

# Open Source Technologies in Clinical Research

Bruce Wienckowski (GSK)

Michael S. Rimler (GSK)

Mike Stackhouse (atorus)

TT10 – 2020 PHUSE US Connect

# Agenda



## Open Source Technologies in Clinical Research (OSTCR)

---

- What is OSTCR?
- What is Open Source?
- Implication of FOSS (Free and Open Source Software) in a Regulatory Environment
  - Qualification, Quality Assurance, and Validation
  - Assessing Risk
  - Package Management
- The Big Players: Python and R
  - R Versus Python
  - Regulatory Environments
- Conclusion

# What is OSTCR?

## Open Source Technologies in Clinical Research (OSTCR)

---



*Open Source Technologies in Clinical Research (OSTCR) is a new project under the PHUSE working group Emerging Trends and Technologies. Our goal is to broaden the acceptance and general level of comfort with these technologies in the industry to assist in increasing their level of adoption. Our approach is to demonstrate value through use cases, while providing implementation and validation guidance along with best practices.*

# What is Open Source?



## Open Source Technologies in Clinical Research (OSTCR)

---

- What is proprietary software?
  - Non-free, typically has licensed limitation on how it can be used
  - Intellectual property rights retained by an individual or organization
  - Copyright of the code, sometimes patents
  - Source code typically protected and not accessible to the consumer
- What is Open Source software?
  - Includes permission to use the source code, design documents, or content of the product
  - Depending on licensing terms, may be able to download, modify, and publish new versions of the software back to community
  - **Open Source != Free**
    - Free and Open-Source Software (FOSS) refers to the liberty of the code, not the cost

A large, glossy, orange abstract shape, resembling a teardrop or a drop, is located on the left side of the slide. It has a gradient from a lighter orange at the top to a darker orange at the bottom, with a white highlight on its upper left edge, giving it a three-dimensional appearance.

# The Implication of FOSS in a Regulatory Environment

# Qualification, Quality Assurance, and Validation



## The Implication of FOSS in a Regulatory Environment

---

### – Qualification

- FDA defines qualification as establishing confidence<sup>1</sup>:
  - Installation: Was the software installed correctly and according to developer's instruction?
  - Operational: Can the software operate consistently through established limits and tolerances
  - Process Performance: Is the process effective and reproducible?
  - Product Performance: Through testing, does the product meet all release requirements for functionality and safety?

### – Quality Assurance

- Assurance that something conforms to establish technical requirements
- How does the developer ensure the product meets performance requirements and adheres to standards and procedures?
- Process by which maintains some degree of confidence in data integrity and accuracy throughout product lifecycle
- Individually and collectively, product meets expectations

### – Validation

*“...establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specifications and quality attributes.”<sup>1</sup>*

# What is Open Source?

The Implication of FOSS in a Regulatory Environment

---



*What is “truth”?*

# Assessing Risk



## The Implication of FOSS in a Regulatory Environment

---

- In qualified systems, we care about:
  - Established Vendor
  - Quality Management of Documentation
  - Standard and “Trusted” Code Resources
  - Proof of Source Code Control
- Need Installation and Operational Qualification to demonstrate that there exists:
  - Assurance of source code management of version releases of the base source code
  - Proper testing in place which the requirements are properly addressed and issues identified and added to a backlog
  - Highly reputable resources who create source code and packages
  - Provide proper validation and verification of the use of the coding language and the associated packages

# Package Management



## The Implication of FOSS in a Regulatory Environment

---

- It's up to your company to determine:
  - How will you define “trusted resources”
    - Trusted resources could be external vendors or individual package developers
  - How do you control the intake of “trusted” packages and maintain the dependencies?
  - What will be the cadence of library/package releases?
  - How do you track and collect incidents for these packages?

# Package Management



## Example – Package Management Inventory List (PMIL)

---

- Environment (Development, Test, Validation, Production)
- Build (2019-01, 2019-02, 2020-01, 2020-02, etc. – The delivery cycle)
- Item Identification
- Package Name
- Package Description
- Package Version
- Vendor / Author
- Owner – The Requester(s)
- Component ID – Company Change Management ID
- Parent Components (Item ID, Ver, Start Date and End Date, Current Status)
- Child Components (Item ID, Ver, Start Date and End Date, Current Status)
- Status (Active, Requested, Testing, Retired, etc.)

A large, glossy, orange abstract shape, resembling a teardrop or a drop, is positioned on the left side of the slide. It has a gradient from a lighter orange at the top to a darker orange at the bottom, with a white highlight on its upper left edge.

# The Big Players: Python and R

## The Big Players: Python and R

---

- Originally created in 1989, first release in 1991
- High-level, general-purpose programming language
- Versatile, applicable and used in a wide range of different domains
- Strong data science toolkit:
  - NumPy
  - Pandas
  - SciPy Ecosystem
  - Scikit-Learn
- Curated distributions from organizations like Anaconda
  - Additionally provide package management tools to maintain dependencies

## The Big Players: Python and R

---

- Originally branched off the statistical programming language S
- Developed by the R Development Core Team as an official part of the Free Software Foundation's GNU Project
- Major component contributing to the popularity is the presence of RStudio
  - Provides the popular RStudio IDE
  - Maintains and develops packages in the Tidyverse
    - Tidyverse is an opinionated collection of R packages designed around data science
    - Shares an underlying design philosophy, grammar, and data structures
  - Powerful analytic tools available like Shiny and RMarkdown

# R Versus Python

## The Big Players: Python and R

---



*“Generally, there are a lot of people who talk about R versus Python like it’s a war that either R or Python is going to win. I think that is not helpful because **it is not actually a battle.** These things exist independently and are both awesome in different ways.”*

*Hadley Wickham, Quartz, 2019*

# Regulatory Environments



## The Big Players: Python and R

---

*FDA does not require use of any specific software for statistical analyses, and statistical software is not explicitly discussed in Title 21 of the Code of Federal Regulations [e.g., in 21CFR part 11]. However, the software package(s) used for statistical analyses should be fully documented in the submission, including version and build identification.*

*As noted in the FDA guidance, E9 Statistical Principles for Clinical Trials (available at <http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/default.htm>), “The computer software used for data management and statistical analysis should be reliable, and documentation of appropriate software testing procedures should be available.” Sponsors are encouraged to consult with FDA review teams and especially with FDA statisticians regarding the choice and suitability of statistical software packages at an early stage in the product development process.*

*Statistical Software Clarifying Statement, FDA, 2015*

# Regulatory Environments



## The Big Players: Python and R

---

- A large amount of work and consideration has already been done
  - R Foundation for Statistical Computing released guidance for the use of R in regulated clinical trial environments<sup>27</sup>
  - R Validation Hub white paper: A Risk-Based Approach for Assessing R Package Accuracy Within a Validated Infrastructure<sup>28</sup>

# Questions?

---



**Bruce Wienckowski**

GlaxoSmithKline

[bruce.d.wienckowski@gsk.com](mailto:bruce.d.wienckowski@gsk.com)

**Michael S. Rimler**

GlaxoSmithKline

[michael.s.rimler@gsk.com](mailto:michael.s.rimler@gsk.com)

**Mike Stackhouse**

ator

[Mike.Stackhouse@atorusresearch.com](mailto:Mike.Stackhouse@atorusresearch.com)

# Questions / Statements



A large, glossy, orange teardrop-shaped graphic is located on the left side of the slide. It has a gradient from a lighter orange at the top to a darker orange at the bottom, with a bright highlight along its upper edge, giving it a three-dimensional appearance.

**Additional Slides**

# References



1. Office of Regulatory Affairs. (n.d.). Glossary of Computer System Software Development Terminology (8/95). Retrieved from <https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/inspection-guides/glossary-computer-system-software-development-terminology-895>
2. R Programming Software: ValidR. (n.d.). Retrieved from <https://www.mango-solutions.com/products/validr/>
3. (n.d.). Retrieved from <https://mran.microsoft.com/open>
4. Open Source Technologies in Clinical Research. (n.d.). Retrieved from [https://www.phusewiki.org/wiki/index.php?title=Open\\_Source\\_Technologies\\_in\\_Clinical\\_Research](https://www.phusewiki.org/wiki/index.php?title=Open_Source_Technologies_in_Clinical_Research)
5. Sharwood, S. (2018, July 13). Python creator Guido van Rossum sys.exit()s as language overlord. Retrieved from [https://www.theregister.co.uk/2018/07/13/python\\_creator\\_guido\\_van\\_rossum\\_quits/](https://www.theregister.co.uk/2018/07/13/python_creator_guido_van_rossum_quits/)
6. Musienko, Y. (2020, January 13). Outstanding Devs: Top Programming Languages to Learn in 2020 - Merehead 4539. Retrieved from <https://merehead.com/blog/most-popular-programming-languages-2020/>
7. NumPy. (n.d.). Retrieved from <https://numpy.org/>
8. Python Data Analysis Library. (n.d.). Retrieved from <https://pandas.pydata.org/>
9. Scientific computing tools for Python. (n.d.). Retrieved from <https://www.scipy.org/about.html>

# References



10. ScitKit Learn. (n.d.). Retrieved from <https://scikit-learn.org/stable/about.html>
11. RStudio: Open source & professional software for data science teams. (2020, January 7). Retrieved from <https://rstudio.com/>
12. Tidyverse. (n.d.). Retrieved from <https://www.tidyverse.org/>
13. Shiny. (n.d.). Retrieved from <https://shiny.rstudio.com/>
14. R Markdown. (n.d.). Retrieved from <https://rmarkdown.rstudio.com/>
15. Kopf, D. (2019, August 19). What's next for the popular programming language R? Retrieved from <https://qz.com/1661487/hadley-wickham-on-the-future-of-r-python-and-the-tidyverse/>
16. Admin. (2017, January 13). How Python Powers Dropbox. Retrieved from <https://www.pythonpeople.nl/how-python-powers-dropbox/>
17. API Documentation: TensorFlow Core r2.1. (n.d.). Retrieved from [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs)
18. Spark Overview. (n.d.). Retrieved from <https://spark.apache.org/docs/2.2.0/index.html>
19. Allaire, J. J. (n.d.). TensorFlow for R. Retrieved from <https://tensorflow.rstudio.com/>
20. (n.d.). Retrieved from <https://spark.apache.org/docs/2.2.0/api/R/index.html>

# References



21. sparklyr. (n.d.). Retrieved from <https://spark.rstudio.com/>
22. McKinney, W. (2018, April 19). Announcing Ursa Labs: an innovation lab for open source data science. Retrieved from <https://wesmckinney.com/blog/announcing-ursalabs/>
23. Interface to 'Python'. (n.d.). Retrieved from <https://rstudio.github.io/reticulate/>
24. U.S Food & Drug Administration. (2015, May 6). Statistical Software Clarifying Statement. Retrieved from <https://www.fda.gov/media/109552/download>
25. Bowsher, P. (2019, June 16). R for Drug Development: An Introduction to the Tidyverse, Shiny & R Markdown. Retrieved from <https://colorado.rstudio.com/rsc/content/2519/#62>
26. Smith, D. (2017, June 29). How R is used by the FDA for regulatory compliance. Retrieved from <https://blog.revolutionanalytics.com/2017/06/r-fda.html>
27. Schwartz, M., Harrel, F., Rossini, A., & Francis, I. (2018, March 25). R: Regulatory Compliance and Validation Issues A Guidance Document for the Use of R in Regulated Clinical Trial Environments. Retrieved from <https://www.r-project.org/doc/R-FDA.pdf>
28. R Validation Hub. (n.d.). Retrieved from <https://www.pharmar.org/about/>