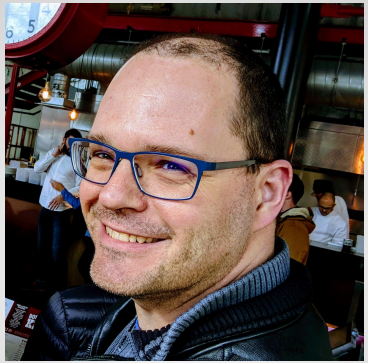
Escaping Legacy

Bringing a legacy system under test

Who are we?



Wouter Lagerweij

@wouterla

Karel Boekhout

@karelboekhout

Agenda

- Introduction
- Sprint 1: The Heat Map
- Sprint 2: The spine of a Story Map
  - **Break**
- Sprint 3: Story Map and slices into key journies
- Sprint 4: Example Mapping: uncovering the details
  - **Break**
- Sprint 5: Priorities: value and complexity, slices and stories
- Sprint 6: Validation: testing with low and high resolution
- Closing

You know this one...

Source: Jeff Patton

Let’s create this one!

- Usually not greenfields, but a big ball of legacy application
- No tests available
- Practical: you can start today

What do we have?

Do you care?

What does this %^]#$ application do?

Let's start with a Mental Breakdown

Will it keep working?

What does it do?

Does it do what you think it does?

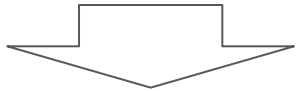
Agenda

- Introduction
- **Sprint 1: The Heat Map**
- Sprint 2: The spine of a Story Map
  - **Break**
- Sprint 3: Story Map and slices into key journies
- Sprint 4: Example Mapping: uncovering the details
  - **Break**
- Sprint 5: Priorities: value and complexity, slices and stories
- Sprint 6: Validation: testing with low and high resolution
- Closing

Start a ‘Heatmap’

The Functional Breakdown

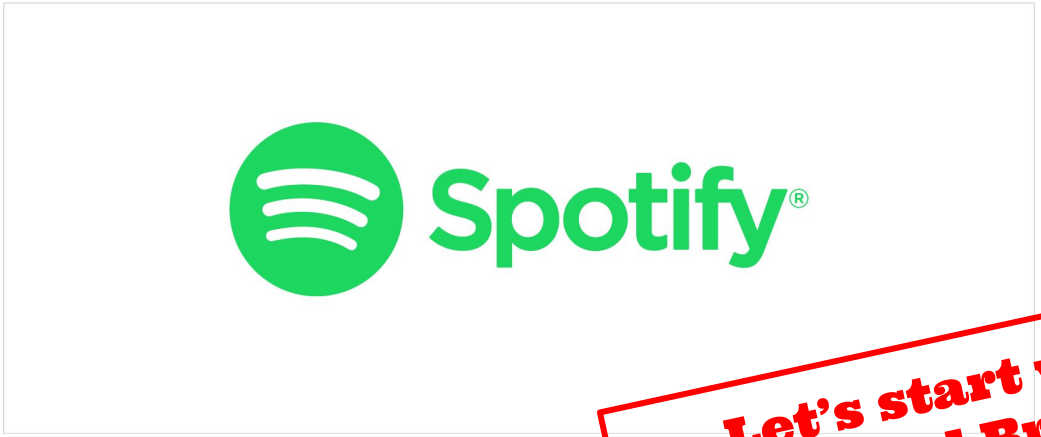
Colour coded for the testing we have available



The Heatmap

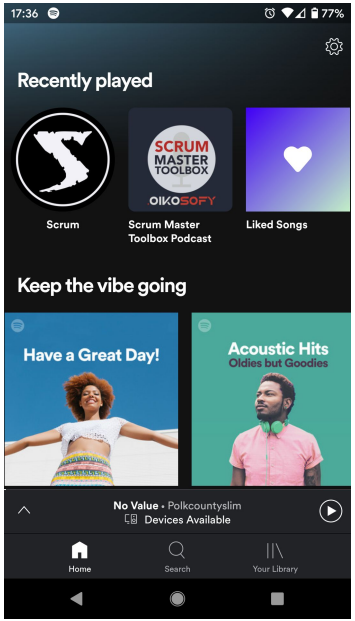
- Our overview for the level of control over our system:
  - What are we testing?
  - How are we testing it?

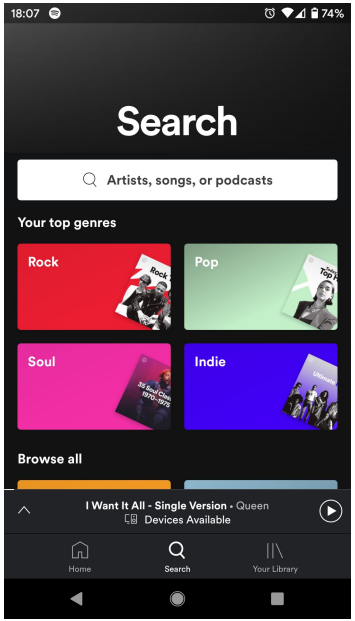
Our example

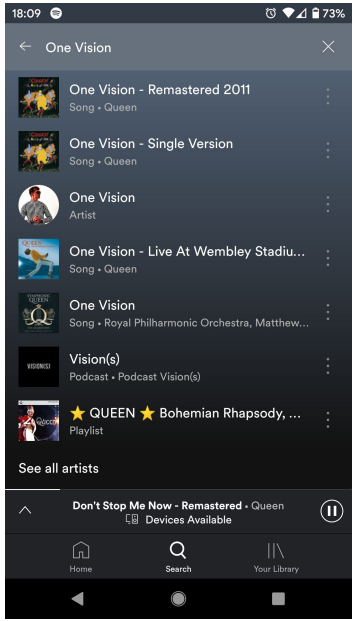


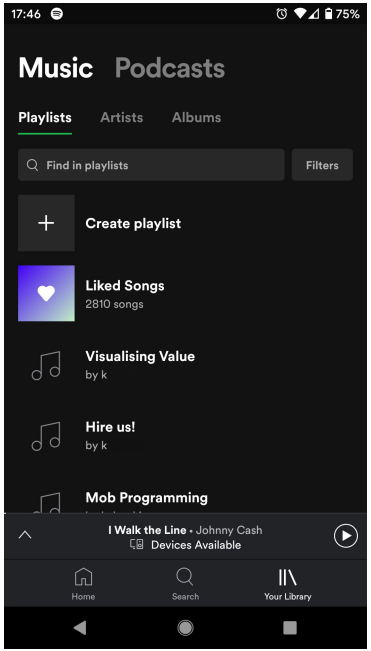
Let's start with a Functional Breakdown

Spotify


17:36
77%
Recently played
Scrum
Scrum Master Toolbox Podcast
Liked Songs
Keep the vibe going
Have a Great Day!
Acoustic Hits
Oldies but Goodies
No Value • Polkcountyslim
Devices Available
Home
Search
Your Library

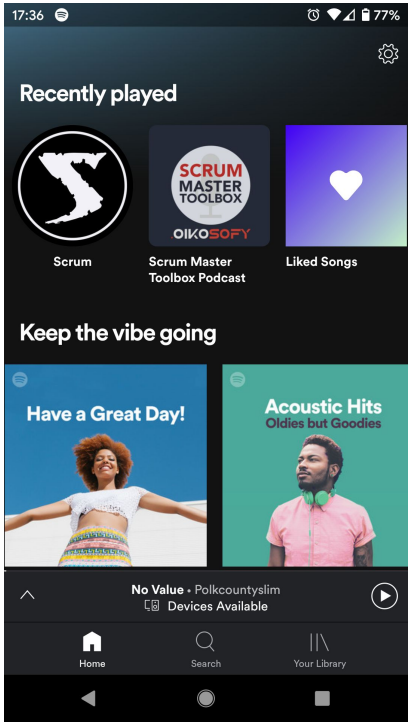

18:07
74%
Search
Artists, songs, or podcasts
Your top genres
Rock
Pop
Soul
Indie
Browse all
I Want It All - Single Version • Queen
Devices Available
Home
Search
Your Library


18:09
73%
One Vision
One Vision - Remastered 2011
Song • Queen
One Vision - Single Version
Song • Queen
One Vision
Artist
One Vision - Live At Wembley Stadiu...
Song • Queen
One Vision
Song • Royal Philharmonic Orchestra, Matthew...
Vision(s)
Podcast • Podcast Vision(s)
★ QUEEN ★ Bohemian Rhapsody, ...
Playlist
See all artists
Don't Stop Me Now - Remastered • Queen
Devices Available
Home
Search
Your Library


17:46
75%
Music
Podcasts
Playlists
Artists
Albums
Find in playlists
Filters
Create playlist
Liked Songs
2810 songs
Visualising Value
by k
Hire us!
by k
Mob Programming
I Walk the Line • Johnny Cash
Devices Available
Home
Search
Your Library


19:47
65%
PLAYING FROM SEARCH
"just pu" in Songs
Just Push Play
Aerosmith
0:04
3:51

Spotify: Home



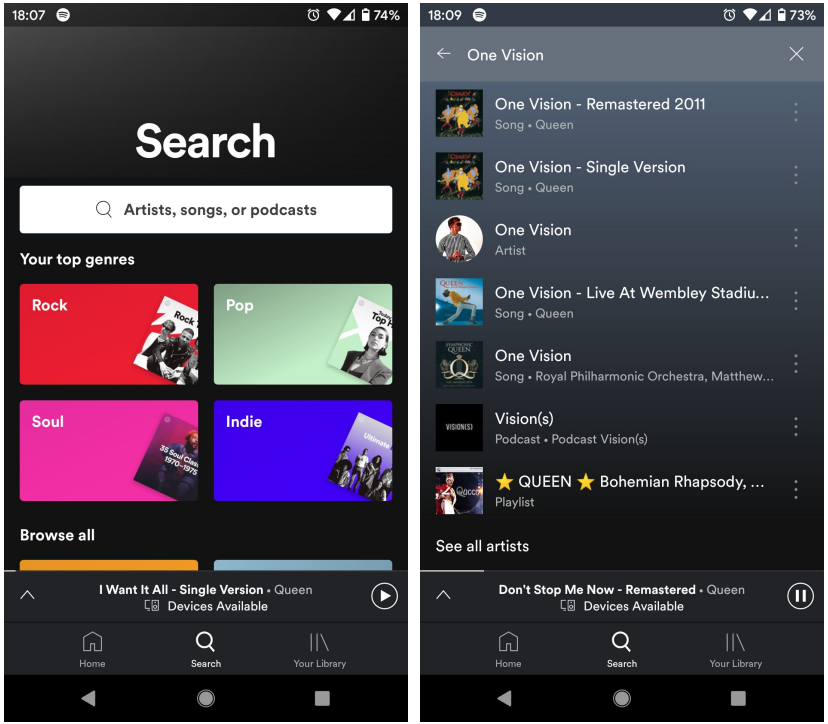
Home screen

- Your heavy rotation
- Mood
- Recently played
- Keep the vibe going
- Made for ….
- More of what you like
- <suggestions>

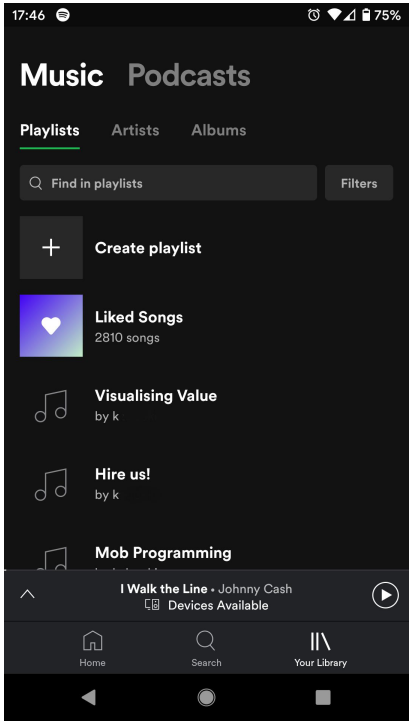
Spotify: Search

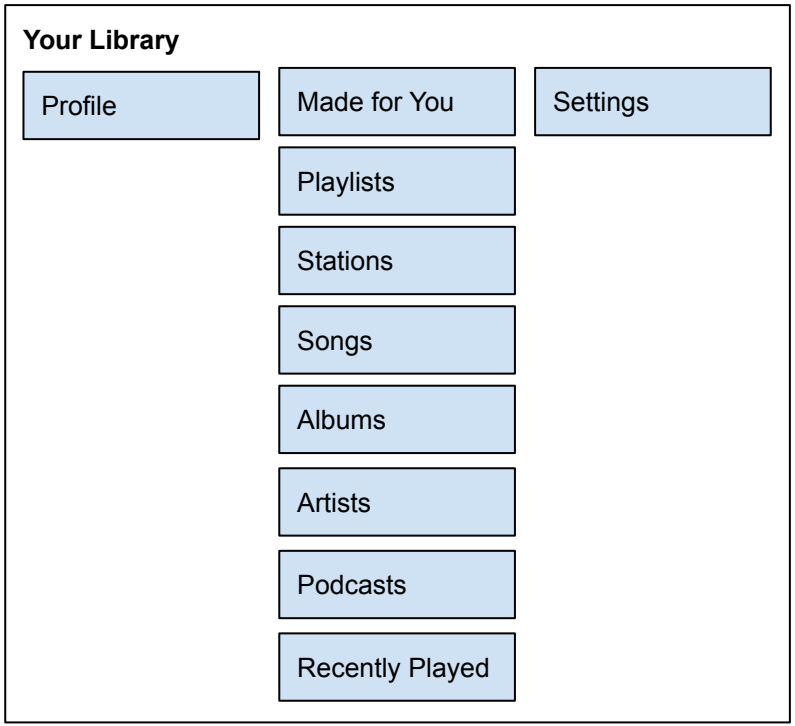
Search

- Search bar
- Your top genres
- Browse all



Spotify: Library



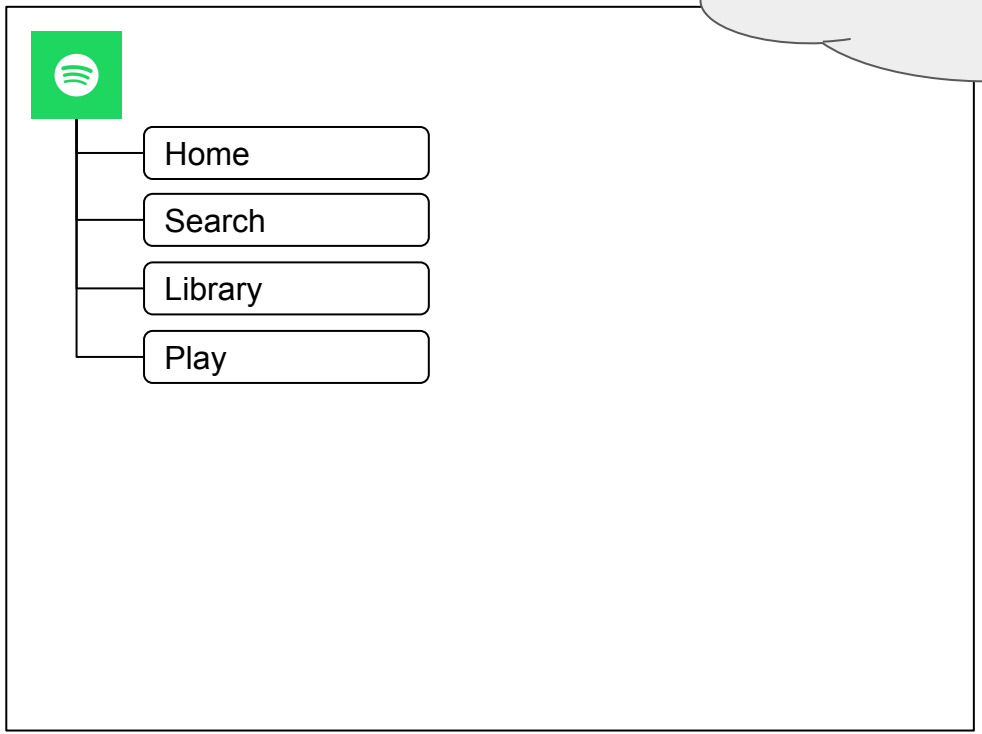


Spotify: Play

Player

Back to Home	Playlist	Share to socials
	Cover	
Add to my library	Title	Extra actions
	Artist	
	Player controls	
Manage devices	Behind the lyrics	

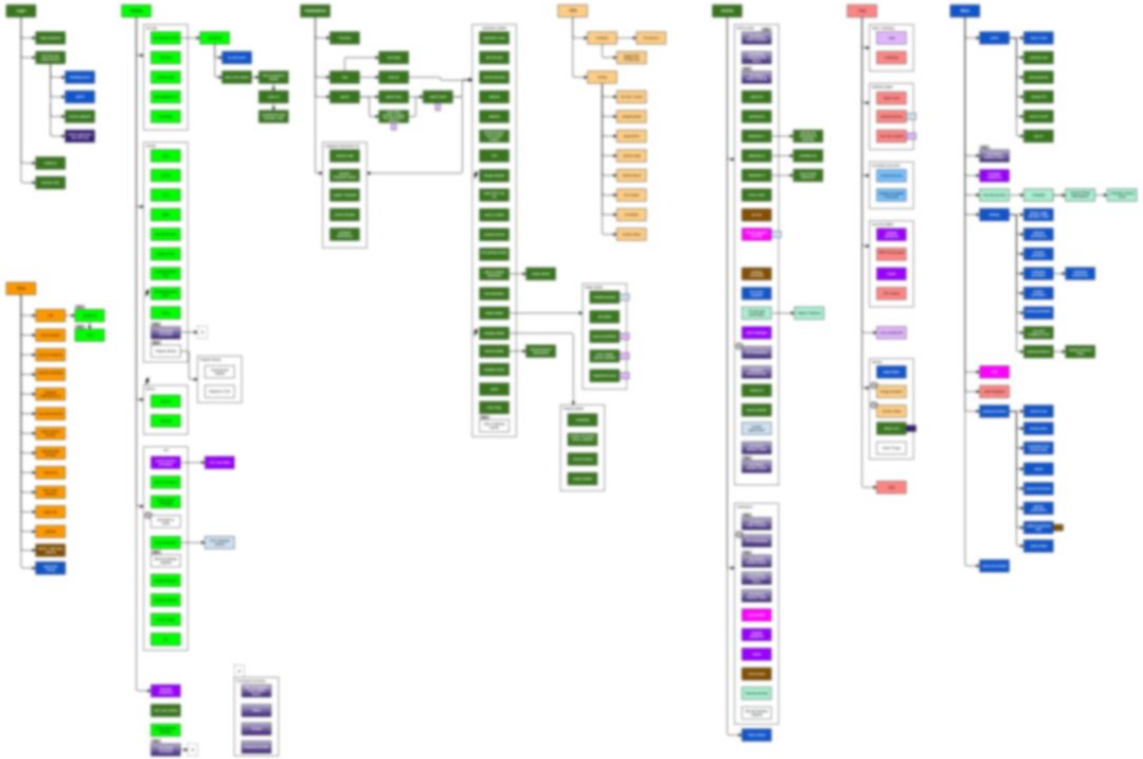
Heatmap



These are the existing tests.

We trust they are well executed. 😇

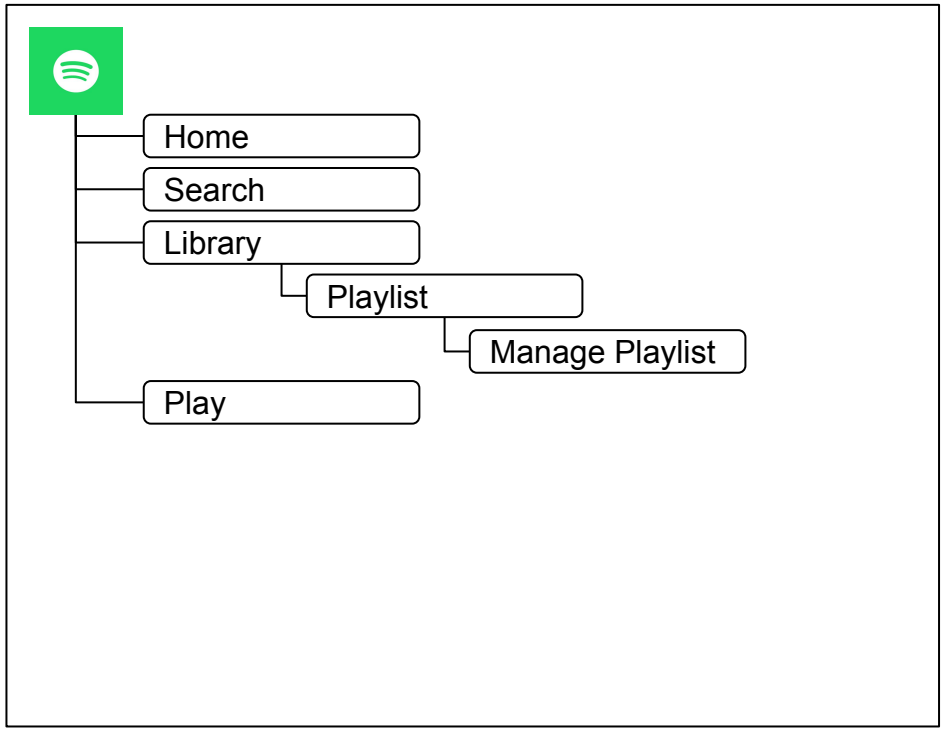
Example of advanced Heatmap



Agenda

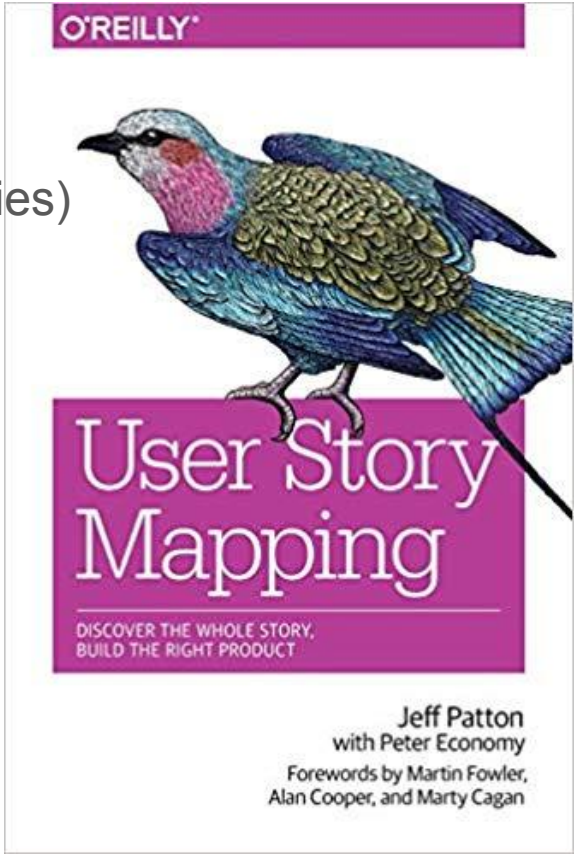
- Introduction
- Sprint 1: The Heat Map
- **Sprint 2: The spine of a Story Map**
  - **Break**
- Sprint 3: Story Map and slices into key journies
- Sprint 4: Example Mapping: uncovering the details
  - **Break**
- Sprint 5: Priorities: value and complexity, slices and stories
- Sprint 6: Validation: testing with low and high resolution
- Closing

Back to the Heatmap



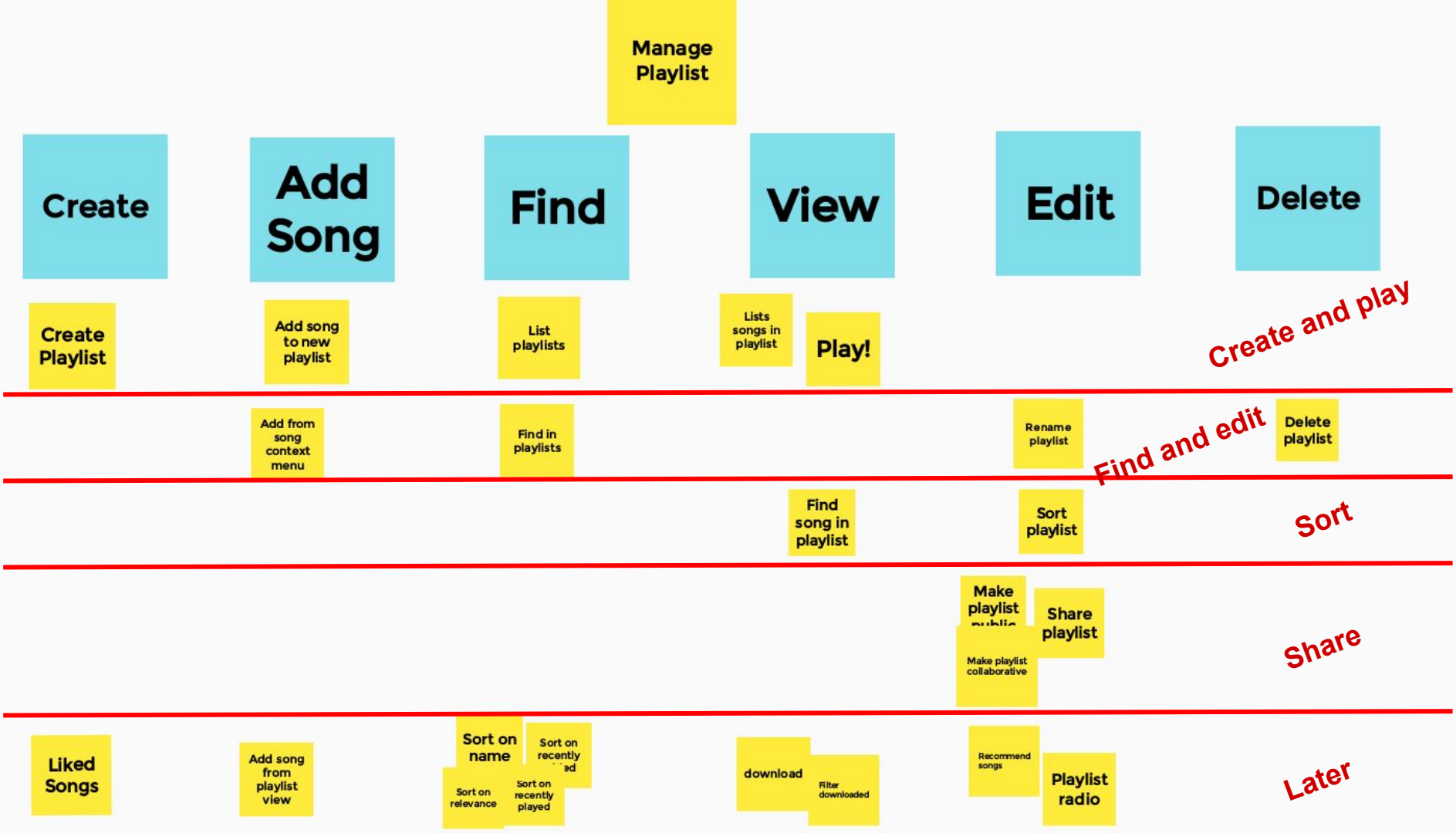
What is a Story Map?

- Follow the flow through the application (user activities)
- Specific User Stories in that flow (detailed actions)
- Slices (things that go well together)
- Priority (what do we do/test first?)
- Difference with building new functionality:
  - Priorities
  - Testing effort instead of implementation effort
  - Dependencies
  - Granularity
  - We know more!

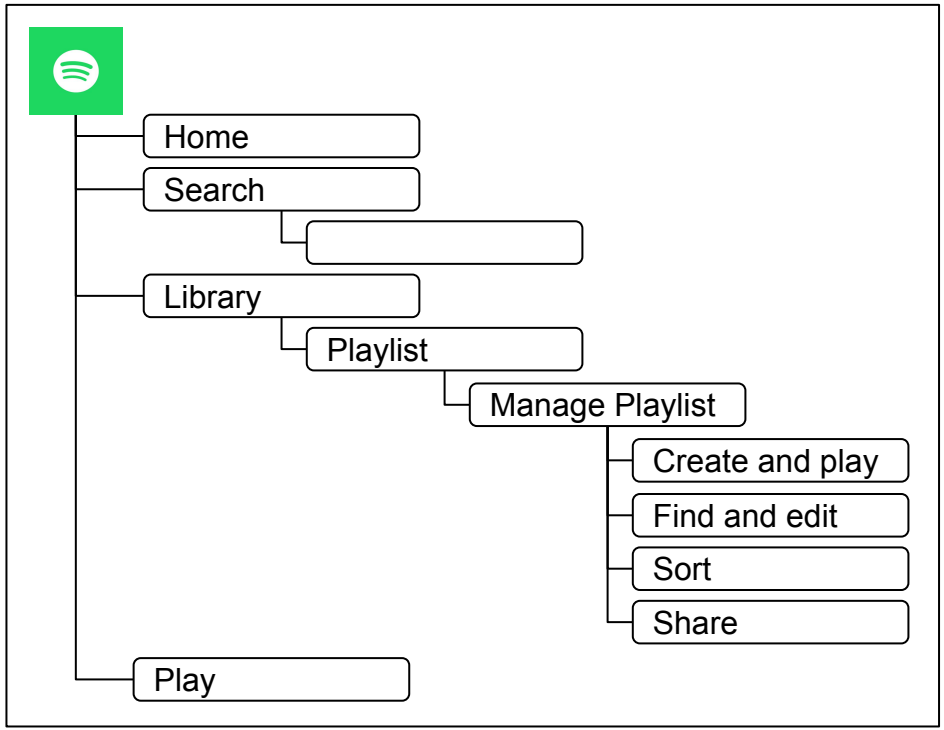


User Story Mapping by Jeff Patton

Elements of a Story Map


Manage Playlist
Create
Add Song
Find
View
Edit
Delete
Create Playlist
Add song to new playlist
List playlists
Lists songs in playlist
Play!
Create and play
Add from song context menu
Find in playlists
Rename playlist
Delete playlist
Find and edit
Find song in playlist
Sort playlist
Sort
Make playlist
Share playlist
Make playlist collaborative
Share
Liked Songs
Add song from playlist view
Sort on name
Sort on recently
Sort on relevance
Sort on recently played
download
Filter downloaded
Recommend songs
Playlist radio
Later

Back to the Heatmap



How to make a Story Map

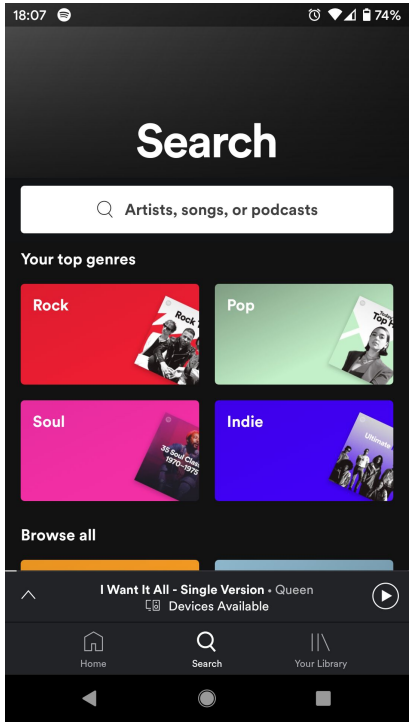
- We start with flow horizontally (orange)
- Each step in the flow can have multiple stories (yellow)
- Use the vertical to create different levels of priority (slices) (pink)
- Make each horizontal ‘slice’ something you can give a name to

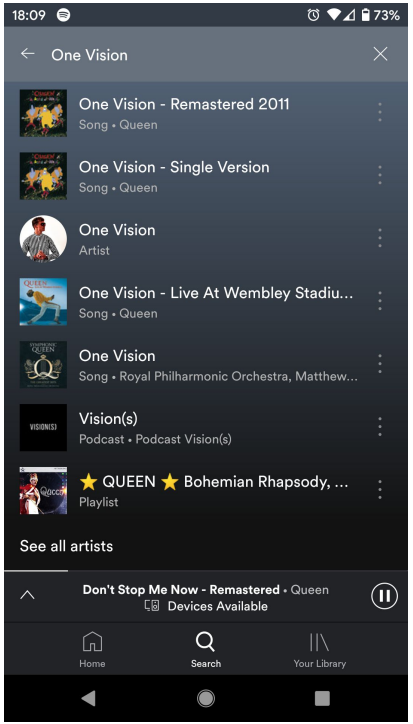
- We do this together (three amigos)

Story Mapping of Spotify Search - Part 1

Now you try!

- Put the Spine cards (orange) in the right order





Using old requirements

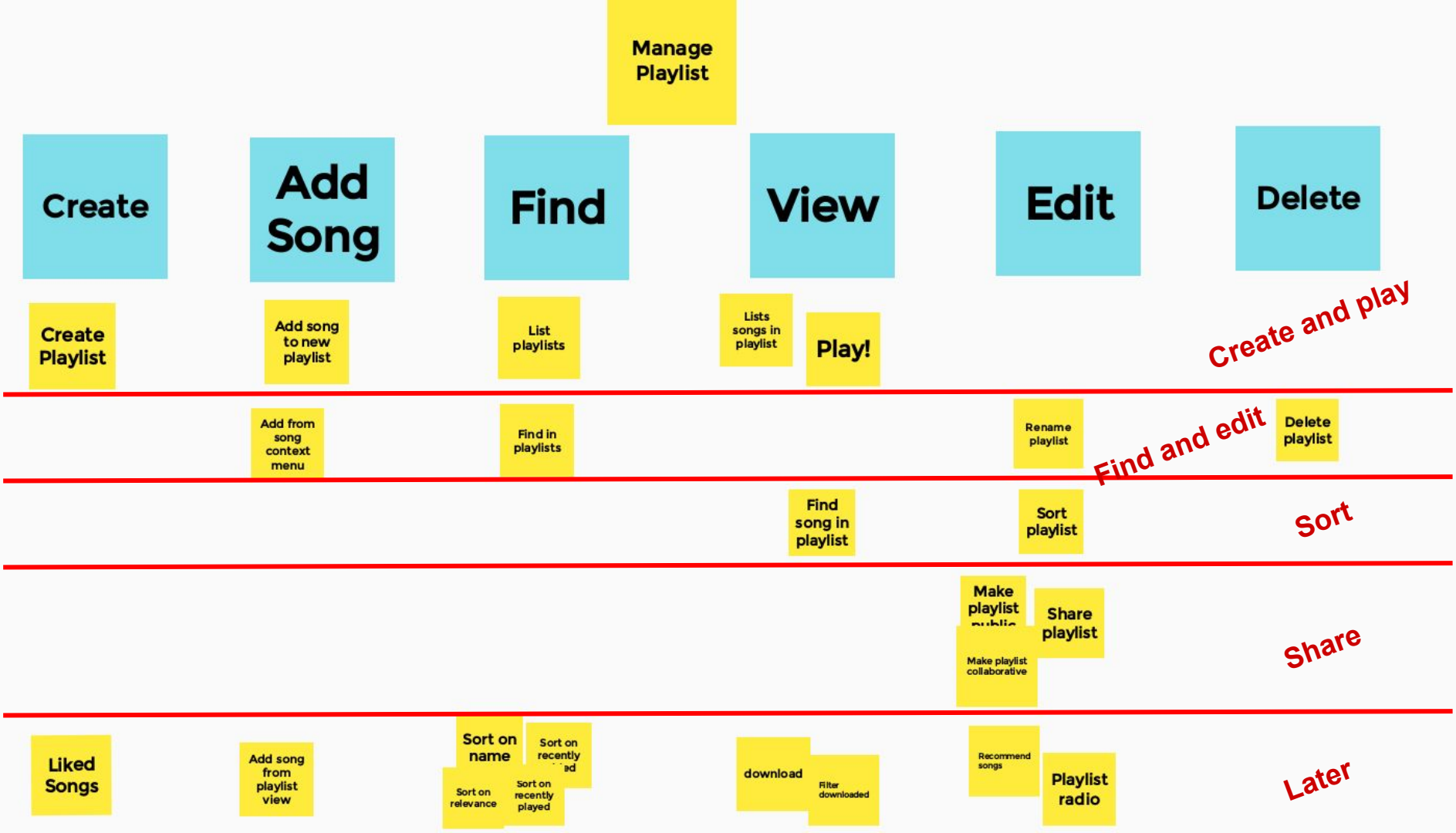
We have some acceptance criteria from when the system was originally built.

Can you cluster them?

How do they relate to the Story Map’s spine?

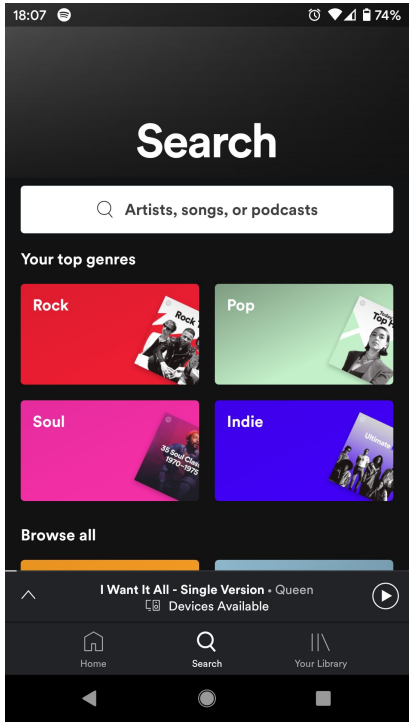
Agenda

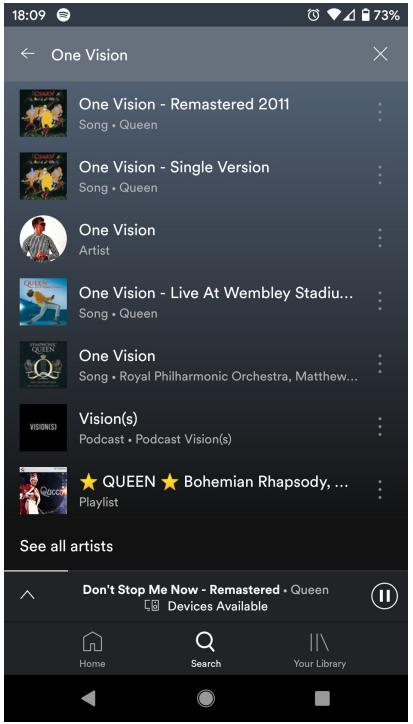
- Introduction
- Sprint 1: The Heat Map
- Sprint 2: The spine of a Story Map
  - **Break**
- **Sprint 3: Story Map and slices into key journies**
- Sprint 4: Example Mapping: uncovering the details
  - **Break**
- Sprint 5: Priorities: value and complexity, slices and stories
- Sprint 6: Validation: testing with low and high resolution
- Closing


Manage Playlist
Create
Add Song
Find
View
Edit
Delete
Create Playlist
Add song to new playlist
List playlists
Lists songs in playlist
Play!
Create and play
Add from song context menu
Find in playlists
Rename playlist
Delete playlist
Find and edit
Find song in playlist
Sort playlist
Sort
Make playlist
Share playlist
Make playlist collaborative
Share
Liked Songs
Add song from playlist view
Sort on name
Sort on recently
Sort on relevance
Sort on recently played
download
Filter downloaded
Recommend songs
Playlist radio
Later

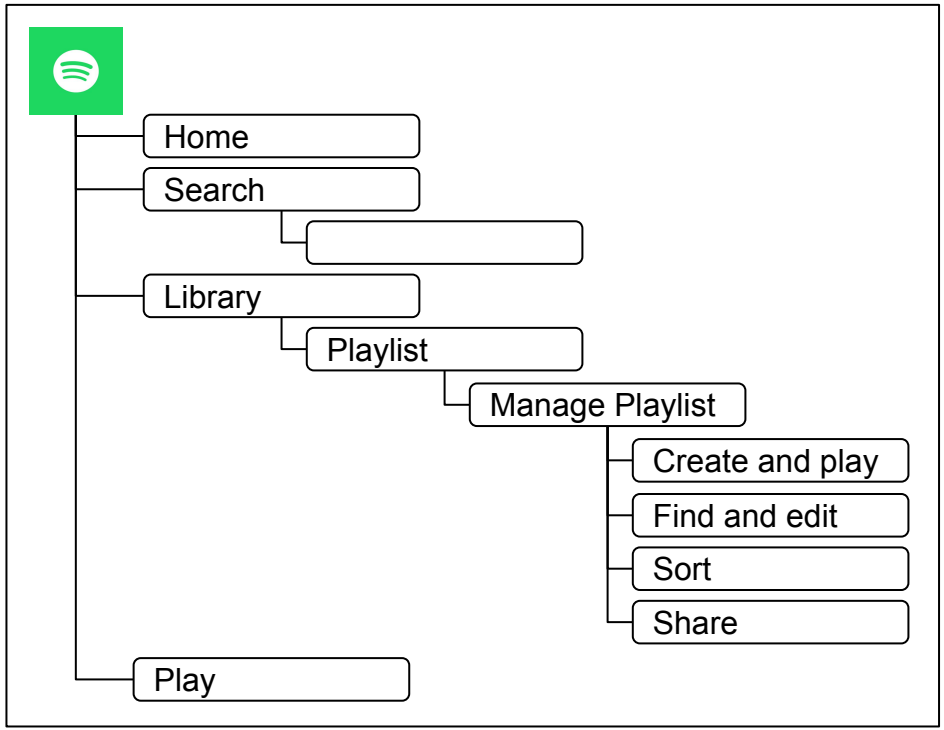
Story Mapping of Spotify Search - Part 2

- Use the Story Cards (yellow) and create your Story Map
- Find paths through the spine using the Stories, creating horizontal grouping
- Give each group a name describing the key flow it represents

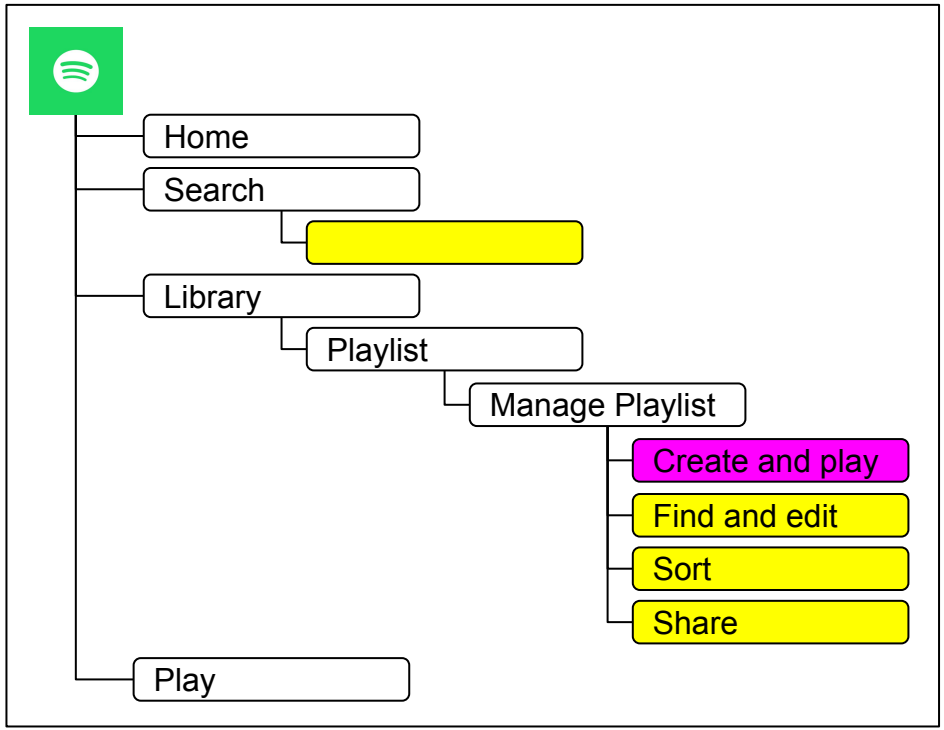


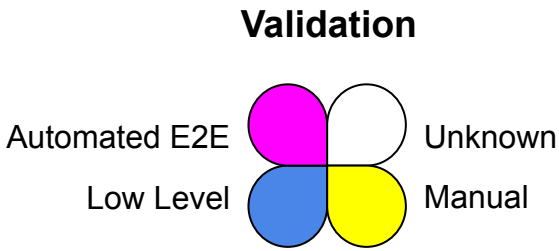


Back to the Heatmap



Back to the Heatmap



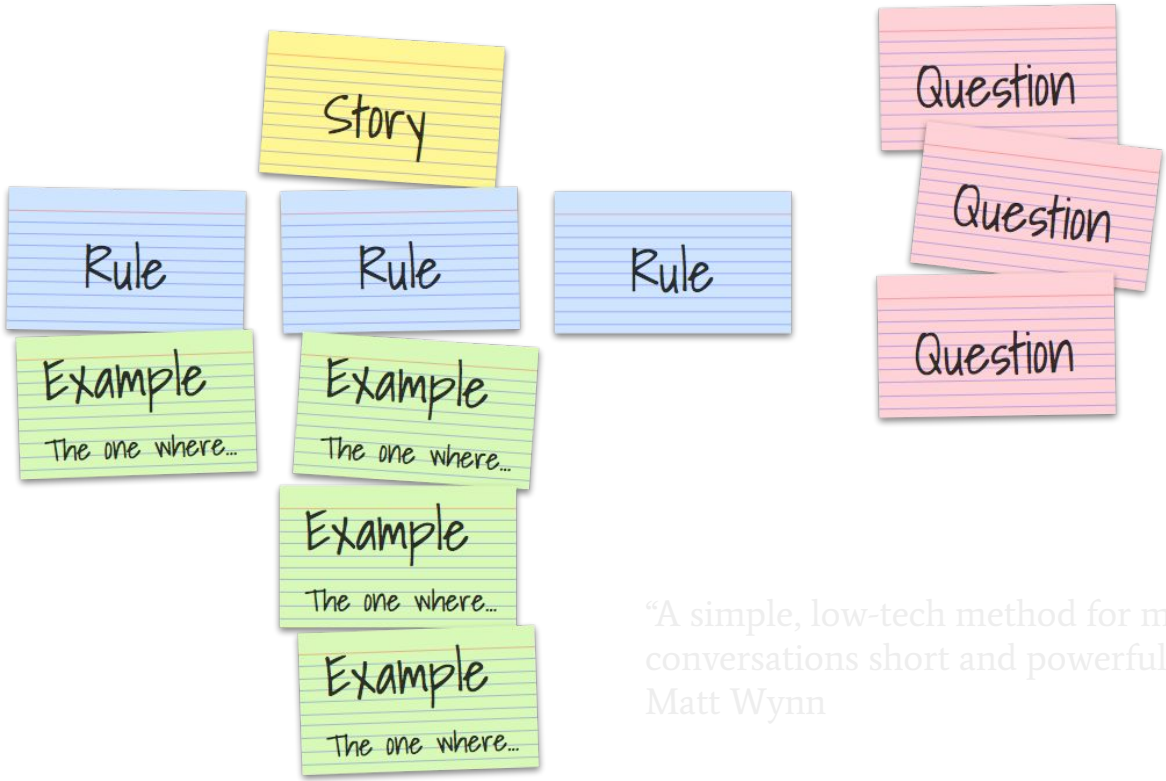


- Put each named Slice on the Heat Map

Agenda

- Introduction
- Sprint 1: The Heat Map
- Sprint 2: The spine of a Story Map
  - **Break**
- Sprint 3: Story Map and slices into key journies
- **Sprint 4: Example Mapping: uncovering the details**
  - **Break**
- Sprint 5: Priorities: value and complexity, slices and stories
- Sprint 6: Validation: testing with low and high resolution
- Closing

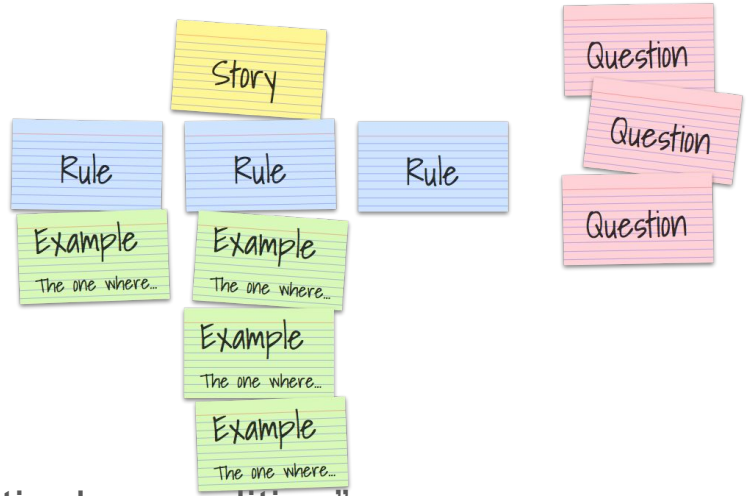
Example Mapping



"A simple, low-tech method for making conversations short and powerfully productive" -- Matt Wynn

https://cucumber.io/blog/bdd/example-mapping-introduction/

Example Mapping



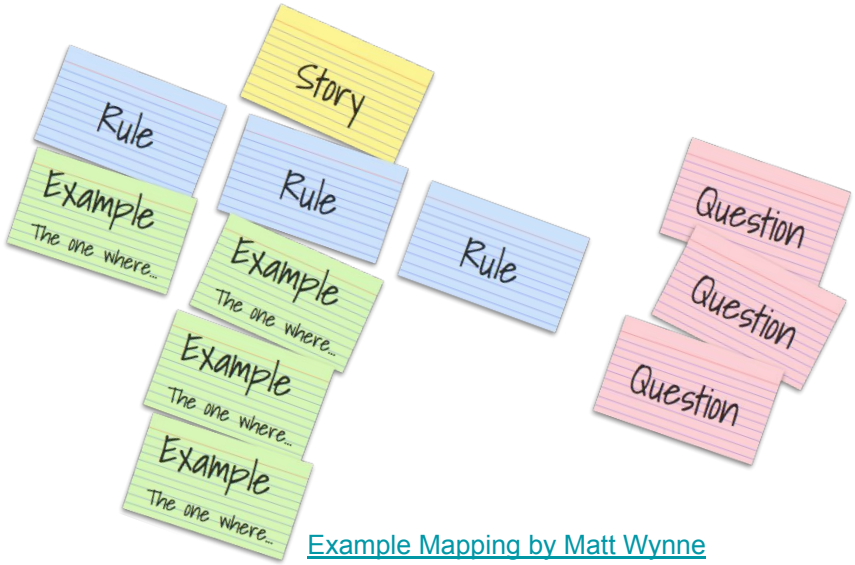
- Examples on GREEN cards
  “Illustrate concrete behaviour of the system”
- Rules on BLUE cards
  “Logical groupings of examples around a particular condition”
  Often: acceptance criteria, business rules, requirements
- Questions or Assumptions on RED cards
  “Topics that would block the discussion”
- User Stories on YELLOW cards
  “We often split off separate stories to discuss at a later stage”

https://cucumber.io/blog/bdd/example-mapping-introduction/

Validation: Example Mapping

For a story:

- List the business rules
- Provide examples that illustrate the rule
- Capture open questions



Example Mapping by Matt Wynne

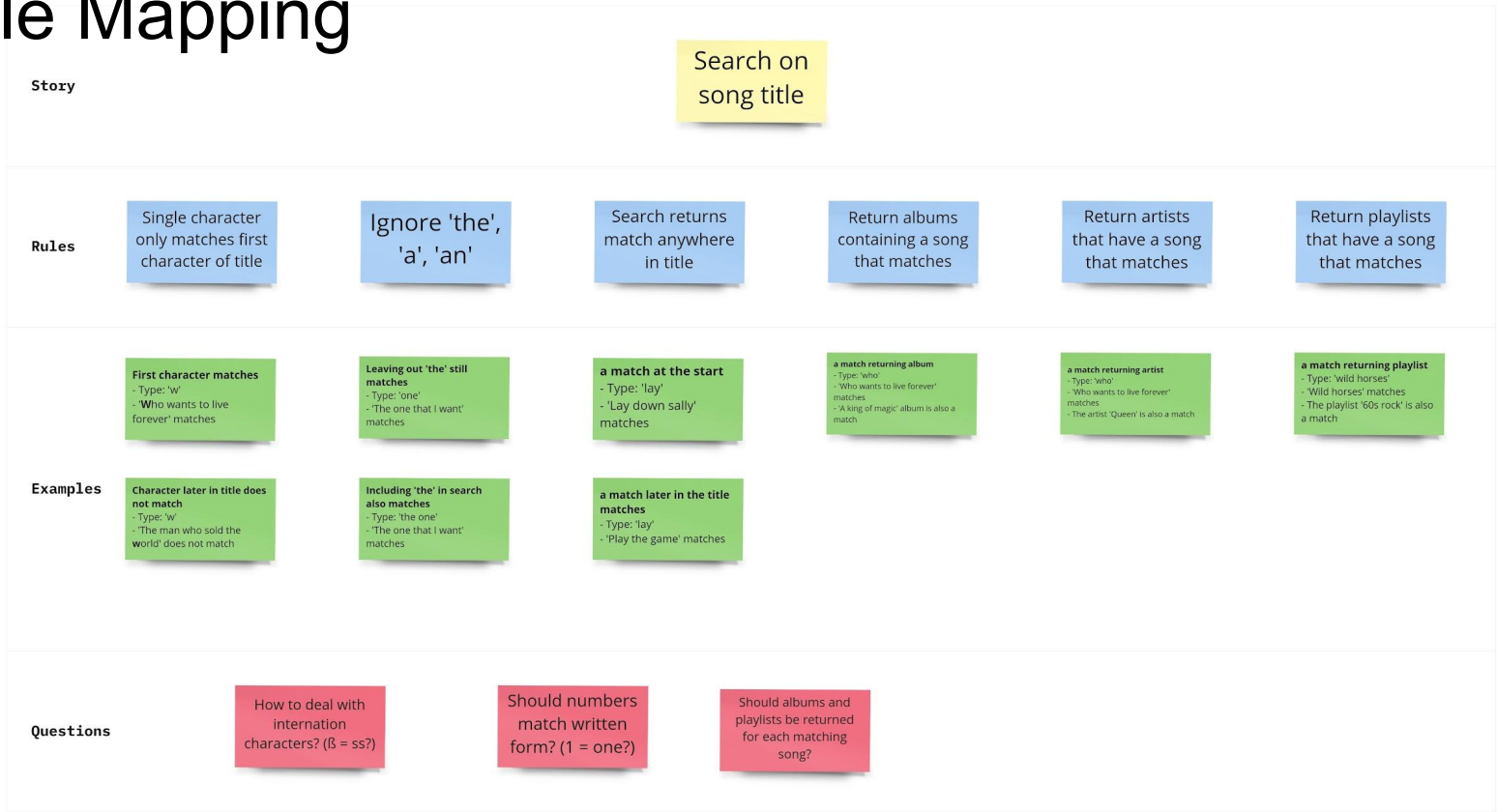
Centered around business rules

Examples can be formalised (in Gherkin, or otherwise) in the form of a test:

Input -> Action -> Outcome (Given/When/Then, Arrange/Act/Assert)

Examples of Examples

Example Mapping



Example Mapping

Now you try!

- Pick one story from your Story Map
- Write as many of the Rules as you can think of

What’s a good example?

- Concrete
- All the necessary information
- Illustrates the rule
- Agreed

Example Mapping

Now you try!

- For two rules - Write examples

Agenda

- Introduction
- Sprint 1: The Heat Map
- Sprint 2: The spine of a Story Map
  - **Break**
- Sprint 3: Story Map and slices into key journies
- Sprint 4: Example Mapping: uncovering the details
  - **Break**
- **Sprint 5: Priorities: value and complexity, slices and stories**
- Sprint 6: Validation: testing with low and high resolution
- Closing

Choose or lose: Priorities

Recent: new features, new areas of code are more vulnerable

Core: essential functions must continue to work

Risk: some areas of an application pose more risk

Configuration sensitive: code dependent on settings can be vulnerable

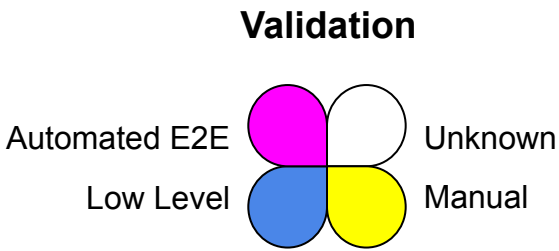
Repaired: bug fixes can introduce new issues

Chronic: some areas in an application may be perpetually sensitive to breaking

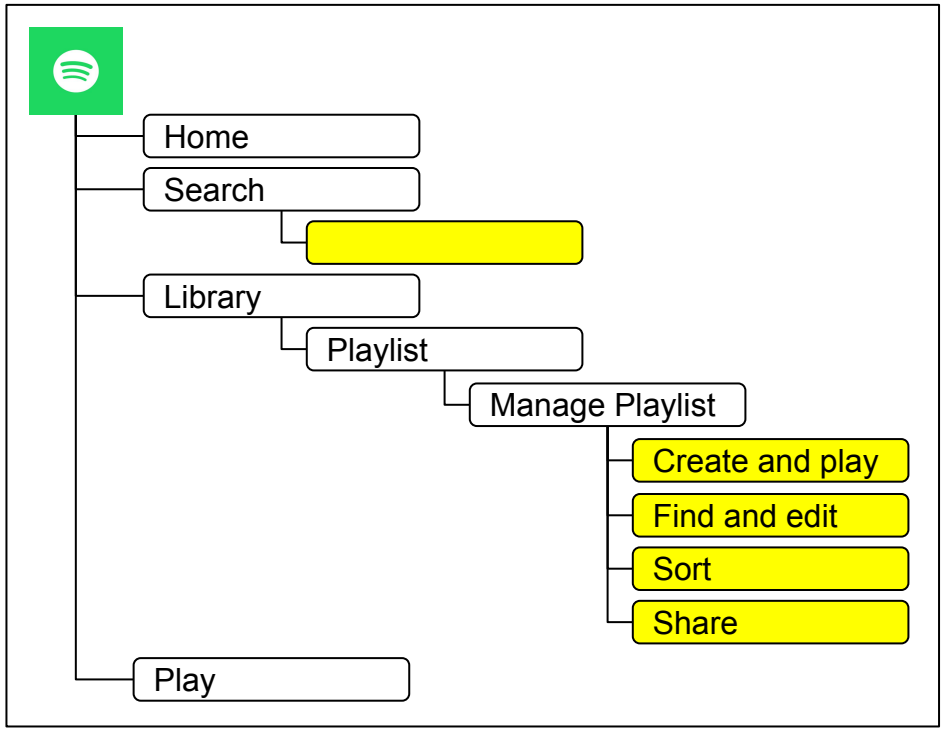
RCRCRC by Karen Johnson, via Matt Heusser

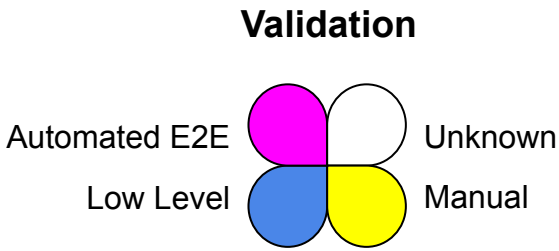
Using the outcomes of Story Map

- Each slice can be named as a separate box in the functional overview
- Each separate box can be an end-to-end test
- Actually creating those tests turns the heatmap yellow
- Stories generated can be starting points for more detailed exploration



Back to the Heatmap





The devil is in the stories

Stories hold the details

Which of the stories on your story map do you think are the most complex?

Which of the stories are more important than others in the story map? Why?

How difficult will these stories be to test comprehensively? If it’s done manually? Automated end-to-end? Unit tested?

Agenda

- Introduction
- Sprint 1: The Heat Map
- Sprint 2: The spine of a Story Map
  - **Break**
- Sprint 3: Story Map and slices into key journies
- Sprint 4: Example Mapping: uncovering the details
  - **Break**
- Sprint 5: Priorities: value and complexity, slices and stories
- **Sprint 6: Validation: testing with low and high resolution**
- Closing

Validation

Each slice from the Story Map can be a test

Search for a Song

<assumes app opened, user logged in>

Go to Search
Search for (part of) a Song’s name
Tap the Song to start it playing

Validation

Each slice from the Story Map can be a test

Recent Searches
Go to Search
Search for any term
Go back to Home
Go back to Search
See term appear on top of Recent Searches
Delete term from Recent Searches
Check by restarting Search
Delete all items from Recent Searches
Check by restarting Search

Automate?

Wait.

We’re finding out what functionality there is to be tested

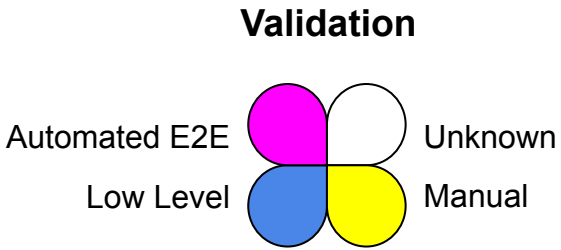
Let **humans** test, don’t restrict them more than necessary

Once you’ve discovered what is interesting about the functionality, ***maybe***, automate

Or, **better**, go into detail with an example map to get more high resolution tests

Validation

- High-level, low resolution, ***human*** end-to-end testing
- High-level, low resolution, automated end-to-end testing
- Low-level, high resolution, automated **example based** tests



Examples can be formalised into Scenarios

```
Scenario: Playlist without name is auto-numbered
Given existing Playlists
| My Playlist #1 |
| My Playlist #2 |
When a new playlist without name is added
Then that new playlist is given the name ‘My Playlist #3’

Scenario: Playlist without name gets next available slot
Given existing Playlists
| My Playlist #1 |
| My Playlist #3 |
When a new playlist without name is added
Then that new playlist is given the name ‘My Playlist #2’
```

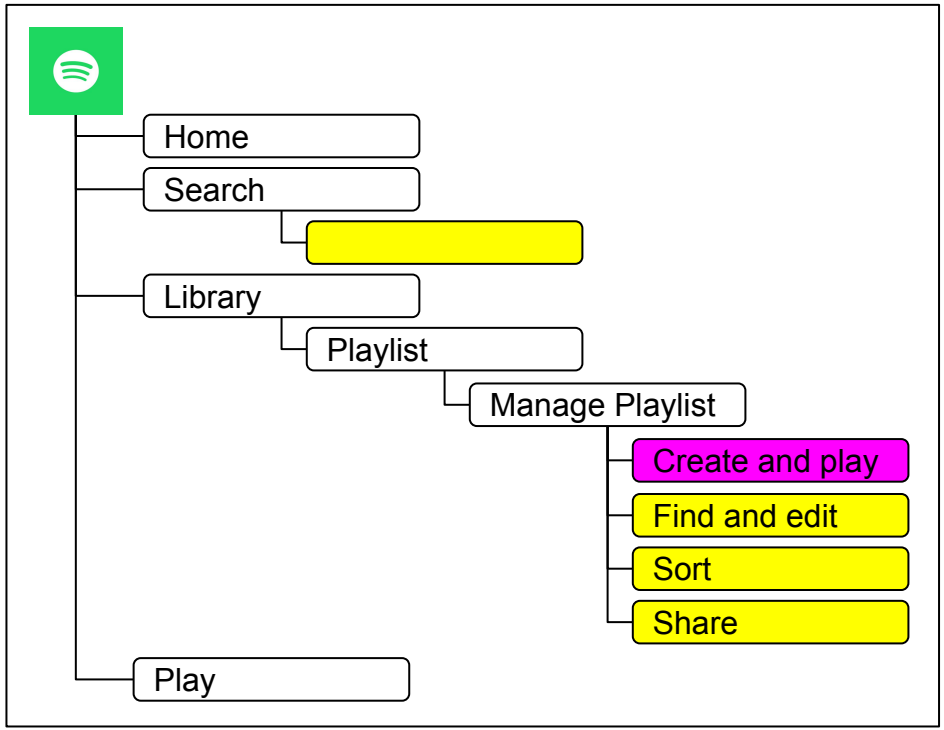
'My Playlist #1' and 'My Playlist #2' exist, so new one is 'My Playlist #3'

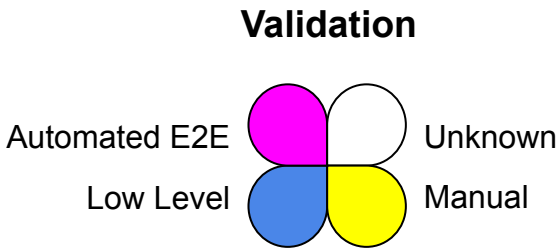
'My Playlist #1' and 'My Playlist #3' exist, so new one is 'My Playlist #2'

Scenarios are implemented at same level the functionality lives

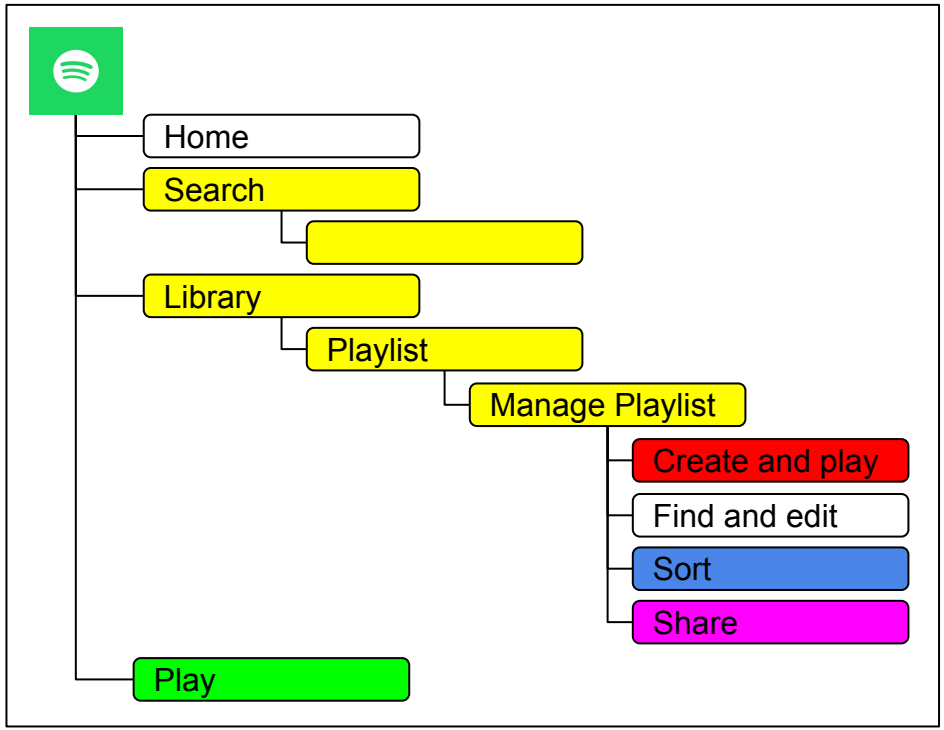

```
Scenario: Playlist without name is auto-numbered
Given existing Playlists
| My Playlist #1 |
| My Playlist #2 |
When a new playlist without name is added
Then that new playlist is given the name 'My Playlist #3'

function testPlaylistNameAutoNumber() {
    playlists.addAll('My Playlist #1', 'My Playlists #2')
    playlists.add('')
    expect(playlists.getAll()).toContain('My Playlist #3')
}
```

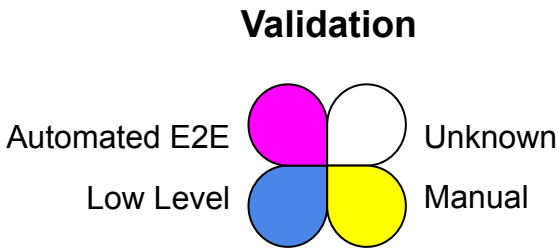
Back to the Heatmap



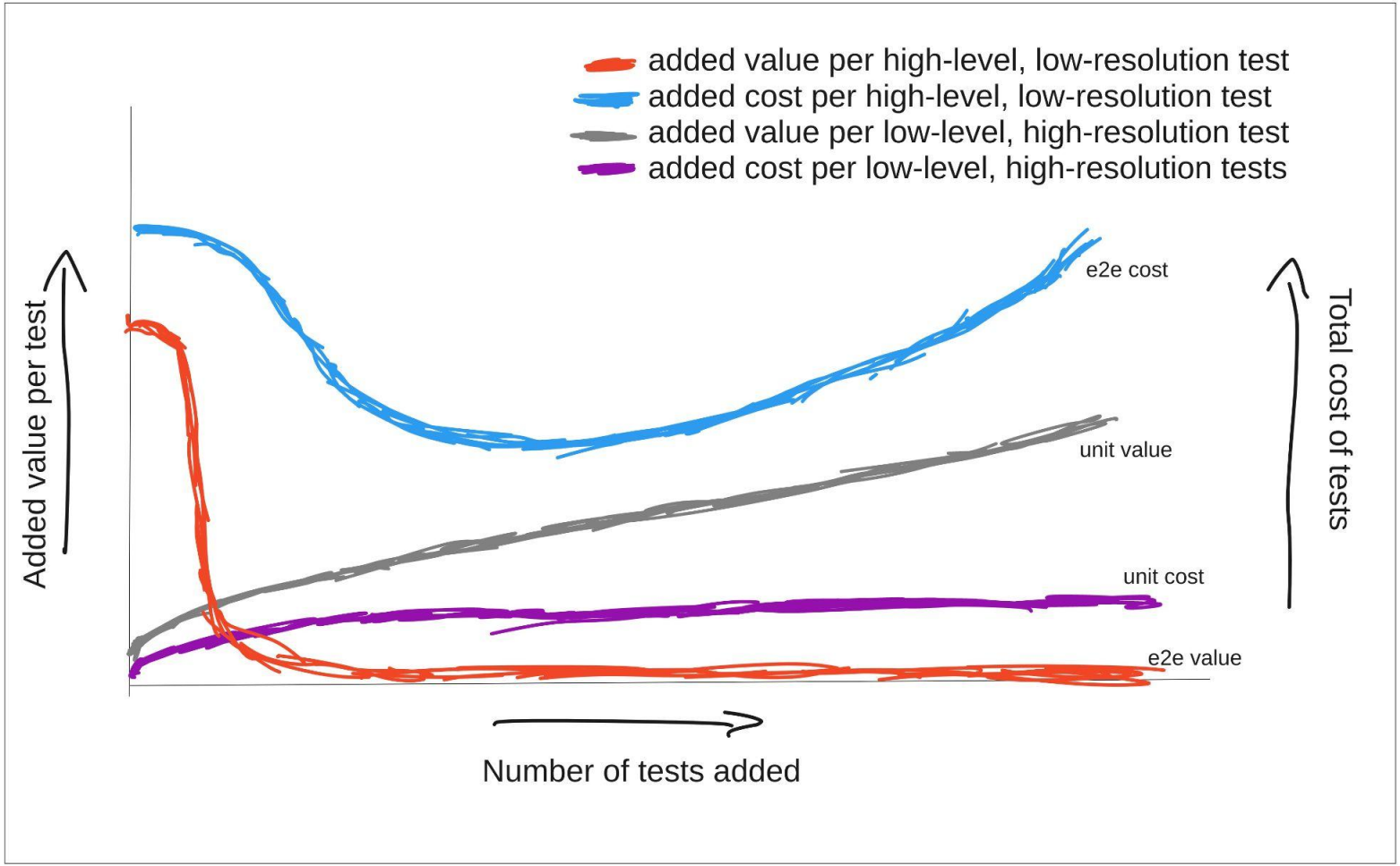


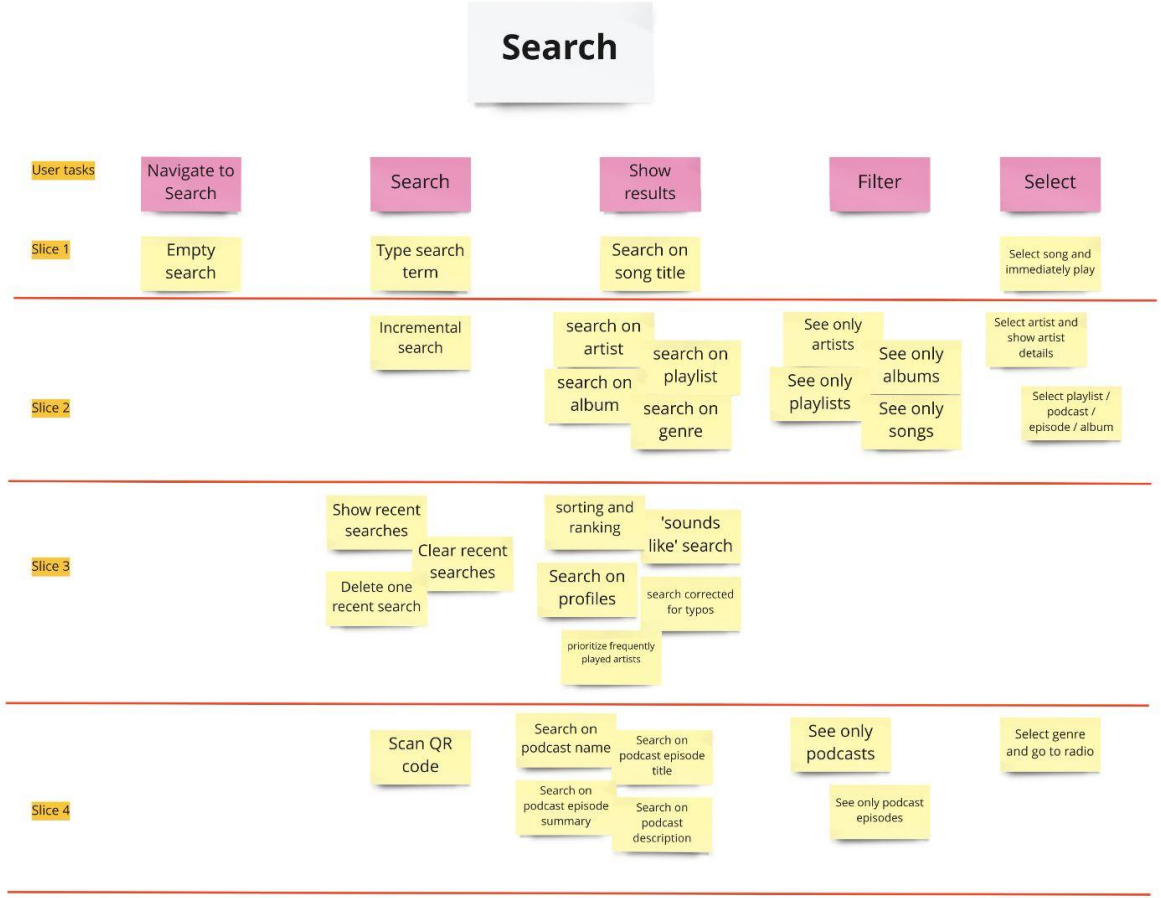
Future?

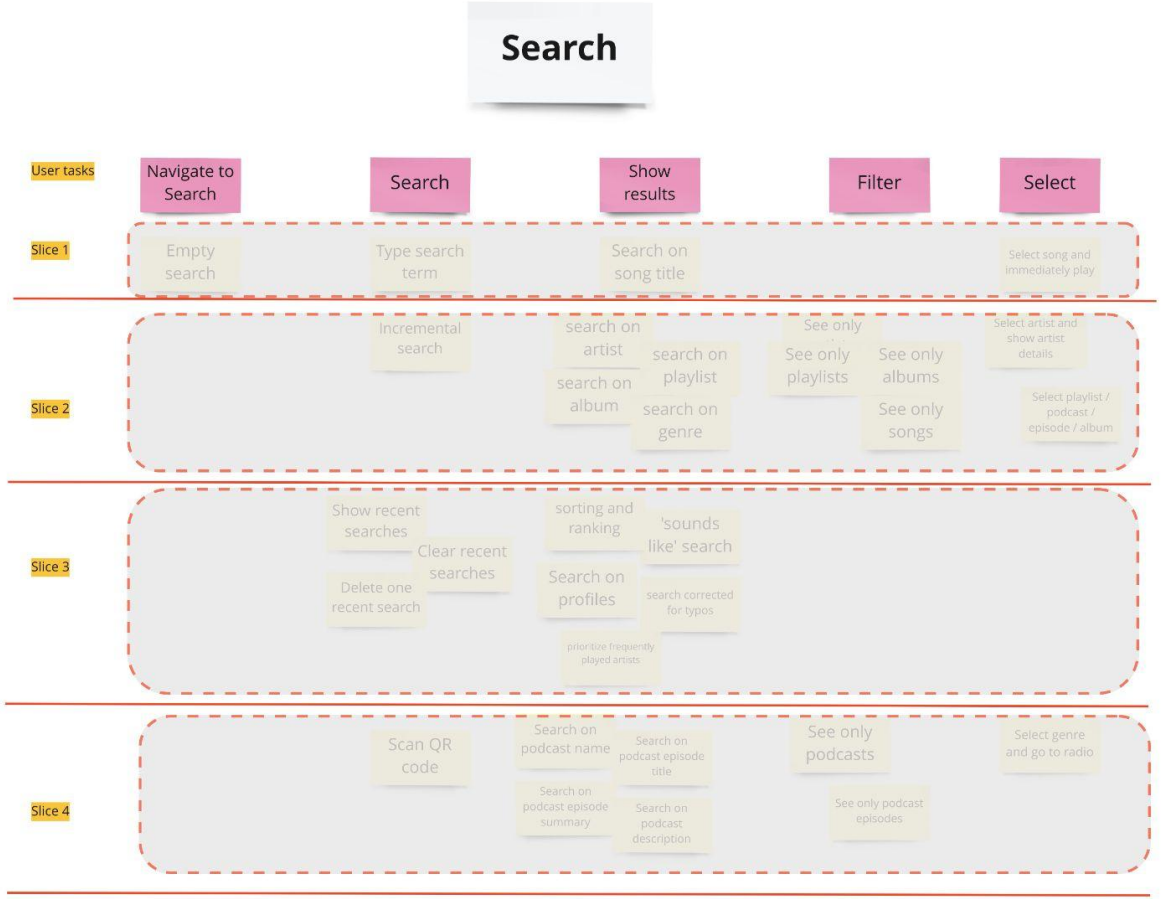


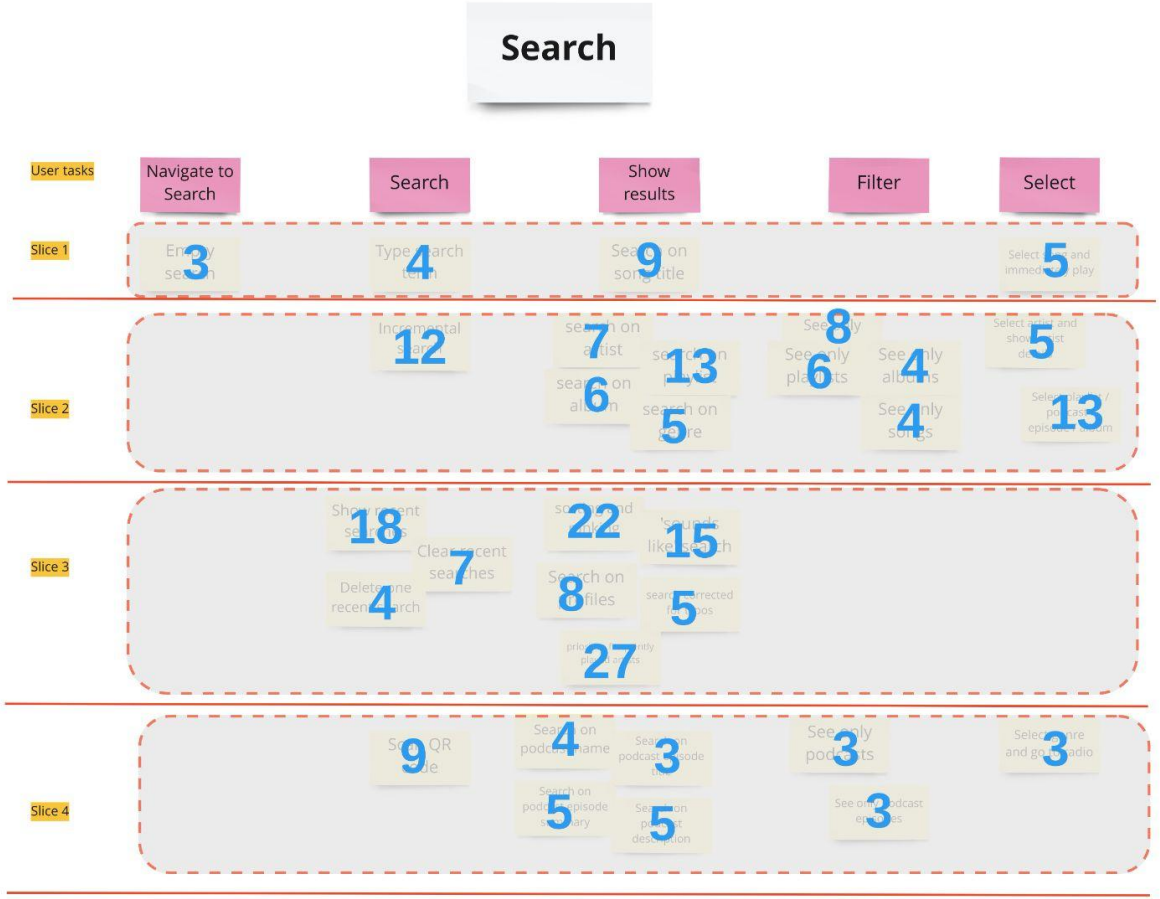


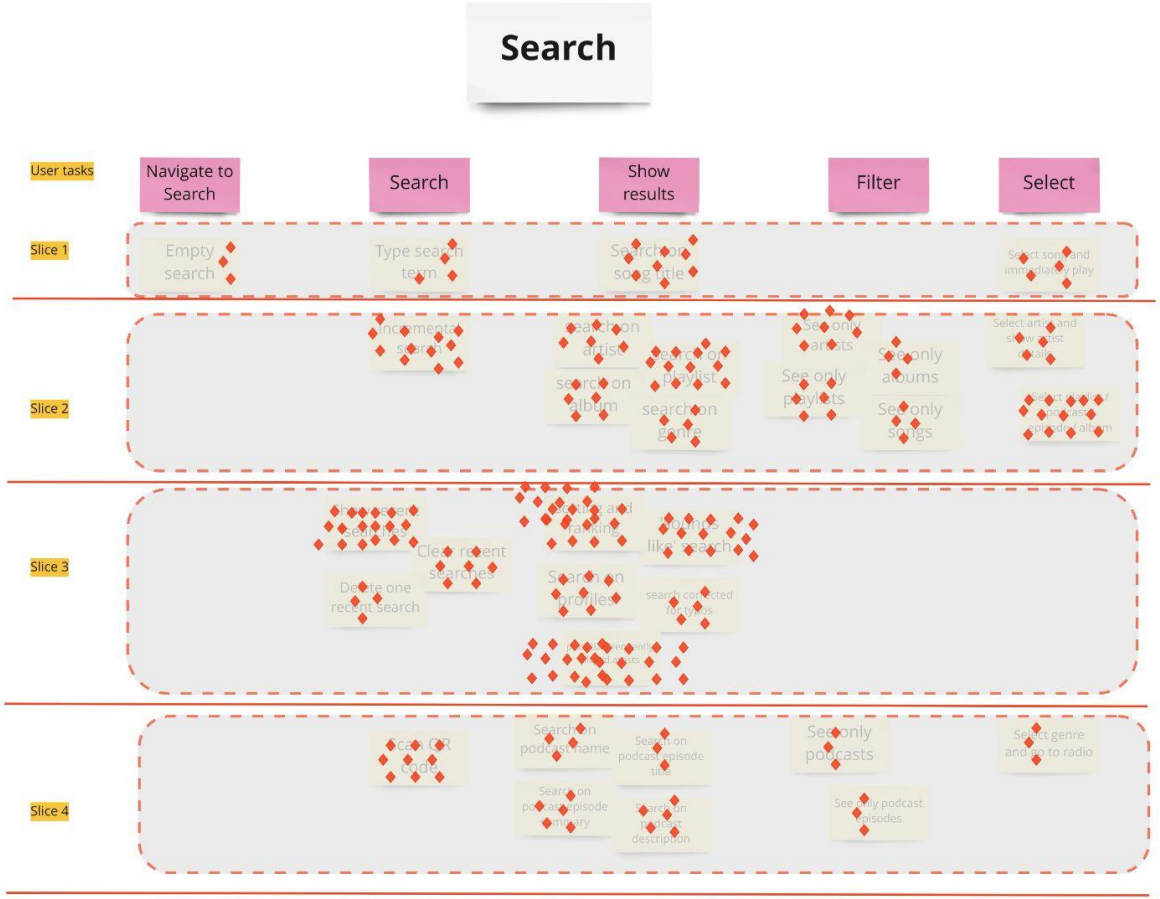
Test Result	
	Failed
	Attention
	😄

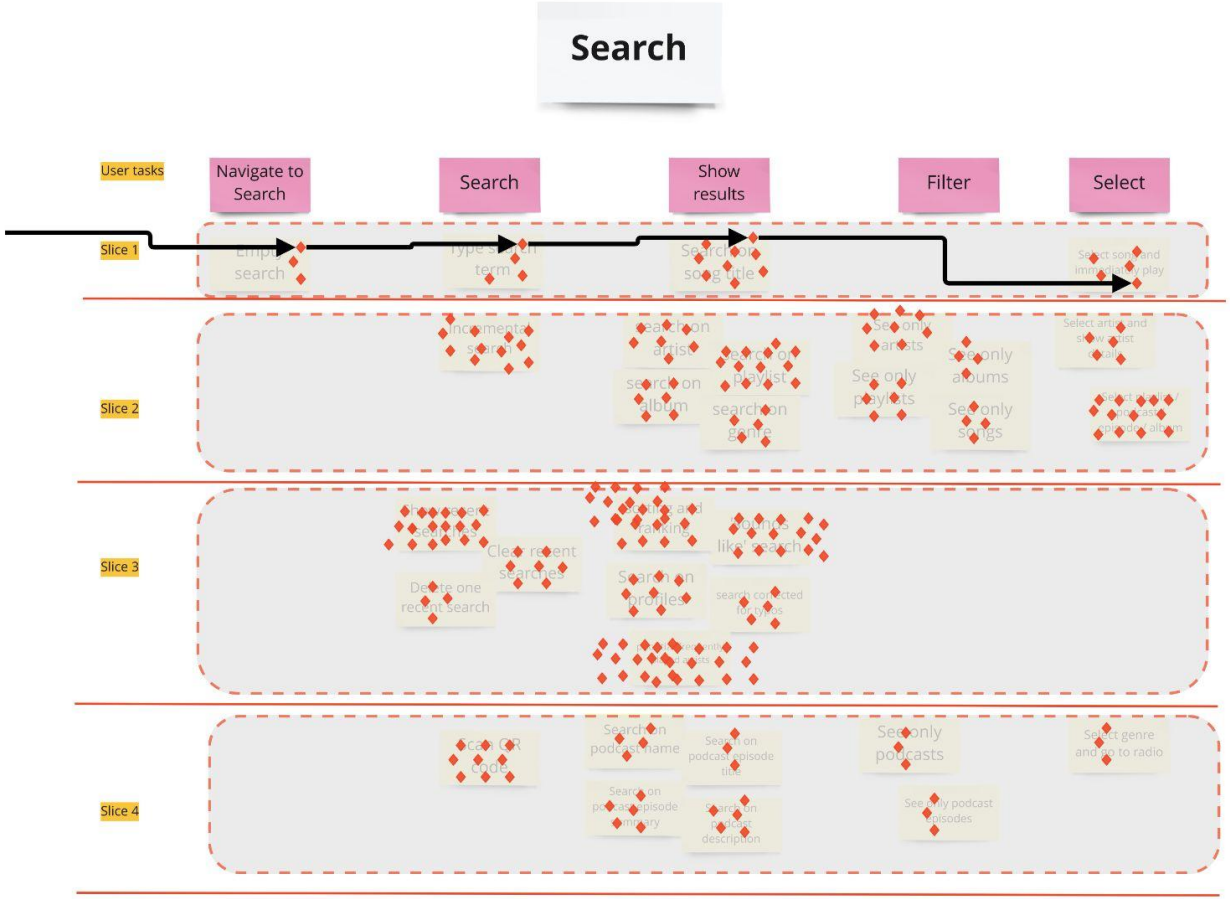

added value per high-level, low-resolution test
added cost per high-level, low-resolution test
added value per low-level, high-resolution test
added cost per low-level, high-resolution tests
e2e cost
unit value
unit cost
e2e value
Added value per test
Total cost of tests
Number of tests added

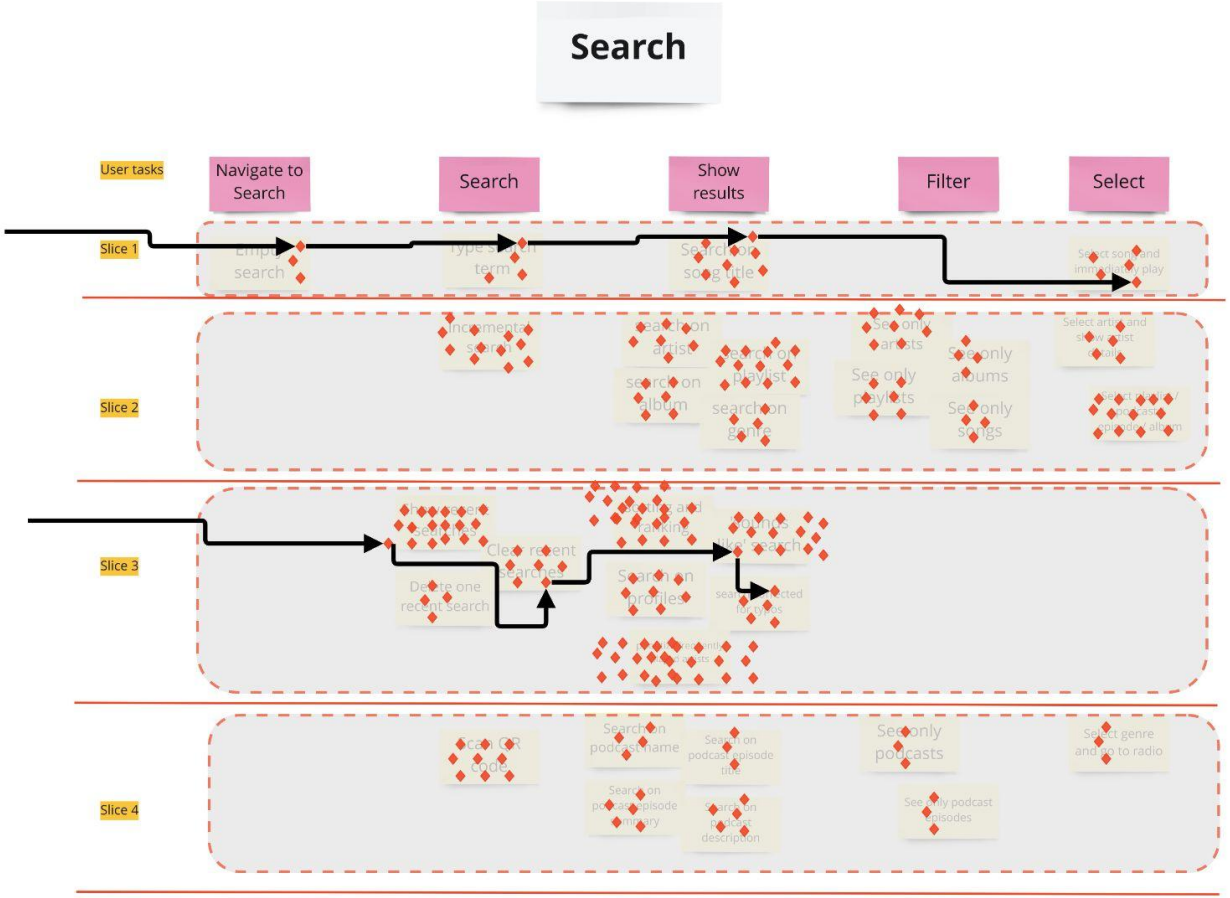

Search
User tasks
Navigate to Search
Search
Show results
Filter
Select
Slice 1
Empty search
Type search term
Search on song title
Select song and immediately play
Slice 2
Incremental search
search on artist
search on playlist
search on album
search on genre
See only artists
See only albums
See only playlists
See only songs
Select artist and show artist details
Select playlist / podcast / episode / album
Slice 3
Show recent searches
Clear recent searches
Delete one recent search
sorting and ranking
'sounds like' search
Search on profiles
search corrected for typos
prioritize frequently played artists
Slice 4
Scan QR code
Search on podcast name
Search on podcast episode title
Search on podcast episode summary
Search on podcast description
See only podcasts
See only podcast episodes
Select genre and go to radio


Search
User tasks
Navigate to Search
Search
Show results
Filter
Select
Slice 1
Empty search
Type search term
Search on song title
Select song and immediately play
Slice 2
Incremental search
search on artist
search on playlist
search on album
search on genre
See only playlists
See only albums
See only songs
Select artist and show artist details
Select playlist / podcast / episode / album
Slice 3
Show recent searches
Clear recent searches
Delete one recent search
sorting and ranking
'sounds like' search
Search on profiles
search corrected for typos
prioritize frequently played artists
Slice 4
Scan QR code
Search on podcast name
Search on podcast episode title
Search on podcast episode summary
Search on podcast description
See only podcasts
See only podcast episodes
Select genre and go to radio


Search
User tasks
Navigate to Search
Search
Show results
Filter
Select
Slice 1
Slice 2
Slice 3
Slice 4
3
4
9
5
12
7
13
6
5
8
6
4
4
5
13
18
7
22
15
4
8
5
27
9
4
3
5
3
3
5


Search
User tasks
Navigate to Search
Search
Show results
Filter
Select
Slice 1
Empty search
Type search term
Search on song title
Select song and immediately play
Slice 2
Incremental search
Search on artist
Search on playlist
Search on album
Search on genre
See only artists
See only playlists
See only albums
See only songs
Select artist and show artist detail
Slice 3
Show recent searches
Clear recent searches
Delete one recent search
Search on profiles
Slice 4
Search on podcast name
Search on podcast episode title
Search on podcast episode category
Search on podcast description
See only podcasts
See only podcast episodes
Select genre and go to radio


Search
User tasks
Navigate to Search
Search
Show results
Filter
Select
Slice 1
Slice 2
Slice 3
Slice 4
Empty search
Type search term
Search on song title
Select song and immediately play
Incremental search
Search on artist
Search on playlist
Search on album
Search on genre
See only artists
See only albums
See only playlists
See only songs
Select artist and show artist page
Select podcast, episode, album
Clear recent searches
Delete one recent search
Search on profiles
search on podcast name
Search on podcast episode title
Search on podcast episode summary
Search on podcast description
See only podcasts
See only podcast episodes
Select genre and go to radio


Search
User tasks
Navigate to Search
Search
Show results
Filter
Select
Slice 1
Slice 2
Slice 3
Slice 4
Incremental search
Search on artist
Search on playlist
Search on album
Search on genre
See only artists
See only playlists
See only albums
See only songs
Clear recent searches
Delete one recent search
Search on profiles
Search on podcast name
Search on podcast episode title
Search on podcast episode library
Search on podcast description
See only podcasts
See only podcast episodes
Select genre and go to radio

Another look at priorities

- Which slices would you automate as end-to-end tests? Why those?
- Which stories would you prioritize for unit/acceptance tests? Why those?

Takeaways: Get back in control

- **What does it do?** (functional breakdown)
- **Is it tested?** (heatmap)
- **Go broad for overview** (Story Map, manual/e2e tests)
- **Dive deep where necessary** (Example Map, detailed tests)
- **Iterate!**

Thank you!

And...

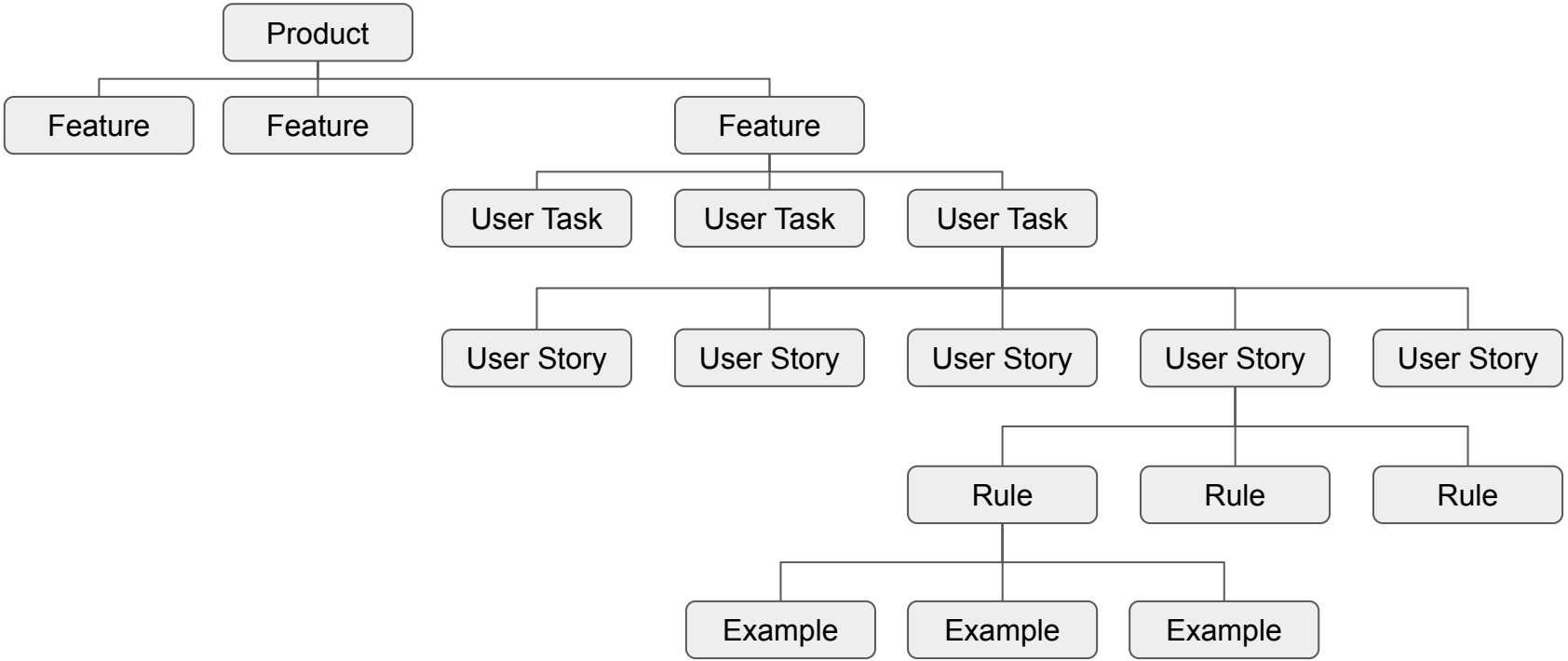
If you have a problem

If no one else can help

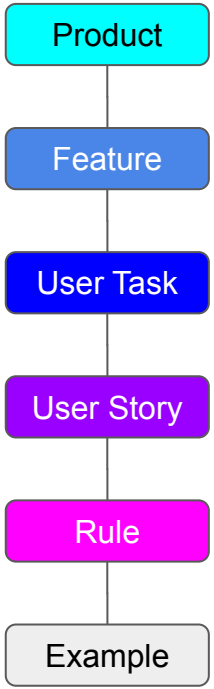
This is how you find us:

wouter@lagerweij.com
karel@hedgefields.com

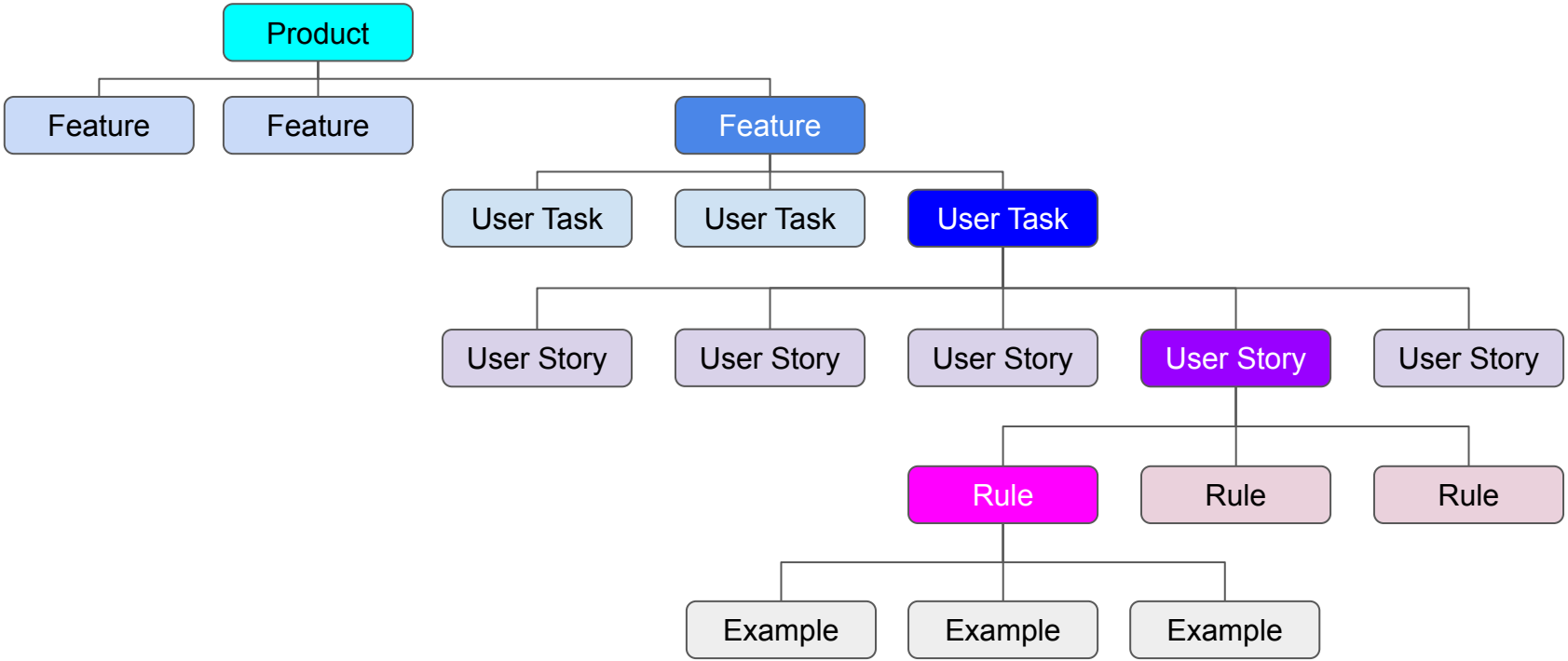
<<deze niet gebruiken, maar dupliceren>>



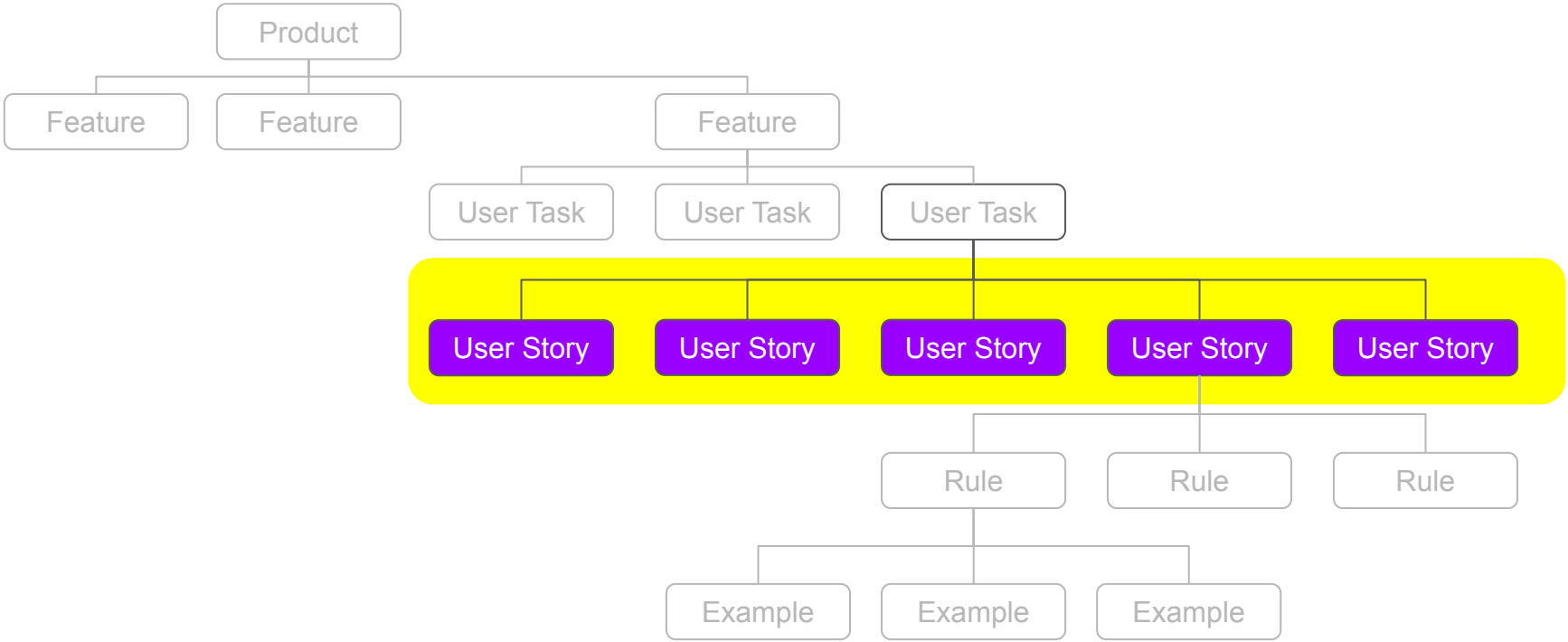
How the elements relate



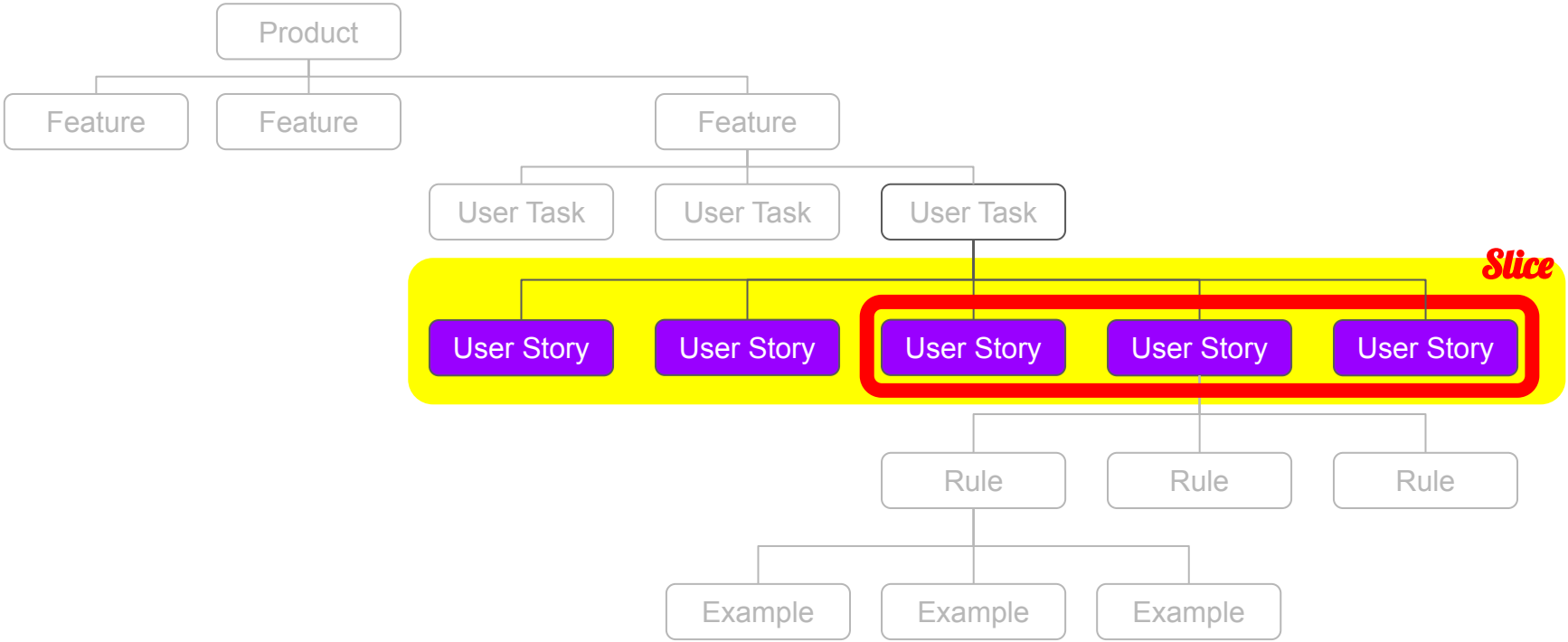
Visual Complexity



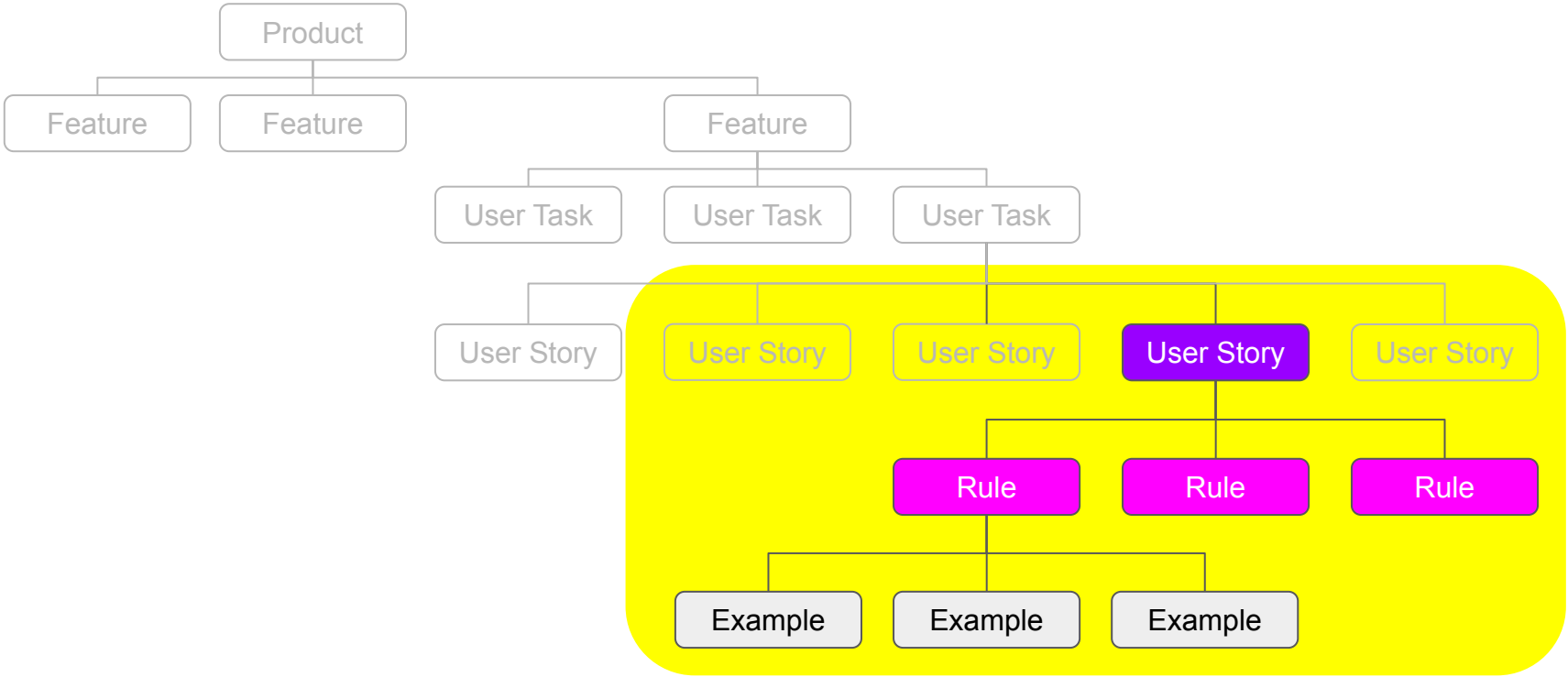
Story Map


Product
Feature
Feature
Feature
User Task
User Task
User Task
User Story
User Story
User Story
User Story
User Story
Rule
Rule
Rule
Example
Example
Example

Story Map + Slice



Example Map


Product
Feature
Feature
Feature
User Task
User Task
User Task
User Story
User Story
User Story
User Story
User Story
Rule
Rule
Rule
Example
Example
Example