



2018

HEURISTIC EVALUATION

Deep Dive Training

IT | HUE

Honeywell Internal

Honeywell

THE POWER OF **CONNECTED**

Agenda



Intro to the Method

Process

Examples

Q&A



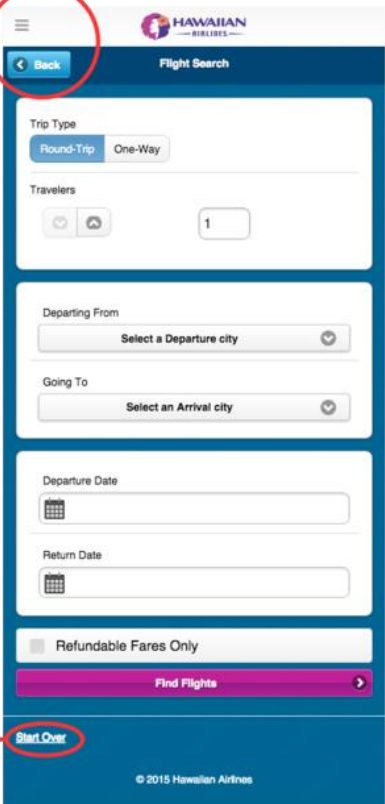
Heuristic Evaluation

Intro to the Method

What is a Heuristic Evaluation?

- A Heuristic Evaluation is a usability inspection technique
- Purpose is to find usability problems in a user interface so that they can be attended to as part of an iterative design process
- One or more persons evaluate a user interface against a set of design principles also called Heuristics

HAWAIIAN AIRLINES MOBILE UI EVALUATION



Back and Hamburger button have similar functionality, unsure why both options are there.

Only allows scrolling to choose, no typing to select airport. Also, what about if user wants to switch Departure/Going to cities?

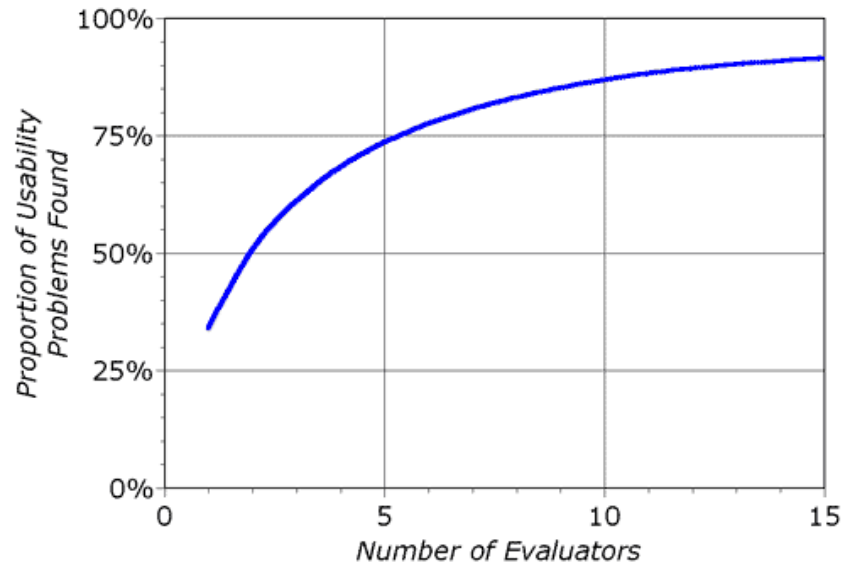
Way too long; a lot of wasted space. Screen length requires scrolling, which is fine if content necessitates it. However, it looks like everything can fit on one screen, especially if containers are taken away.

Also visual language as a whole can be updated to current app design standards, colors and buttons look garish and make the page look dated.

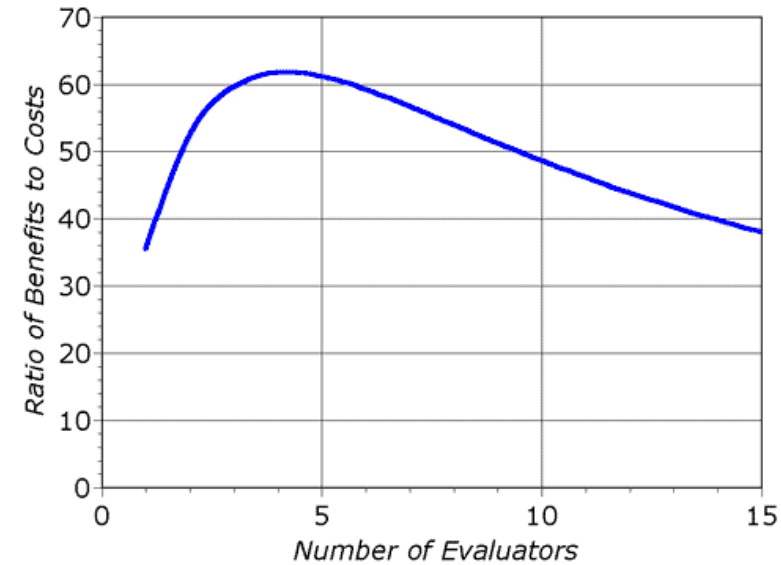
Easy to miss the "Start Over" button.

What is a Heuristic Evaluation?

- This is a qualitative method – few evaluators work fine



- Diminishing return between the number of evaluators and the volume of issues found
- 5 evaluators find around 75% of the usability problems.
- Doubling the number of evaluators to 10 only yields 85% of identified problems.

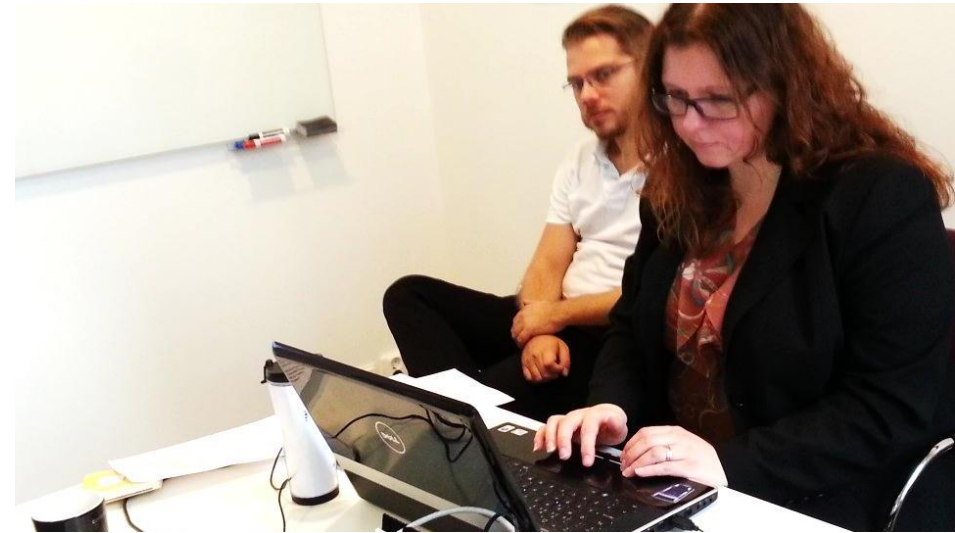


- Curve showing how many times the benefits are greater than the costs for heuristic evaluation of a sample project.
- The optimal number of evaluators in this example is four, with benefits that are 62 times greater than the costs.

Why Heuristic Evaluation?

The Challenge

Usability Testing is the single-best method for UX evaluation



But what if you don't have representative test users available?

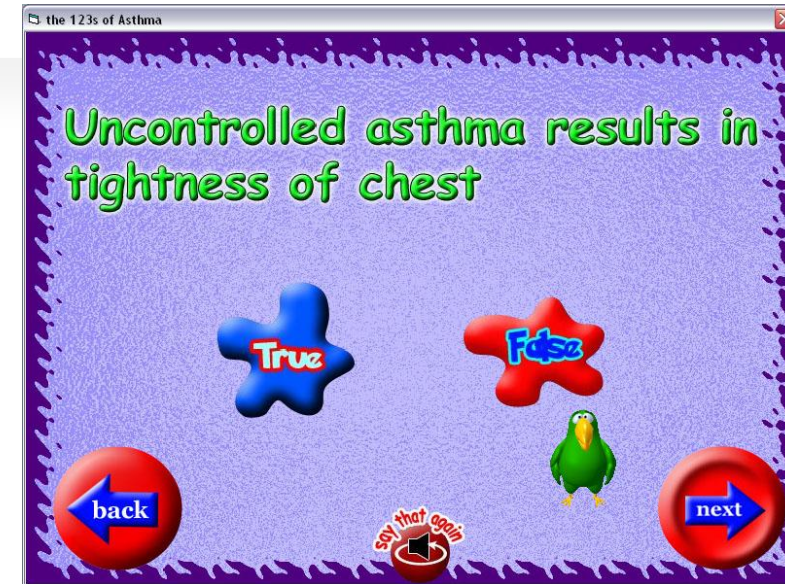
- You can test with colleagues (wrong audience!)
- You can use heuristic evaluation

Why Heuristic Evaluation?

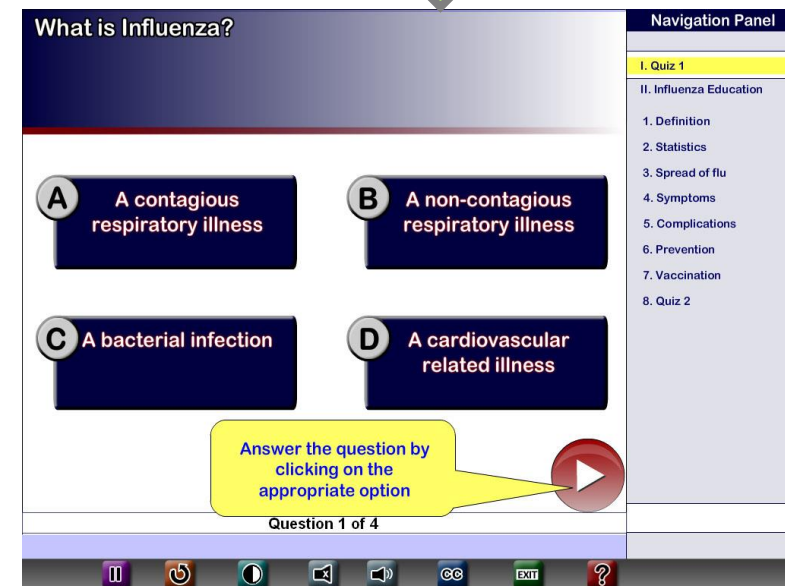
Advantages

- Easy to orchestrate since no target users are required
- Heuristics have been shown to find a significant number of problems
- Heuristics cover a wide range of the types of problems that can occur
- **Biggest advantage**
 - Almost always cheaper than usability testing

Before HE



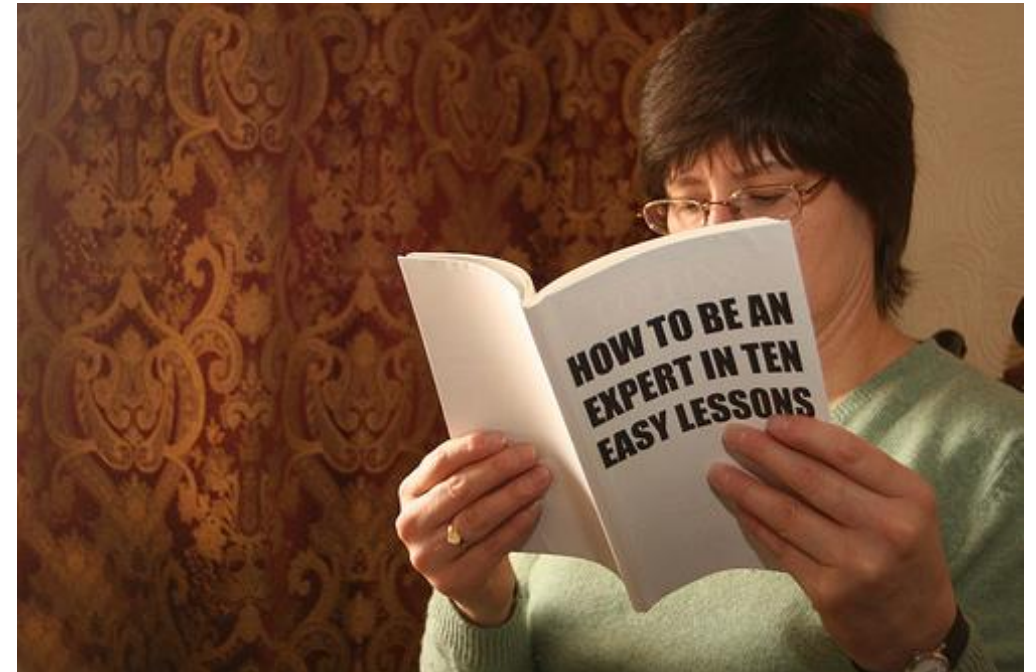
After HE



Why Heuristic Evaluation?

Disadvantages

- No established way of running a heuristics study
- No good definition of what is a good evaluator
- Heuristics do not cover domain specific issues – domain experts may be required but may be hard to find
- **Biggest disadvantage**
 - Validity – evaluators are NOT the end users
 - Important problems may get missed while oftentimes trivial problems are identified



Heuristic Evaluation

Process



How to carry out a Heuristic Evaluation

Two approaches, both work fine – also in combination

Go through the product step by step

- Review product step 1
- Cross-check heuristics
- Assess compliance
- Document findings
- Assign severity ratings
- Ideate mitigation
- Repeat for steps 2 ... n

and/or

Go through the heuristics one by one

- Review heuristic 1
- Cross-check product
- Assess compliance
- Document findings
- Assign severity ratings
- Ideate mitigation
- Repeat for heuristic 2 ... n

How to carry out a Heuristic Evaluation

Collaboration between several evaluators

1. Each evaluator works separately first
2. All evaluators share, aggregate and discuss their findings
 - Clarify issues
 - Remove redundancies
 - Discuss and agree on severity ratings
 - Discuss and agree on mitigations

The Heuristics – Overview

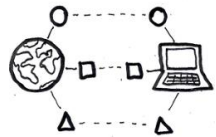
There are different sets of design principles. The most popular set are Nielsen's 10 Heuristics

Developed in 1990 by Jakob Nielsen and Rolf Molich these heuristics have been refined by Nielson based on a factor analysis of 249 usability problems to derive a set of heuristics with maximum explanatory power.



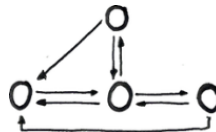
1. Visibility of System Status

Give the users appropriate feedback about what's going on.



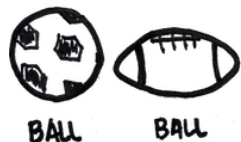
2. Match System & Real World

Use real-world words, concepts, and conventions familiar to users in a natural, logical, order.



3. User Control & Freedom

Support, undo, redo, and exit points to help users leave an unwanted state caused by mistakes.



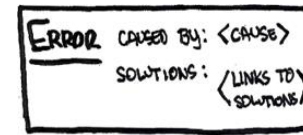
4. Consistency & Standards

Use consistent words, situations, and actions throughout.



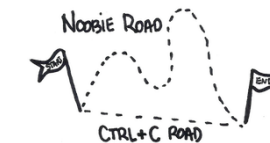
5. Error Prevention

Eliminate error-prone conditions or check for them before users commit to an action.



6. Recognition rather than Recall

Make objects, actions, and options visible at the appropriate time to minimize users' memory load.



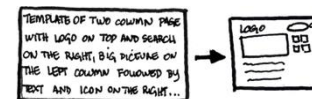
7. Flexibility & Efficiency of Use

Make the system efficient for different experience levels through shortcuts, advanced tools, and frequent actions.



8. Aesthetic & Minimalist Design

Don't show irrelevant or rarely needed information. Extraneous elements diminish the relevance of the others.



9. Help Users Recognize, Diagnose, & Recover

Express error messages in plain language and suggest solutions.



10. Help & Documentation

Make necessary help and documentation easy to find.

How to carry out a Heuristic Evaluation

The guidelines are somewhat ambiguous and interpretive.

There will be **OVERLAP**

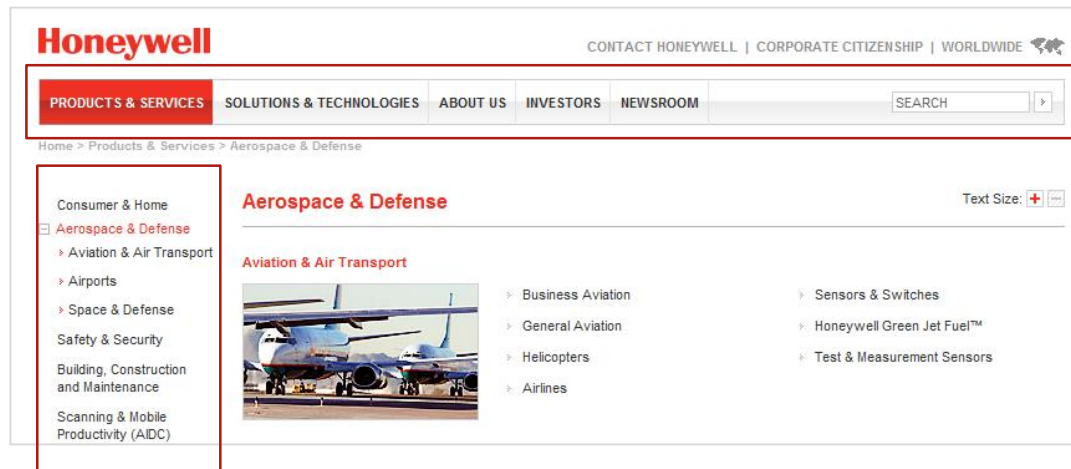
There will be **SUBJECTIVITY**

There will be **CONFLICTS**

1. Visibility of System Status



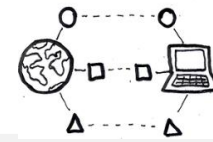
The system should always keep user informed about what is going on, through appropriate feedback within reasonable time.



Clearly mark the user's current location

Password	<input type="password"/>	⇒ 6 characters or more (be tricky!)
Password	<input type="password" value=".."/>	⇒ Too short
Password	<input type="password" value="....."/>	⇒ Too obvious
Password	<input type="password" value="....."/>	Weak
Password	<input type="password" value="....."/>	Good
Password	<input type="password" value="....."/>	Strong
Password	<input type="password" value="....."/>	Very Strong

2. Match between System & Real World



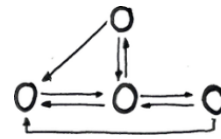
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.



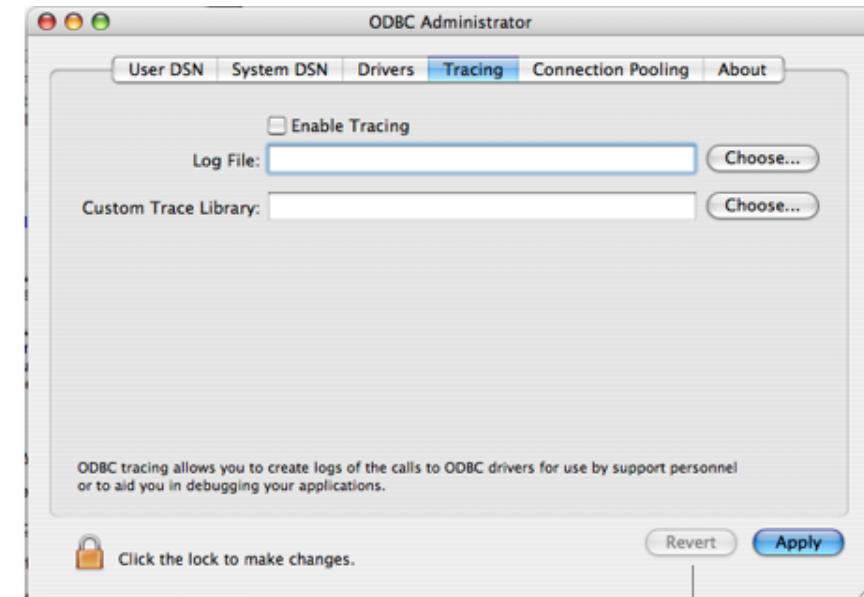
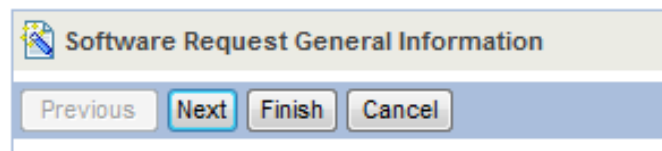
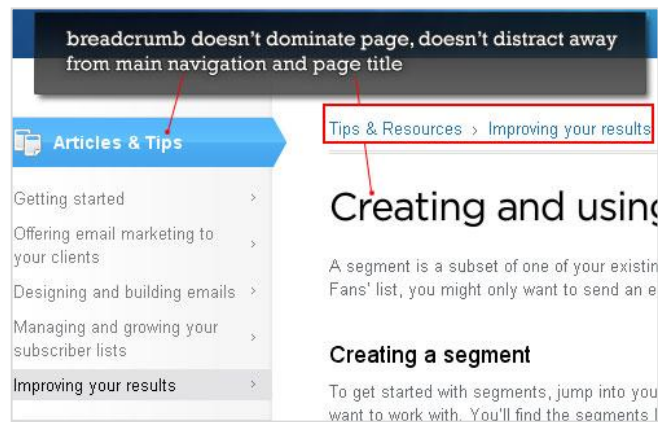
PRODUCTS	PROBLEMS WE SOLVE	SUPPORT	IN THE NEWS	ABOUT US	NEW
Thermostats Programmable 7 Day 5-1-1 Day 5-2 Day 1 Week Non Programmable Digital Manual	Accessories Entry/Exit Remote Indoor Temperature Sensor Wireless AquaReset Outdoor Reset Kit Wireless Outdoor Air Sensor Portable Comfort Control	Air Cleaning Whole-House HEPA Whole-House Enhanced Whole-House Electronic Whole-House Replaceable with Filter Reminder Whole-House	Humidifiers Whole-House Steam Whole-House Evaporative Whole-House Fan-Powered Whole-House Drum Replacement Parts	Dehumidifiers Whole-House (120 pint) Whole-House (90 pint) Whole-House (65 pint) Ventilation Energy-Efficient	

Use terminology in menus and content with words, phrases and concepts familiar to the user

3. User Control & Freedom



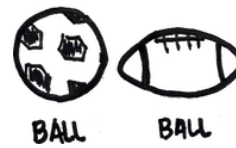
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.



The secondary action button is inactive due to its transparent look.

Clearly indicate the affordances available to the user

4. Consistency & Standards



Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.



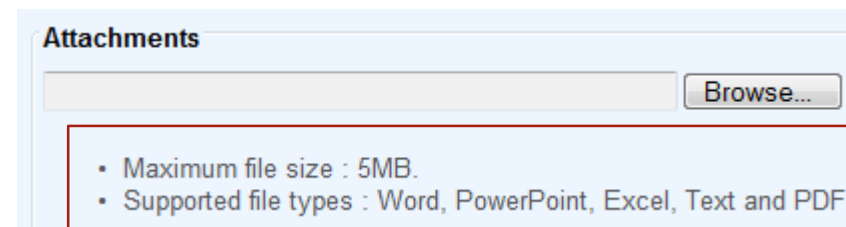
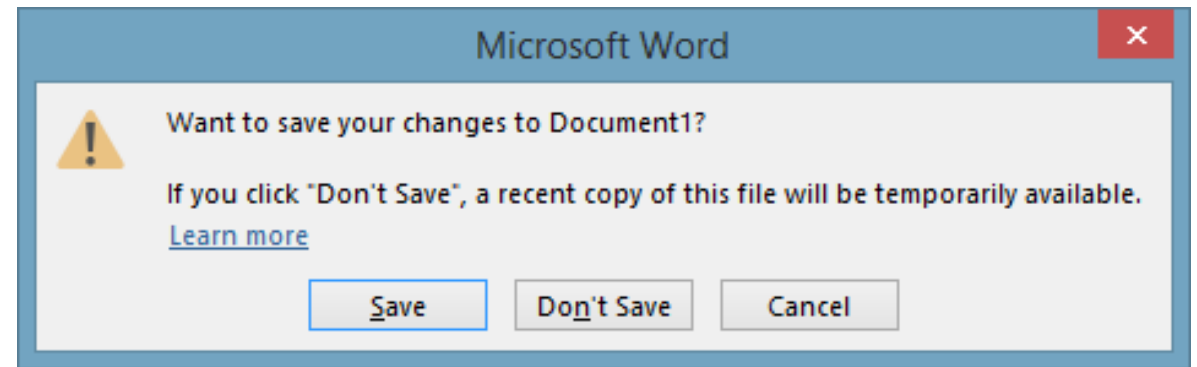
Follow naming convention and design themes consistently



5. Error Prevention

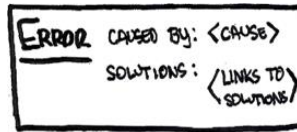


Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

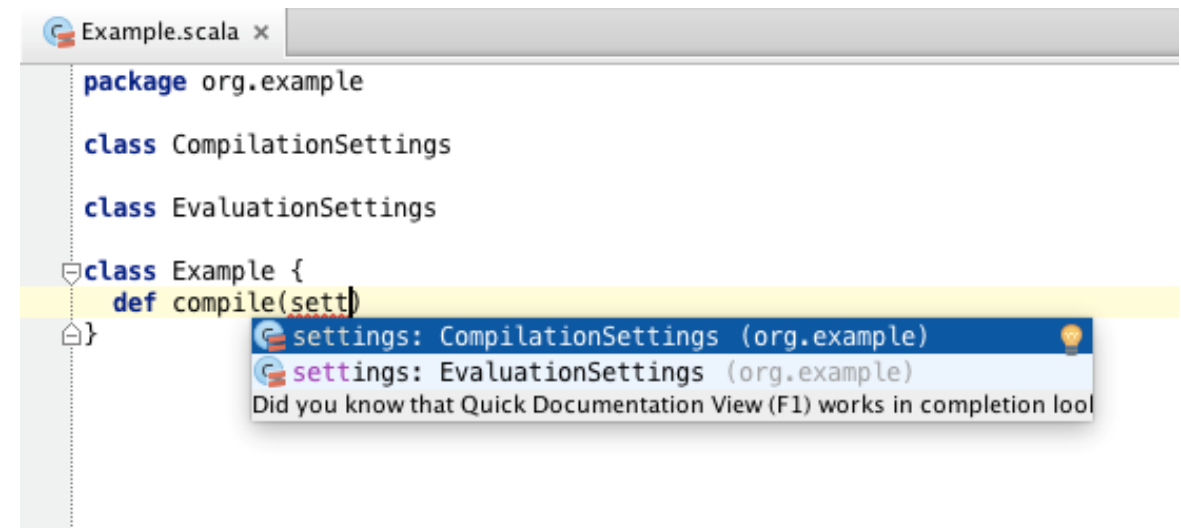
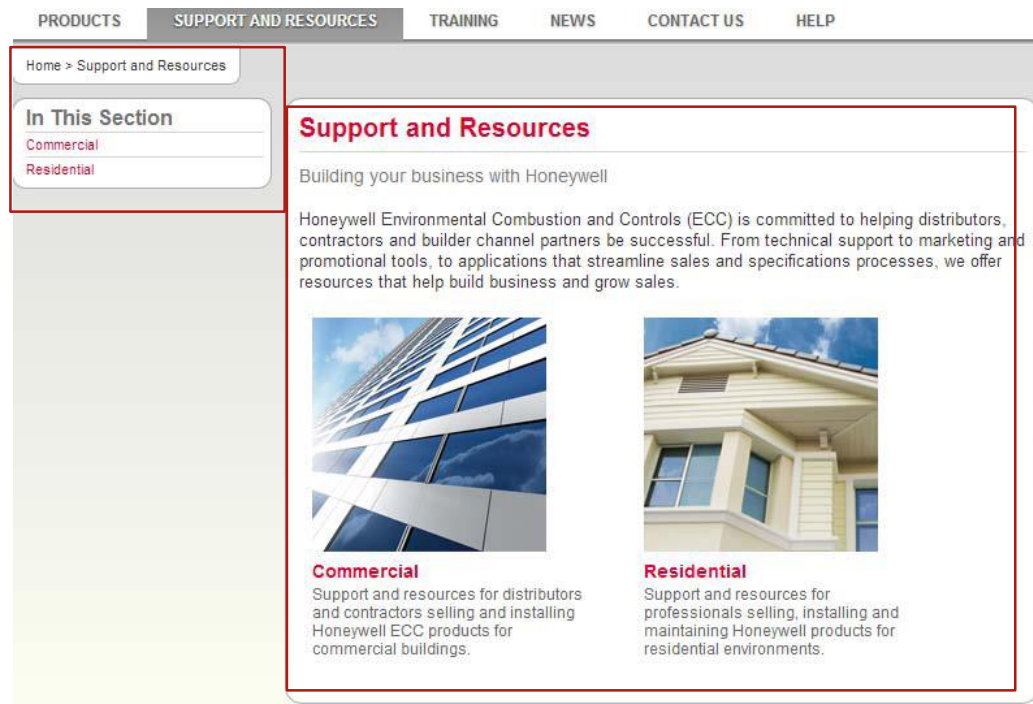
A form with several input fields. The "Name" field has two sub-fields labeled "First" and "Last". The "Time" field has two sub-fields labeled "HH" and "MM", followed by a dropdown menu for "AM/PM". The "Email" field has a sub-field labeled "Please use your office email address." The "Date" field has three sub-fields labeled "MM", "DD", and "YYYY". The "Address" field has three sub-fields labeled "Street Address", "Street Address Line 2", and "City". The "Region" field is a separate input field. A red box highlights the "Name", "Time", "Email", "Date", and "Address" fields.

Reduce opportunities for errors by educating users pro-actively

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.



Use unique styles for each level of hierarchy on an interface

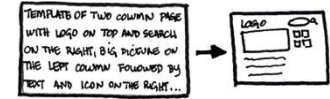
8. Aesthetic and minimalist Design



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.

Answer Rules

Each security answer must satisfy the following rules:

- Must be between 3 and 20 characters long
- Cannot contain any word in the question
- Cannot be the same as any other security answer

Note: Security answers are not case sensitive.


Error: The date is not in the correct format.
Enter a date in the same format as this example: 29/Nov/1998
Example: 29/Nov/1998

* HireDate

* JobId

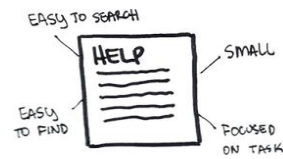
Salary

Complete Certificate Request

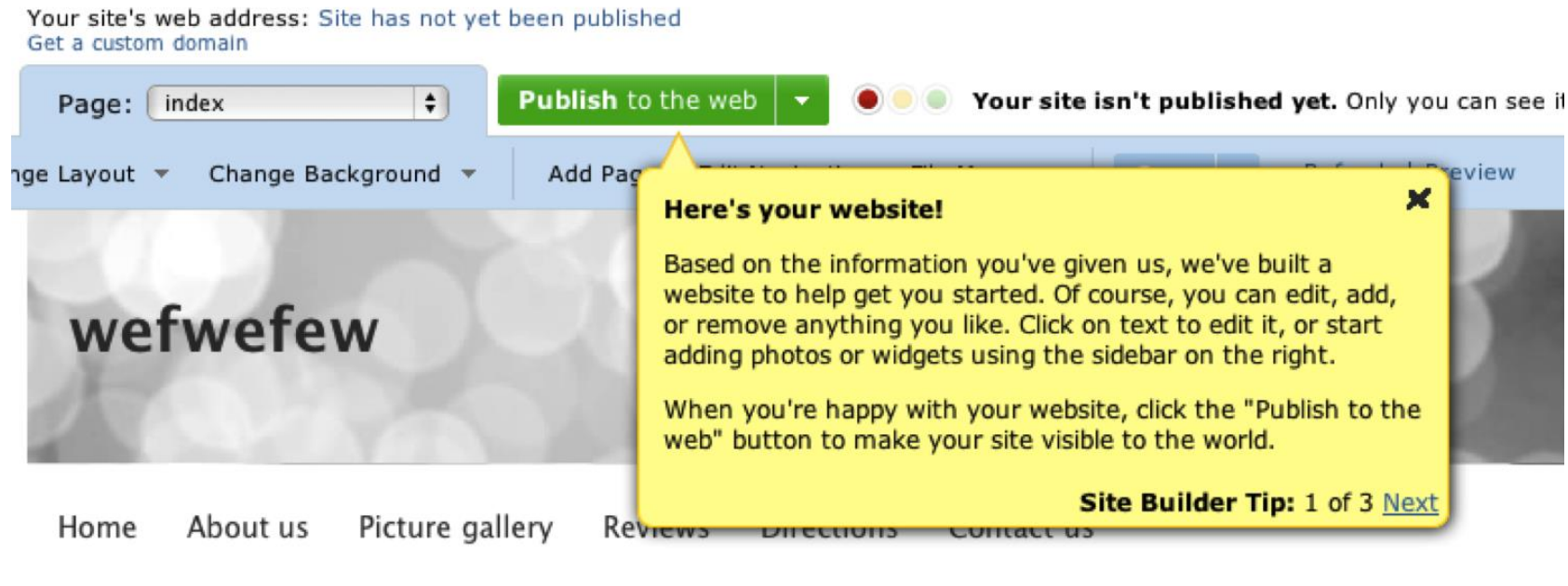
 There was an error while performing this operation.
Details:
CertEnroll::CX509Enrollment::p_InstallResponse: ASN1 bad tag value met. 0x8009310b (ASN: 267)

Give contextual, informative and concise error messages as soon as possible

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.



Prioritizing Findings

- Typically, you will not have enough time to mitigate all usability issues found
- Applying severity ratings to the findings allows the project team to determine which issues should be mitigated and which can be deferred to later or omitted

Severity Ratings (from Nielsen & Mack '94)

0 - don't agree that this is a usability problem

1 - cosmetic problem

2 - minor usability problem

3 - major usability problem; important to fix

4 - usability catastrophe; imperative to fix

A woman with blonde hair in a ponytail, wearing a blue plaid shirt, is writing on a glass wall covered in colorful sticky notes. Two men are standing behind her, looking at the wall. The man in the foreground is wearing a blue button-down shirt, and the man in the background is wearing a light blue polo shirt. The scene is brightly lit, suggesting a window or large glass door in the background.

Heuristic Evaluation

Examples

Examples

[Heuristic 7 – Flexibility and Efficiency of Use] [Severity 1]

- The redundancy of the notification is confusing. The second notification should suffice, no need for the first one.
- This is a cosmetic issue since it appears only once when opening app, it is not a persistent problem where it pops up each time you open the app.

Severity Ratings (from Nielsen & Mack '94)

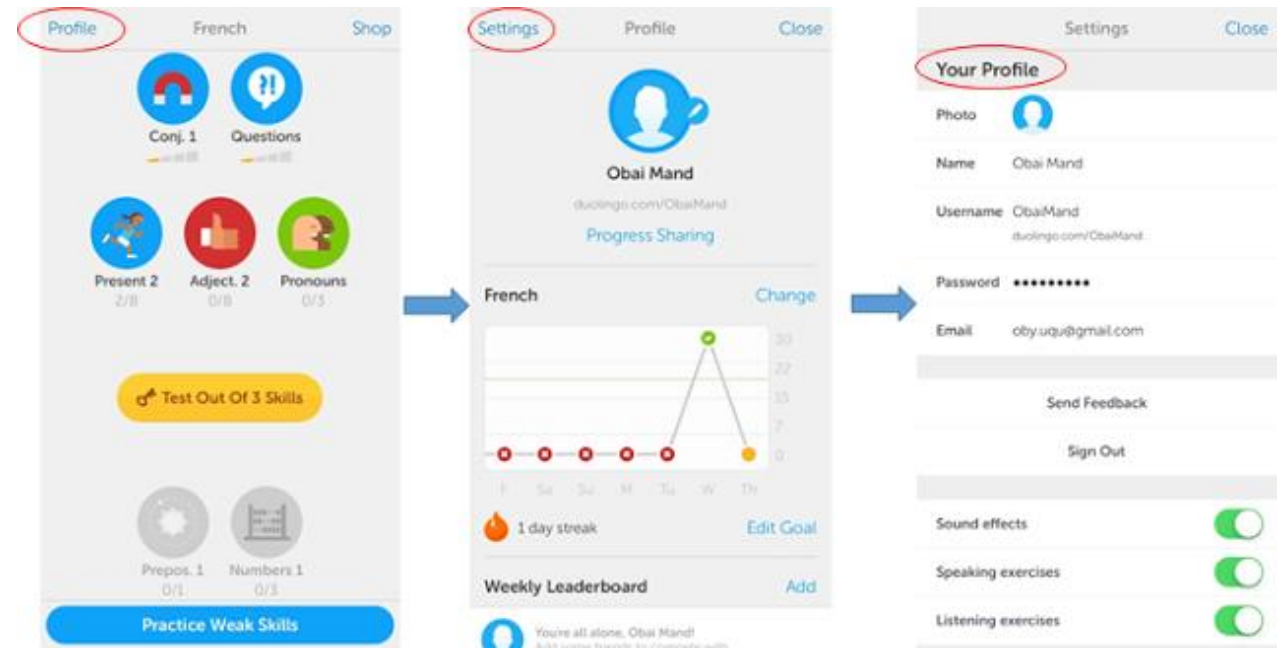
- 0 - don't agree that this is a usability problem
- 1 - cosmetic problem
- 2 - minor usability problem
- 3 - major usability problem; important to fix
- 4 - usability catastrophe; imperative to fix



Examples

[Heuristic 7 – Flexibility and Efficiency of Use] [Severity 2]

- When user wants to go to the settings, s/he must go through the Profile page first, then go to Settings page.
- This conflicts with common sense. If a user needs to change app settings, there should be a clear button on the main page, not hidden under another page.
- This is a minor issue since the user always has to take these navigation steps.



Severity Ratings (from Nielsen & Mack '94)

- 0 - don't agree that this is a usability problem
- 1 - cosmetic problem
- 2 - minor usability problem
- 3 - major usability problem; important to fix
- 4 - usability catastrophe; imperative to fix

Recommended Reads

- Heuristic Evaluation Details: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>
- HON Design Language System: <https://www.honeywell.com/brand/digital/patterns>
- Heuristic Evaluation Report Template:
<https://honeywellprod.sharepoint.com/:p:/s/ITHUECorporate/Ec7PIdmwZltDoTUhK4iL2v8B5n33oELe8EkBw1qVg4M0fw?e=sbpBVP>
- Heuristic Evaluation Handbook including Heuristics for Mobile:
https://honeywellprod.sharepoint.com/:w:/s/ITHUECorporate/EZJoHsj54k9JmXrQKH-FpFEBoWFjD5jeHJYA-_MTjoOxyQ?e=ihEdih

Heuristic Evaluation

Q&A



Thank You!



For More Information, Please visit below link :

<https://in.honeywell.com/BusinessFunction/IT/ITHUEPortal/Contents/Home.html>