax

THREE TYPES OF CONSISTENCY

Internal consistency: is the interface consistent with itself?

External consistency: is the design consistent with similar types of applications/applications on the platform?

Metaphorical consistency: is the design consistent with the similar real-world entity/object?

NIELSEN'S 10 HEURISTICS

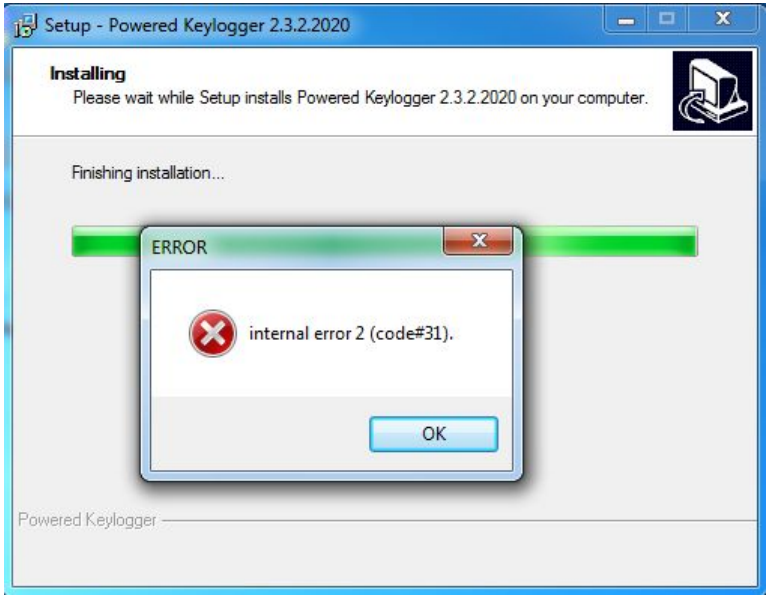
1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. **HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS**
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

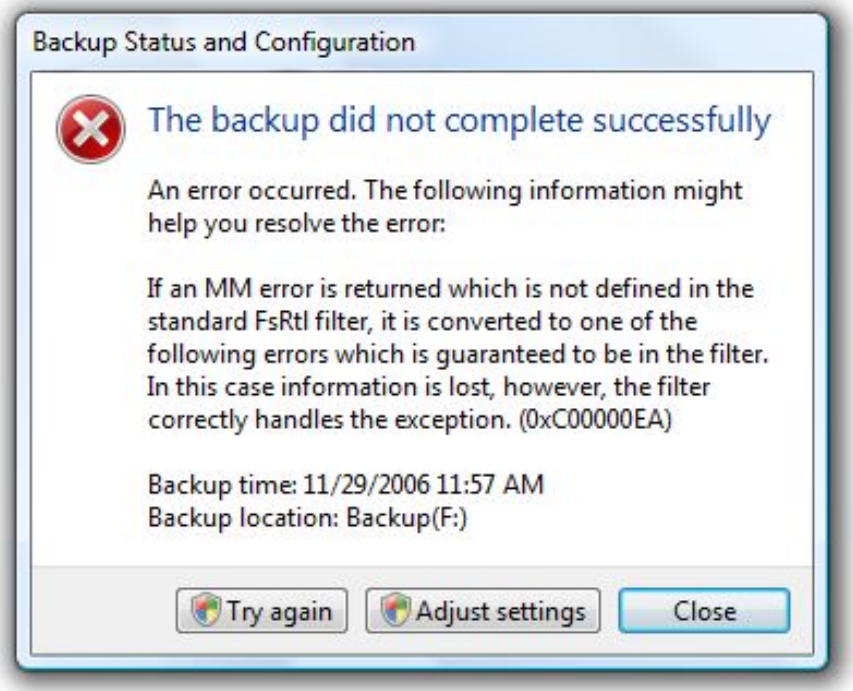
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS

Use plain language

Identify the problem

Constructively suggest a solution


Setup - Powered Keylogger 2.3.2.2020
Installing
Please wait while Setup installs Powered Keylogger 2.3.2.2020 on your computer.
Finishing installation...
ERROR
internal error 2 (code#31).
OK
Powered Keylogger


Backup Status and Configuration
The backup did not complete successfully
An error occurred. The following information might help you resolve the error:
If an MM error is returned which is not defined in the standard FsRtl filter, it is converted to one of the following errors which is guaranteed to be in the filter. In this case information is lost, however, the filter correctly handles the exception. (0xC00000EA)
Backup time: 11/29/2006 11:57 AM
Backup location: Backup(F:)
Try again
Adjust settings
Close

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. **ERROR PREVENTION**
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

6. 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.

CART RETURN LANE

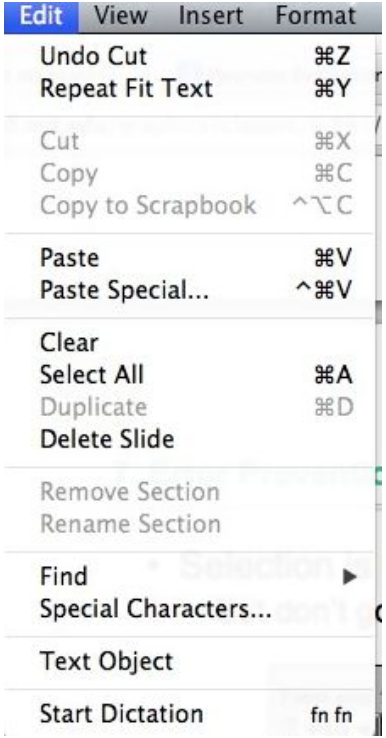
Constraining width of the lane makes it possible to avoid “get it wrong”

PREVENTION TECHNIQUES

Grey out commands

Select rather than type

Avoid typing errors through selection



More Error Prevention

Provides auto-complete suggestions for city

Date picker for dates (while still allowing for text-entry)

Greys out inappropriate “check out” dates given “check in” dates

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. **RECOGNITION RATHER THAN RECALL**
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

7. RECOGNITION RATHER THAN RECALL

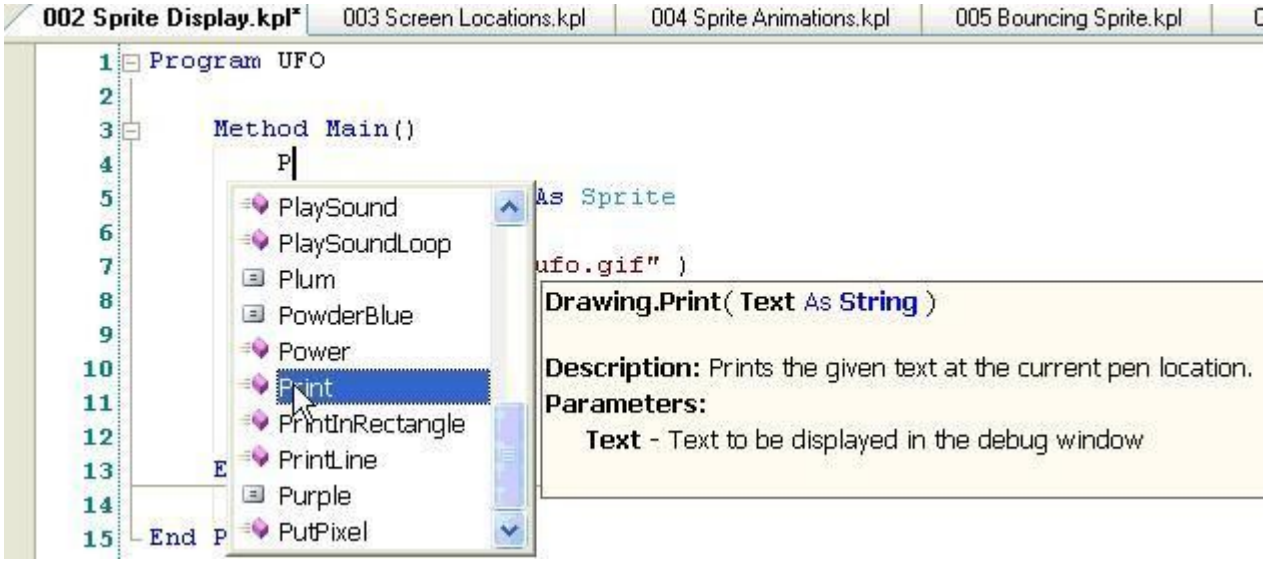
Making objects, actions, and options visible to minimize the user's memory load

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


Arno Pro
Ayuthaya
Baghdad
✓ Bank Gothic
Baskerville
Baskerville Old Face
Bastion
Batang

Autocomplete is a nice example of how the system aids you. It is easier to recognize symbols rather than recall them from scratch.



NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. **AESTHETIC AND MINIMALIST DESIGN**
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

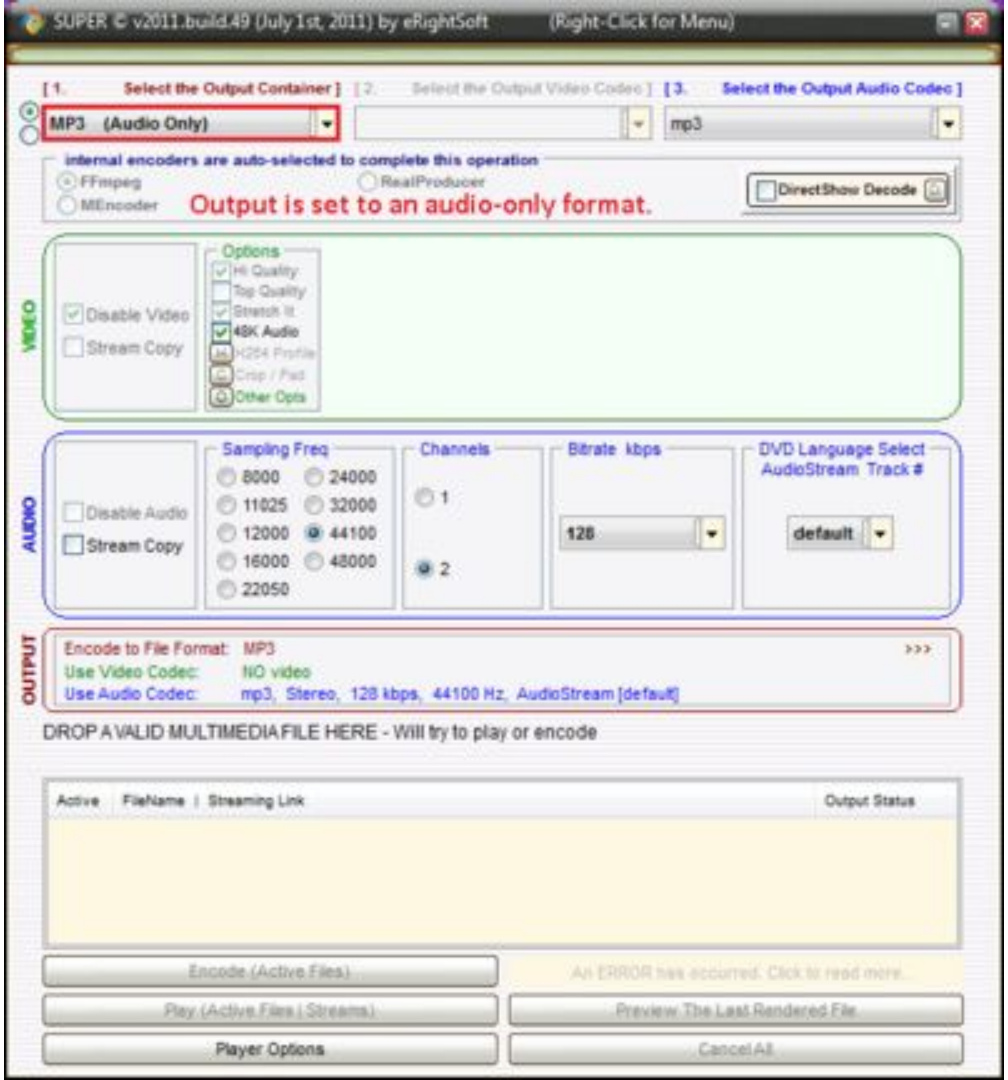
8. AESTHETIC AND MINIMALIST DESIGN

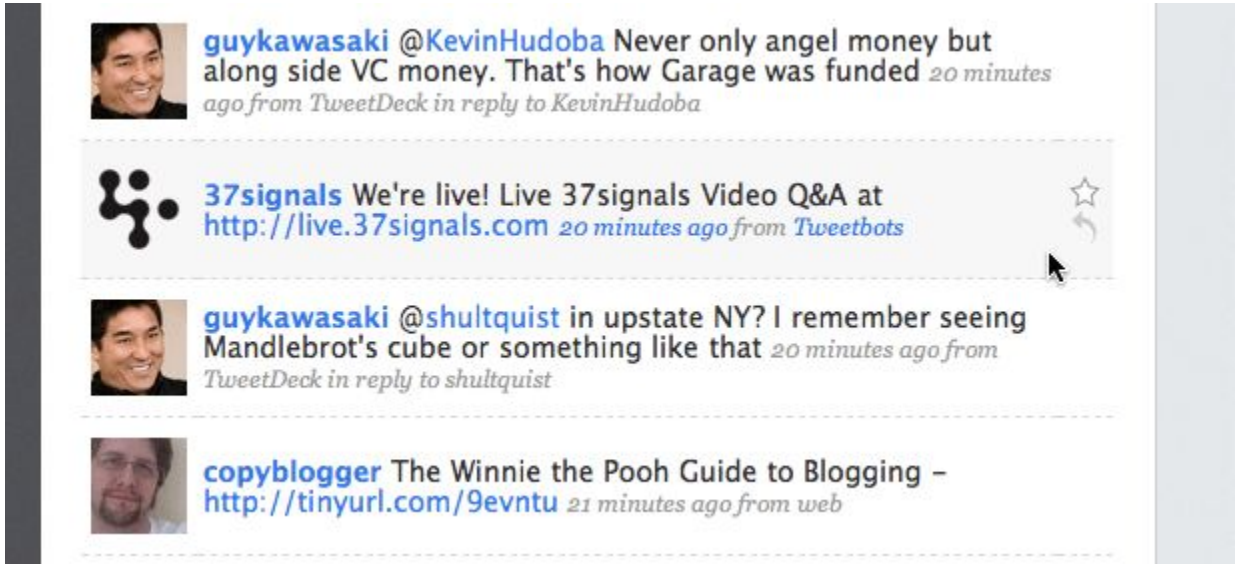
Dialogues should not contain information which is irrelevant or rarely needed

Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

SUPER is an audio/video converter tool

Although I have selected “audio only”, all of the video stuff is still there.





Hover controls only appear when they are likely to be used (i.e. when the mouse is hovering over the activation area).

What is the downside of this approach? Discoverability can be compromised

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. **FLEXIBILITY AND EFFICIENCY OF USE**
10. HELP AND DOCUMENTATION

9. 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

- Macros
- Double clicking to like in Instagram
- Keyboard shortcuts

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. **HELP AND DOCUMENTATION**

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.

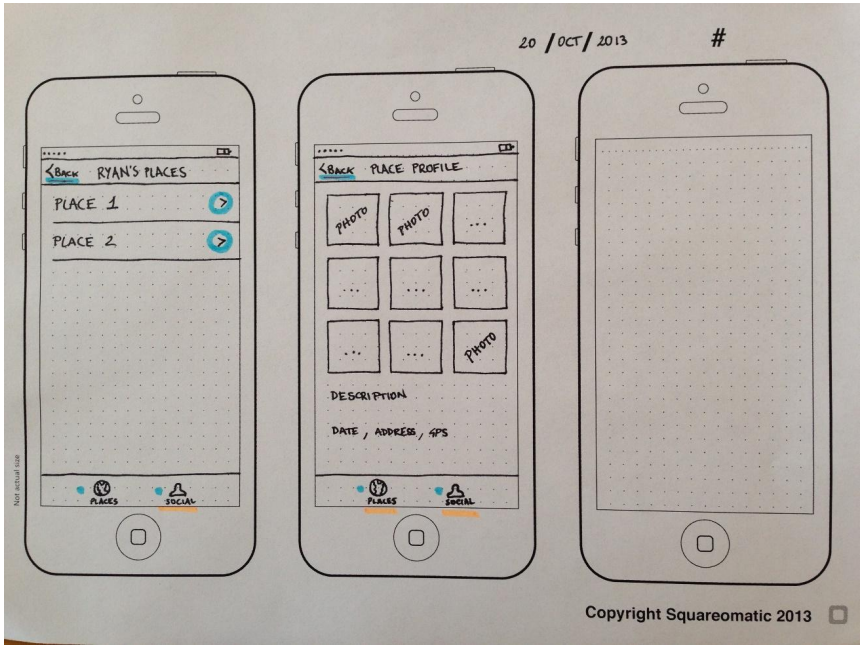
NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

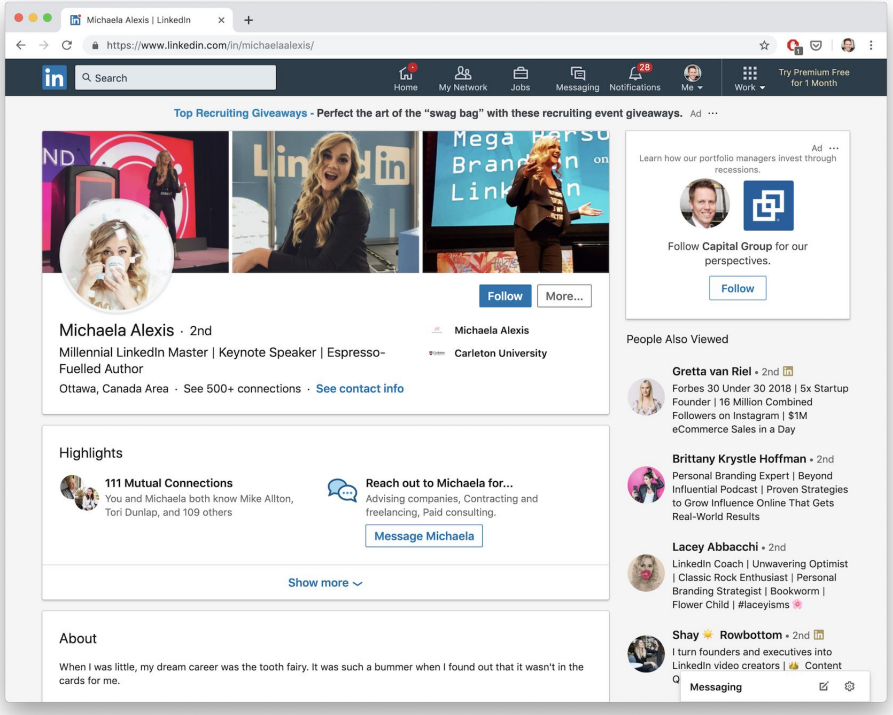
WHY EVALUATE WITH PARTICIPANTS?

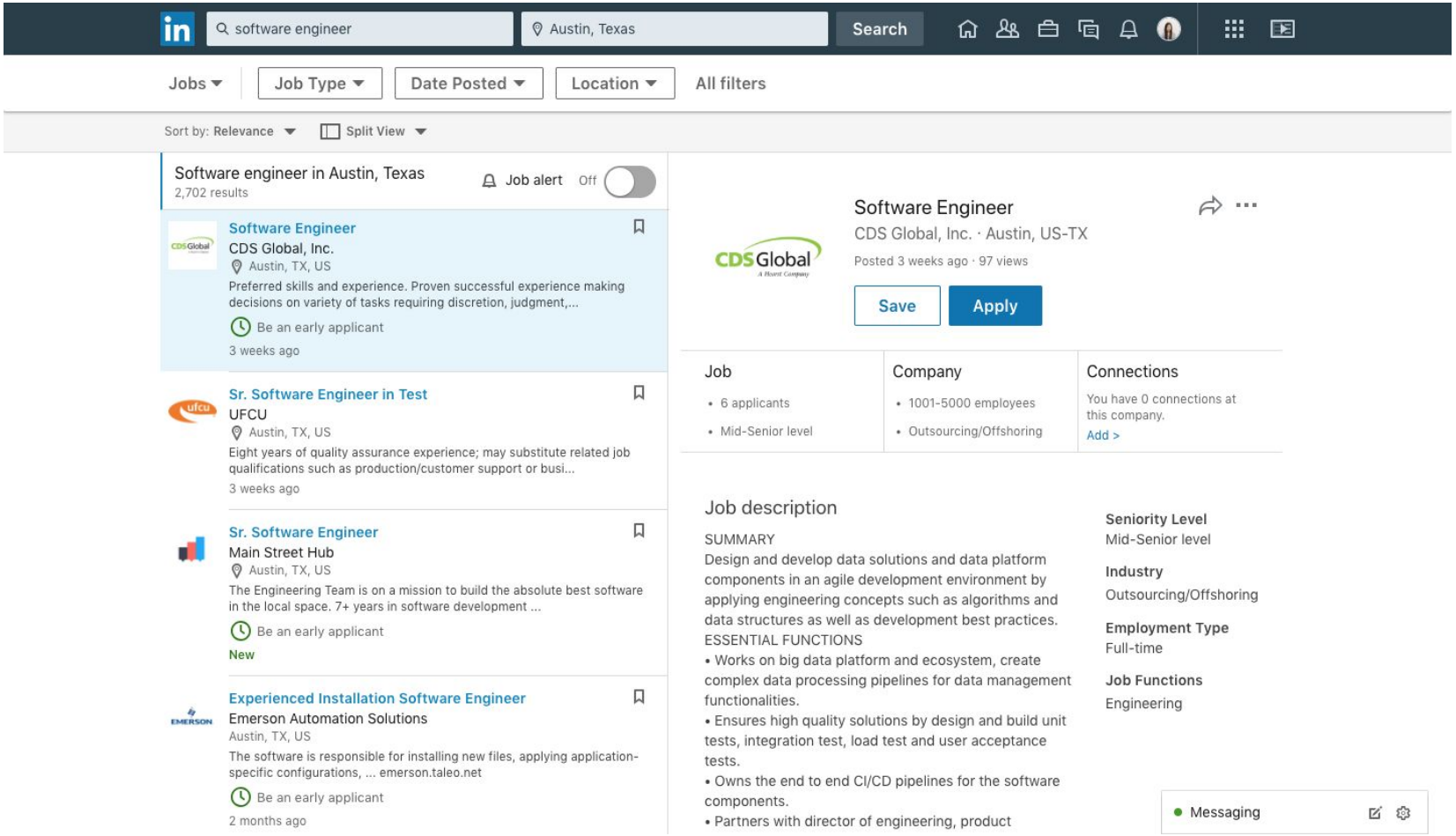
- People expect a system to be usable and beyond
- From a business perspective – well designed products sell
- From a designer perspective – evaluation allows to transition from “getting the right design” to “getting the design right”

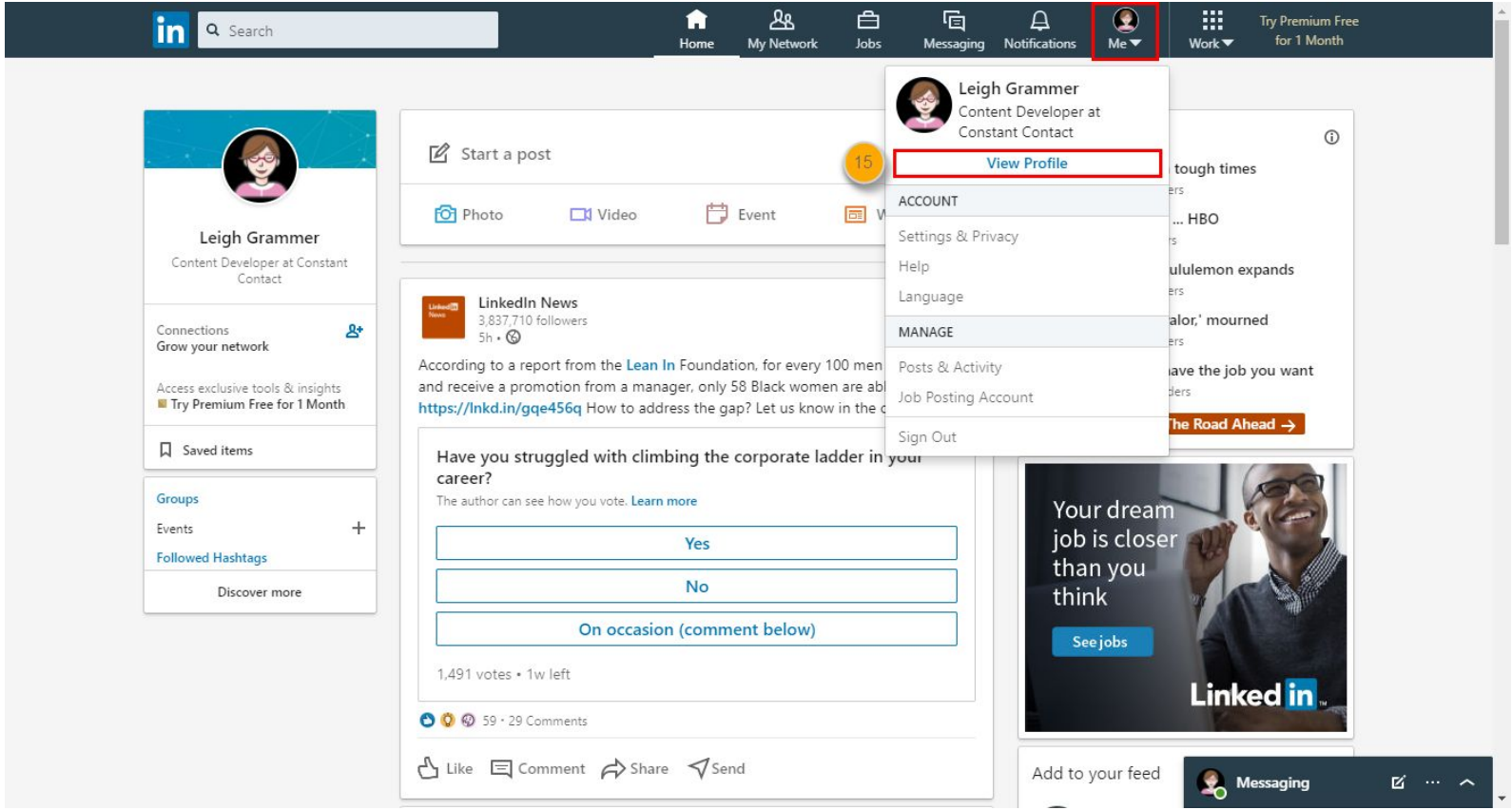
WHAT TO EVALUATE?



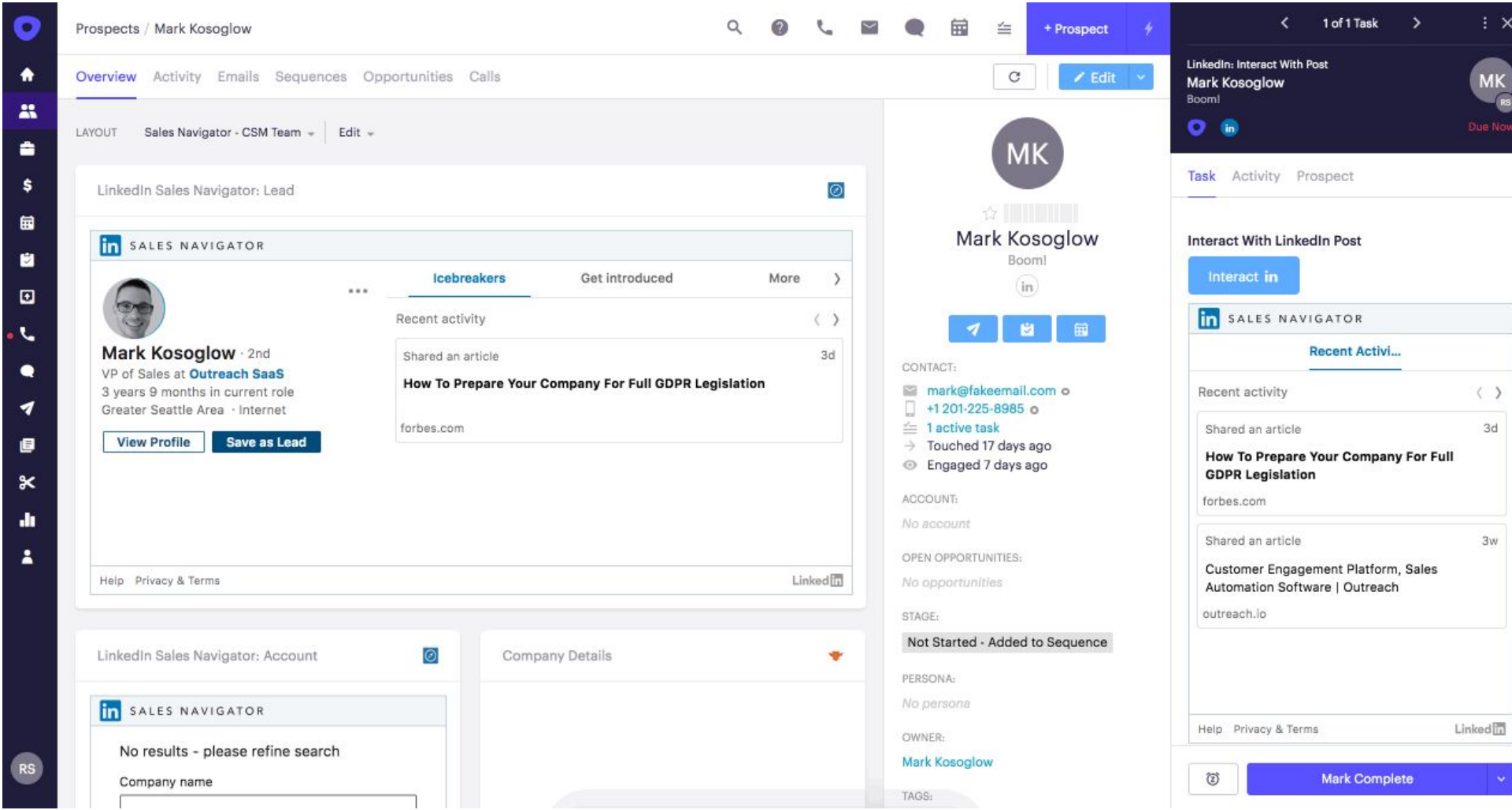
https://koenig-media.raywenderlich.com/uploads/2013/10/second-sketch.jpg




software engineer
Austin, Texas
Search
Jobs
Job Type
Date Posted
Location
All filters
Sort by: Relevance
Split View
Software engineer in Austin, Texas
2,702 results
Job alert
Off
Software Engineer
CDS Global, Inc.
Austin, TX, US
Preferred skills and experience. Proven successful experience making decisions on variety of tasks requiring discretion, judgment,...
Be an early applicant
3 weeks ago
Sr. Software Engineer in Test
UFCU
Austin, TX, US
Eight years of quality assurance experience; may substitute related job qualifications such as production/customer support or busi...
3 weeks ago
Sr. Software Engineer
Main Street Hub
Austin, TX, US
The Engineering Team is on a mission to build the absolute best software in the local space. 7+ years in software development ...
Be an early applicant
New
Experienced Installation Software Engineer
Emerson Automation Solutions
Austin, TX, US
The software is responsible for installing new files, applying application-specific configurations, ... emerson.taleo.net
Be an early applicant
2 months ago
Software Engineer
CDS Global, Inc. · Austin, US-TX
Posted 3 weeks ago · 97 views
Save
Apply
Job
6 applicants
Mid-Senior level
Company
1001-5000 employees
Outsourcing/Offshoring
Connections
You have 0 connections at this company.
Add >
Job description
SUMMARY
Design and develop data solutions and data platform components in an agile development environment by applying engineering concepts such as algorithms and data structures as well as development best practices.
ESSENTIAL FUNCTIONS
• Works on big data platform and ecosystem, create complex data processing pipelines for data management functionalities.
• Ensures high quality solutions by design and build unit tests, integration test, load test and user acceptance tests.
• Owns the end to end CI/CD pipelines for the software components.
• Partners with director of engineering, product
Seniority Level
Mid-Senior level
Industry
Outsourcing/Offshoring
Employment Type
Full-time
Job Functions
Engineering
Messaging



https://imgssl.constantcontact.com/kb/ThirdParty/linkedin-home-tab-me-menu-expanded-and-view-profile-button-step15.png


Prospects / Mark Kosoglow
+ Prospect
Overview
Activity
Emails
Sequences
Opportunities
Calls
Edit
LAYOUT
Sales Navigator - CSM Team
Edit
LinkedIn Sales Navigator: Lead
SALES NAVIGATOR
Icebreakers
Get introduced
More
Mark Kosoglow · 2nd
VP of Sales at Outreach SaaS
3 years 9 months in current role
Greater Seattle Area · Internet
View Profile
Save as Lead
Recent activity
Shared an article
3d
How To Prepare Your Company For Full GDPR Legislation
forbes.com
Help
Privacy & Terms
LinkedIn Sales Navigator: Account
SALES NAVIGATOR
No results - please refine search
Company name
Company Details
MK
Mark Kosoglow
Boom!
CONTACT:
mark@fakeemail.com
+1 201-225-8985
1 active task
Touched 17 days ago
Engaged 7 days ago
ACCOUNT:
No account
OPEN OPPORTUNITIES:
No opportunities
STAGE:
Not Started - Added to Sequence
PERSONA:
No persona
OWNER:
Mark Kosoglow
TAGS:
1 of 1 Task
LinkedIn: Interact With Post
Mark Kosoglow
Boom!
Due Now
Task
Activity
Prospect
Interact With LinkedIn Post
Interact
SALES NAVIGATOR
Recent Activi...
Recent activity
Shared an article
3d
How To Prepare Your Company For Full GDPR Legislation
forbes.com
Shared an article
3w
Customer Engagement Platform, Sales Automation Software | Outreach
outreach.io
Help
Privacy & Terms
Mark Complete
RS

AT ALL LEVELS YOU CAN ASK MULTIPLE QUESTIONS:

Is it usable?

Is it safe?

Is it aesthetically well designed?

Is it efficient and flexible?

…….

WHEN TO EVALUATE?

FORMATIVE

Refers to evaluation conducted during design to check that a product continues to meet users' need

e.g., TCSD walkthrough, heuristic evaluation of low-fi prototypes

SUMMATIVE

Refers to evaluations carried out to assess the success of a (almost) finished product

e.g., field testing, in-the-wild studies

TYPES OF EVALUATION

Controlled settings directly involving users – e.g., usability testing

Natural settings involving users – e.g., in-the-wild studies

Any settings not directly involving users (“discount” methods) – e.g., heuristics evaluation, task centered walkthroughs [in the TCSD paper posted to Brightspace]

FOR ASSIGNMENT 4 & 5

Same rules as for Assignment 2 apply:

- Participants can only be adults who can consent by themselves
- Don’t force anyone – invite 2 or 3 per team member and recruit on first-come-first-serve basis
- Make the study time reasonable. Anything more than 30 minutes might be too much for course projects; but too short sessions will not give you enough data. Aim for 15-20 minutes.

USABILITY TESTING

USABILITY TEST

Usability test is a formal method for evaluating whether a design is learnable, efficient, memorable, can reduce error and meets user's expectations.

Important to note: **Users are not being evaluated**, the design is being evaluated

HOW TO CONDUCT A USABILITY TEST?

- Determine goals of usability test
- Determine testing timeframe
- Determine target audience & recruitment plan
- Develop testing plan
  - What are the most important things you want to know?
  - Conceptual model extraction
  - Provide non-leading questions
  - Simple/realistic scenarios
  - Prepare any written materials (audience-specific, if necessary)
- Run a pilot study
- Run your test with real participants

WHAT CAN BE THE GOALS OF A USABILITY TEST?

Learnability/Discoverability: How easy is it for users to accomplish basic tasks the first time they encounter the design?

Efficiency: Once users have learned the design, how quickly can they perform the tasks?

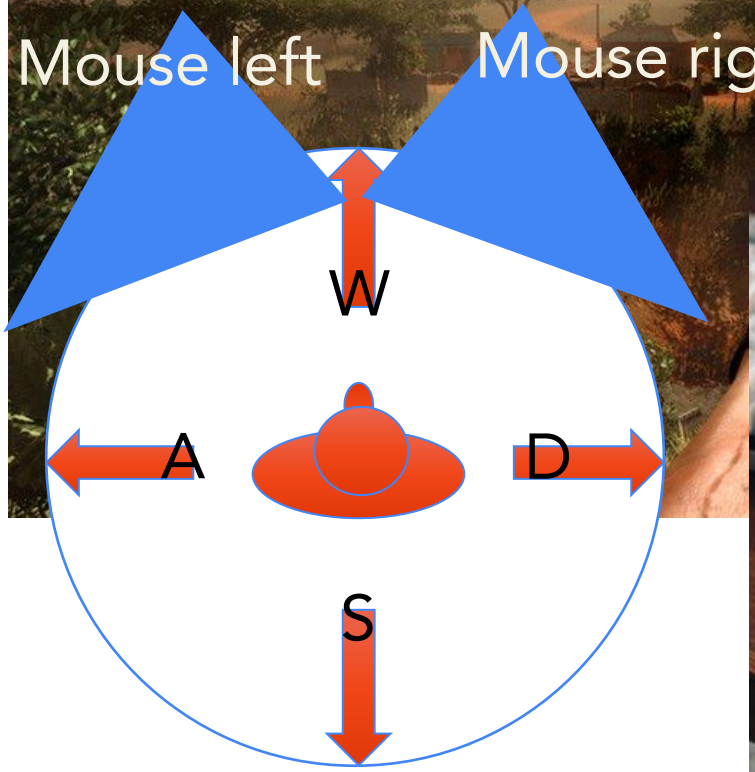
Memorability: When users return to a design after a period of not using it, how easily can they re-establish proficiency?

Errors: How many errors do users make, where are these errors occurring, and how easy is it to recover from these errors?

Satisfaction: How pleasant is it to use?

Learnable,
memorable,
but inefficient

Difficult to learn, but efficient



http://droid-n.com/wp-content/uploads/2011/06/wpid-acer-m500-circular-ui.jpg

USABILITY TESTING

USERS

USABILITY TESTING: USERS

- Who?
  - Depends on your test needs
  - **Goal**: get the people that will be using it, or people that represent those that will be using it

- How many?
  - Considerable debate in the community. Rule of thumb: ~5

https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

HOW MANY USERS?

A good rule of thumb – small number of users + iterative testing

If you have budget for 15 participants run 3 studies with 5 participants each

Ultimate goal is to improve the design and not just document its weakness

WHEN TO TEST WITH MORE USERS?

Several distinct groups of users – e.g., website for children and parents

Nielsen recommends,

- 3–4 users from each category if testing two groups of users
- 3 users from each category if testing three or more groups of users

USABILITY TESTING

TASKS AND ENVIRONMENT

START WITH A LIST OF TASKS

Ensure you have good task descriptions ready.

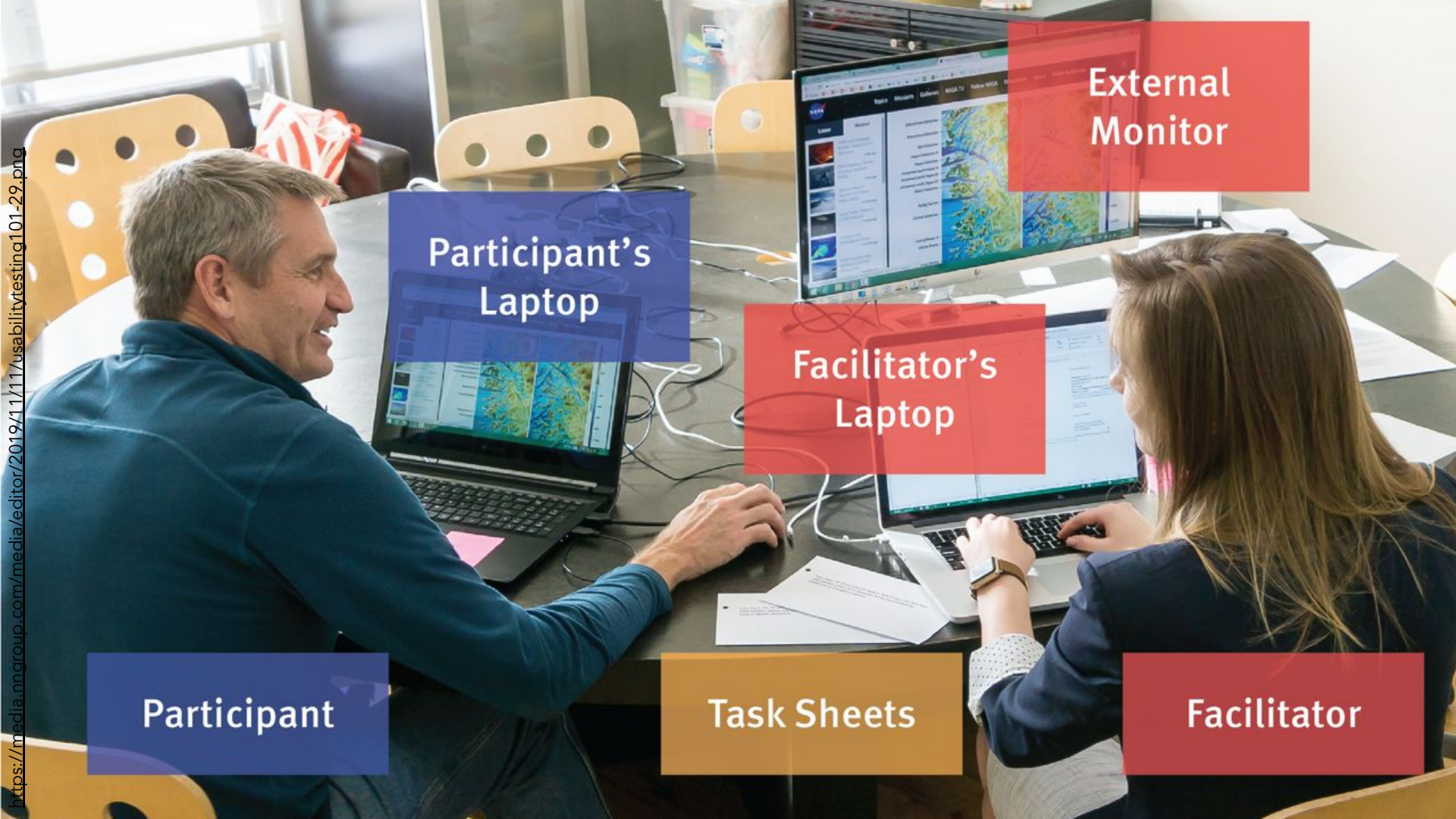
Keywords to keep in mind: Task is specific, describes a complete job and is about the user steps

Pick 2-3 key tasks your system supports for evaluation and check your original task description (with no reference to an interface) against your current UI and see how the user accomplishes the task

Think about the task scope

TESTING ENVIRONMENTS (IN-PERSON)

https://files-storage.easygenerator.com/image/477006e3-e094-4d06-bd26-9b681942c5d9.jpg



https://media.nngroup.com/media/editor/2019/11/11/usabilitytesting101-29.png

https://spin.atomicobject.com/2014/08/22/mobile-usability-testing-camera/

https://image3.slideserve.com/5732108/georgia-tech-s-aware-home-n.jpg

http://awarehome.gatech.edu/

Georgia Tech's Aware Home