

Accessible and User-Friendly Drag-and-Drop Questions

Emilie Berglund, Senior Manager, Accessibility Implementation
Varun Goyal, Lead UI Engineer

McGraw Hill

Introduction

Emilie Berglund, Senior Manager,
Accessibility Implementation

emilie.berglund@mheducation.com

Varun Goyal, Lead UI Engineer

varun.goyal@mheducation.com

Session summary

Drag-and-drop questions are popular but often inaccessible. In an effort to create drag-and-drop questions for all students, McGraw Hill updated Ordering and Matching question types for WCAG compliance in 2018. In 2019, we focused on usability improvements for screen reader users. We will share our progress and lessons learned.

Example 1: Ordering Type 1

Student reorders choices within a single list.

Order the fruits alphabetically.



apple

banana

cherry

Example 2: Ordering Type 2

Student move choices from one list to a set of ordered drop zones.

Order the fruits alphabetically.

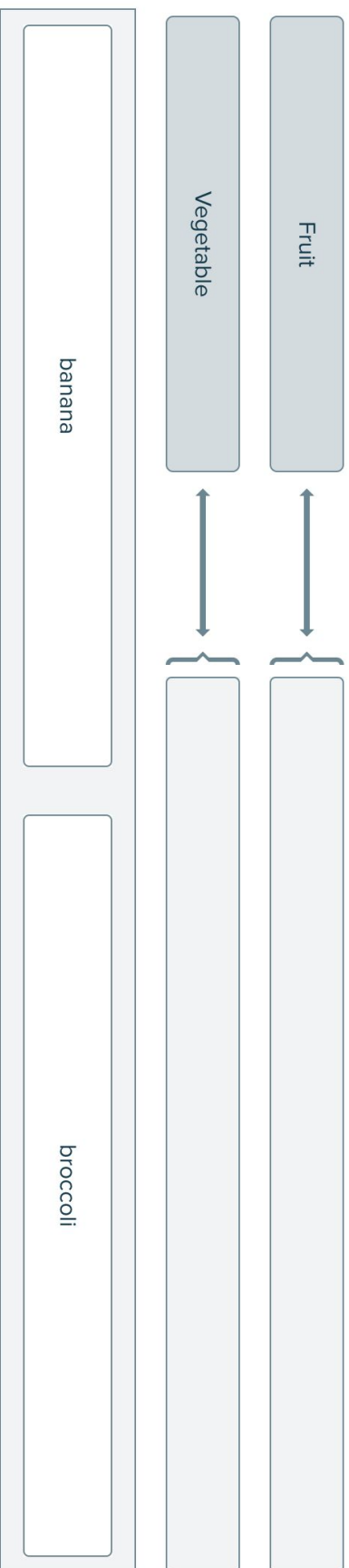
An interactive interface for ordering fruits. It features a source list on the left and three target drop zones on the right. The source list contains three items: 'cherry', 'banana', and 'apple'. The target zones are empty, intended for the user to drag and drop the items into them in alphabetical order.

cherry
banana
apple

Example 3: Matching 1 to 1

Student matches choices to prompts where each prompt has exactly one matching choice.

Match each food to its food group.

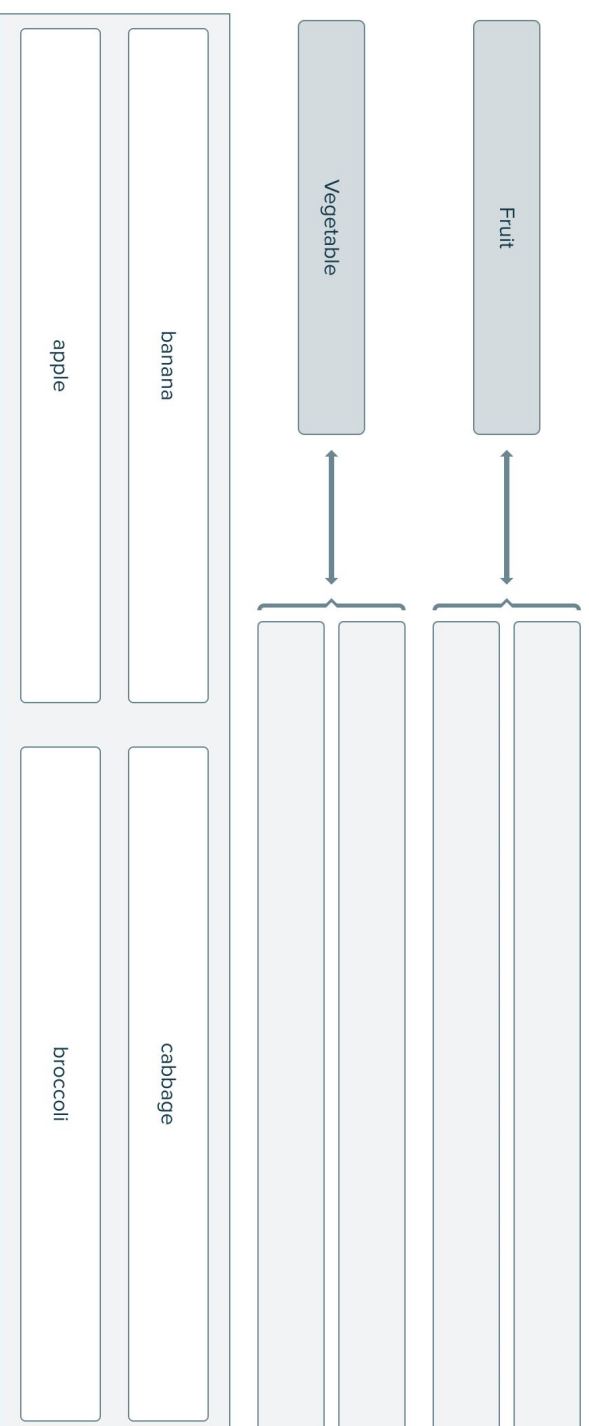


Example 4: Matching 1 to Many

Student matches choices to prompts where each prompt can have more than one matching choice.

Match each food to its food group. Each food group has more than one matching choice.

①



Today's Topics

Project background

Initial challenges

Accessibility implementation process

- Defining screen reader behavior
- Third-party audit
- User testing (for screen reader)

Demonstration of the “final” results with JAWS

Terminology

MH: McGraw Hill

Assess: name of the MH assessment product updated

RBD: REACT Beautiful DND

Assignment versus assessment:

- Used interchangeably today
- Assignment of the “assessment” type

Project Summary and Scope

High-level project summary

- Initiated in June 2018
- Update Assess, the existing assessment product for MH's new platform, to be WCAG 2.1 AA compliant
- Incremental rollout

Project scope

- Student assignment experience, including “take” mode and “post-submission” mode
- Assignment pages, features within the pages, question types

Project Unknowns

How would we create compliant and accessible Ordering and Matching questions?

MH had millions of dollars worth of content in drag-and-drop questions. Could that content be saved?



Where We Started

Initial Challenges

Engagement with Team

Existing Content

Existing Code

Initial Challenges

Limited time on a large project

1. Partnering with a team without experience in accessibility implementation
2. Requirement to support existing content
3. Existing code base with multiple teams involved

Development of accessible Ordering and Matching questions = NEW!

4. Defining screen reader behavior
5. Interacting with content via screen reader



Challenge 1: Lack of A11y Experience

We addressed this through

- In-person training with the product team on a11y implementation
 - Importance of supporting accessibility--empathy and laws
 - Testing tools, assistive technologies
 - Deep dive into feature-specific development
- Partnering with the product team throughout the process
 - UX discussions
 - Defining expected behavior
 - Testing

Challenge 2: Existing Content

“Blank slate” was not possible given thousands of existing Ordering and Matching questions. We needed to

- NOT “break” content,
- NOT require authoring updates.

We decided to

- Maintain current UI,
- Determine IF we could make compliant,
- Conduct user testing to refine the experience.

Challenge 3: Existing Code Base

Existing code written in REACT.

The REACT Beautiful DND (RBD) open-source library was selected after investigation, understanding that customization of RBD might be needed.

Key was to figure out

- What accessibility support did RBD provide?
- What accessibility support did we need to add?

Challenge 4: Interacting with Content

Choices: How would we interact with them when using a screen reader?

- Not many accessible models to follow
- Research and audits provided contradicting information

Final result was implementation of role="application"

- Move to choice: TAB
- Selection and dropping: SPACEBAR
- Move a selected choice: ARROW KEYS
- Cancel selection or movement: ESC

Challenge 5: Defining Screen Reader Behavior

Where would we start?

Could we reference what already exists?

Ultimately, the strategy that prevailed:

- Working through items with JAWS and used empathy to define the experience.
- Starting with least complex, transition through to most complex types

Screen Reader Behavior Common to All 4

The active element: choices



Every choice had

- Label
- Role: toggle button
- States: pressed or not pressed (draggable)

Screen Reader Behavior: Ordering Type 1

Where am I?

2 of 3

Order the fruits alphabetically.




cherry
banana
apple

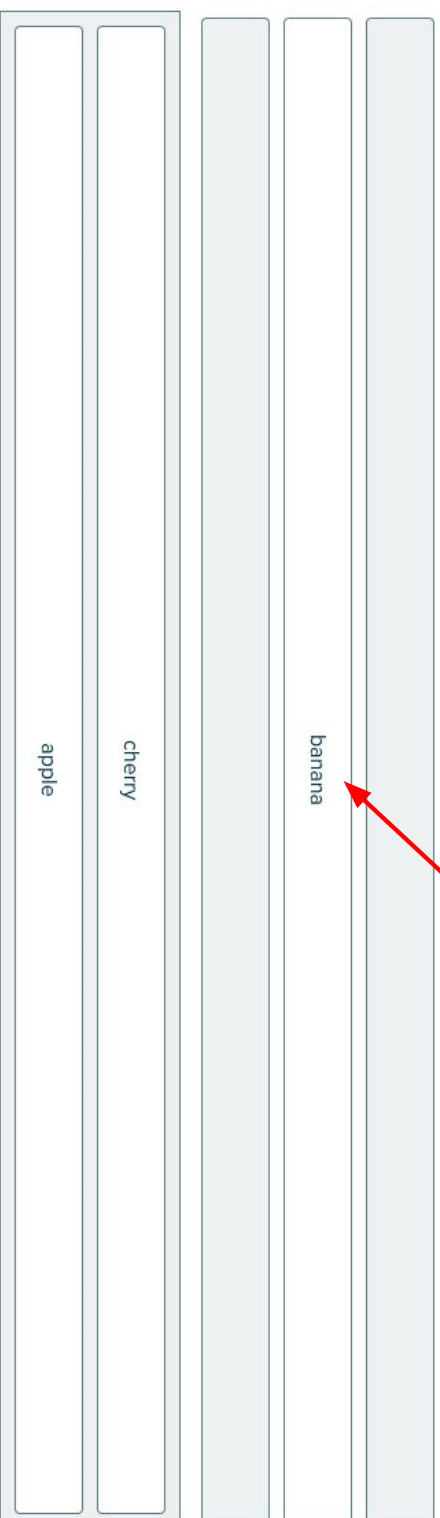


Screen Reader Behavior: Ordering Type 2

Where am I?

Order the fruits alphabetically. 

drop zone 2 of 3



banana

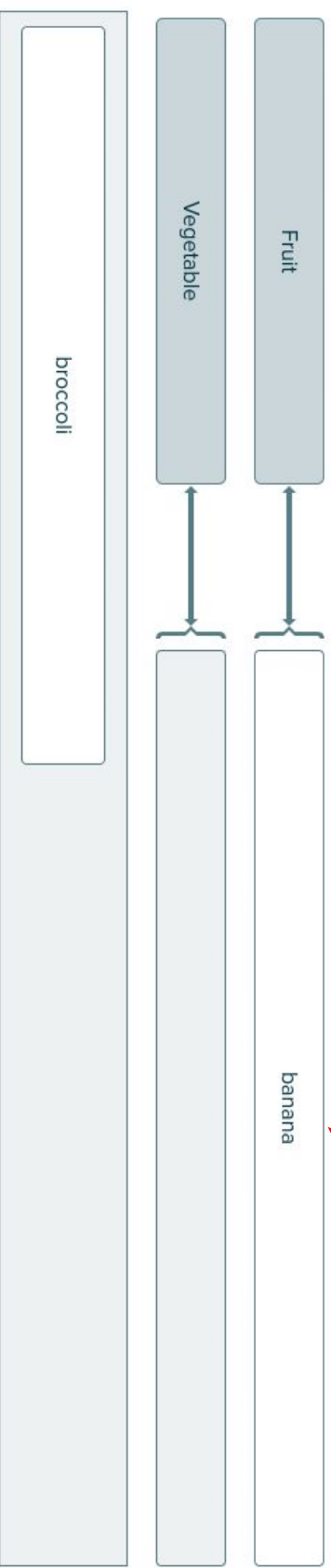
cherry

apple

Screen Reader Behavior: Matching 1 to 1

Where am I?

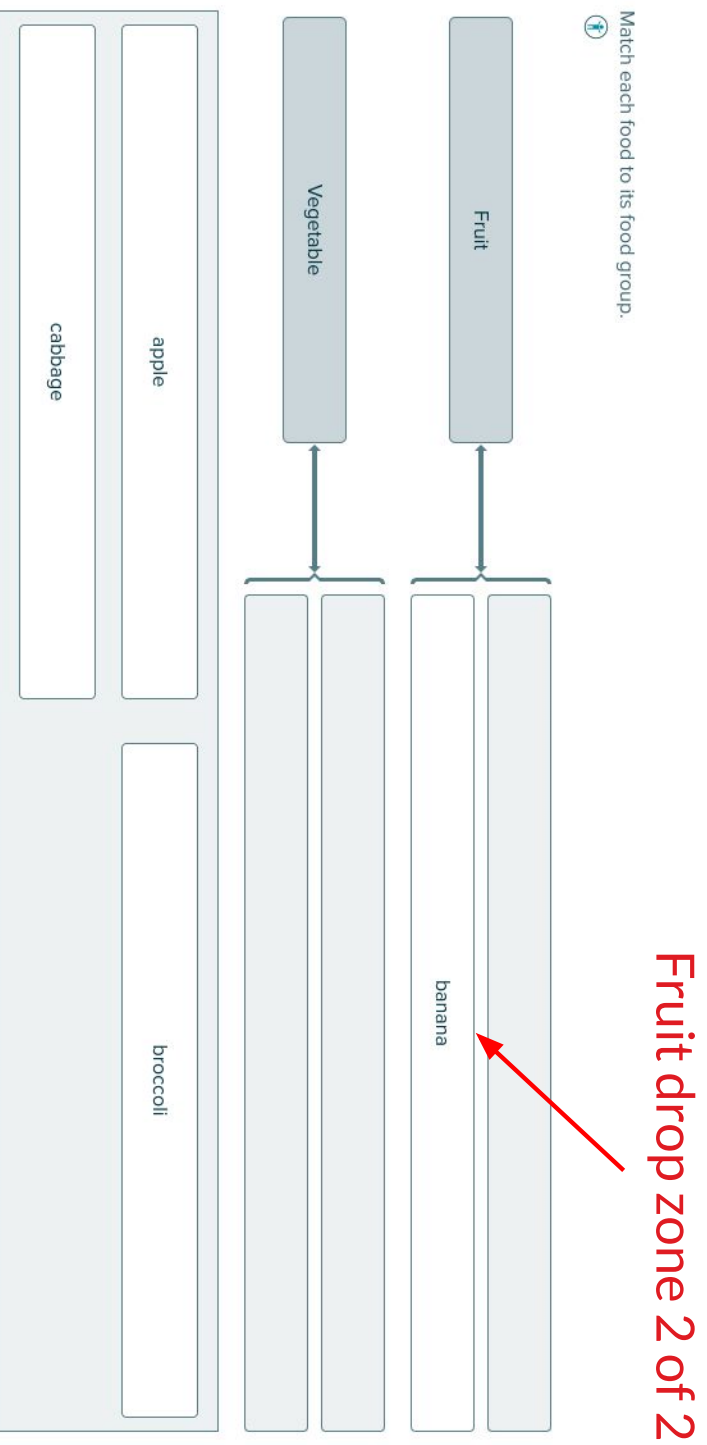
Match each food to its food group.



Fruit drop zone

Screen Reader Behavior: Matching 1 to Many

Where am I?



Audit and User Testing

WCAG 2.0 AA Compliance Audit

User Testing

Audit Results

Two third-party vendors audited.

Issues reported were

- Use of role="application",
- Minor code issue,
- High-contrast mode,
- Wordiness--too much and not enough,
- Answering lengthy questions with magnification at 200% is challenging.

Compliant but *What about Accessible?*

Final grade was compliant!

Now... How could we improve experience for screen reader users?

User Testing

User testing completed on a representative sample of Ordering and Matching questions:

- Testing completed by the MH user testing group both in person and remotely
- 8 screen reader-dependent students
- Ages ranged from 7th grade to college age
- JAWS screen reader was used
- Session length was 1 hour
- Script defined by group for testing

Overall Observations

1. Age and maturity impacted behavior and patience to work through the content.
2. Order of the questions impacted understanding.
3. Matching questions – the most challenging to define – were the simplest question types for the student testers.
4. Our goal was to identify areas for improvement for screen reader users. We had achieved compliance. MH group mentality shifted to finding fault.
5. One unsuspecting feature caused the most issues unexpectedly!

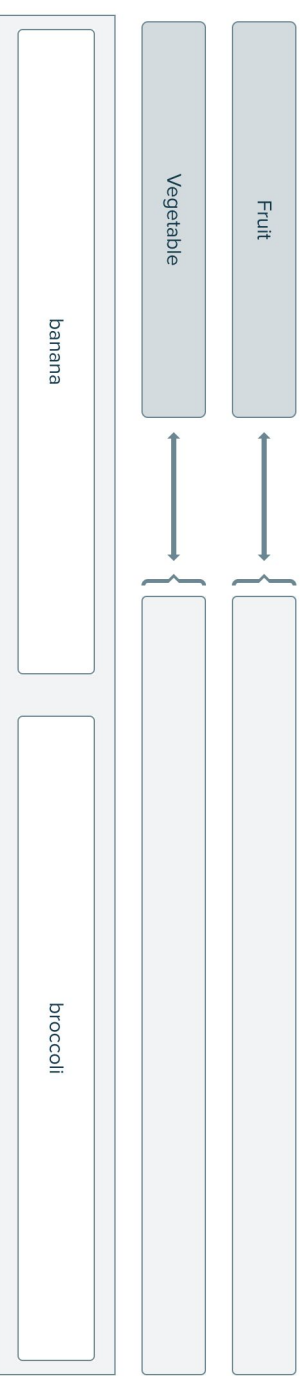
Challenges Users Experienced

The most challenging portions of the testers' experiences were

- Selection of choice via ENTER
- Movement of choices within choices list
- Location of drop zones
- Accessibility tooltip

Match each food to its food group.

①



Biggest Impact: Tooltip

Original: Accessibility Tooltip



Press TAB to move the focus to a choice. Press SPACEBAR to select a choice, DOWN ARROW and UP ARROW to move the selected choice, and SPACEBAR again to drop the choice. Press Escape to cancel your action.

Update: Instructions Toggle Tip



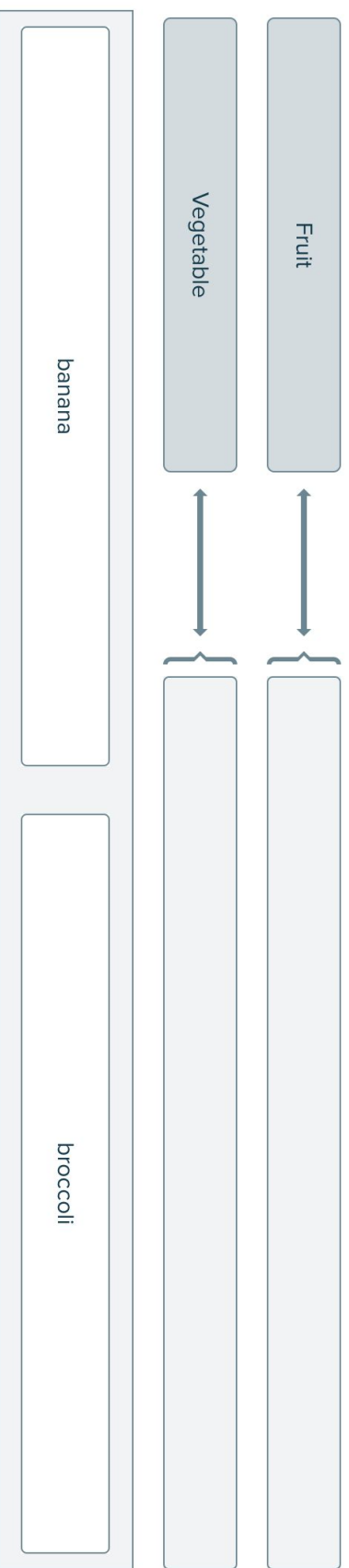
There are 3 choices to reorder.
If using your keyboard to complete this question, press TAB to move the focus to a choice. Press SPACEBAR to pick up or drop a choice. Press DOWN ARROW and UP ARROW to move a choice. Press ESCAPE to cancel your action.

Results per User Testing and Next Steps

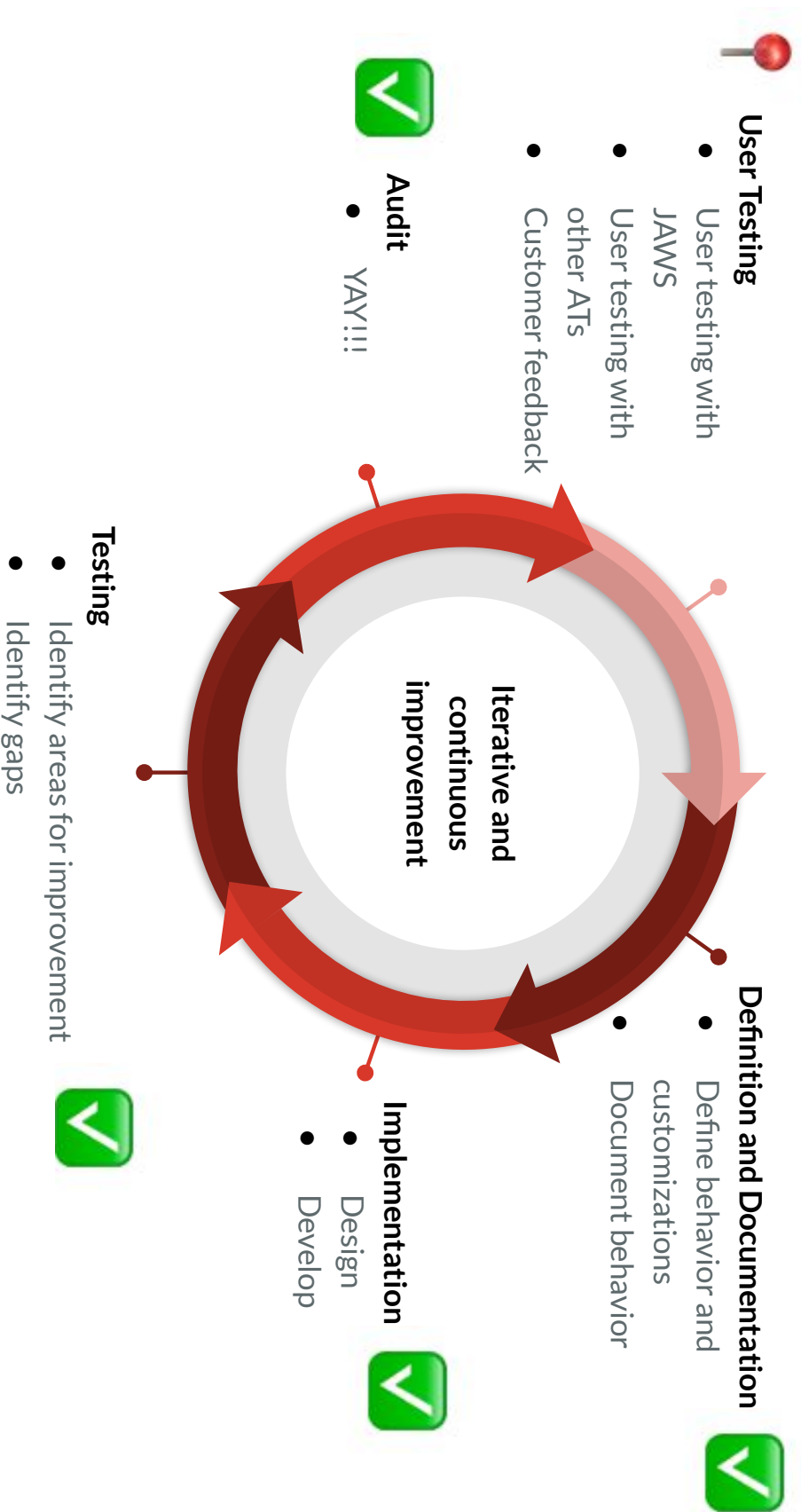
We synthesized the feedback, identifying

- Straight-forward updates and
- Other challenges requiring investigation and further testing.

Match each food to its food group.



Our Process



Demonstration

JAWS Screen Reader
