Here is a common example of an enabler and a spike (*assuming a proper User Story at the start, which is maybe a big assumptions*):

It starts with a PO working with a client and mapping a feature:

The example feature is something like "Diabetic Cost Savings Campaign". The POs map out the different things a campaign power user needs to do from writing queries, to needing to export results in different ways, to creating or having reports associated with a campaign. PO wouldn't need to write out full stories or Gherkin Acceptance Criteria during the mapping. It might be a more basic form of a story.

Example Story A during Feature Mapping:

Story: Find current diabetic members

AC1: Find members with a prescription

AC2: Find members with a high A1C

When the feature is mapped, it may be that the campaign is broken apart into more than one feature right away:

Campaign A, Feature 1: Finding diabetic members (a set of related stories here)

Campaign A, Feature 2: Exporting campaign contacts for communication (stories about exporting results for a campaign communication tools)

Campaign A, Feature 3: Diabetic member savings reporting (tracking who they sent mailers or emails to, who opened emails, who responded, etc.)

A mapped feature sets the general scope of the feature. The feature needs to achieve a managed scope before it comes to the team in refinement, so mapping helps break down the essential scope.

The features should be prioritized. In many case, they prioritize naturally (first CA-F1, then CA-F2, and then CA-F3). But teams may be working on multiple campaign initiatives. So the PO needs to walk away and think of all concurrent initiatives and prioritize the campaigns prioritized features into what already exists:

1. CB-F1
2. CA-F1
3. CB-F2
4. CA-F2
5. CB-F3
6. CA-F3

Now it is time to refine a mapped feature with the team. The PO brings the next prioritized feature. In this case, let’s imagine the team just completed refinement of CB-F1 stories. It is time to bring CA-F1 to the team. Let’s imagine the team is refining Example Story A above.

User Story: (The PO may have updated the mapped story to read like this for the team)

As a Diabetes Campaign Power User, I need to be able to find active members who have diabetes so I can recommend lower price drug treatment alternatives.

Acceptance Criteria: (The team and the PO… as well as sometimes the client … fill out the AC details like below for Example Story A above)

1. *Find members with a prescription*

GIVEN I have access to current member plan information

AND I have access to historical pharmacy member prescription information

WHEN I search for current members taking diabetic treatment drugs

THEN I see a list of member content information with their current prescription

2. *Find members with a high A1C*

GIVEN I have access to current member plan information

AND I have access medical health check-up details

WHEN I search for current members A1C marker of over 7%

THEN I see a list of member content information

The team would ask a ton of questions and help "clarify" the Story into a state what we see above. The team might think of more examples of scenarios, especially if something could go wrong, like data was missing and what message to show the Campaign Power User (that would become a new AC added to the story).

The team would then start thinking about the potential solutions, where the conversation could sound like...

* "Ok, looking at the Acceptance Criteria, it sounds like we have three data needs: current member data, member prescription history, and member health data."
* "I don't believe we have ever worked with member health data. I think we need a spike to research that a little bit."
* "I agree. We cannot estimate this story until we can see the data structures and how we can ingest that data."
* "Here is my recommendation for a spike Product Backlog Item... Spike - how do we ingest the member health data, specifically A1C data?"
* "Let's break that apart. First... Spike - Can we get A1C health data?"
* Ok, and then maybe at the same time a second spike... Spike - How do we ingest A1C health data?"
* "Wait - we don't need a spike on the first one. I can answer that right now in refinement. I am looking right now and apparently that data is over in the X database. I don't believe we have worked with that one yet."

The team and the PO would go on to refine other stories in the feature (or the next priority Feature set by the PO) and that story would have to wait until the spike could go into a sprint and an Agile Dev Team Member with analysis and/or programming skills could take a look at the procedure for ingesting that data in the Spike.

Let's imagine the next sprint started two days later and the team took on that second spike. Two days later they have done an experiment and have answered the question. So they bring it to the next refinement that is happening in the current sprint...

* "We completed the *Spike - Can we get A1C health data?* And it looks like we can get the data ingested."
* The PO might ask, "Are we ready to estimate the story now that we are out of questions?"
* (The team votes by discussing how the story is like the work they did on another similar story previously completed and they comparatively size this story using relative sizing)
* The team members who participated in the Spike share a new insight, "Now we need an enabler for our story! We need to setup permissions to access the existing Web Service that will get us the A1C data. And we need to extend the runway to hold the ingested data to use in the story."
* PO might ask "So what do you want for an enabler?"
* The Spike team members, *"Enabler - Setup for Member Health Data Ingestion"*
* Enablers still need acceptance criteria, so they write out Gherkin statements for the technical enabler, limiting their activity to setting up the runway...  
  + GIVEN we have a user profile that can connect to the A1C data service  
    AND there is new data since our last ingestion  
    WHEN the ingestion process requests data  
    THEN the TARGET model receives fresh data  
    AND the INGESTION AUDIT LOG is updated with the new ingestion count
  + GIVEN we have a user profile that can connect to the A1C data service  
    AND there is no new data since our last ingestion  
    WHEN the ingestion process requests data  
    THEN the TARGET model receives NO NEW DATA  
    and the INGESTION AUDIT LOG is updated with a 0 new row count

During next Sprint planning, the team must finalize all the tasks they will do to accomplish the Enabler first. They may have started writing task ideas into the Enabler even back during Refinement. But the goal is to finalize those tasks to formulate a proper plan (the order of tasks, who will work on which task, how long they feel it will take in hours). This research and refinement by the team, in this order, is what creates stable sprints.

Assuming they get the enabler completed in the coming sprint, the user story will go into the sprint after that. The User Story would not be attempted in the same sprint as the predecessor Enabler if it can be avoided. Trying to get two dependencies done in the same sprint is possible but it increases the risk of not accomplishing the work in the sprint.