Agenda

01 Introduction
02 Why open source the project?
03 How to prepare for open sourcing
04 Project health
05 Recap
Introduction
<table>
<thead>
<tr>
<th>Open Source @ Uber</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>350+</strong> Repositories on GitHub</td>
</tr>
<tr>
<td><strong>2,000+</strong> Contributors worldwide</td>
</tr>
<tr>
<td><strong>117K+</strong> Stars on GitHub. Average 365 per repository</td>
</tr>
<tr>
<td><strong>5</strong> Community projects: Jaeger, Horovod, Pyro, Apache Hudi, Kepler.gl</td>
</tr>
<tr>
<td><strong>4</strong> 4 of the top open source projects on InfoWorld's awards list</td>
</tr>
</tbody>
</table>
Why open source the project
Why open source the project?

Who are you?
The motives, reasons and arguments may be different for different maintainer categories. Are you an individual maintainer or a company?

What do you expect to achieve?
As an individual maintainer, sharing your work with the community can be rewarding on multiple levels. It can be an opportunity to grow the project beyond your own resources, and it can also be good personal branding.

As a company, open sourcing projects can furthermore support the image as a good open source citizen, it can attract talent, foster collaboration and gain influence.
Why open source the project?

Is it useful for the community?
It may seem like an obvious question, but is the project useful for others?
If the project fulfills a need, solves a problem or simply make developers lives easier, there is a good chance it will benefit from being open sourced.

Do you have the resources to support the project?
An open source project do require attention, also after flipping the switch from private to public on GitHub.
Bug fixes, developer support, handling issues and pull requests etc. are ongoing efforts maintainers have to allocate time and resources for.

What’s the goal of open sourcing the project?
Before open sourcing the project, define why the project should be open sourced, and what the goal is.
Is it a side project, that partially is used to showcase your skills and expertise, the goals may be very different from the goals a company will set, when they open source a project.
How to prepare for open sourcing
How to prepare for open sourcing

Checklist
- Public repository
- License type selected
- README file
- Documentation
- How to contribute
- How to file bugs
- Sample apps
- Tutorials
- Website
- ……
- ……
How to prepare for open sourcing

Choosing a licence
Open source software is free, so no need to think of licenses, right? Not correct!
Licenses are just as important for open source software, as it is for closed source software. Licenses defines the conditions that applies to the software, and what you can do with it, and what you are obligated to do, if you use it.

The Developer Journey
The Developer Journey describes the steps developers take, from they initially find the project, to fully adopting the project in their software.
Looking at the project, and the resources the project provides to help developers try, use and integrate project, is a good exercise.
Choosing a license

In open source software, licenses define rights, obligations, limitations, permissions and more for both maintainers and consumers.
Choosing a license

In open source software, licenses define rights, obligations, limitations, permissions and more for both maintainers and consumers.
# Choosing a license

## Examples:

<table>
<thead>
<tr>
<th>CAN</th>
<th>MIT</th>
<th>Apache 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial Use</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Modify</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Distribute</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Sublicense</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Private Use</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Use Patent Claims</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Place Warranty</td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CANNOT</th>
<th>MIT</th>
<th>Apache 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Liable</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Use Trademark</td>
<td></td>
<td>●</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MUST</th>
<th>MIT</th>
<th>Apache 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include Copyright</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Include License</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>State Changes</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>Use Notice</td>
<td></td>
<td>●</td>
</tr>
</tbody>
</table>

Uber’s Open Source projects a typically licensed under MIT or Apache 2.0.

Source: https://tldrlegal.com
Choosing a license

“Tesla inches toward GPL compliance in low gear: Source code forcibly ejected into public”

https://www.theregister.co.uk/2018/05/21/tesla_inches_toward_gpl_compliance/

https://github.com/teslamotors/linux
The Developer Journey

Developer Journey:
The journey, or the steps you take, from you initially discovering a project to successfully adopting it into your project.
The Developer Journey

1. Explore
2. Getting Started
3. Experience Features
4. Integration

If the developer gets stuck on any of those four steps, there’s a high risk the developer will not adopt a project.
The Developer Journey

Let’s look at some examples of what typically should be provided to make sure the developer is supported throughout the journey.
README File
The Developer Journey

An open source project’s GitHub repository is often one of the first places developers go to learn more about the project.

It’s the project’s landing page.

So, what should be in the README file?
A good approach is to answer the following questions:

- What is it, and why does it matter?
- What are the most important features and how does it work?
- How do I install it?
- How do I use it?
- Can you help me test it?
- Now, how do I deploy it?
Getting Started
Getting started
The Getting Started section should go through the process of doing a minimalistic installation or implementation.

Horovod documentation
Horovod improves the speed, scale, and resource utilization of deep learning training.

Get started
Choose your deep learning framework to learn how to get started with Horovod.

<table>
<thead>
<tr>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>TensorFlow</td>
</tr>
</tbody>
</table>

To use Horovod with TensorFlow on your laptop:

1. Install Open MPI 3.1.2 or 4.0.0, or another MPI implementation.
2. If you've installed TensorFlow from PyPI, make sure that the g++-4.8.5 or g++-4.9 is installed.
   If you've installed TensorFlow from Conda, make sure that the gxx_linux-64 Conda package is installed.
3. Install the Horovod pip package: `pip install horovod`
4. Read Horovod with TensorFlow for best practices and examples.

Or, use Horovod on GPUs, in Spark, Docker, Singularity, or Kubernetes (Kubeflow, MPI Operator, Helm Chart, and FDSL).

<table>
<thead>
<tr>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keras</td>
</tr>
<tr>
<td>PyTorch</td>
</tr>
</tbody>
</table>
The Developer Journey

Getting started
At this point we have provided guidance, which enables developers to get hands on experience with the project.
Tutorials
The Developer Journey

Tutorials
The most common tutorial is Hello World!

It’s super simple, and it walks through the steps from installation to creating some kind of output.

Good tutorials will show, step-by-step, how to do basic things with the project, which can easily be replicated by the developer.
The code is revealed as the tutorial explains the steps. This breaks larger code blocks down to smaller and more chunks, which are easier to comprehend.
The Developer Journey

Tutorials
At this point, the developer has replicated a series of steps, a working application, and a basic understanding of how things work.
Sample Code
The Developer Journey

Sample code
In contrast to the tutorials, sample code doesn’t walk through a typical use case by breaking the code down to small chunks with detailed explanations.

Sample code is complete projects, which shows best practice of implementation of selected features.
The Developer Journey

Sample code
Sample code is typically independent projects on GitHub.
Documentation
The Developer Journey

Documentation
My guess is that a lack of good documentation is a Top 3 reason why developers choose not to adopt a project.

There are many opinions on how good documentation is structured, and what it contains…

… and many of them are right.
Documentation
The reference can look very different, depending on the project type.

PySpark & SQL
A set of Spark specific helper functions for the petastorm dataset

```
petastorm.spark_utils.dataset_as_rdd(dataset_url, spark_session, 
  schema_fields=None, hdfs_driver='libhdfs3')
```

Retrieve a spark rdd for a given petastorm dataset

- **dataset_url** – A string for the dataset url (e.g. hdfs://path/to/dataset)
- **spark_session** – A spark session
- **schema_fields** – list of unschema fields to subset, or None to read all fields.
- **hdfs_driver** – A string denoting the hdfs driver to use (if using a dataset on hdfs). Current choices are libhdfs (java through JNI)
  or libhdfs3 (C++)

Returns:
A rdd of dictionary records from the dataset

Row queries
Predicates for petastorm

```
class petastorm.predicates.PredicateBase
  Base class for row predicates

  get_fields()
```

Train
This command lets you train a model from your data. You can call it with:

```
ludwig train [options]
```

or with
```
python -m ludwig.train [options]
```

from within Ludwig's main directory.

These are the available arguments:

- **usage**: ludwig train [options]
  This script trains a model.

  **optional arguments:**
  -h, --help show this help message and exit
  --output.directory OUTPUT_DIRECTORY directory that contains the results
  --experiment.name EXPERIMENT_NAME experiment name
  --model.name MODEL_NAME name for the model
  --data.csv DATA_CSV input data CSV file. If it has a split column, it will be used for splitting (0: train, 1: validation, 2: test), otherwise the dataset will be randomly split
Support
Support
Is there any way to get in contact with the project maintainer?

- GitHub
- Slack
- Gitter
- Email
- Etc.
Project Health
Project Health

Metrics

Metrics about community engagements can be used to assess the overall health of the project.
Examples of relevant questions to seek answers to from metrics

- How does the project do?
- What is the project’s value?
- How big is the project community?
- Should we continue developing it?
- How can we improve the project?
Project Health

Working with Uber’s Open Source project metrics has shown a clear trend.

All projects cannot be treated the same way. We have to ask different questions for projects that are in different stages of their lives.

Projects have different maturity levels, and if we use the same metrics to assess their success, we will typically not get the right picture of less mature projects.
Project Health

New Project
- Stars
- Issues
- Clones

Mature Project
- Pull Request
- Contributors
- Engagement
While popularity metrics like stars may not have much value for mature projects, it’s a good indicator of interest in newer projects.

Another good indicator for newer projects is how many times developers have tested the project - measured in clones, installs etc.
Project Health

Projects in later stages of their lives are expected to have more contributions, and a lot of the metrics we are interested are centered around contributions.

We look not only the quantity of contributions, but also the demographics and level of the contributor engagement.
Project Health

In the early stages of an open source project’s life, we want to look for *intend* to adopt, engage and contribute.

In later stages we want to see *action*. 
Project Health

The path to reporting metrics

- Strategy
- Analytics
- Customization
- Reporting
Project Health

**Strategy**
The goals defined for the project. Examples could be:

- Collaboration
- Attract talent
- Gain influence
- Give back
- Foster OSS participation
Project Health

Analysis
Define the questions you want answered.

- Where are contributors coming from?
- How much engagement do my projects get?
- How many core, regular, and casual contributors do my projects have?
- How quickly are we handling external contributions?
- What is the company’s OSS footprint?
- What is contributors path in our projects?
Project Health

Customization
Identifying the sources of insights

- +30 data sources supported
- Predefined and customizable dashboards
- Contributors identity information management
- Rest API for data consumption
- 100% free, open source software
Project Health

Reporting
Where contributors are coming from?
Project Health

Reporting
How much engagement do my projects get?
Project Health

Reporting

How much engagement do my projects get?
Project Health

Reporting

How many core, regular and casual contributors do I have?
Reporting

How fast I am dealing with external contributions?
Project Health

Reporting
What is contributors path in our projects?

Users
External people asking questions

Contributors
External people answering questions
External people submitting patches

Maintainers
External people committing code
Metrics can reveal areas where we may need to improve, where there’s opportunities and the overall health of the community around the project.

Metrics can also be used to answer business questions.
Recap
Recap

- Before open sourcing a project, define why you want to do it, what the goal is and if you have the resources to support the project as open sourced.

- Make sure you are providing the resources developers need to try, evaluate and adopt the project.

- Monitor the project’s health, and evaluate against your goals.
Thank you

Carsten Jacobsen  
Open Source Developer Advocate, Uber  
carstenj@uber.com  
@CarstenJacobsen

opensource.uber.com