

Software Engineering: A Jr. Developer's Guide July 2018

- Now: QA Specialist
- Then: Accountant & uni instructor w/ an MBA
- https://github.com/kojoidrissa/pyohio_2018

The Issue

- Career changers seek internships or "Jr. Developer" jobs
- Jr. Developer == ???
- “When am I ready?”

The Issue

- MOAR Programming!
- You don't become an author or journalist by memorizing a dictionary.

Secondary Issue

- How do you mentor a new developer?
- I have a talk on that...we'll see if it gets accepted anywhere.

Programming

- Individuals
- Author
- Less robust

Software Engineering

- Teams
- Others
- More robust

Interjection(!): “Real” coders

- If your code runs & does what you want it to do, you’re a “real” coder.
- Choose your style
- Lone Genius Myth is Myth

Functional Details

- Version Control
- Documentation
- Testing
- Dependency Management/Deployment
- Development Environment

Version Control

- Why?
- Protect yourself from mistakes
- Enable yourself to try new things
- Learn to work w/ a team

Version Control (Git)

- When?
- 10 lines or more
- working code you're scared to break

Version Control (Git)

- What/How?
- Solo usage
- Basic feature branch/merge workflow

Version Control (Git)

- Reference
 - Pro Git Chapters 1-3
 - <https://git-scm.com/book/en/v2>
 - Free!

Documentation

- In Code vs. *About* Code?
- Start with In Code

(!) Self-Documenting Code

- Kinda, not really
- Naming is hard
- Start to think about it now

Documentation

- Refine your thinking
- Help your future self
- Avoid technical debt
- Help your teammates help you

Documentation

- When?
- Writing code using a new technique
- Writing functions, especially if they call other functions

Documentation: How

- Docstrings for functions (minimums)
 - What the function does
 - What is it's input (include data type)
 - What is it's output (include data type)

Documentation

- Learn a markup language
- ReStructured Text or Markdown
- Learn about Python documentation standards
- <https://devguide.python.org/docquality/>

Testing & TDD

So.

Many.

Reasons.

Why Testing: The Big Picture

- Tests -> Automated tests
- Automated Tests -> CI
- CI -> CD (Continuous Delivery)
- CD -> CD (Continuous Deployment)

Testing

- Why should YOU write tests?
- Protection from future changes breaking things
- Assurance past fixes still work

Testing & TDD

- Test Driven Development
- Write tests BEFORE you write code
- TDD: forces you to think before you type

Testing & TDD

- When?
- As soon as possible
- Art + Science: experience helps

Testing & TDD

- How?
- doctests & unittest
- Third-party: pytest

Dependency Mgmt & Deployment

- Virtual Env
- Containers
- Virtual Machines (VMs)

Other

- Development Environment
- CLI vs. GUI: can you navigate?
- Terminal multiplexer
 - screen, tmux, byobu
- A shell language

Demo Time!

- Git log from PyVideo
- Docs & Tests: My project
- My PyOhio 2017 talk
 - Dep. Mgmt
 - Dev. Environment

Finding New Developers

To:

- Engineering Managers
- Dev Leads
- Others trying to hire developers...

Next Steps for You

- Last 2 talks today in Cartoon 2: testing
- Sunday 2:15 Cartoon 2: Containers talk
- “A Hands-On Guide to Teaching...”

Next Steps for Me

- Working on a tutorial version of this talk
- Is there anything I overlooked?

Questions/Comments?

- @Transition on Twitter
- kojoidrissa.com
- https://github.com/kojoidrissa/pyohio_2018