REDUCING THE LINES: A VISUAL DAG EDITOR
Why does this process take so long?

-- Every product owner, at every company
Writing DAGs can be a time consuming process. Checking all of the parameters for inclusion and accuracy, and creating task dependencies is time-consuming, and error prone.
THREE Impediments to writing DAGs quickly

Verbosity

Fluency

Complexity
1. VERBOSITY

Mountains of detail
DAG Metrics

DAG Length
Most DAGs were between 1000 and 2000 lines long, with many reaching up 4000 lines.

Task Relationships
Representing a many-to-one dependency ends up with many repetitions of very similar function calls.

- task_1 >> task_4
- task_2 >> task_4
- task_3 >> task_4

More info on DAG parameters at:
2. COMPLEXITY

Where does that go?
Task Metrics

Number Task Parameters

Many task parameters are repeated and use standard or common parameters.

Providing clear values for default boolean parameters requires extra lines of code.

Confusing Order for Parameters

As DAGs are written by different developers and/or updated over time, parameter order can become confusing, or subject to personal preferences.

More info on operator parameters to use this template at:
3. FLUENCY

Not everyone speaks Python
Language Adoption

Python Developers
Approximately 8MM† developers worldwide use Python. Out of a total global workforce of 3 Billion*, 0.002%

Business Power Users
Number of users familiar with a gui and browser. 1.5 Billion‡

Citations from:
†wikipedia(https://en.wikipedia.org/wiki/Global_workforce#—_text=As%20of%202012%2C%20the%20global%20workforce%2C%20around%202%20billion%20unemployed)
‡zdnet(https://www.zdnet.com/article/programming-languages-python-developers-now-outnumber-java-one)
§madeup stat to prove my point
A SOLUTION

In three parts
Questions: How can we enable?

Grouping
How can common tasks be grouped together?

Isolated Configuration
Can the creation of a DAG be driven dynamically?

Non-technical Authors
Can a someone without Python experience edit a DAG?
THREE Stages

SubDAGs  Dynamic DAGs  A Visual Editor
A complex DAG with 80 + tasks can weigh-in at near 5000 lines.
Adding SubDags for repetitive tasks can bring this down to less than 1000 lines.
Using Dynamic DAGs can help reduce this further.
RABIX: VISUAL EDITOR

Airflow plugin to allow the use of Rabix: a visual editor using open standards for workflow definition.
A complete DAG

Huge reductions in length.
< 20

For all DAGs.
Where did the **LINES** go?

**Code Modules.**
SubDags help you to apply the DRY principle to your DAGs.

Duplicate lines are hidden behind abstractions.

**Meta Data Files.**
Configuration values are stored in metadata descriptions of your tasks and DAG.
Common Workflow Language
The Common Workflow Language (CWL) is an open standard for describing analysis workflows and tools...
https://github.com/common-workflow-language/common-workflow-language

Rabix
Rabix Composer: a powerful, open source, graphical editor allowing visual programming in CWL.
https://github.com/rabix/composer
https://rabix.io/
THANKS!

MY NAME IS TRAEY HATCH
I am here because I love Airflow.
You can find me at:

@trejas2
linkedin.com/in/trejas
github.com/trejas