Project Justification

After you complete the graphic organizer below, use this project justification document to explain how you used computational thinking in your project.

Problem Identification. For each iteration of your problem, please explain how you arrived at your identified problem.

The first problem was to uderstand how a company could send maintenance before a machine breaks. The second was finding how to read the datas from the sensors. The third was how to fix possible errors in the data.

Decomposition. For each iteration where you decomposed an identified problem, please explain how this decomposition helped you solve your identified problem.

Decompositions was useful, in each iteration a sub-problem was identified. The all helped in finding a solution.

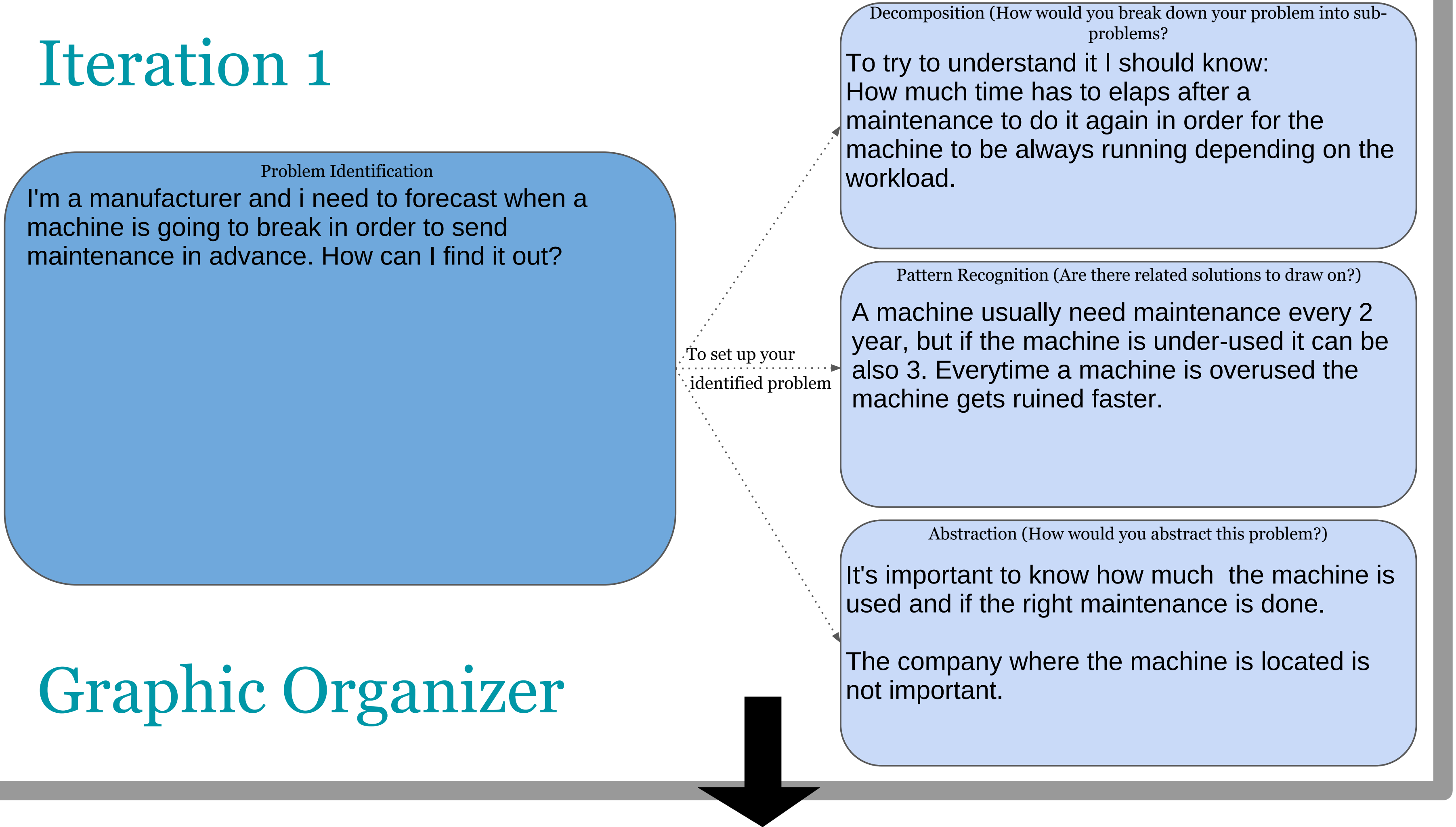
Pattern Recognition. For each iteration where you recognized patterns in data, please explain how these patterns helped you solve your identified problem.

Pattern recognition was essential in order to solve the problem because information were based on past trends.

Abstraction. For each iteration where you abstracted information, please explain how abstraction allowed you to solve your identified problem.

Abstraction was absolutely needed in order to identify which where the important data in order to solve the problem

Iteration 1



Problem Identification

I'm a manufacturer and i need to forecast when a machine is going to break in order to send maintenance in advance. How can I find it out?

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?

To try to understand it I should know:
How much time has to elaps after a maintenance to do it again in order for the machine to be always running depending on the workload.

Pattern Recognition (Are there related solutions to draw on?)

A machine usually need maintenance every 2 year, but if the machine is under-used it can be also 3. Everytime a machine is overused the machine gets ruined faster.

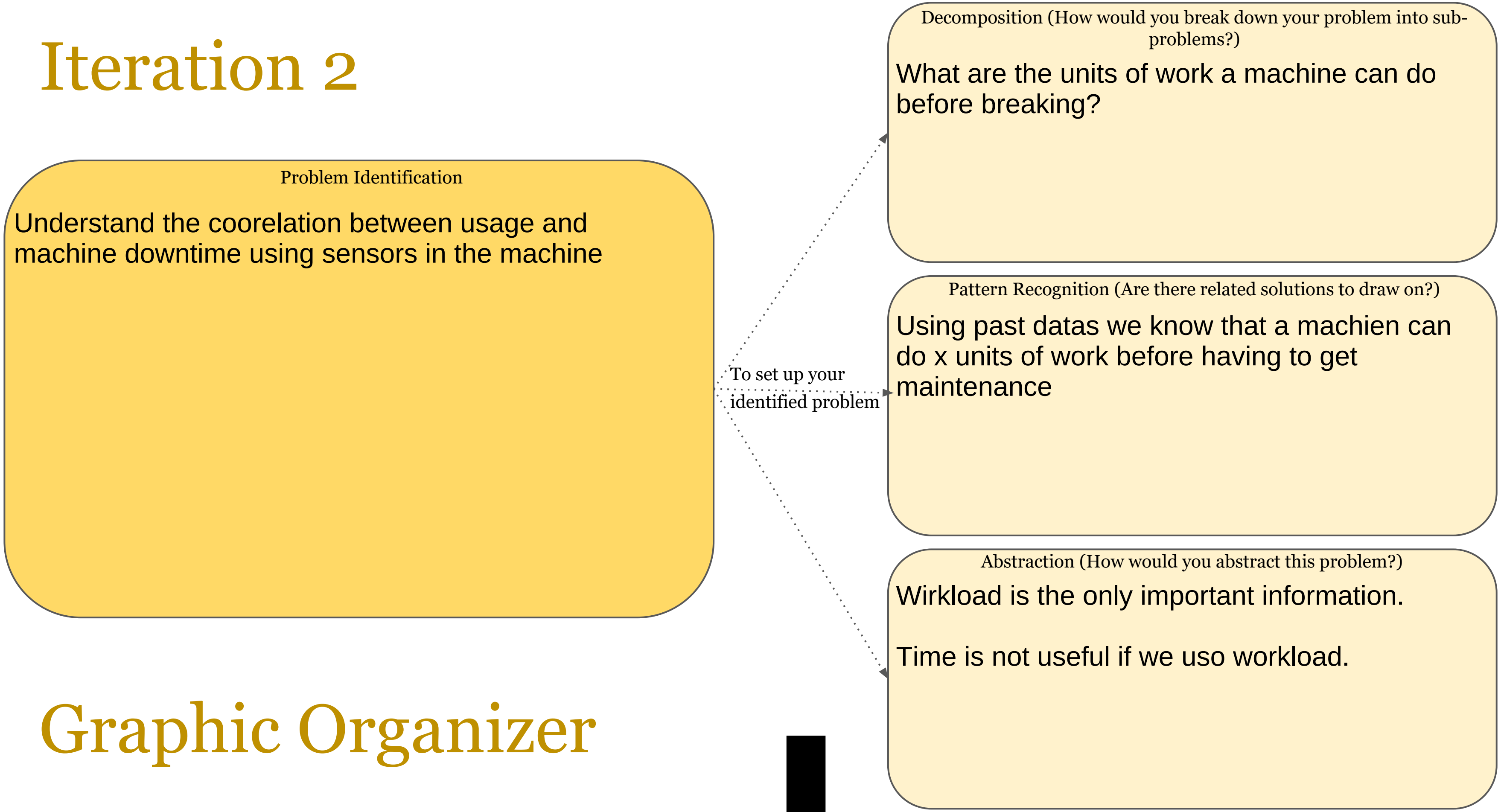
Abstraction (How would you abstract this problem?)

It's important to know how much the machine is used and if the right maintenance is done.

The company where the machine is located is not important.

Graphic Organizer

Iteration 2



Problem Identification

Understand the coorelation between usage and machine downtime using sensors in the machine

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

What are the units of work a machine can do before breaking?

Pattern Recognition (Are there related solutions to draw on?)

Using past datas we know that a machien can do x units of work before having to get maintenance

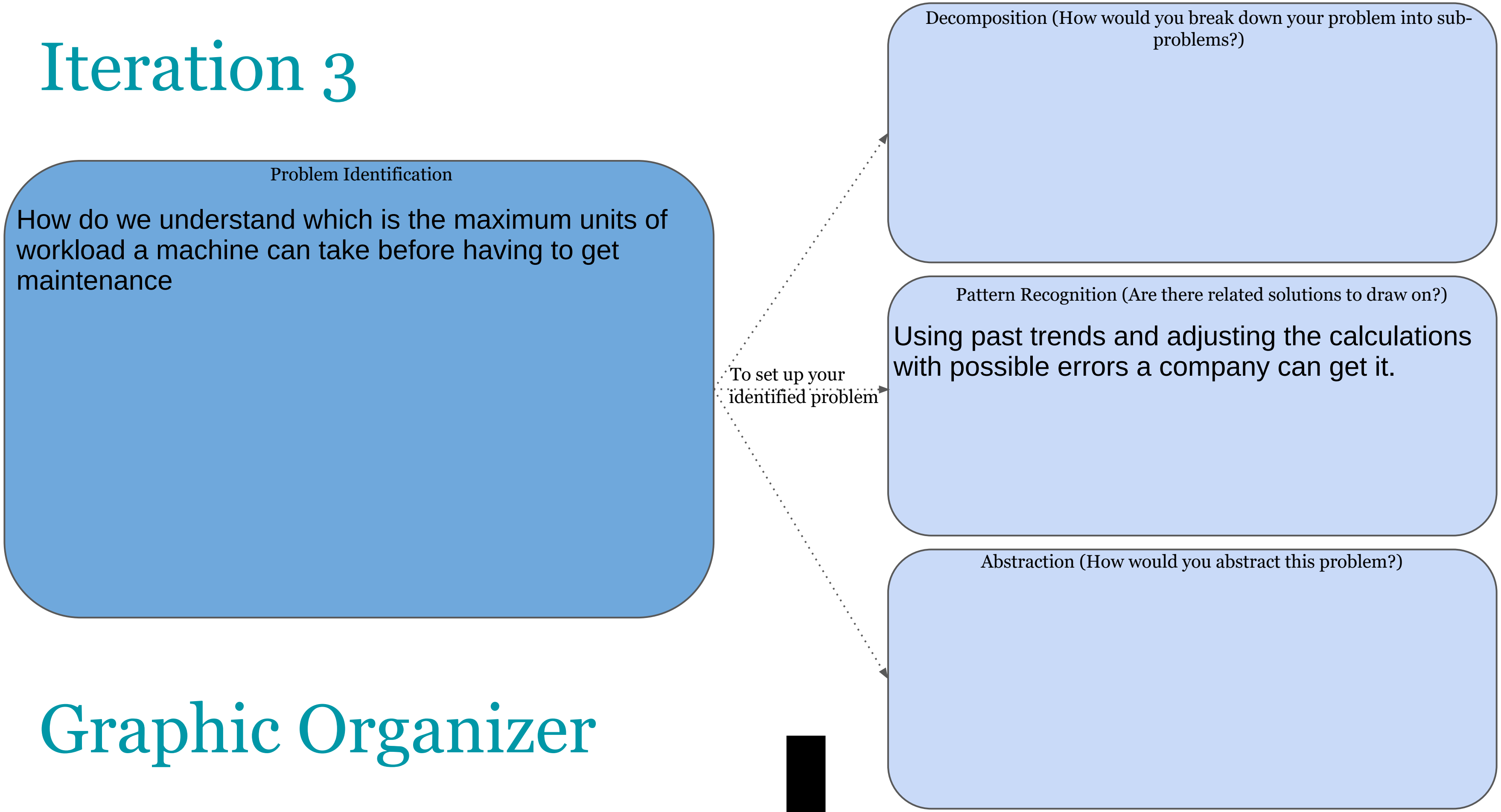
Abstraction (How would you abstract this problem?)

Wirkload is the only important information.

Time is not useful if we uso workload.

Graphic Organizer

Iteration 3



Problem Identification

How do we understand which is the maximum units of workload a machine can take before having to get maintenance

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

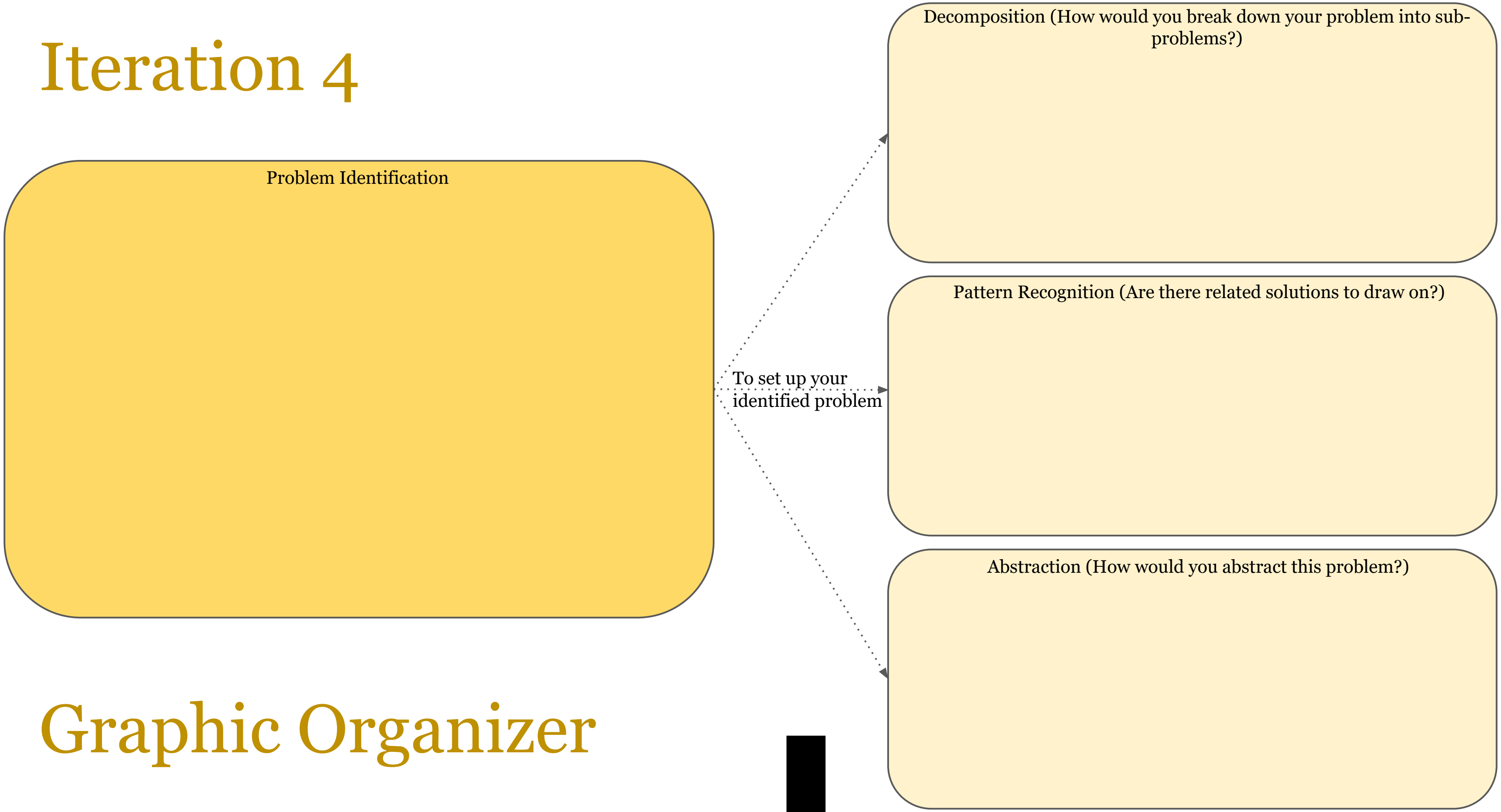
Pattern Recognition (Are there related solutions to draw on?)

Using past trends and adjusting the calculations with possible errors a company can get it.

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 4



Problem Identification

To set up your identified problem

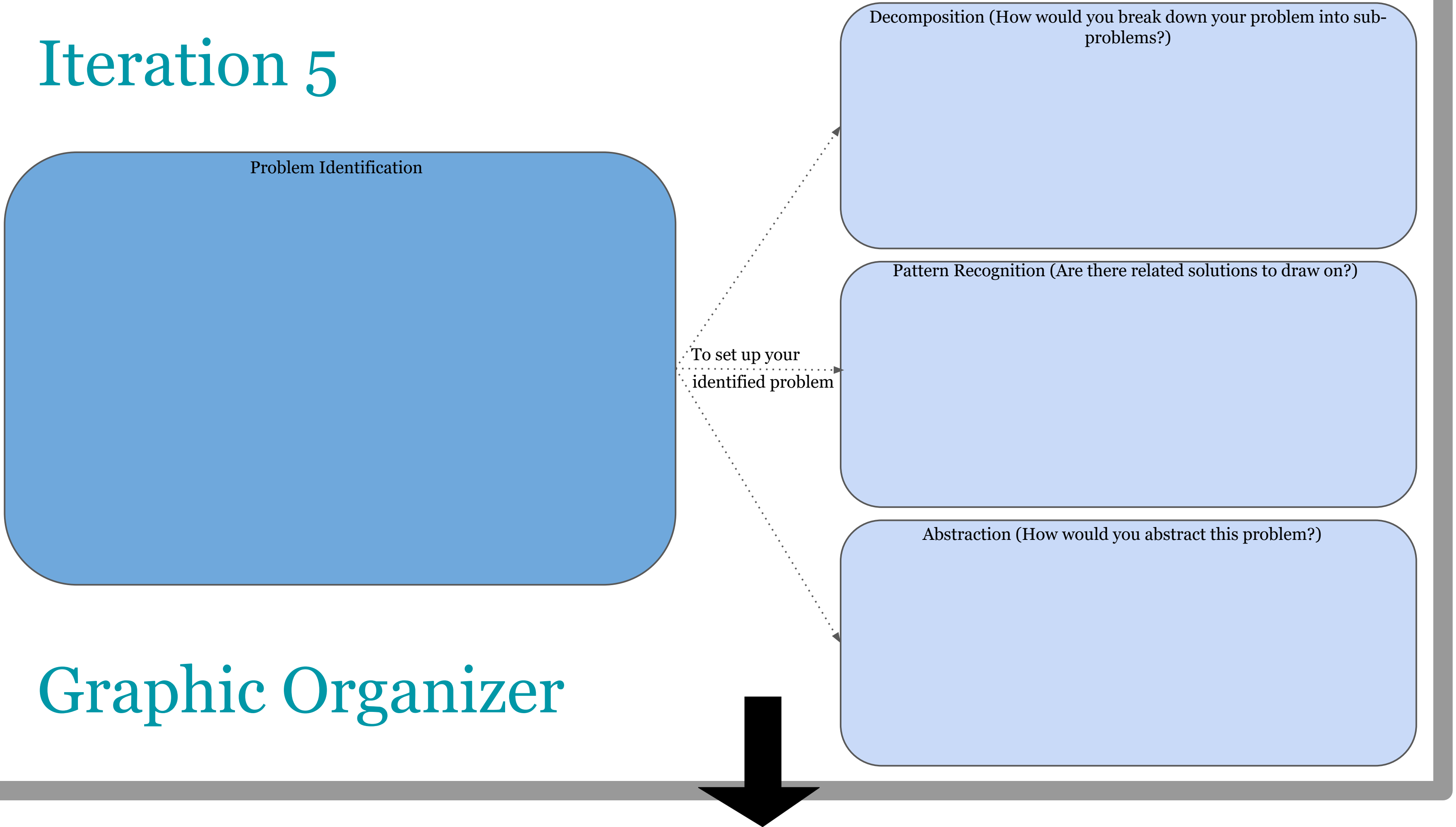
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 5



Problem Identification

To set up your identified problem

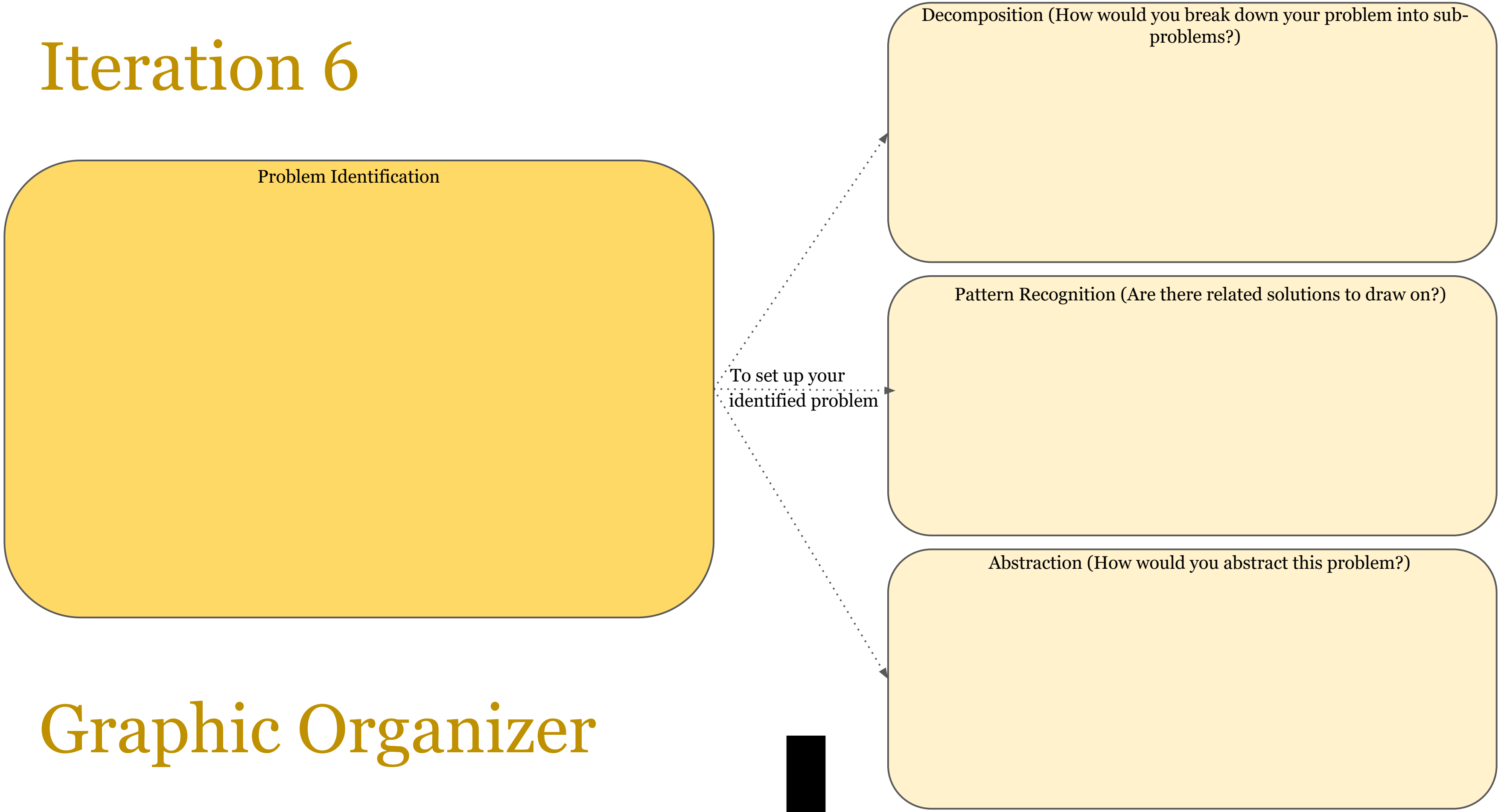
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 6



Problem Identification

To set up your identified problem

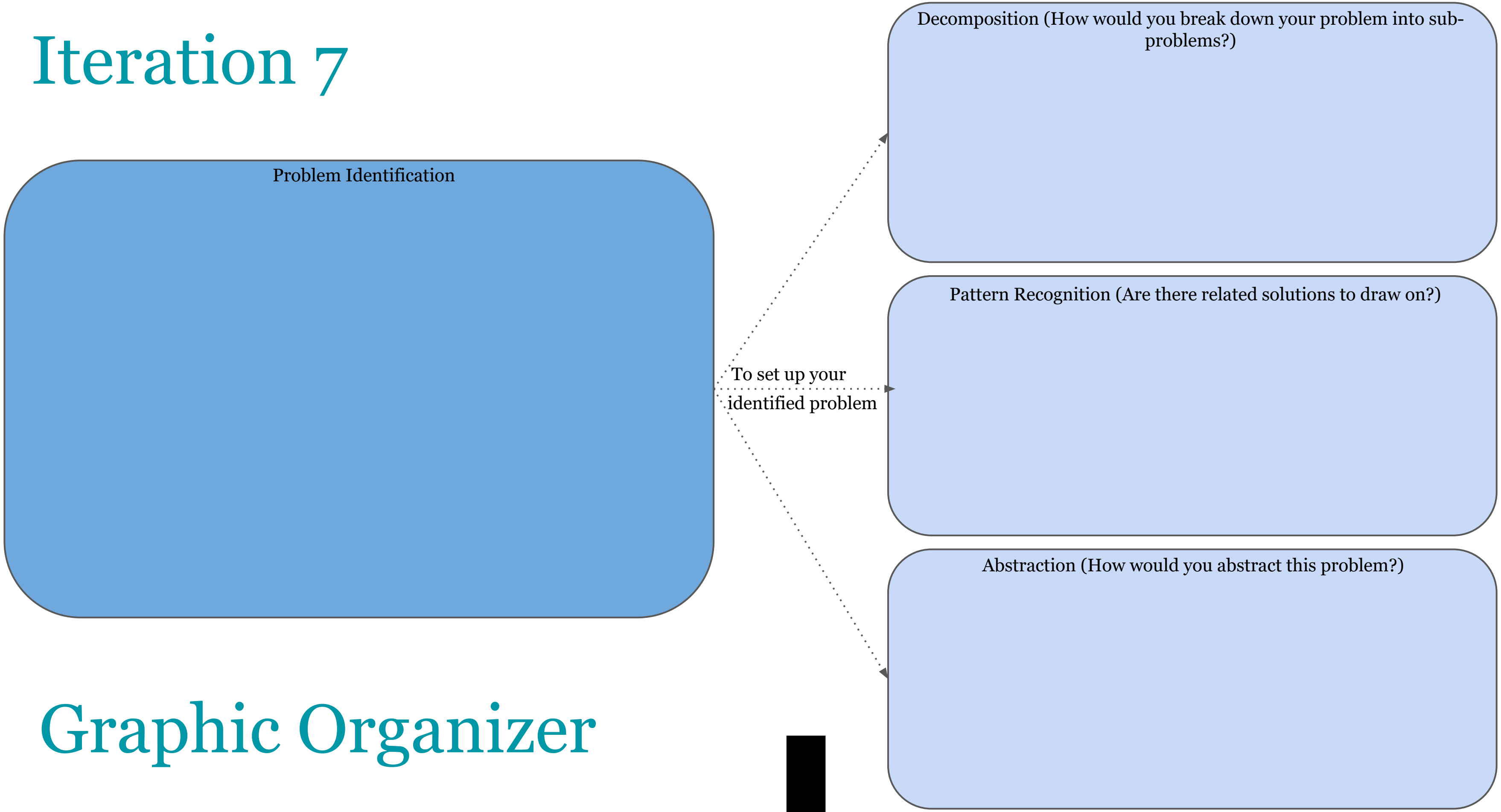
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 7



Problem Identification

To set up your identified problem

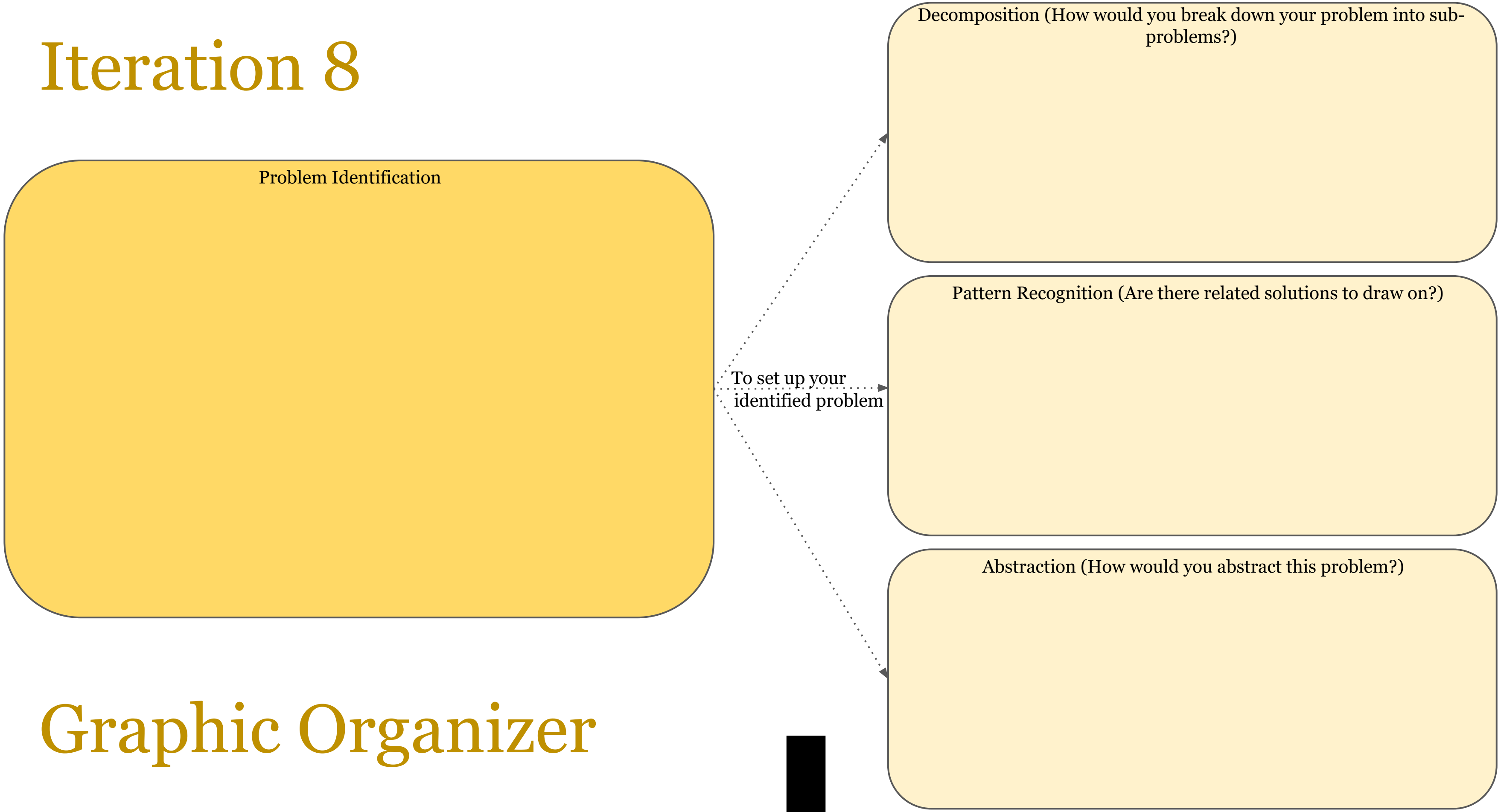
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 8



Problem Identification

To set up your identified problem

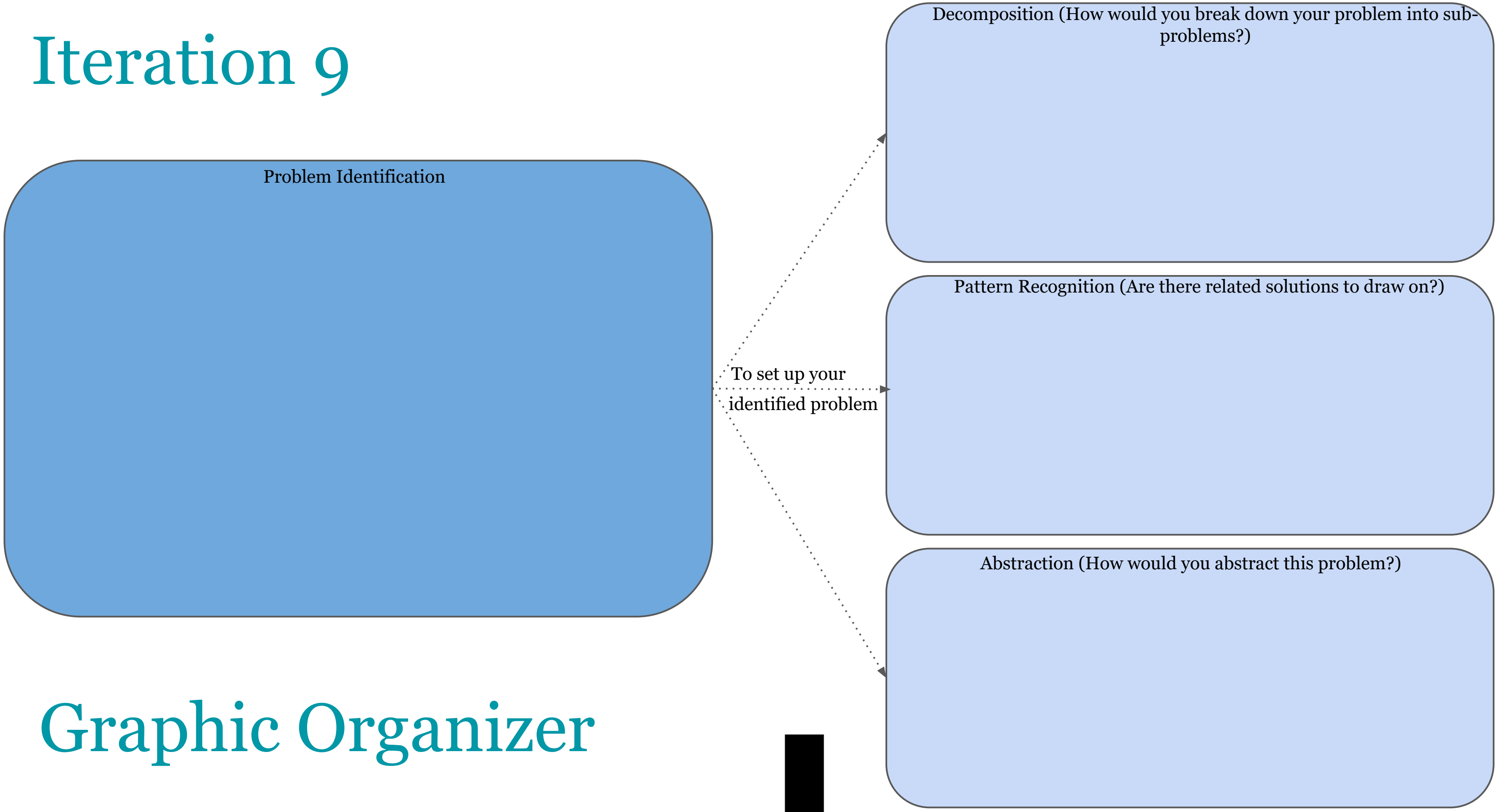
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 9



Problem Identification

To set up your identified problem

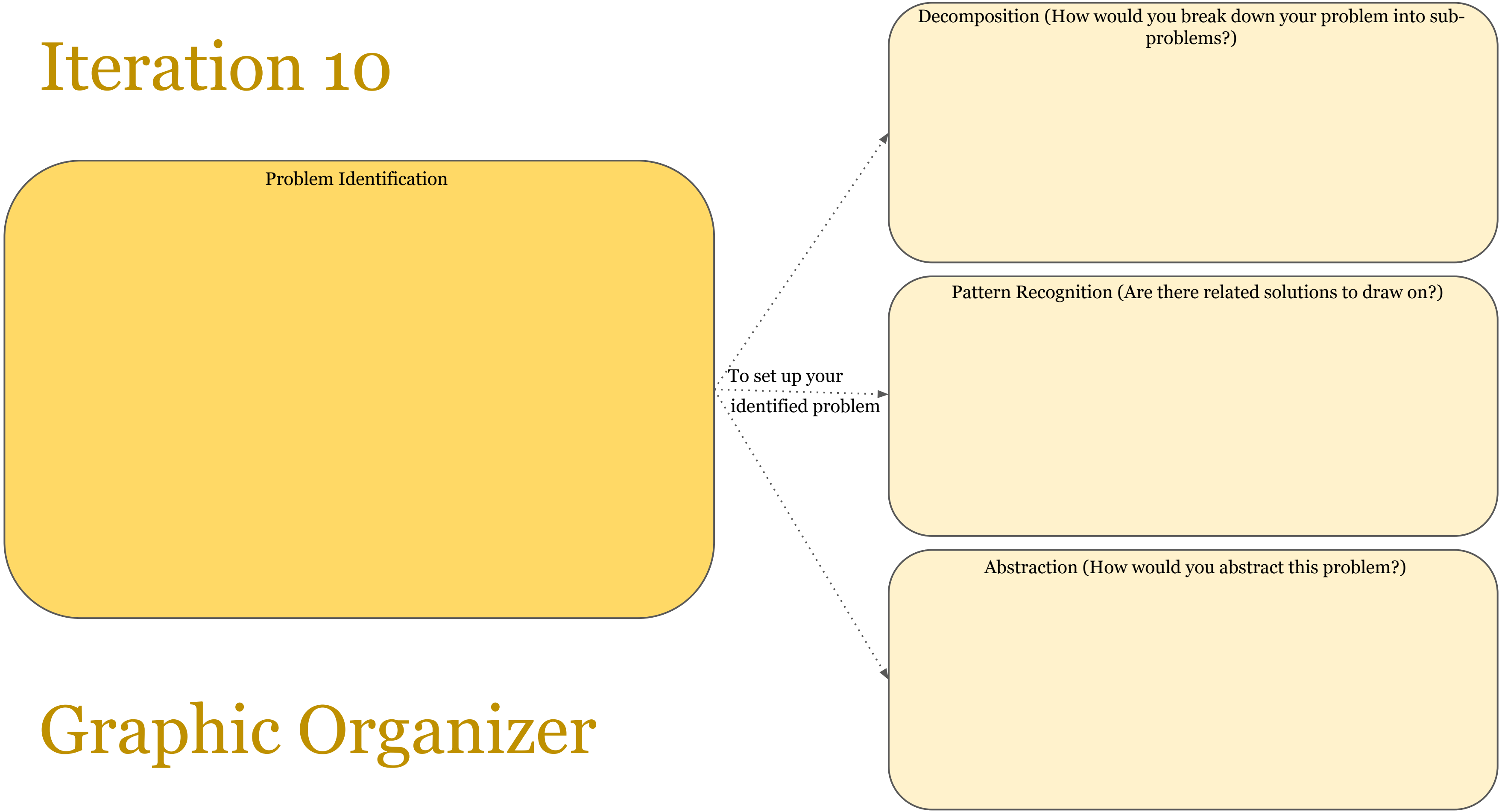
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 10



Problem Identification

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer