Software in modern science is comparable to any other kind of experimental apparatus. It must be checked and validated to assure its correctness. Software publications are subject to neither QA nor peer review. Scientists typically develop their own software spending typically 30% of their time. It must be checked and validated to assure its correctness. Although it is common to relax the exact version requirement of software components, and permit some variability, each departure from the coherent set should be the object of a revalidation or reassessment of the application. Reciprocally, users should be careful to entrust results comparing with validated results. For this purpose applications should include test suites with validated results.

Ensuring that correct and reproducible results are produced is a matter of scientific integrity. As computational scientists, and research enablers, we must provide a reliable way-to-use software environment which meets the principles of scientific integrity.

**Software Environment Properties**

**Reproducibility:** To be able to reproduce historical software environments. Software environments must be archived or maintained for some years after newer versions have become available.

**Coherency:** An application is normally designed and deployed to operate with a pre-defined set of software components of a specific version. These constitute the application’s coherent software stack. Although it is common to relax the exact version requirement of all components and permit some variability, each departure from the coherent set should be the object of revalidation or reassessment of the application.

**Stability:** An application can be said to be stable if it is able to produce consistent results when its coherent software environment is augmented with foreign components. In practice, an application’s stability can only be guaranteed if it is free of excess components in its coherent software environment. That is, an application’s coherent environment must be limited to its coherent software stack.

**Correctness:** An application must run pre-defined problems to verify it produces expected results by comparing with validated results. For this purpose applications should include test suites with validated results. Reciprocally, users should be careful to entrust results only for problems that lie within the scope of an application’s tested boundaries. Any extension of an application’s problem solving spectrum requires formal validation.

**Consistency:** Application produce same results when using same software components.

**Software Stack vs Software Environment.**

Traditional Linux Environment Modules, LMOD or Easybuild can’t minimize software environment related misuse risks.

**Software Stack Time Machine:**

- Go back in time and recall exactly the same environment used to run the original simulations.

**References**

1. David B. Resnik, Scientific Research and the Public Trust. Science And Engineering Ethics, Volume 17, September 2011, pp 399-409

**Conclusion.**

We reviewed the existing software environment management tools in use and concluded they are insufficient to mitigate misuse risks. So we developed a software stack management tool through a combination of rules and tcl scripts that extend the existing environment modules tools. This method can implement all properties necessary to minimize software environment related misuse risks.

---

**Scientific Integrity**

**Facts:**

- Software stack in modern science is comparable to any other kind of experimental apparatus
- It must be checked and validated to assure its correctness
- Scientific software publications are subject to neither QA nor peer review
- Scientists typically develop their own software spending typically 30% of their time
- 90% or more of them are self-taught, lacking the knowledge of software development practices

In my opinion, scientific integrity means fully disclosing all potential areas of bias, curtailing blatant scientific misconduct, and our mandate, as scientists, to ensue quality science is published in our journals.**

**Software Stack vs Software Environment.**

Enhanced Software Management Environment based on Linux Modules aimed to enforce scientific integrity in complex software environments

J.Naranjo, B. Marchand, M. Al-Barwani, HPC Center NYU Abu Dhabi