Authoring Process *Skills: user flows, information architecture*

Sero! is a **web-based assessment tool** for teachers and students. It uses an alternative to multiple choice — **concept mapping** — as the assessment method.

Brief

The most interaction-dense part of Sero! is the process of **authoring assessments**. It involves creating a concept map and deciding how to assess students on the content inside it. As more options were added to this feature over time, it became harder for the team to **visualize and iterate its design and development.**

My goals for this project were to **strengthen my mental model of the authoring process** and **identify areas with room for improvement.**

Creating user flows

I made four user flows, one to reflect each step of the authoring process:

Map setup: title the assessment and choose from options that will shape the concept map.

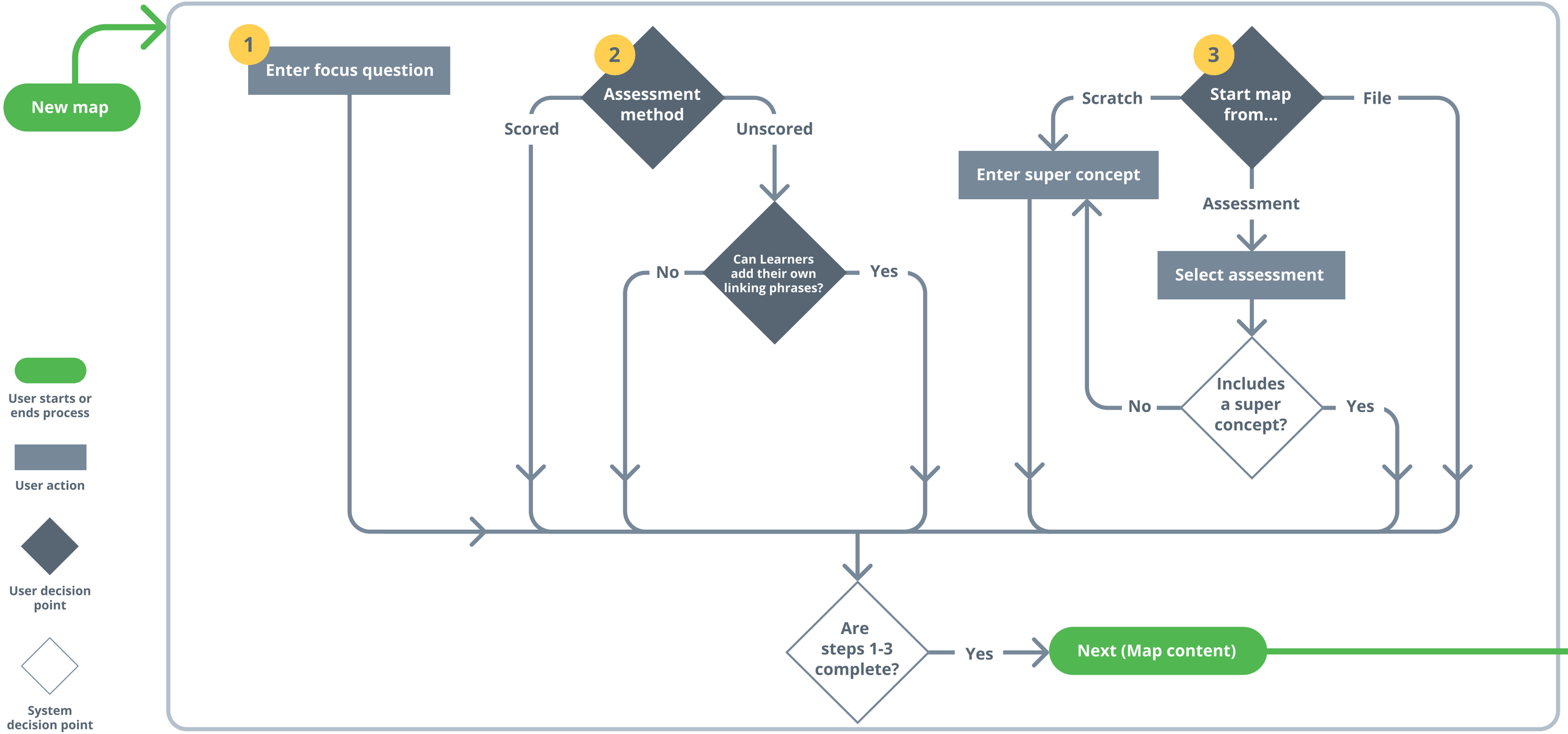
Map content: create and edit a concept map.

Assessment: decide how to assess students on the map content.

Assign: choose whom to assign to and choose from options that affect what students can see after taking the assessment.

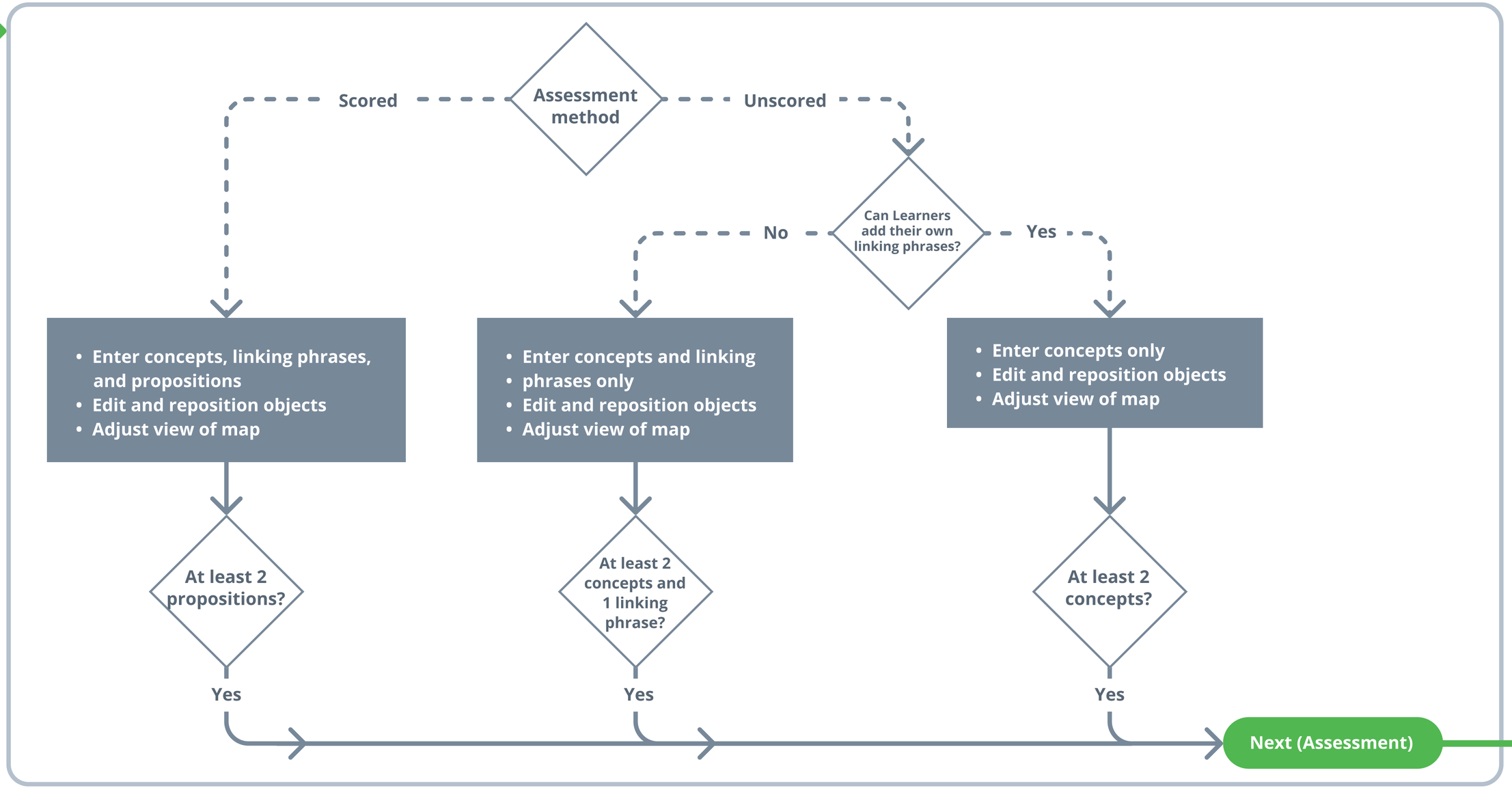
Authoring Process *Skills: user flows, information architecture* 3

Map setup



Authoring Process *Skills: user flows, information architecture*

Map content



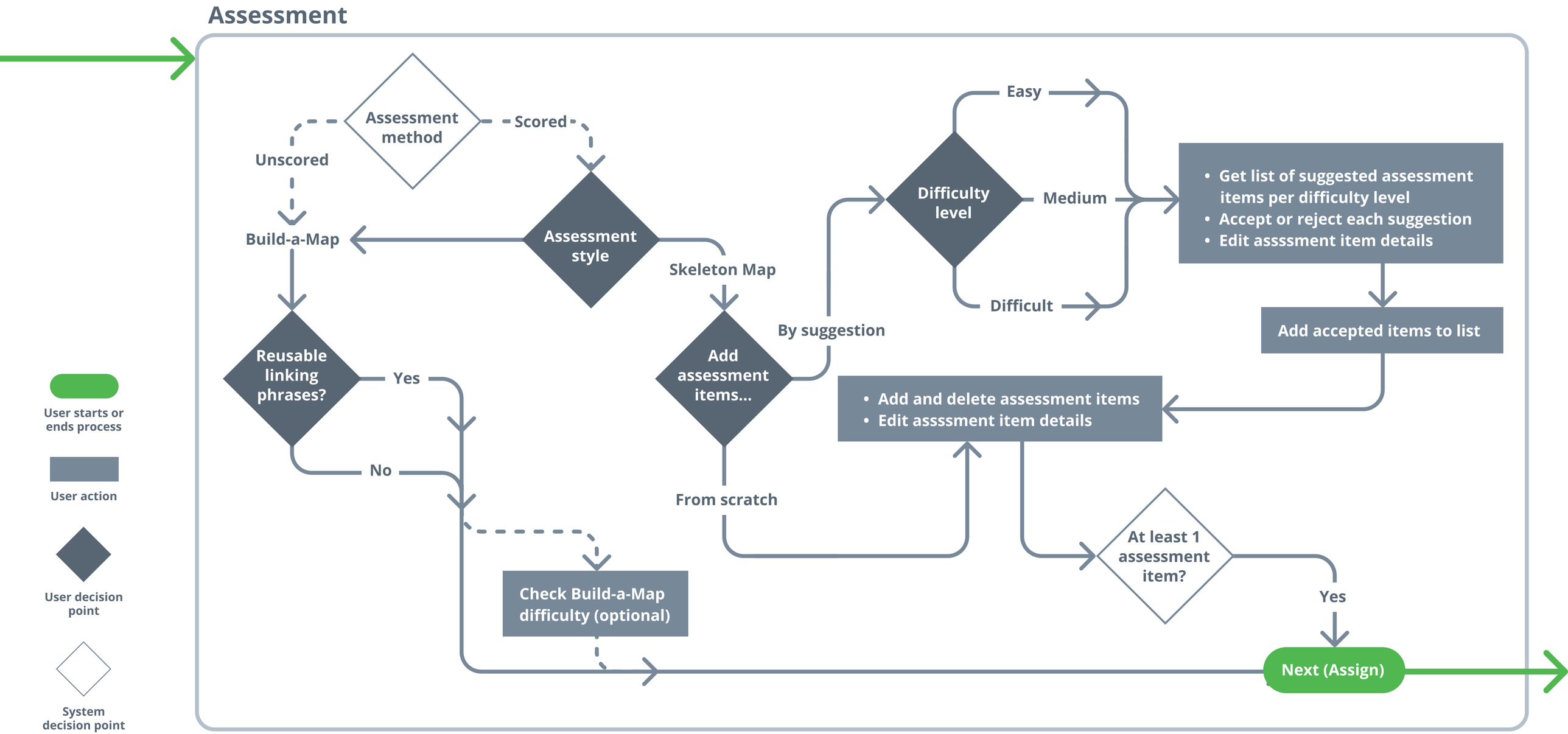
User starts or ends process

User action

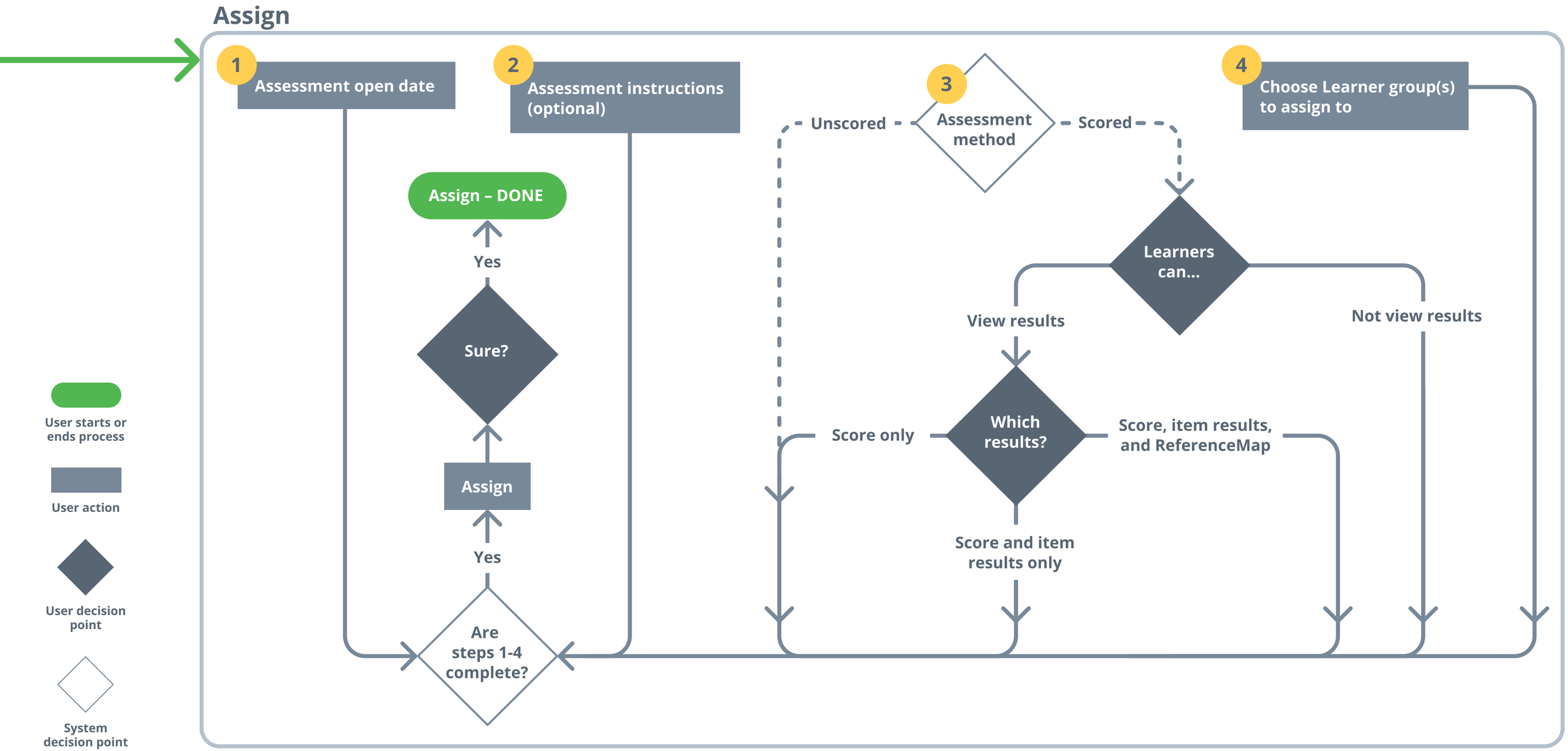
User decision point

System decision point

Authoring Process *Skills: user flows, information architecture*



Authoring Process *Skills: user flows, information architecture*



Concluding thoughts

Creating and analyzing these user flows helped me step back from the details and realize a few things about the bigger picture:

Many options decided in one step don't take effect until a later step. Could it benefit the user to arrange some options closer to where they impact the experience?

Configuring one type of assessment requires far more steps than configuring the other type of assessment.
Could giving more screen space to the latter make that step more manageable?

Making user flows can help bridge the gap between my and developers' understanding of the authoring process.
Could simplifying the steps of these user flows also simplify the code and developers' maintenance of it? How might that affect the end user?