

## Opening Remarks

information on the conf - where to find quiet room, etc.

----

Keynote: Django the Dinosaur: A Story About A Web Framework.

-----

Mixing reliability with Celery for delicious async tasks

celerty is a distributed system.

there can be task loss between the web process and broker.

If there's a connection error, a task might never be queued

Or the task commit can fail, too

-----

Postgres Performance: From Slow to Pro

RAM is a big issue

you can use pgtune to get rec settings

not using statement timeouts can cause issues, too

keep an eye on IOPS

remember to do vacuums

a single record result should return in 1ms

Nothing should take more than 5 seconds to return

-----

Empathetic testing: Developing with compassion and humility.

switched to ipmp and ppmp in numpy and their test stopped passing

git blame to find out who wrote these tests

They needed to upgrade Python

When they fix the app the test breaks and vice versa

Problem is Implementation Bias - it's testing the way it was built, not the way it works.

Need to write better tests

things change so you have to treat your code as discardable

pyquery can help us see the data better to understand how to write the tests

a mock was obscuring an issue with the code

-----

Powering Energy Storage Beyond Excel

When energy is created, but doesn't need to be used right now - how is it stored? (eg from green sources)

needs a 20-25 year lifecycle for the batteries

they were working with an energy company that put everything into Excel, but there are problems:

- no scaling
- no reliable versioning
- no protection of business know-how
- no fault tolerance
- no field permissions
- no tests

So they built a Django solution with Postgres and Django REST-framework

how they transitioned

#### 1. Eval

- gather **all** the formulas in the spreadsheet. Document them - understand what they mean. Add examples to make sure you understand it.

#### 2. Prototype

#### 3. Review

check with jupyter

#### 4. Build

consider using a cookiecutter template

#### 5. Switch

Don't create a web version of Excel

-----

Don't Buy the "A.I." Hype

First "AI" was Eliza

hype cycle chart is similar to dunning kruger effect chart

-----

There's More to Open Source than Code

make sure that people know they can contribute in other ways than code - for example

- documentation
- community help
- blogging about software or how to
- writing bug reports, helping a project manage their bug reports and issues
- help on forums
- volunteer at events

-----

BDD To The Bone: Acceptance Testing with Behave and Selenium

Sometimes the requirements can be like a game of telephone.

Speaker warns against non-testable requirements (could substitute with non-measurable)

Gherkin - a language used to express requirements (see slide to see the format of a requirement)

Can we take a Gherkin requirement and run it? Behave

Behave can map Gherkin to runnable code

It uses decorators with the Gherkin keywords

Using Gherkin gives a shared vocabulary to the requirements. Non-devs can write requirements and it can easily be put into the Behave tests.

BDD - behavior driven tests

-----

The Programmer's Imagination

starts off talking about dinosaurs and neanderthal art to show how the art changes based on culture (not just science)

think about the art you're creating - is it going to look bad in a few decades?

-----

An approach to lightweight tenancy management using Django Rest Framework

many tenants, one server

in multitenant the isolation is at the applicaiton layer

lightweight - we want to share the database and application instance

-----

Introduction to GitHub Actions: Understanding Key Terms and Building Your First GitHub Action

Workflow

.github/workflows/ dir

it runs one or more jobs

Job - series of steps inside a workflow

by default, jobs run in parallel

Contexts is how we access information after the workflow