



The poster features a dark blue background with a light blue geometric pattern of lines and dots. At the top, three logos are displayed: Brookhaven National Laboratory, Jefferson Lab, and HEP Software Foundation. The main title 'SOFTWARE & COMPUTING' is in large white capital letters, with 'round table' in a white script font below it. The event details 'Best Practices • March 2, 11:00 a.m. (EST) / 5:00 p.m. (CET)' are in green. Three circular portraits of the speakers are shown, each with their name and affiliation below. The topics for each speaker are listed at the bottom.

BROOKHAVEN
NATIONAL LABORATORY

Jefferson Lab

HSF
HEP Software Foundation

SOFTWARE & COMPUTING

round table

Best Practices • March 2, 11:00 a.m. (EST) / 5:00 p.m. (CET)

Henry Schreiner
IRIS-HEP
CMake: Best practices

Carina Haupt
German Aerospace Center
Good practices for sustainable software development

Ben Morgan
University of Warwick
Organizing a (small) software project

These are the live notes for the [Software & Computing Round Table on March 2, 2021](#). This document will be used in advance to give the discussions on “**Best Practices for Software Design**” structure and focus, as well as during the meeting itself to moderate and record the discussion and gather input from all participants.

Discussion

[Good practices for sustainable software development](#)

Speaker: Carina Haupt

Questions

1. Adam Carpenter (Jefferson Lab): Is there administrative enforcement for developers to meet these recommendations?
2. Ben Morgan (University of Warwick): Are there any particular guidelines/recommendations that projects are resistant to (for whatever reason) or struggle to adopt?

[CMake: Best practices](#)

Speaker: Henry Schreiner

Questions

3. Nathan Brei (JLab): Best practices for using pybind11 in a CMake project?

Organizing a (small) software project

Speaker: Ben Morgan

Questions

4. Nathan Brei (JLab): Thoughts on the Doxygen->Breathe->Sphinx->ReadTheDocs pipeline that projects seem to be adopting recently? Looks beautiful but unclear to me if it is worth the complexity and brittleness
 - a. Ben Morgan (University of Warwick) It's been on my "take a look" list for a long time, so afraid I don't have direct experience/thoughts. RTDs does have quite a few advantages over Doxygen (easier to author, though Doxygen does support Markdown), so would be nice if Sphinx worked easily for C++!
 - i. Thanks! I've heard conflicting things from colleagues, haven't tried it out either
 - ii. I feel like there's some weird subtle human factors at play which make RTD "just better" in some sense than a comparable doxygen or ad hoc github pages. Can't put my finger on what it is, but RTD gives me much more confidence as a user that the project is, indeed, documented and understandable. Any idea why that might be?
5. **Nathan Brei (JLab): C++ directory organization holy war: E.g. Does one separate the headers from the source or not? Do directories align with separate build artifacts or not? Where do test cases go? Has there been any convergence on this issue?**