

# Usability

# YOU ARE NOT THE USER!

- Most users consider a mobile device as a tool, not a toy.
- Average users do not explore.
- Users do not notice or use features unless forced to do so.
- Users are not comfortable with anything that is “too” different.
- Users ignore complex or multi-step tasks unless necessary.
- Most users dislike having to change settings.
- Users often lack confidence.
- Users repair applications by deleting and replacing them.





# Reasons For Failure

<https://clearbridgemoible.com/youre-doing-it-wrong-the-top-6-reasons-why-apps-fail/>

1. Poorly researched market and audience.
2. Lack of originality – particularly in saturated markets.
3. Ignoring platforms – not using traditional platform specific conventions and nuances (being too different)
4. Poor user experience.
5. Improper testing that fails to address consumer expectations and needs.
6. Poorly executed launch – no marketing plan.

Note: Quality, bugs, programming are NOT the most common causes for failure. It is the planning, design, and testing that is typically lacking.

---

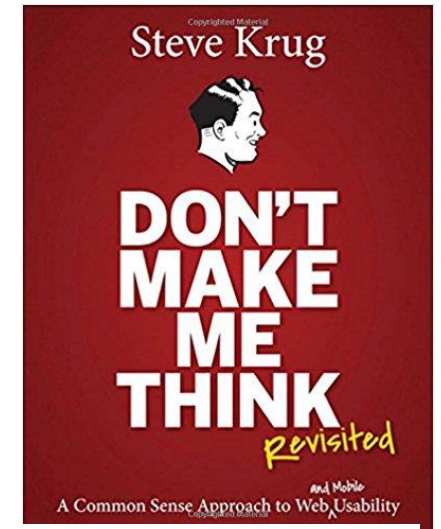
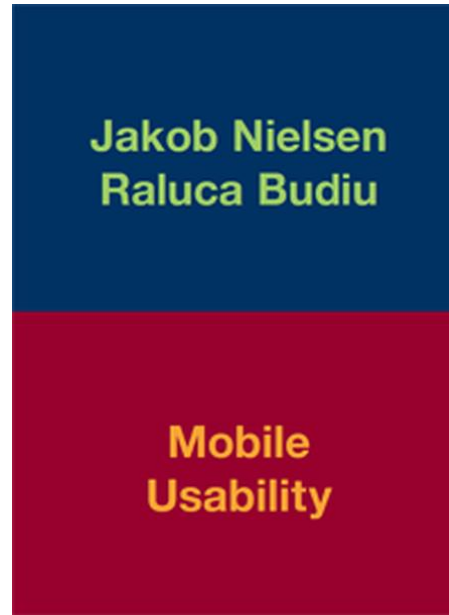
# Who?



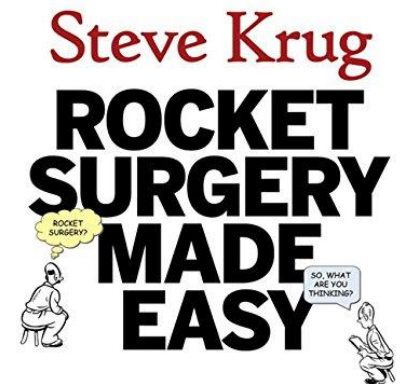
Jakob Nielsen



Don Norman



The how-to companion to the bestselling *Don't Make Me Think!*  
A Common Sense Approach to Web Usability



The Do-It-Yourself Guide to Finding  
and Fixing Usability Problems

- Nielsen Norman Group:  
<https://www.nngroup.com>

HINT: A good resource for your project

For UI/UX enthusiasts

**YOU**  
**≠**  
**USER**

[nngroup.com](https://nngroup.com)

# Neilson's Heuristics


- Heuristics can fail; they are “rules of thumb” and not algorithms (which never fail).
- General and broad; they don't describe specific situations.
- Incomplete; There are other principles which can be used.
- Consider them as guidance/suggestions.
- But .. They are **REALLY GOOD** suggestions!

## **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 the 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.

A decorative footer consisting of two overlapping geometric shapes. On the left is a light gray trapezoid, and on the right is a darker gray trapezoid. They are separated by a white diagonal line that runs from the bottom left towards the top right.



- Communicate clearly to users what the system's state is — no action with consequences to users should be taken without informing them.
- Present feedback to the user as quickly as possible (ideally, immediately).
- Build trust through open and continuous communication.





**NN/g**  
NNGROUP.COM


- Ensure users can understand meaning without having to go look up a word's definition.
- Never assume your understanding of words or concepts will match those of your users.
- User research will help you uncover your users' familiar terminology, as well as their mental models around important concepts.

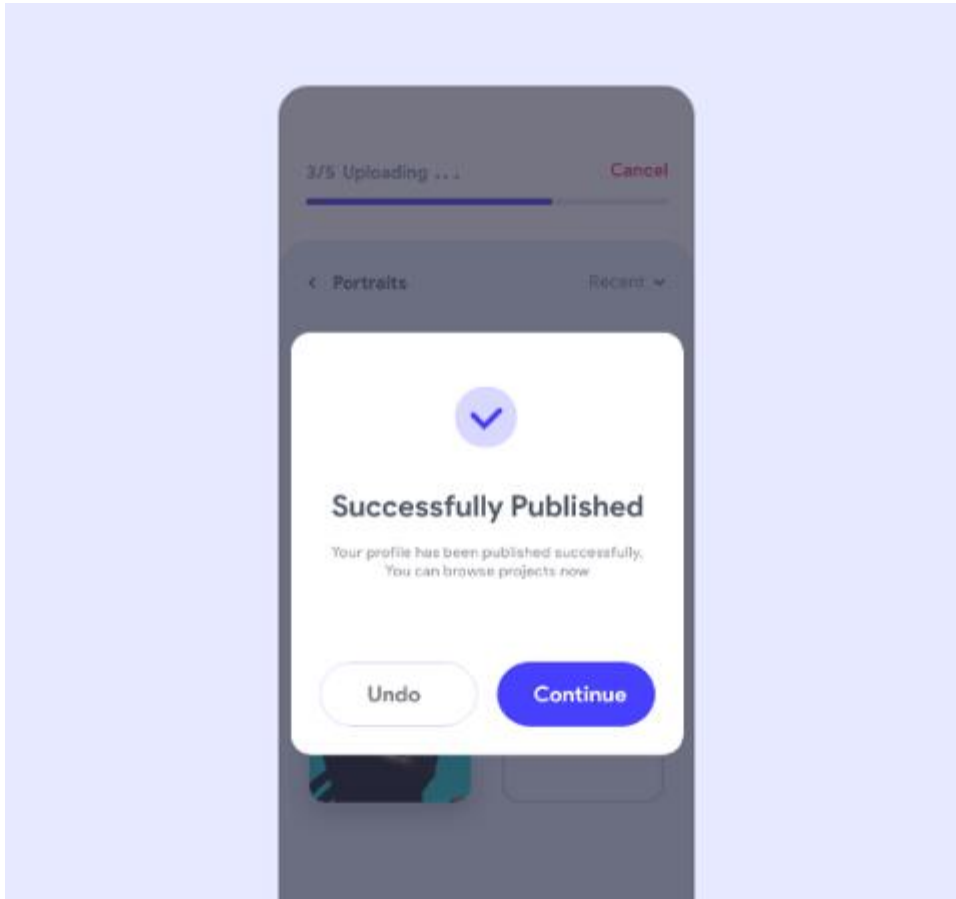
### **3. User Control and Freedom**

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.

### **4. Consistency and Standards**

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

A decorative footer consisting of two overlapping geometric shapes: a light gray trapezoid on the left and a dark gray trapezoid on the right, separated by a white diagonal line.



<https://uxdesign.cc/user-experience-is-one-of-the-hottest-topics-in-day-today-designers-life-fb314978e1ff>

- Support Undo and Redo.
- Show a clear way to exit the current interaction, like a Cancel button.
- Make sure the exit is clearly labeled and discoverable.



<http://www.brandflakesforbreakfast.com/2011/07/do-you-like-or-1.html>




- Improve learnability by maintaining both types of consistency: internal and external.
- Maintain consistency within a single product or a family of products (internal consistency).
- Follow established industry conventions (external consistency).

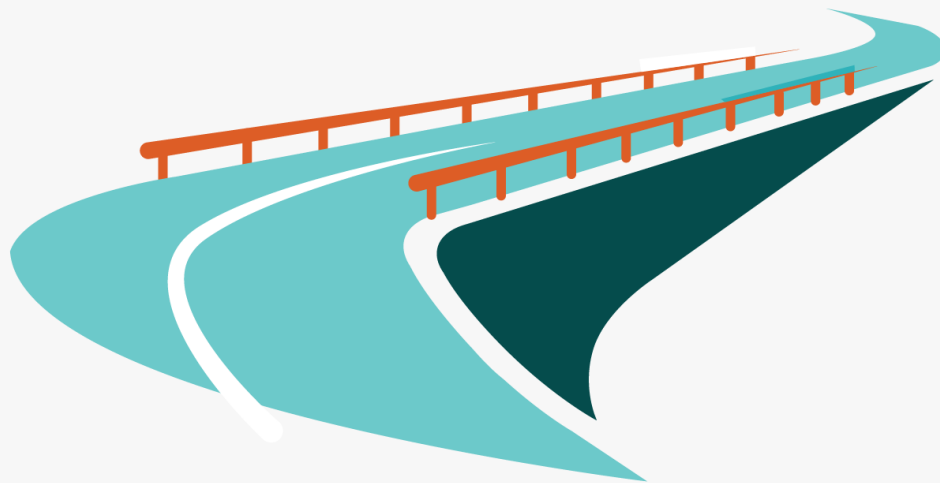
## **5. Error Prevention**

Good error messages are important, but the best designs carefully prevent problems 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**

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 (e.g. context-dependent help) should be visible or easily retrievable whenever appropriate.


A decorative graphic at the bottom of the slide consisting of two overlapping geometric shapes. The left shape is a light gray trapezoid, and the right shape is a darker gray trapezoid, separated by a white diagonal line.



**NN/g**  
NNGROUP.COM

- Prioritize your effort: Prevent high-cost errors first, then little frustrations.
- Avoid slips by providing helpful constraints and good defaults.
- Prevent mistakes by removing memory burdens, supporting undo, and warning your users.

# Google

- Q how do I
- Q how do i **get a scene card**
  - Q how do i **clear my cache**
  - Q how do i
  - Q how do i **contact ikea by phone**
  - Q how do i **contact westjet customer service**
  - Q how do i **contact air canada by phone**
  - Q how do i **take a screenshot**
  - Q how do i **contact cra by phone**
  -  **How Do I Live Without You**  
Song by Trisha Yearwood
  - Q how do i **say goodbye lyrics**

Google Search

I'm Feeling Lucky

[Report inappropriate predictions](#)


- Let people recognize information in the interface, rather than having to remember (“recall”) it.
- Offer help in context, instead of giving users a long tutorial to memorize.
- Reduce the information that users have to remember.

## **7. Provide Flexibility and Efficiency of Use**

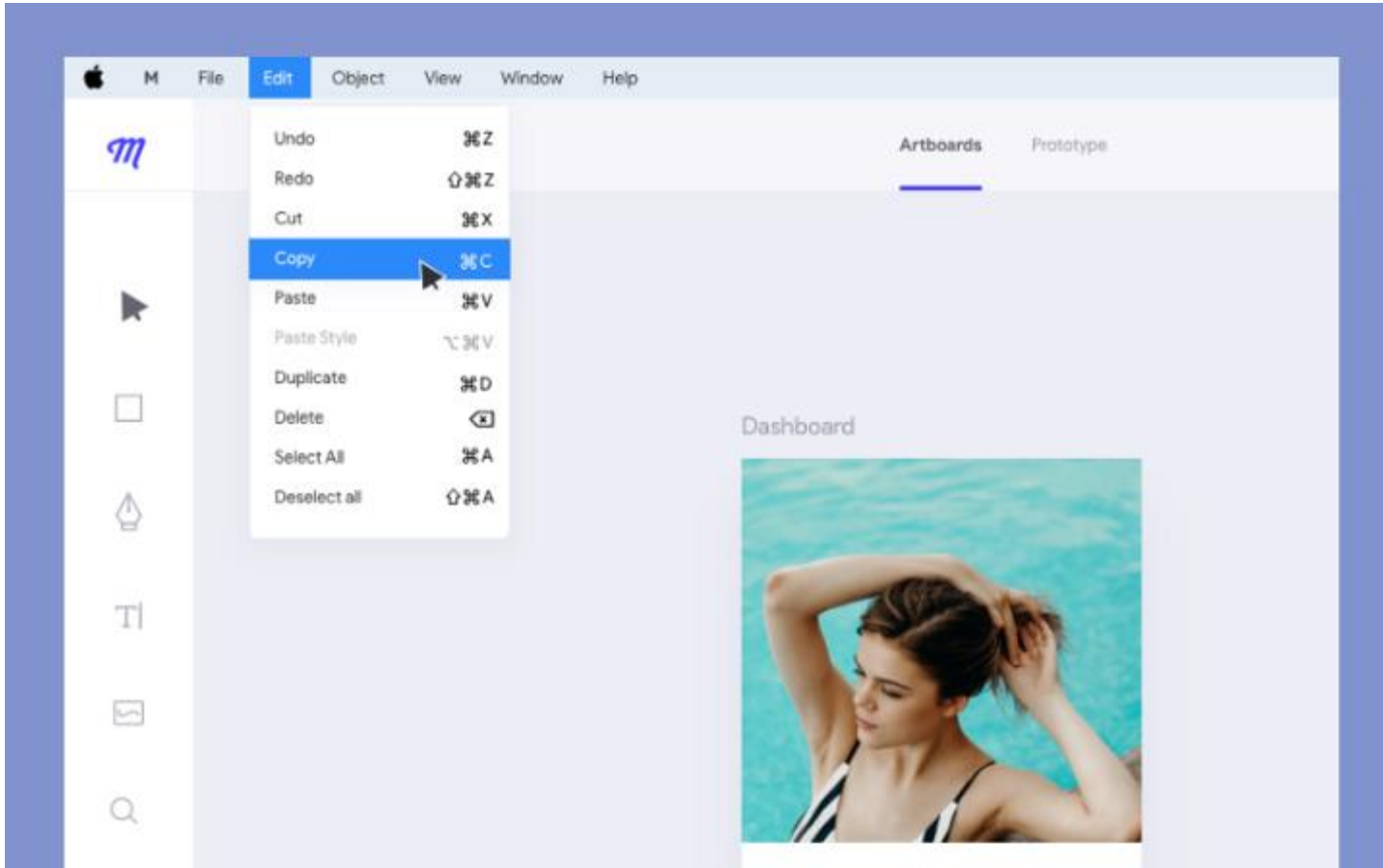
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

## **8. Focus on 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.

A decorative graphic at the bottom of the slide consisting of two overlapping geometric shapes. The left shape is a light gray trapezoid, and the right shape is a darker gray trapezoid. They are separated by a white diagonal line that runs from the bottom left towards the top right.





- Provide accelerators like keyboard shortcuts and touch gestures.
- Provide personalization by tailoring content and functionality for individual users.
- Allow for customization, so users can make selections about how they want the product to work.

<https://uxdesign.cc/user-experience-is-one-of-the-hottest-topics-in-day-today-designers-life-fb314978e1ff>



**NN/g**  
NNGROUP.COM

- Keep the content and visual design of UI focused on the essentials.
- Don't let unnecessary elements distract users from the information they really need.
- Prioritize the content and features to support primary goals.

## **9. Help Users to Recognise, 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.

Two side-by-side login form prototypes illustrating error message placement. Both forms feature a cartoon monkey logo at the top.

**Left Form (Username Error):**

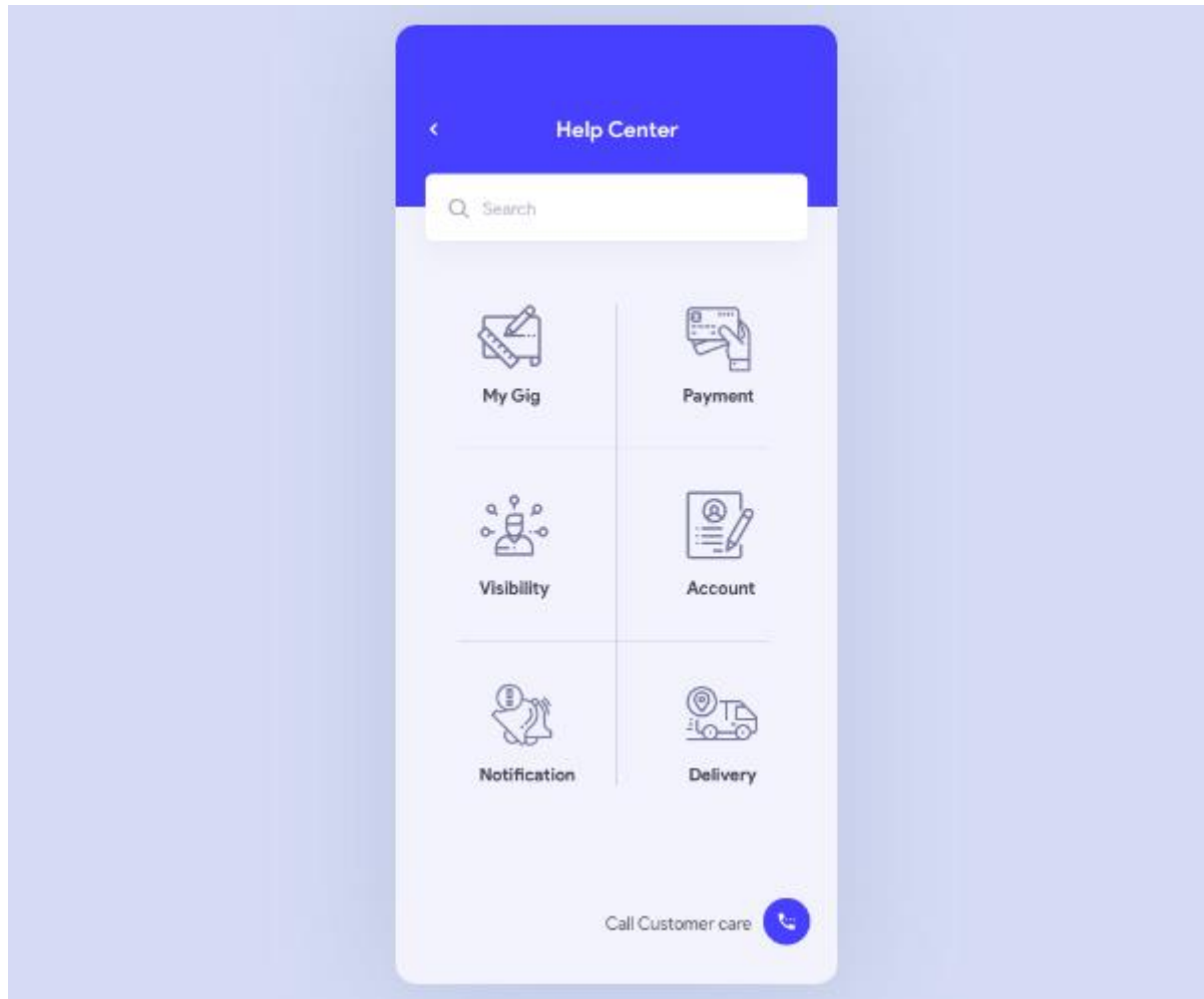
- Error message: "Sorry, we couldn't find an account with that username. Can we help you recover your username?"
- Username field: "freshsparkss" (highlighted with a red border)
- Password field: Empty (highlighted with a red border)
- Buttons: "Log In", "Stay logged in" (checkbox)
- Links: "Create an account", "Trouble logging in?"

**Right Form (Password Error):**

- Error message: "Sorry, that password isn't right. We can help you recover your password."
- Username field: "freshsparks" (highlighted with a red border)
- Password field: Empty (highlighted with a red border)
- Buttons: "Log In", "Stay logged in" (checkbox)
- Links: "Create an account", "Trouble logging in?"

<https://blog.prototypr.io/10-usability-heuristics-with-examples-4a81ada920c>

- Use traditional error message visuals, like bold, red text.
- Tell users what went wrong in language they will understand — avoid technical jargon.
- Offer users a solution, like a shortcut that can solve the error immediately.



<https://uxdesign.cc/user-experience-is-one-of-the-hottest-topics-in-day-today-designers-life-fb314978e1ff>

- Ensure that the help documentation is easy to search.
- Whenever possible, present the documentation in context right at the moment that the user requires it.
- List concrete steps to be carried out.

---

Visibility of system status

---

Match between the system and the real world

---

User control and freedom

---

Consistency and standards

---

Recognition rather than Recall

---

Help Users to recognise, diagnose, and recover from errors

---

Prevent errors

---

Provide flexibility and efficiency of use

---

Focus on aesthetic and minimalist design

---

Help and documentation

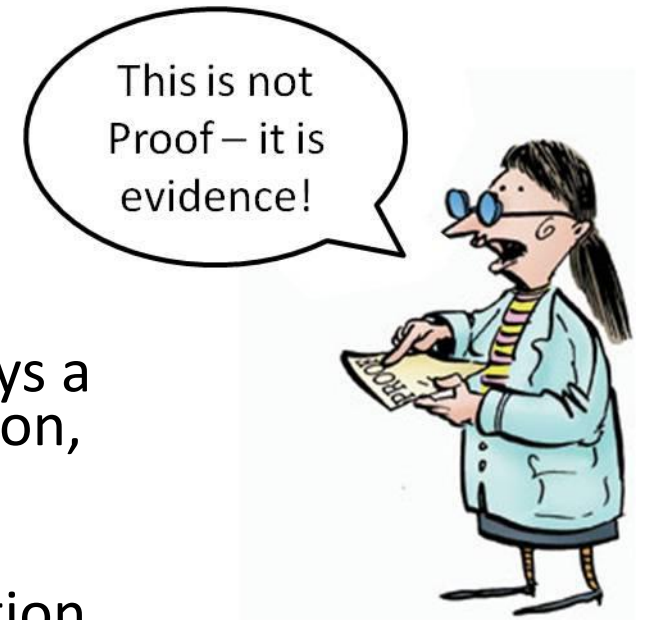
---

# Not only in IT Systems

- [https://www.zenhaiku.com/archives/usability\\_applied\\_to\\_life.html](https://www.zenhaiku.com/archives/usability_applied_to_life.html)
- Not a required reading....

# Using Heuristics

- You cannot really prove that an application obeys a heuristic unless you methodically, for every action, screen, and datum, show that the heuristic is followed (and you miss absolutely nothing).
- In general, you just don't have enough information, time, or resources to show that every heuristic is followed, all the time, and everywhere in the application.
- You can show examples of where a heuristic is NOT followed.
- Proving something is difficult – usually we find evidence that something is true. Disproving something is easy – any counter-example or evidence against a fact disproves it.





## Some Thoughts:

- If a user can do it, a user will do it.
- Double verify anything that can't be undone.
- You are NOT the user; you can't think like a user because you're a programmer.
- What information do you REALLY need? Don't collect information you don't need.
- Did you explore alternatives?
- Learnability is only an element of usability (particularly for novices) – information needs decrease with experience.
- Efficiency does not mean “fast.”
- K.I.S.S.
- UI Design is not the place for creativity!
- Try to balance:
  - The users right to choice vs. excessive need to make choices.
  - Flexibility vs. complexity.
  - Form (aesthetics, appearance) vs. function (effectiveness and utility).

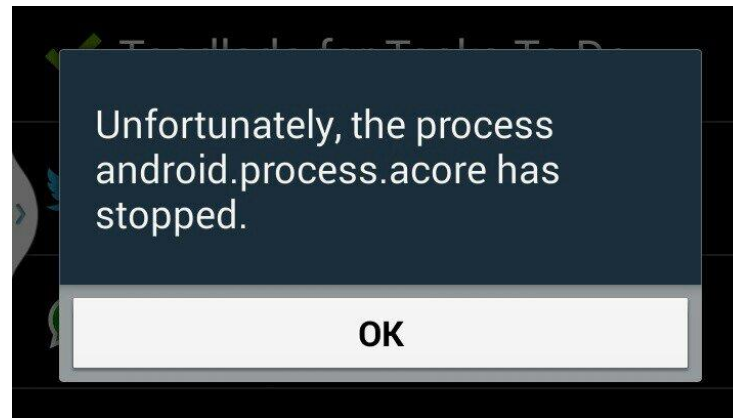
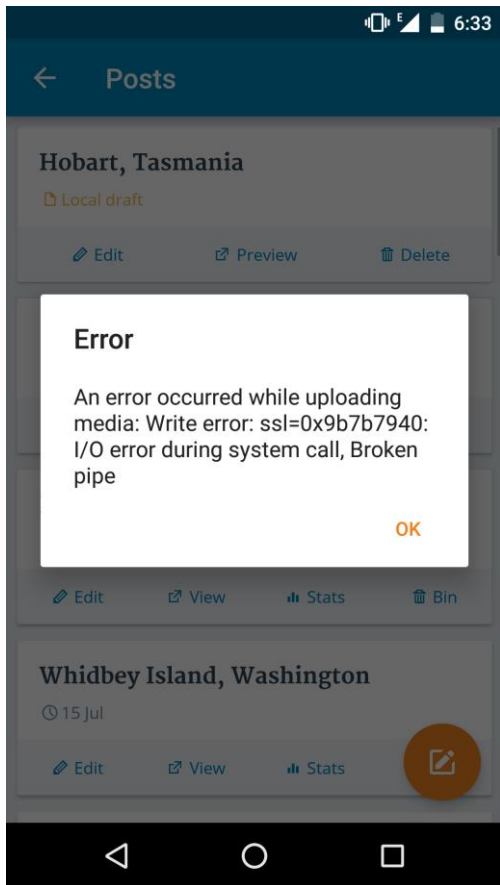
## Advanced Volume Control



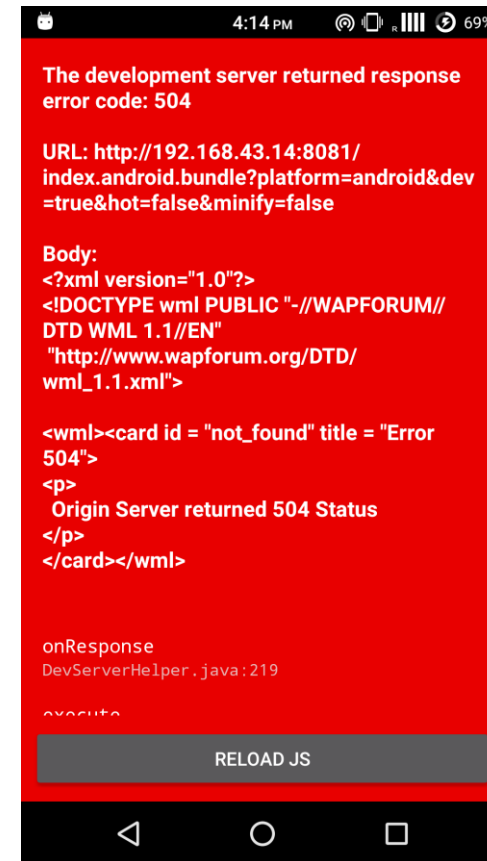
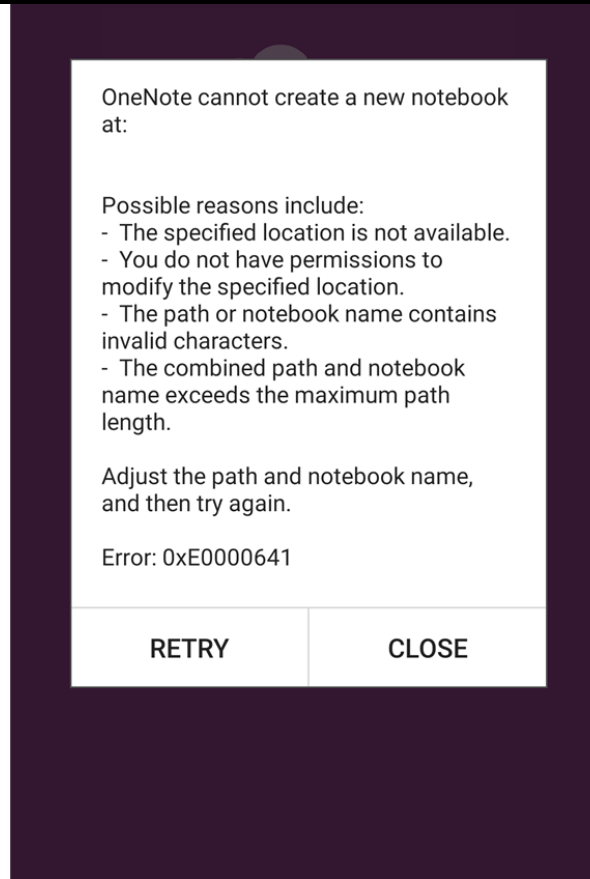
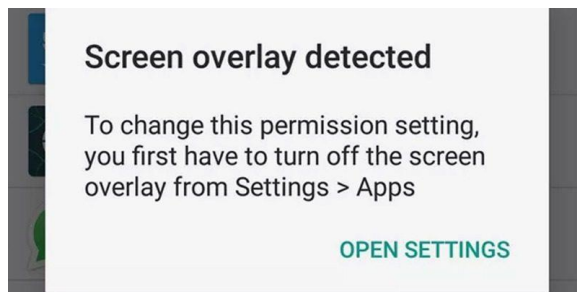
### Volume Control

<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9	<input type="radio"/> 10	<input type="radio"/> 11
<input type="radio"/> 12	<input type="radio"/> 13	<input type="radio"/> 14	<input type="radio"/> 15	<input type="radio"/> 16	<input type="radio"/> 17	<input type="radio"/> 18	<input type="radio"/> 19	<input type="radio"/> 20	<input type="radio"/> 21	<input type="radio"/> 22
<input type="radio"/> 23	<input type="radio"/> 24	<input checked="" type="radio"/> 25	<input type="radio"/> 26	<input type="radio"/> 27	<input type="radio"/> 28	<input type="radio"/> 29	<input type="radio"/> 30	<input type="radio"/> 31	<input type="radio"/> 32	<input type="radio"/> 33
<input type="radio"/> 34	<input type="radio"/> 35	<input type="radio"/> 36	<input type="radio"/> 37	<input type="radio"/> 38	<input type="radio"/> 39	<input type="radio"/> 40	<input type="radio"/> 41	<input type="radio"/> 42	<input type="radio"/> 43	<input type="radio"/> 44
<input type="radio"/> 45	<input type="radio"/> 46	<input type="radio"/> 47	<input type="radio"/> 48	<input type="radio"/> 49	<input type="radio"/> 50	<input type="radio"/> 51	<input type="radio"/> 52	<input type="radio"/> 53	<input type="radio"/> 54	<input type="radio"/> 55
<input type="radio"/> 56	<input type="radio"/> 57	<input type="radio"/> 58	<input type="radio"/> 59	<input type="radio"/> 60	<input type="radio"/> 61	<input type="radio"/> 62	<input type="radio"/> 63	<input type="radio"/> 64	<input type="radio"/> 65	<input type="radio"/> 66
<input type="radio"/> 67	<input type="radio"/> 68	<input type="radio"/> 69	<input type="radio"/> 70	<input type="radio"/> 71	<input type="radio"/> 72	<input type="radio"/> 73	<input type="radio"/> 74	<input type="radio"/> 75	<input type="radio"/> 76	<input type="radio"/> 77
<input type="radio"/> 78	<input type="radio"/> 79	<input type="radio"/> 80	<input type="radio"/> 81	<input type="radio"/> 82	<input type="radio"/> 83	<input type="radio"/> 84	<input type="radio"/> 85	<input type="radio"/> 86	<input type="radio"/> 87	<input type="radio"/> 88
<input type="radio"/> 89	<input type="radio"/> 90	<input type="radio"/> 91	<input type="radio"/> 92	<input type="radio"/> 93	<input type="radio"/> 94	<input type="radio"/> 95	<input type="radio"/> 96	<input type="radio"/> 97	<input type="radio"/> 98	<input type="radio"/> 99
<input type="radio"/> 100										

☐ Mute



Which are good and which are bad? Why? How do you fix them?

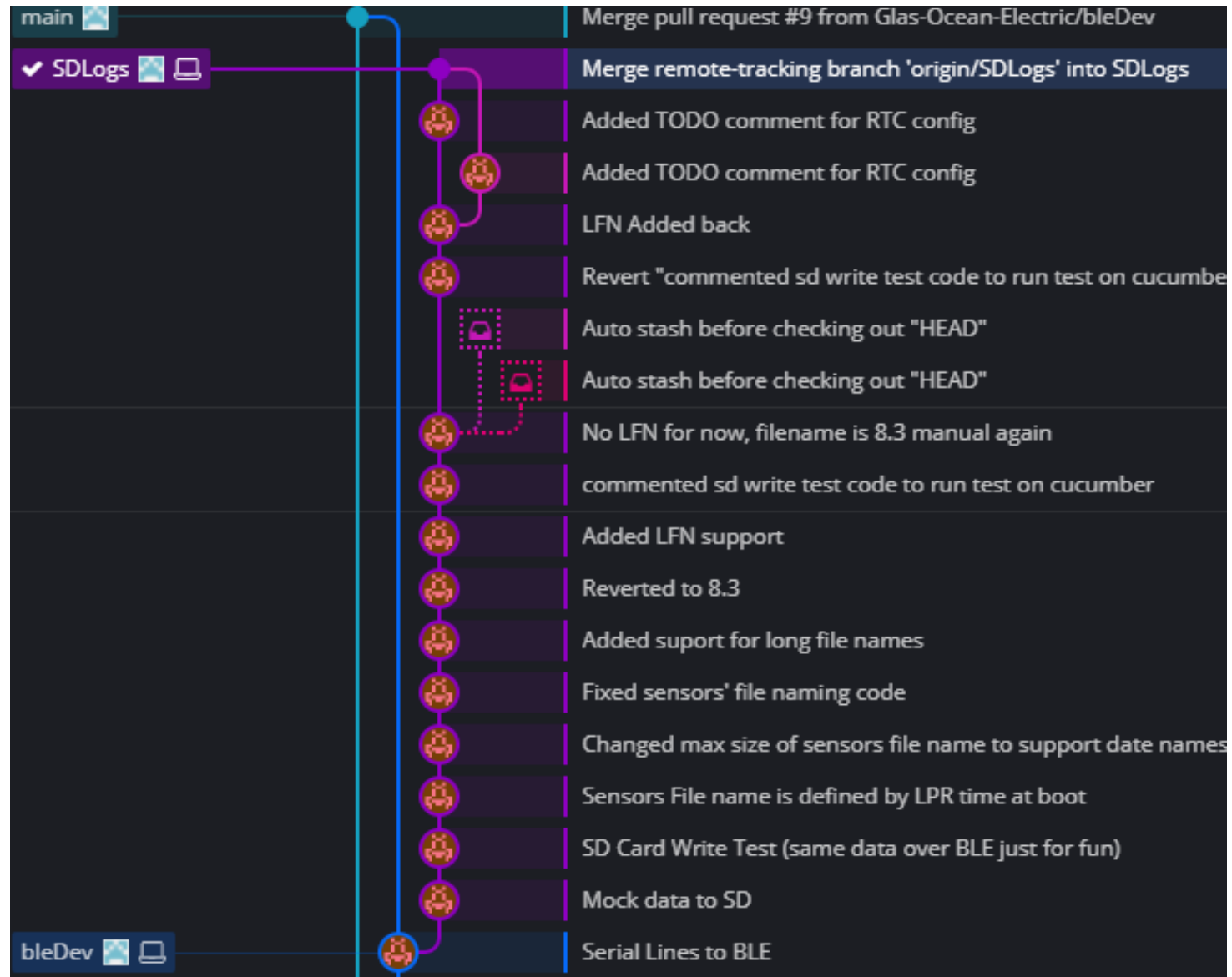


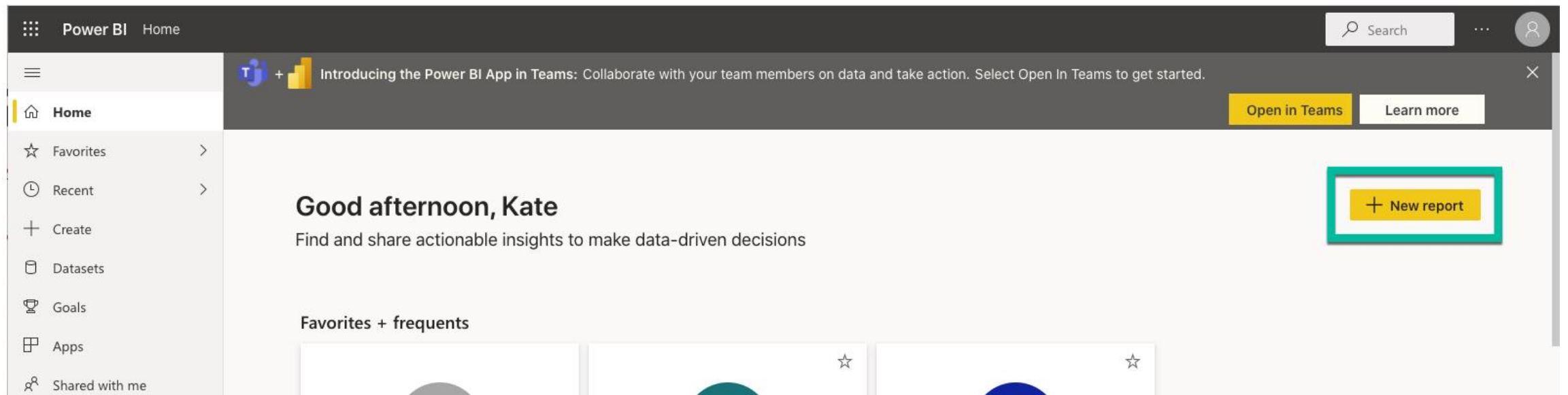
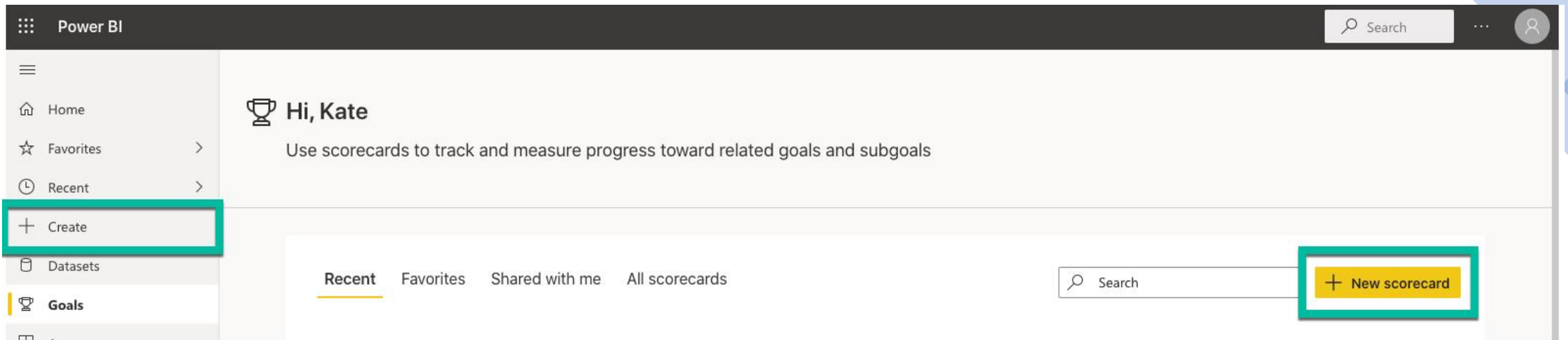
## View Report

Please wait, generating report...









Azure DevOps

AKD-20 Integration

Queries > Shared Queries > PLANNING - Sorted by Date - Ne...

48 work items  
48 top level, 0 linked and 1 selected

Results Editor Charts

Run query + New Save query Rename Save items ...

11 of 48

Due Date	Title	State	Assigned To	Iteration Path	Parent
3/31/2020 12:00 AM	AKD-20 Integration - Conceptual Design	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - System Documentation	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/8/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/9/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/14/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/15/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/15/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		

Azure DevOps

AKD-20 Integration

Queries > Shared Queries > PLANNING - Sorted by Date - Ne...

48 work items  
48 top level, 0 linked and 1 selected

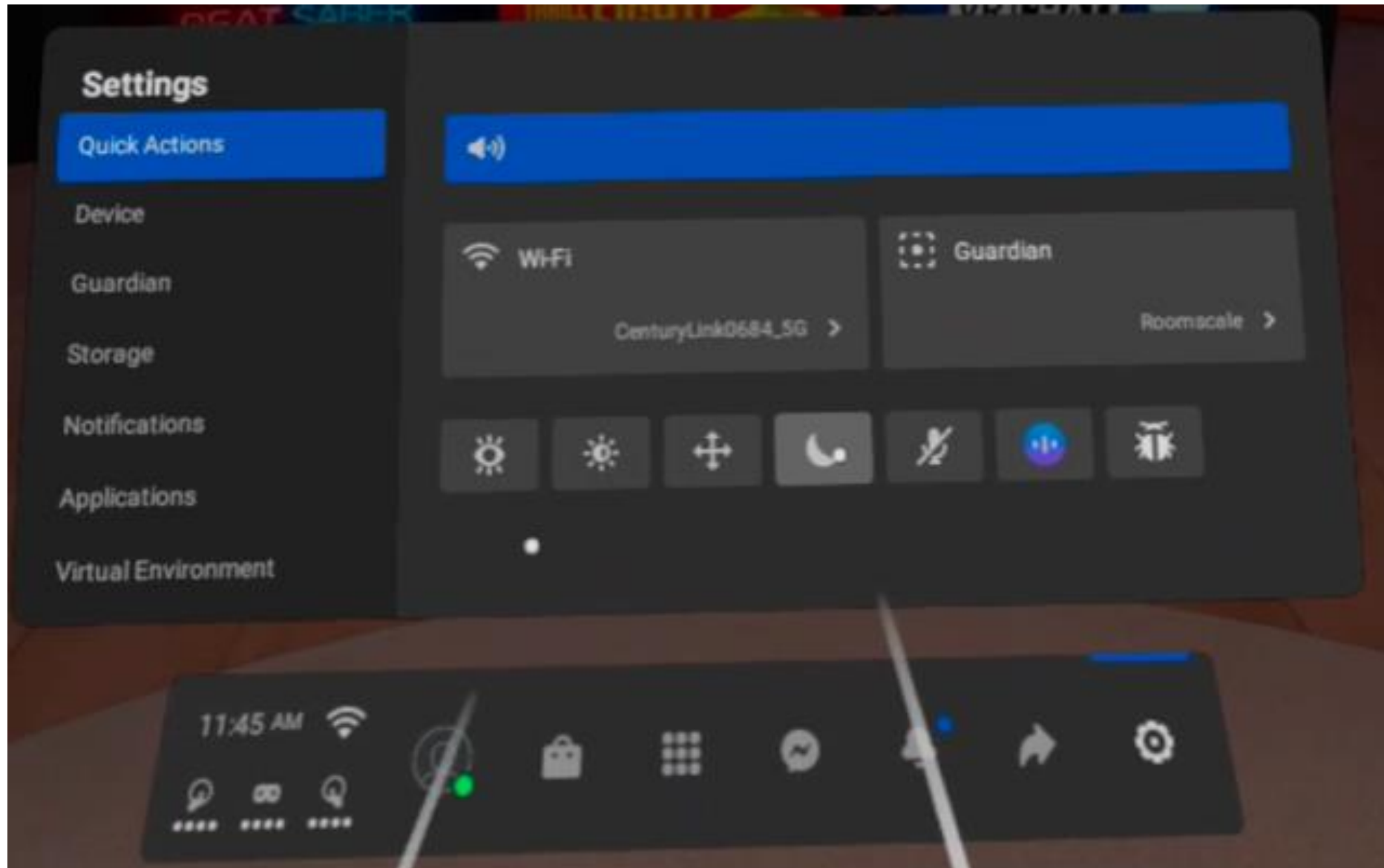
Results Editor Charts

Run query + New Save query Rename Save items ...

11 of 48

Due Date	Title	State	Assigned To	Iteration Path	Parent
3/31/2020 12:00 AM	AKD-20 Integration - Conceptual Design	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - System Documentation	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/1/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/8/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/9/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/14/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/15/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		
4/15/2020 12:00 AM	AKD-20 Integration - User Stories	New	AKD-20 Integration		





## Unable to Deploy

Your project cannot be deployed due to the following invalid item(s). Please correct these errors and try again.

### *New Operation (13)*

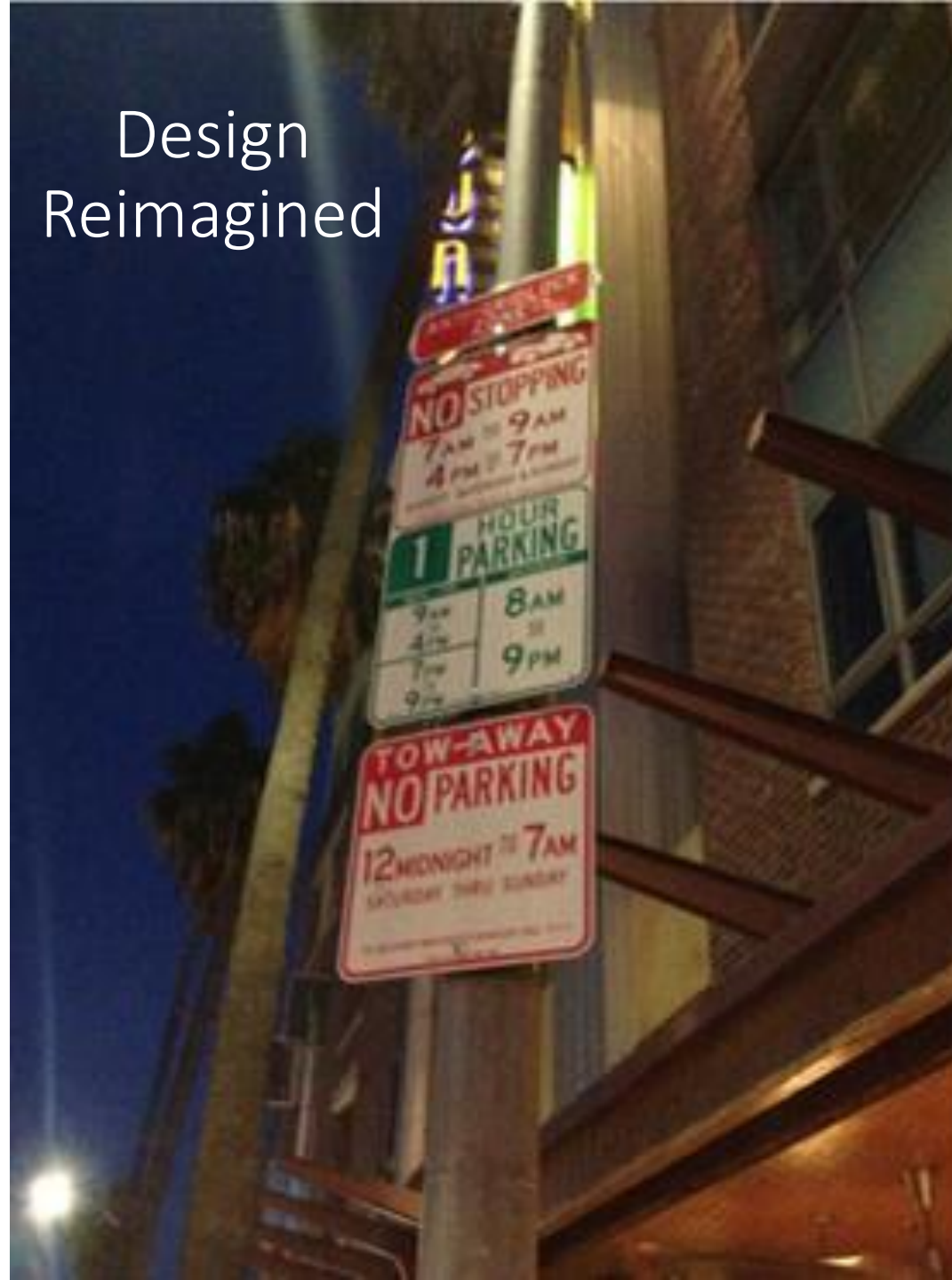
- Operation cannot contain more than one application/SOAP activity Query (5).  
Operation rules and patterns can be found [here](#).

























### *Query (7)*

- Activity is not configured properly

Continue

# Design Reimagined




PARKING SCHEDULE			
	M-F	SAT	SUN
12am			
7am			
8am			
9am			
4pm			
7pm			
9pm			
12pm			





# PARKING SCHEDULE

	M	T-F	Sat/ Sun
8am	 Free	 Free	 Free
10am		 2 hr	
	 2 hr		
6pm	 Free	 Free	

