
Dependency Migration in Practice

— Anjali Pal —

Code.org

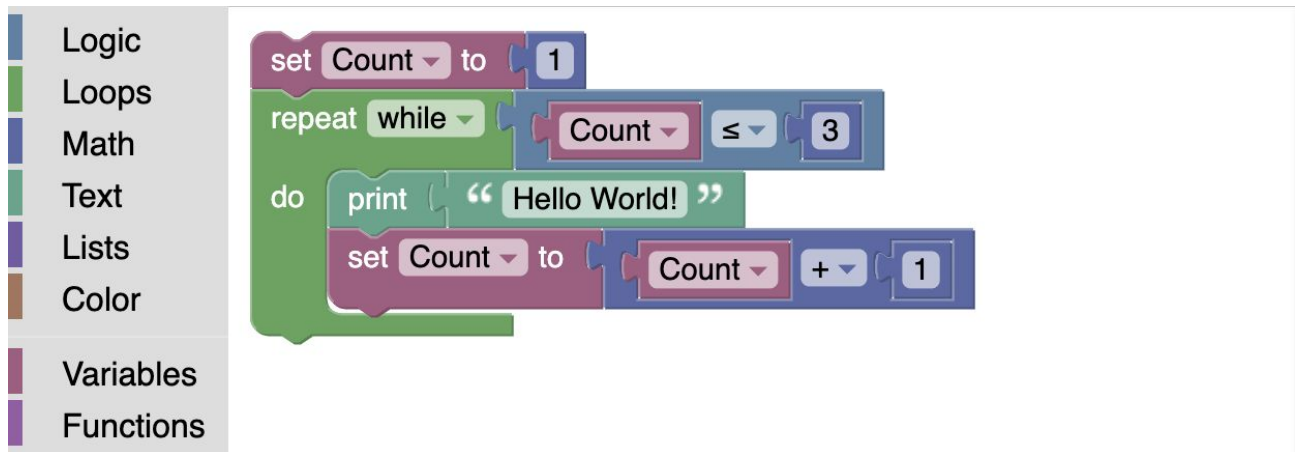
The screenshot displays the Code.org Scratch environment. On the left is a game level with a 5x5 grid. A character is positioned at the bottom right of the grid. Three purple diamonds, each labeled with the number '1', are located at the top-left, top-right, and middle-left positions of the grid. Below the grid are two orange buttons: 'Run' and 'Step'. A goal indicator shows 'Goal' and '0/3' diamonds.

On the right, the 'Blocks' panel shows a script for the character. The script starts with a 'when run' block, followed by a 'repeat 3 times' loop. Inside the loop, the blocks are: 'move forward', 'move forward', 'collect', and 'turn right'. Below the loop, there is a 'collect' block. A 'repeat ??? times' block is also visible in the blocks panel.

```
when run
  repeat 3 times
    do
      move forward
      move forward
      collect
      turn right
  collect
```

Demo

Blockly



The image shows a Blockly code editor interface. On the left is a vertical sidebar with category tabs: Logic, Loops, Math, Text, Lists, Color, Variables, and Functions. The main workspace contains a script with the following blocks:

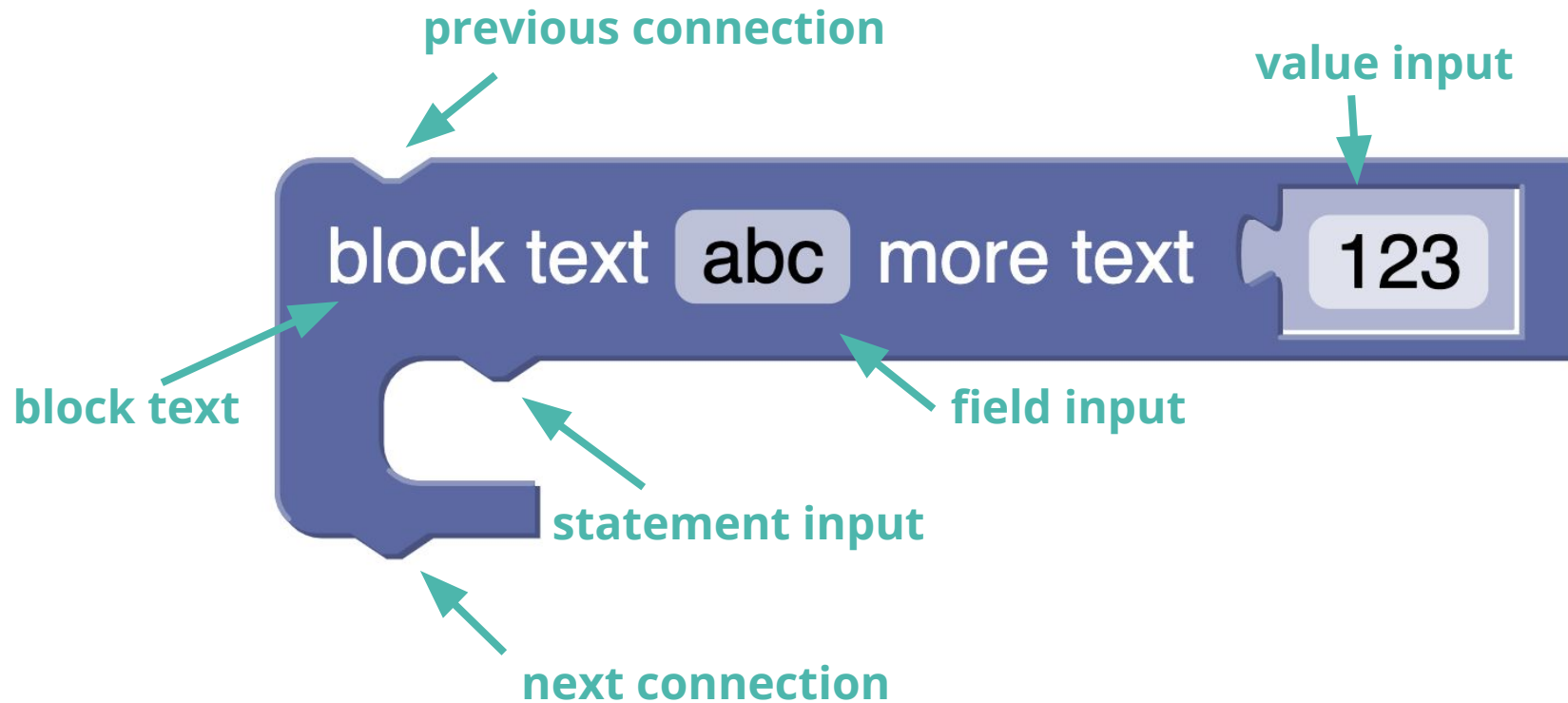
- A purple 'set Count to' block with the value '1'.
- A green 'repeat while' block with a dropdown menu set to 'while'.
- Inside the 'repeat while' block's 'do' area:
 - A green 'print' block with the text 'Hello World!'.
 - A blue 'set Count to' block with a dropdown menu set to 'Count' and a plus sign, followed by a blue '1' block.
- Outside the 'repeat while' block's 'while' area:
 - A blue '≤' (less than or equal to) block with a dropdown menu set to 'Count'.
 - A blue '3' block.

Language: JavaScript ▾

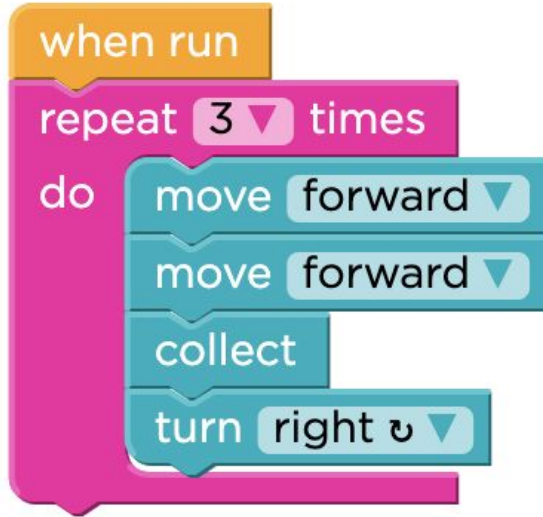
```
var Count;
```

```
Count = 1;
while (Count <= 3) {
  window.alert('Hello World!');
  Count = Count + 1;
}
```

Blockly



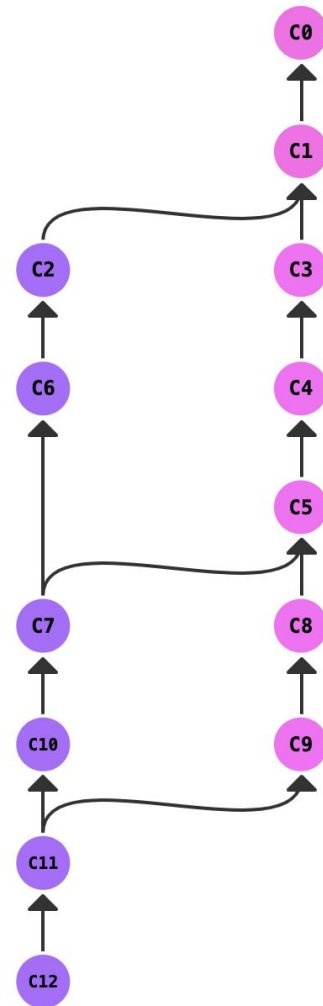
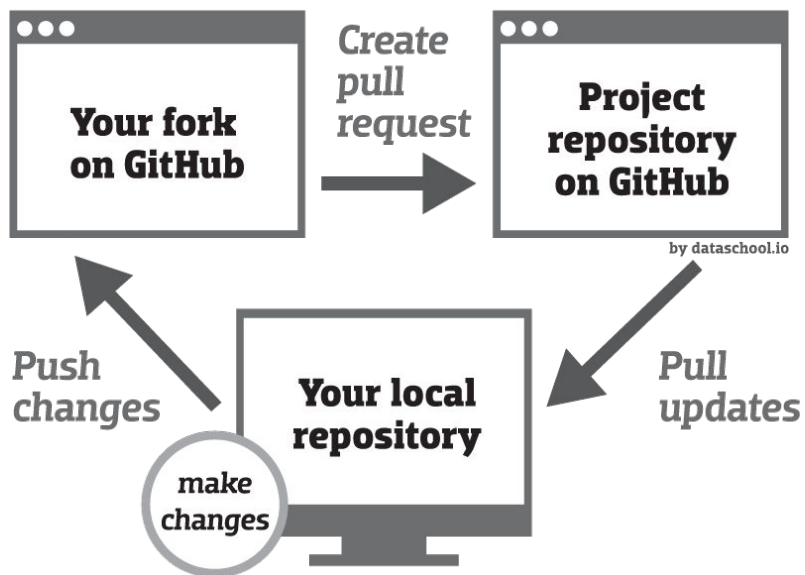
Custom Blocks Example



```
for (var count = 0; count < 3; count++) {  
  moveForward();  
  moveForward();  
  collect();  
  turnRight();  
}
```

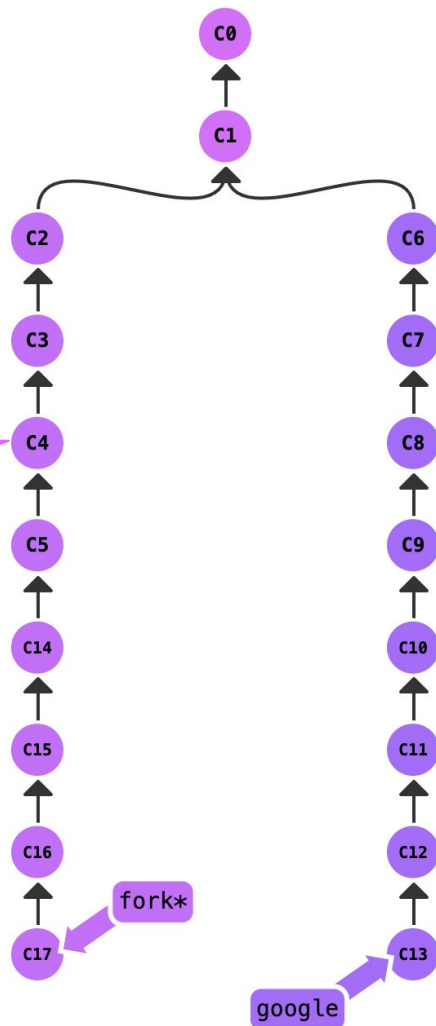
Open Source Fork Model

Expectation



Reality

~2300 commits

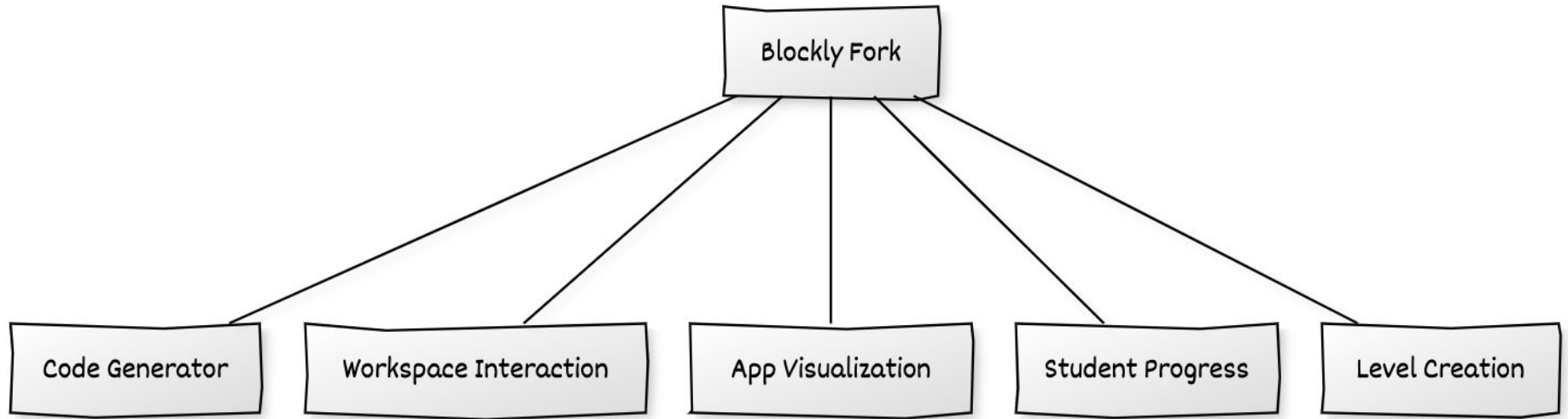


~7200 commits!

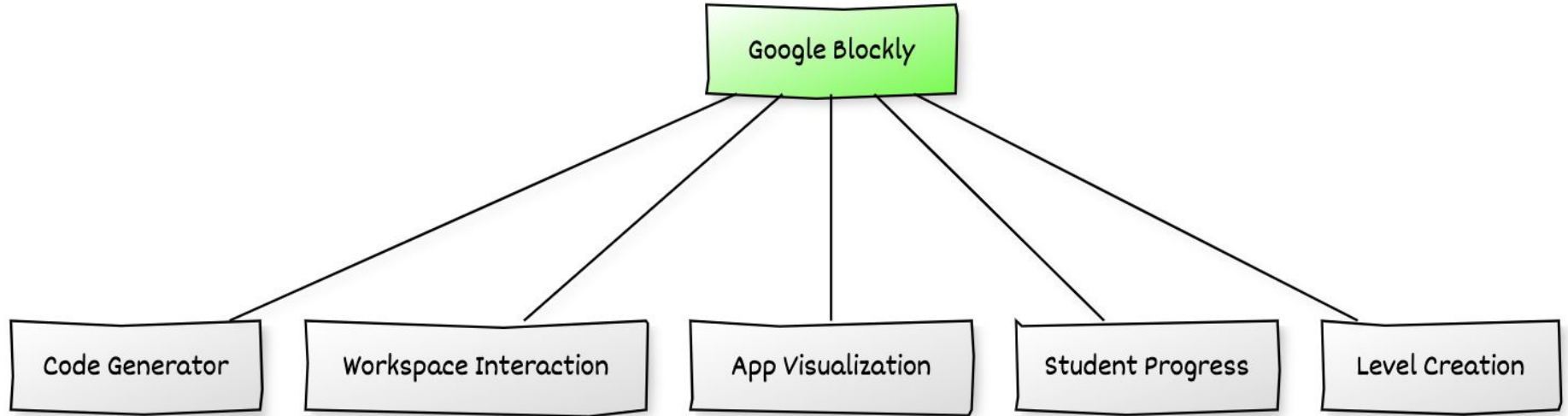
High-Level Goals

- ❖ Consistent Shipping
 - ❖ Incremental Progress
 - ❖ Isolated Changes
 - ❖ Manage Risk/Uncertainty
-

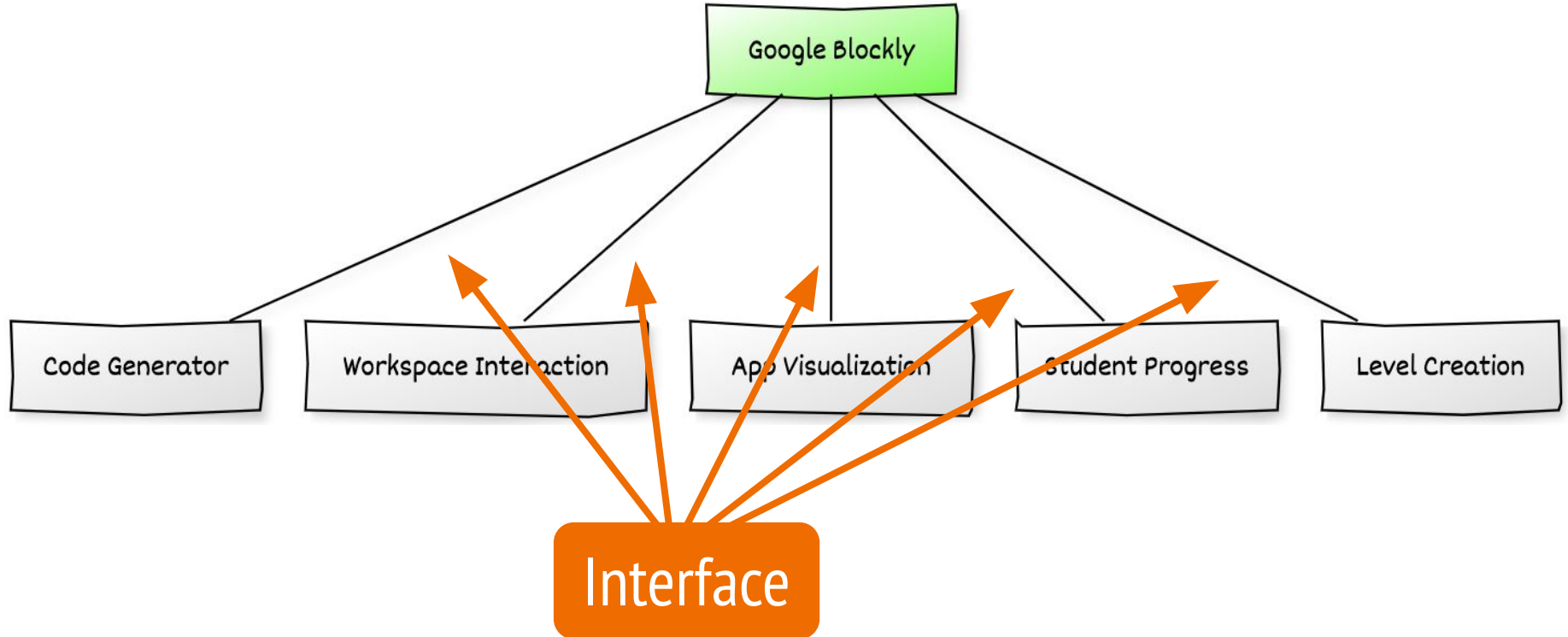
Migration



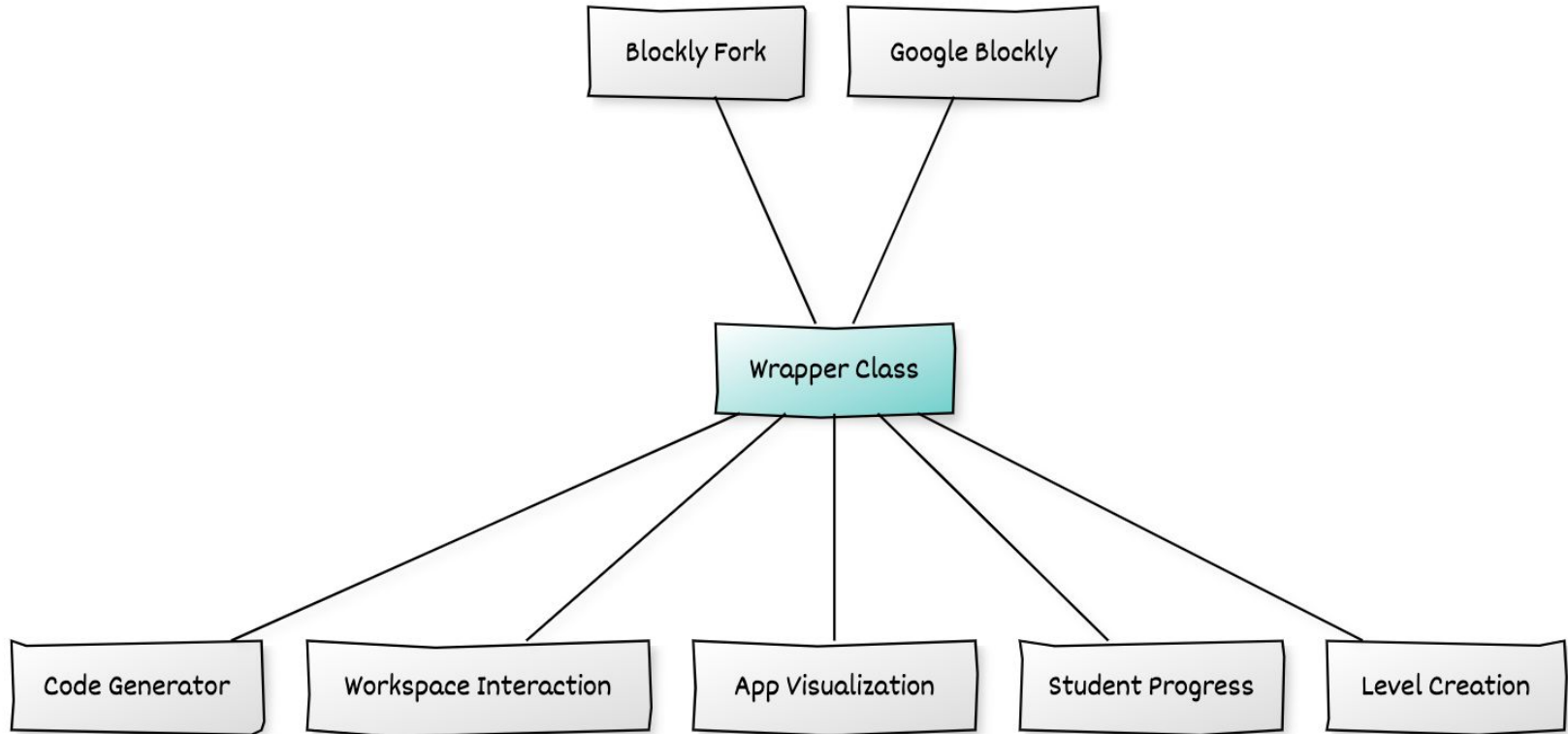
Migration



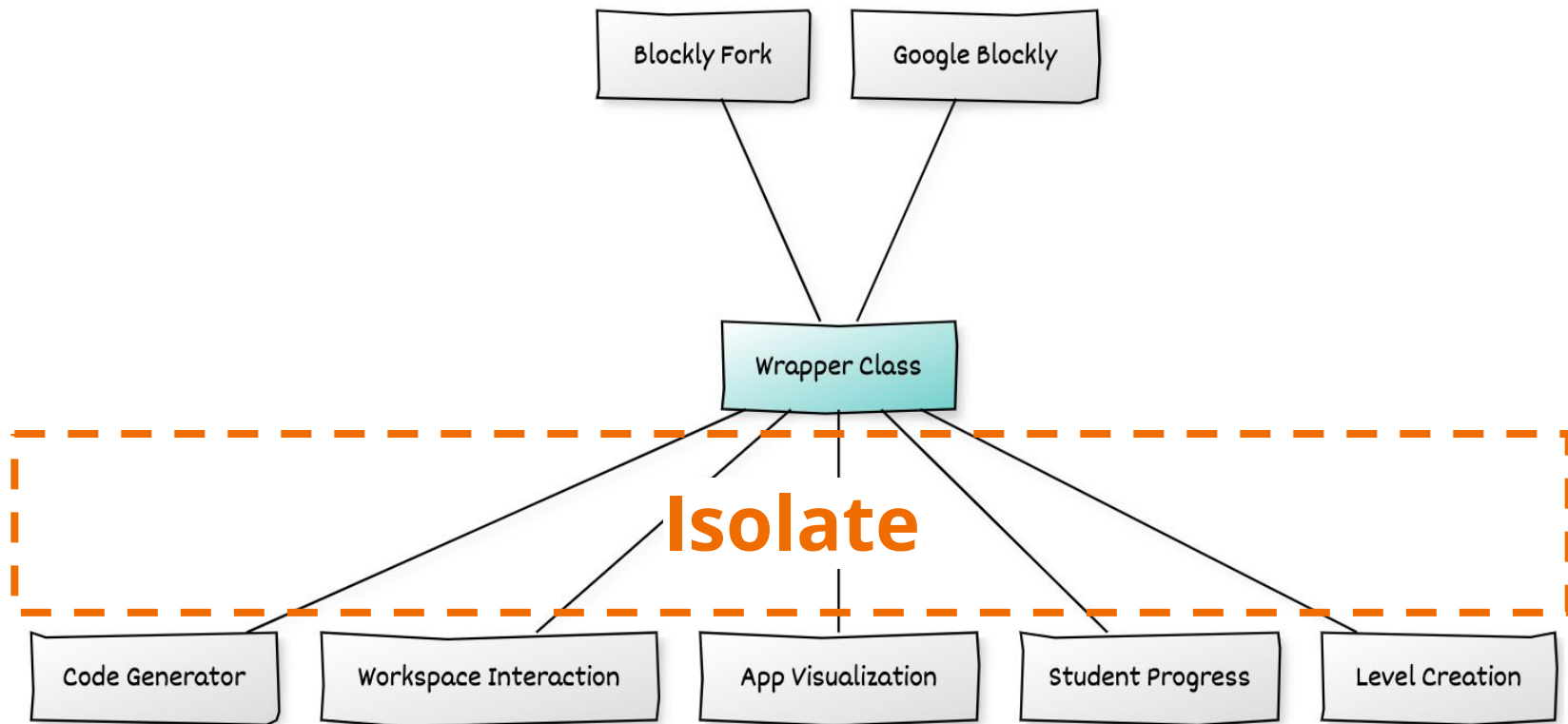
Migration



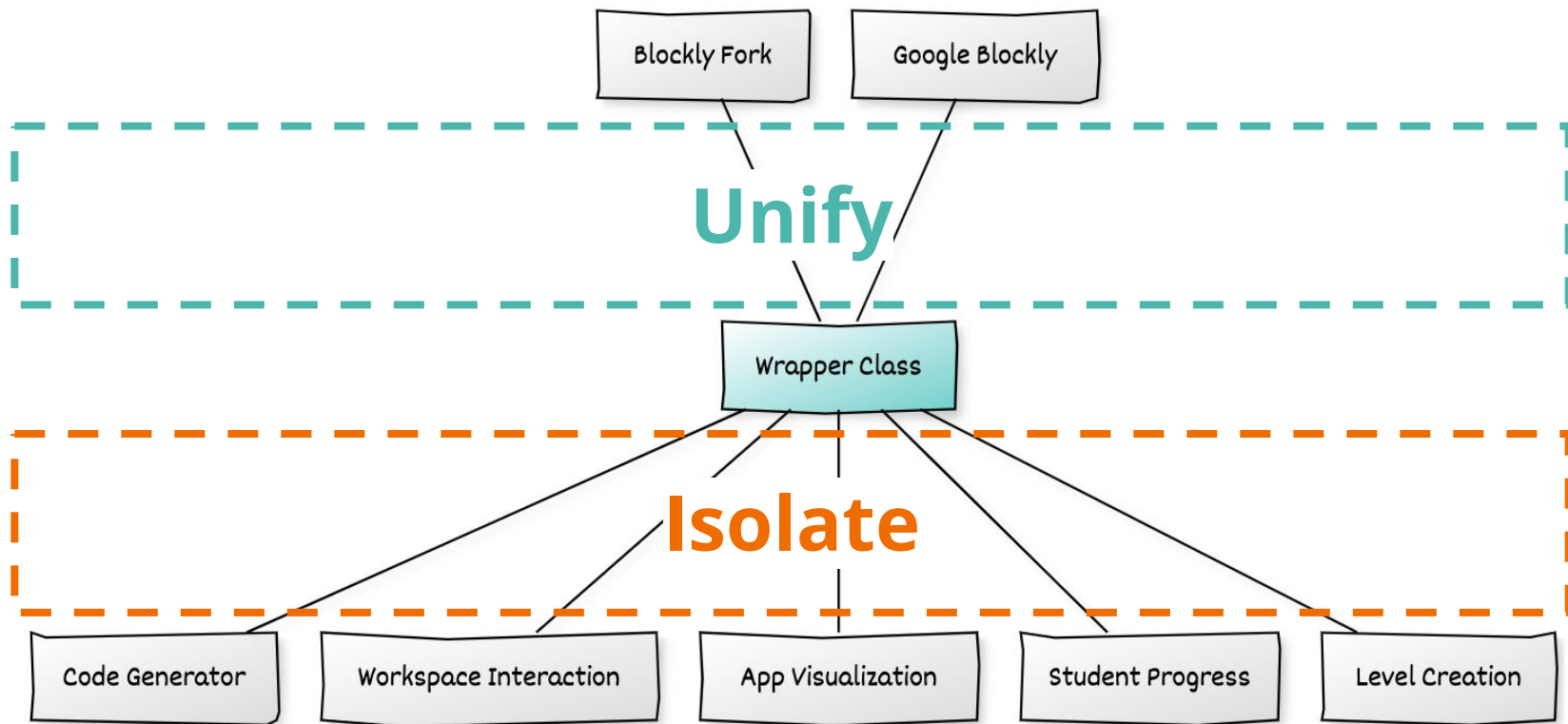
Migration



Migration



Migration



Example: Block Color



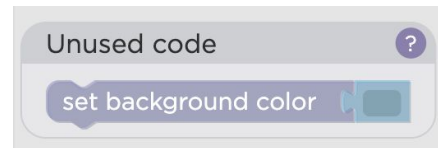
Google Blockly has a method called **Block.setColour**

Code.org Blockly fork has a method called **Block.setHSV**

Solution: Implement **setHSV** on the Google Blockly wrapper class, and pass through to **Block.setColour**

```
setHSV(h, s, v) {  
  return super.setColour(Blockly.utils.colour.hsvToHex(h, s, v * 255));  
}
```


Example: Unused Blocks



Code.org has a custom UI that shows around disconnected blocks, and is rendered when the Run button is clicked.

Solution:

- Implement a new class on the Google Blockly wrapper to handle rendering the unused block frame.
- Add a custom event handler on the run button to render the unused block frame.
- Override **Block.dispose** on the Google Blockly wrapper to also dispose of the unused block fram.

Example: Trashcan



Code.org Blockly positions the trashcan in the same place as the block toolbox

Google Blockly positions the trashcan in the lower-right corner of the workspace.

Solution:

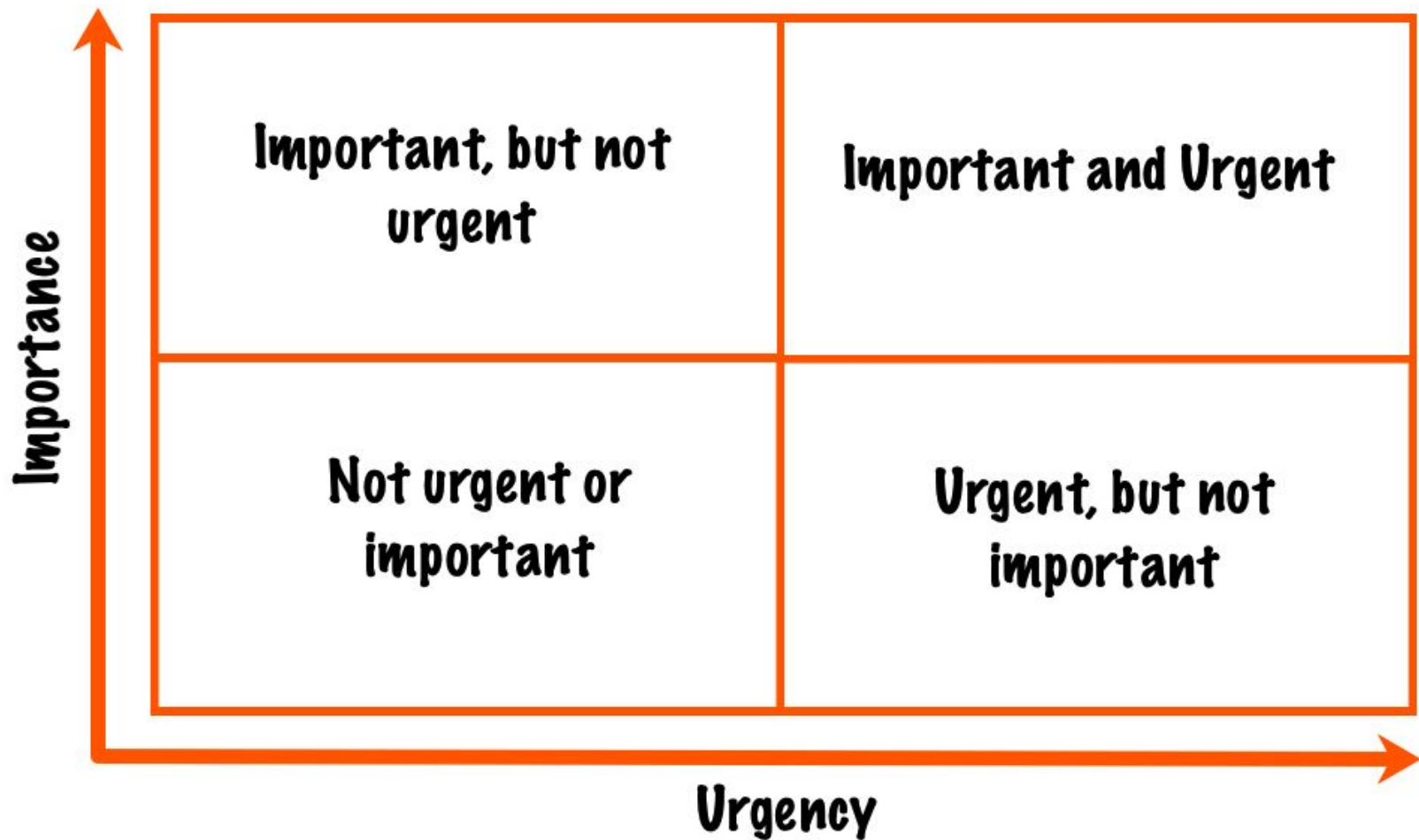
- Extend the Google Blockly Trashcan class to override the **position** method.
- Override **BlockDragger** to show/hide the trashcan while a mouse-drag is in progress.

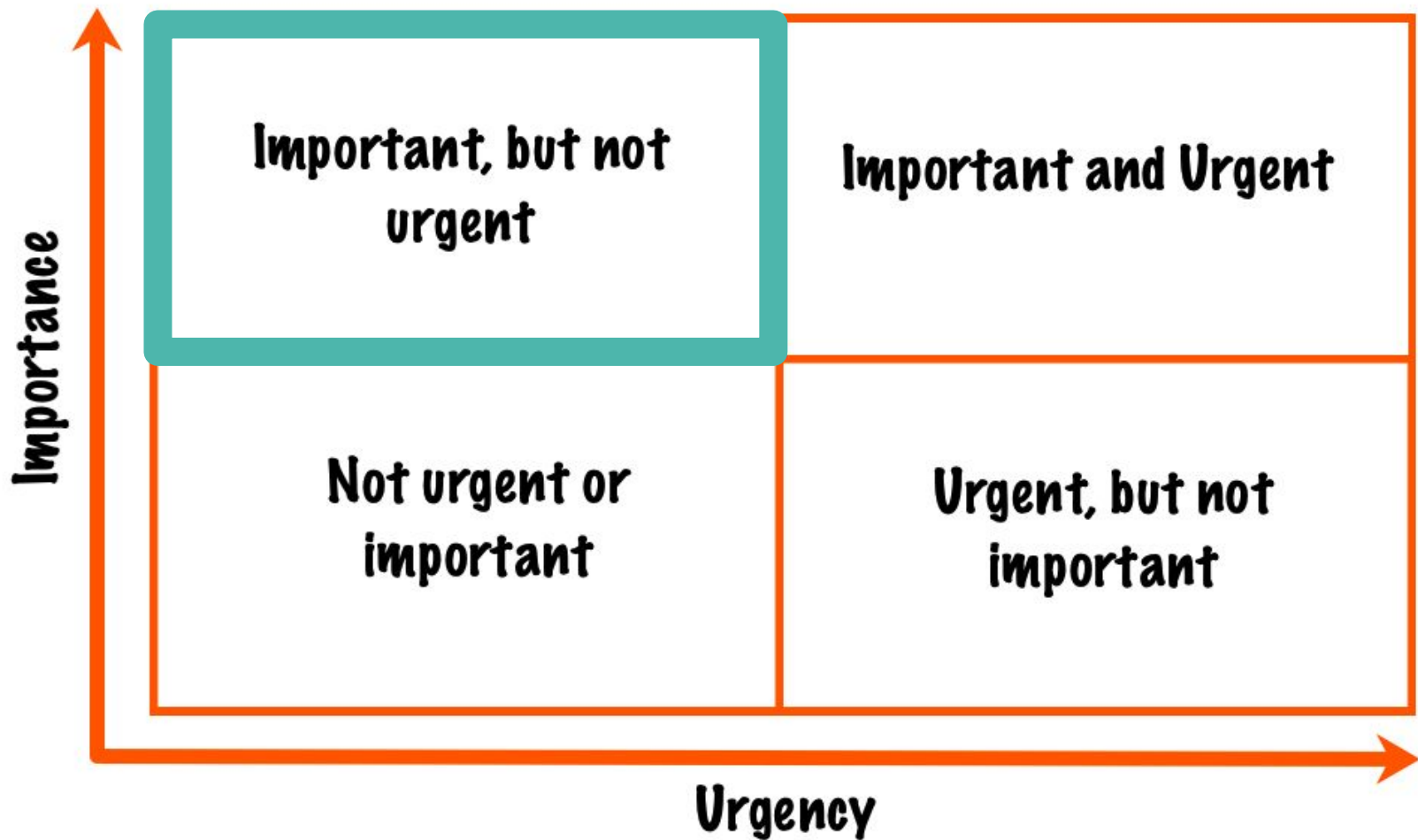
Deploying Software

Rollout

- Single flag configures which version of Blockly is used when the page is loaded.
- Each environment can be individually configured.
- The environment setting can be overridden with a URL query parameter for testing and development.
- We can easily revert back to the Blockly fork without a full production deploy.

Takeaways





Software engineering is not just code

- ❖ Understand history and context of existing code
 - ❖ Manage relationships, goals, incentives across various groups
 - ❖ Advocate for project direction
-

Code is communication

- ❖ Optimize for readability
 - ❖ People come and go on engineering teams, but the code stays
 - ❖ Reading code takes practice
-

Technical writing

- ❖ Strong technical writing is at least as important as coding ability
 - ❖ Know your audience
 - ❖ Explain why you made decisions
 - ❖ Outline long-term plan
-

Thank you!

Questions?