Controlling the Chaos

Establishing Best Practices for Python Scripting

SCANA®
POWER FOR LIVING

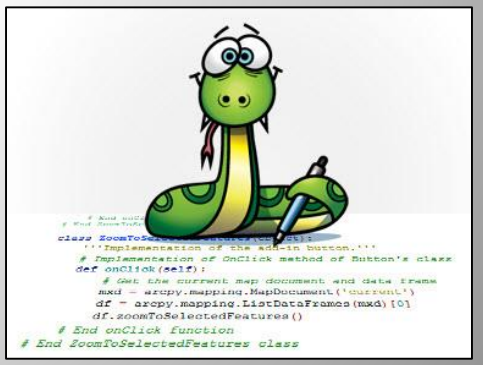
GIS at SCANA

- SCANA is a mid-size utility in South Carolina
- 5.5 GIS developers to support:
  - ArcGIS Desktop
  - GIS data availability
  - ~60 viewers (*Silverlight and JavaScript*)
  - ~90 batch scripts (*VB, C#, and Python*)

SCANA®
POWER FOR LIVING

What's the Python Problem?

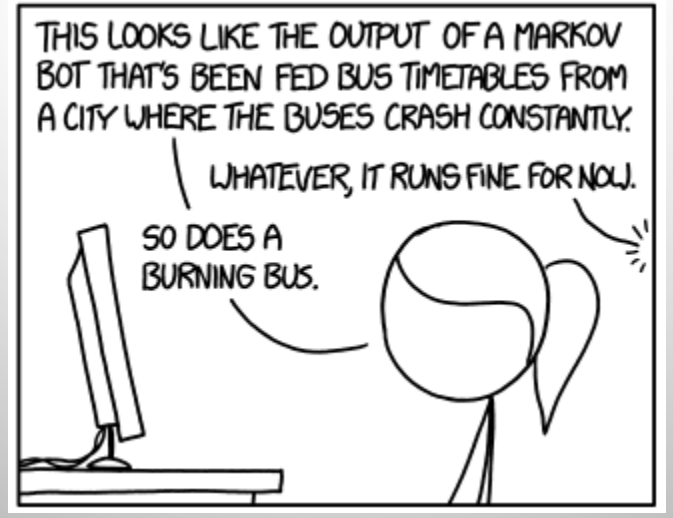
- **Impending Python 3 upgrade** January 2019 (Utility Network and ArcGIS Pro)
- Inadequate documentation
- Scripts running on local machines
- Different versions in different locations
- Unclear ownership
- Inconsistent arcpy functions
- Variation in script design



SCANA® POWER FOR LIVING

How Did We Get Here, Anyway?

- Chaos is not an overnight development
  - Years of different developers with different styles
  - Few standards and little oversight
  - Prioritizing delivery over all else



xkcd.com/1695/

SCANA® POWER FOR LIVING

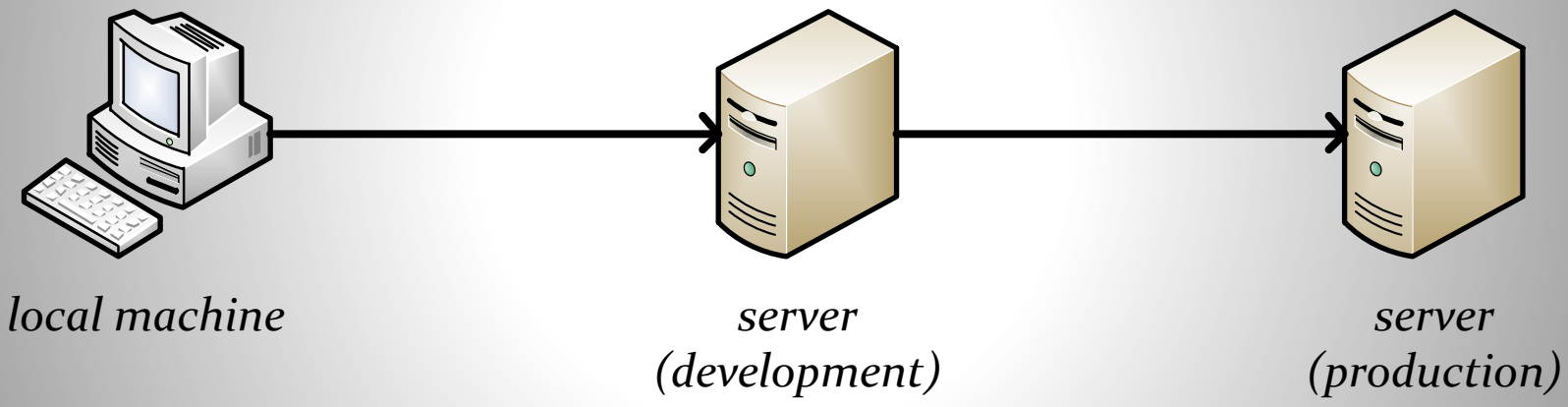
Solution: Best Practice Guidelines

- Defined “best practices” for all Python projects
  - Template
  - Documentation
  - Defined workflows
- New development meets standards
- Old applications
  - Upgraded to best practices as time permits
  - Include best practices with other upgrades

SCANA®
POWER FOR LIVING

Deployment Guidelines

- Testing/QA happens before moving to production



Reduces production bugs, hotfixes, and angry customers

SCANA® POWER FOR LIVING

Documentation

- What does it do, and why does it do it?
  - Customer Requirements
  - Functional Requirements
  - Workflow or modeling design
- Included as part of the repository and deployment packages

Important application details can be found quickly

SCANA® POWER FOR LIVING

QA and Peer Review

- Project is checked for quality and compliance
  - Code review
  - Documentation review
- Peer review provides broader expertise

Enables sanity checks, knowledge transfer, cross-training

SCANA® POWER FOR LIVING

Version Control

- Central repository for all code changes
- Branch from default for any development/updates
- Deployments to production come from specified repository branch

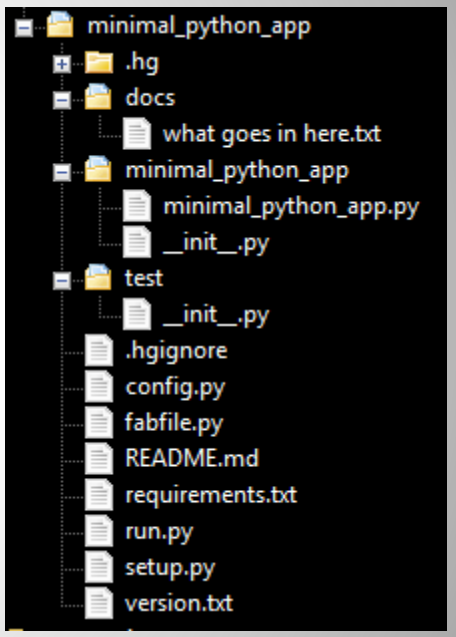
No more lost code
One-stop source for "current" version

SCANA®
POWER FOR LIVING

“Minimal” Python Template

- Template project to start a project
- Uniform structure

```
my_app/
    run.py
    my_app/
        __init__.py
        businesslogic.py
```



Common format for scripts; Headstart on coding

SCANA® POWER FOR LIVING

Custom Python Module

- Developed `sgist` (**S**CANA **GIS** **T**ools) module
  - *Inspiration for our* `sjst` *(SCANA JavaScript Tools) library*
- Standardized methods for commonly used functions
  - Write a log
  - Send email on interesting events
  - Access configuration file
  - Parse input parameters
  - Copying data from SDE to GDB (or `in_memory`)

Developers don't need to reinvent the wheel

SCANA® POWER FOR LIVING

Concluding Thoughts

- Standard workflows reduce confusion, improve efficiency
- Cafeteria style:
  practices that work for your team and applications
- Adapt as new problems arise
- **Ensure team buy-in throughout**

Erica Pfister-Altschul
ep44848@scana.com

SCANA®
POWER FOR LIVING