

There's a Snake in the Birdhouse!

Building a Python Culture at Vrbo


Presenter: Mason Egger

[@masonegger](https://twitter.com/masonegger)

twitch.tv/codingwithmason



Introduction

- Developer Advocate @DigitalOcean 
- Previously a Site Reliability Engineer at Vrbo
- No Python at Vrbo



- Today, I'm going to tell you how I lead the effort to build a Python Culture at Vrbo

Who is this talk for?

- Anyone who wants to
 - Use Python at their job but currently is unable
 - Wants to build a stronger Python community at their job
 - Wants to learn how to introduce and evangelize *any* technology, not just Python
- Hopefully, you'll be able to learn from my successes and mistakes to bring Python to your job.

Prologue

- Engineers at Vrbo deployed applications on an internal PaaS
- Project templates existed for supported languages (Java, Node)
- Could deploy “off road” apps, but no official support from Platform teams

Discovery Meeting

- Met with various teams (Data Science, Machine Learning, API teams) to gauge interest
- **Don't** build something no one wants
- **Don't** develop in isolation or avoid other's opinions
- Result: Build a cookiecutter that creates a default Flask app for the platform including static analysis, unit tests, project documentation, logging, metrics, etc.

Building the Archetype

- **Do** Create a meaningful set of expectations and use cases
 - A base application for “standard” Python and a Data Science specific application with miniconda.
- **Do** Provide an Out-of-Box experience with sensible defaults

Building the Archetype

- **Do** Create good documentation and rationale of decisions
 - Document how to use the archetype, all known bugs and issues, rationale behind choosing certain technologies, and the development journey
- **Don't** Trying to solve every Python use case with one tool
- **Do** Use a few internal or proprietary tools as possible. This makes potentially open sourcing easier later

Internal PaaS 2.0

- Version 2 of Vrbo's internal PaaS was being built concurrently with Python support
- The Python Archetype would only support version 2
- This was used to entice people to migrate to the newer version
- This was good because I could take advantage of the new features
- This was bad because I became dependent on the new PaaS development timeline
- This became both a **success** and **failure**

Testing

- **Do** Find a group of empowered beta testers
 - They'll help you evangelize later
- **Do** Find “first customers” who will actually attempt to use the application in production.
- **Don't** Become upset by negative feedback or bug reports

Evangelism Do's

- **Do** Demonstrate the flexibility of Python
- **Do** Rewrite existing applications that are “troublesome” or “complex” in Python
- **Do** Write new applications or features in Python
- **Do** Be the “temporary” solution
 - “There is no discernable difference between a proof of concept and production software.” – my ex manager

Evangelism Don'ts

- **Don't** say "That language sucks. You should use Python"
- **Don't** Try to use Python to solve problems Python doesn't handle well
- **Don't** Be overbearing. If someone isn't interested that's ok.

Results

- Within the first month 80 applications were deployed using the new archetype in production
- Other people within the company used the development journey documentation to start building support for Golang

Conclusion

- Thank you for tuning in today
- Feel free to tweet me if you have questions @masonegger
- Tune in to my live streams Tuesdays at 1:00 CST <https://twitch.tv/codingwithmason>