

# Methodology & Demonstration of Git-Based Configuration Control in PRA

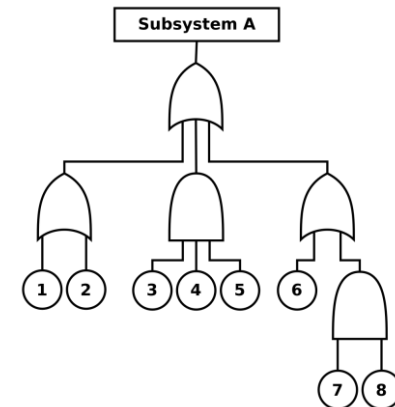
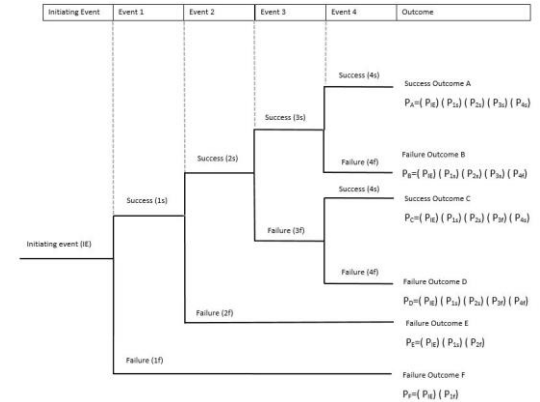
PSAM16

June 29<sup>th</sup>, 2022

Mostafa Hamza  
mmhamza@ncsu.edu

# Background

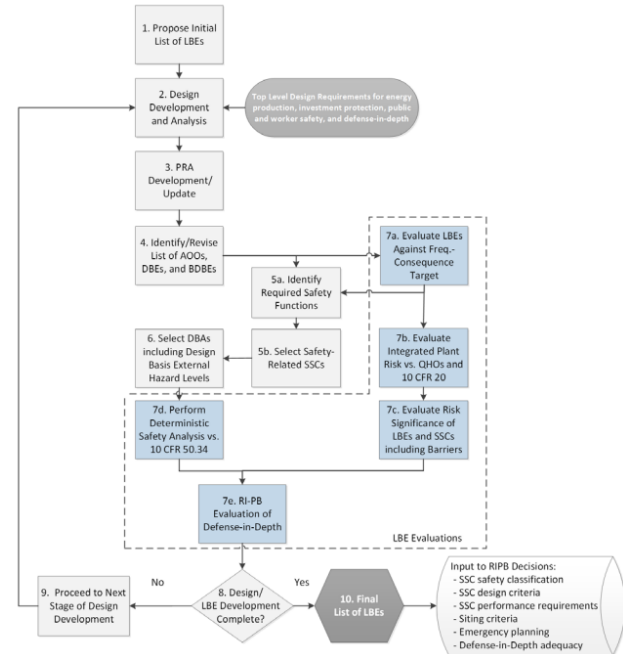
- Probabilistic Risk Assessment (PRA) quantify the risk associated with any large-scale project by answering three questions (Kaplan & Garrick)
  - **What could go wrong?**
  - **How likely is it?**
  - **What are the consequences?**
- Fault Trees (FTs) and Event Trees (ETs) help answer the first two questions.
- For small systems, building FTs and ETs are a straightforward task, however, as the system grows larger, FTs and ETs get more and more complicated.



[https://en.wikipedia.org/wiki/Event\\_tree\\_analysis](https://en.wikipedia.org/wiki/Event_tree_analysis)  
[https://en.wikipedia.org/wiki/Fault\\_tree\\_analysis](https://en.wikipedia.org/wiki/Fault_tree_analysis)

# Motivation (1/2)

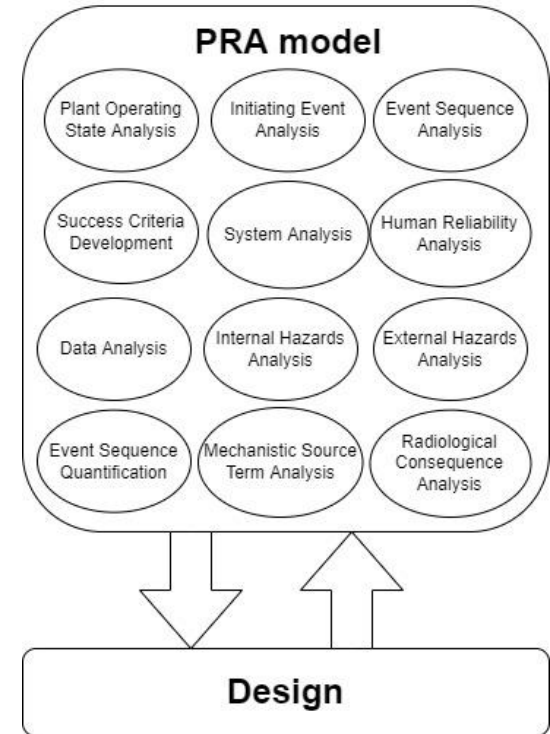
- According to the ASME/ANS standard for advanced non-LWR, a configuration control program is required from PRA to track changes to the plant design and their impact on the risk profile of the design.
- According to the Licensing Modernization Project (LMP), PRA can be used to support the design and licensing of non-LWRs.
- This requires iterative PRA models that model different stages of the design and inform design updates in later stages of design.
- Even though, the configuration control requirements does not require tracking changes to the PRA model, to facilitate the iterative process, tracking changes in the PRA model becomes a necessity.



Moe, W. L. (2019). *NEI 18-04 Risk-Informed Performance-Based Technology Inclusive Guidance for Advanced Reactor Licensing Basis Development* (INL/EXT-19-55375-Rev000). Idaho National Lab. (INL), Idaho Falls, ID (United States). <https://doi.org/10.2172/1557649>

## Motivation (2/2)

- Different iterations of the PRA model consists of multiple elements including system analyses, event sequence analyses, and event sequence quantification.
- These elements are supported by multiple FTs and ETs which are constantly changed and updated to reflect any changes to the design.
- Even though, collaboration in small models is not required, for larger models, collaboration on different elements or different component of the same element present a way of speeding up the modeling process.
- PRA tools (SAPHIRE, CAFTA, RISKMAN) provides limited configuration control, however, none of them provide any collaboration capability.

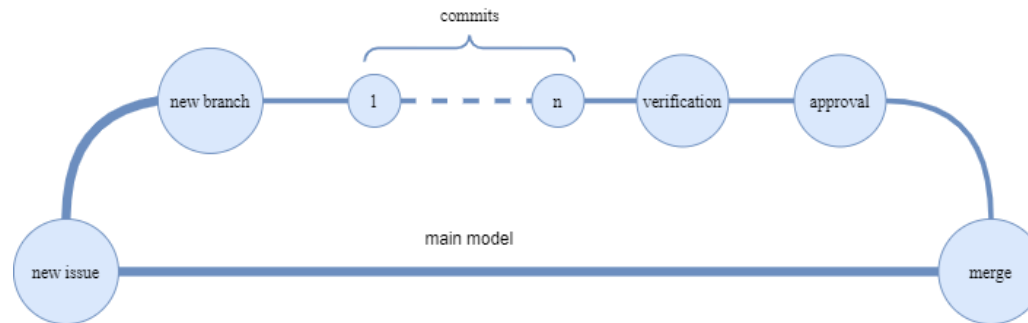


## Version Control Systems and git

- VCS: Tool that manages and tracks different versions of software or **other content**
  - Goes back to 1970s, developed for UNIX distributions
- Distributed VCS is the standard today, and thought alongside coding languages
- **git** is one of the most widely used tools, released in 2005
  - Multi-user development
  - Speed
  - Scalability
  - Immutability
  - Accountability
- VCS and git can be used for version controlling of PRA models, track changes, shorten development in a transparent environment

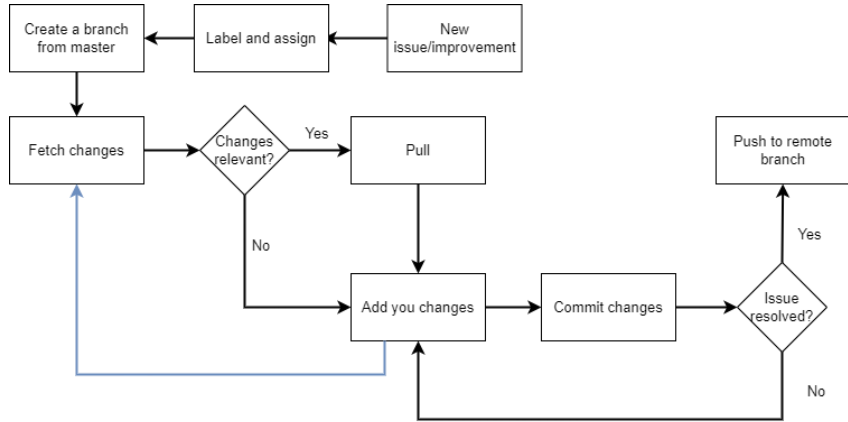
# Methodology Overview

- To fully utilize git, text-based model files must be adopted
- Version history of auxiliary files are still recorded and can be reverted when needed
- Fast tracking other's changes and importing them are effortless

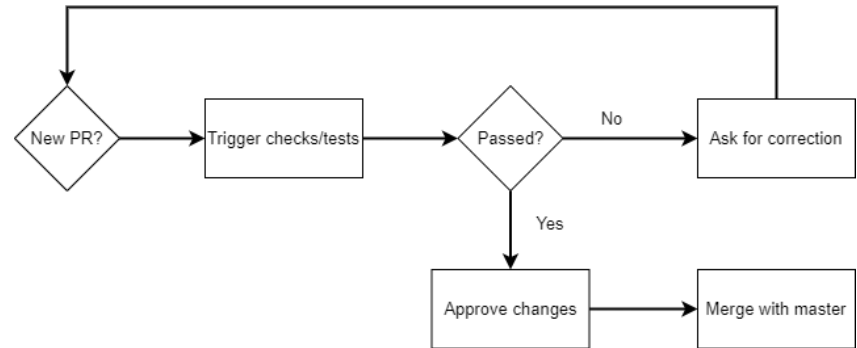


# Methodology for model developer

- Model developer



- Repo maintainer



# Version Control for PRA Models

- This methodology mimics the one introduced by [Vedros et al.](#) in INL's "Enhancement of Industry Legacy Probabilistic Risk Assessment Methods and Tools".
- INL's methodology utilizes git to store and track different revisions of the design documents.
- PRA tools, like SAPHIRE, exports binary files, hence, the application of this methodology is very limited for PRA models.
- So, a text-based mediator is needed.

ERROR.OLD

EVENT.DAT

EVENT.IDX

EVENTREE.BLK

EVENTREE.DAT

EVMODEL.blk

EVMODEL.dat

EVMODEL.idx

FT-READ.FTT

GATE.DAT

GATE.IDX

INIs

RecentActsIniName.ini

OBJPARM.dat

OBJPARM.idx

SYSTEM.BLK

SYSTEM.DAT

current-0.dat

BIN +1.05 KB (150%) EVENT.DAT

Binary file not shown.

BIN +0 Bytes (100%) EVENT.IDX

Binary file not shown.

BIN +9.75 KB (810%) EVENTREE.BLK

Binary file not shown.

BIN +0 Bytes (100%) EVENTREE.DAT

Binary file not shown.

BIN +384 Bytes (180%) EVMODEL.blk

Binary file not shown.

BIN +1.12 KB (150%) EVMODEL.dat

Binary file not shown.

BIN +0 Bytes (100%) EVMODEL.idx

Binary file not shown.

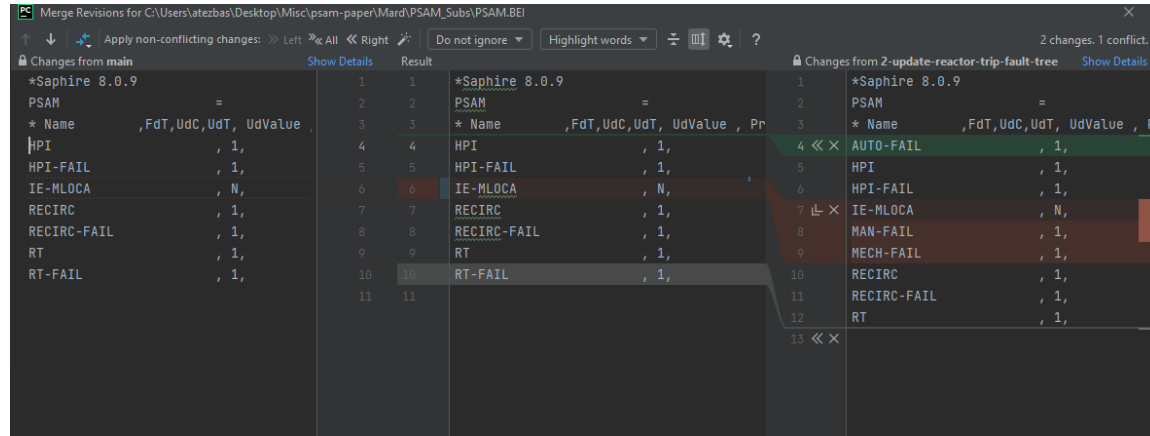
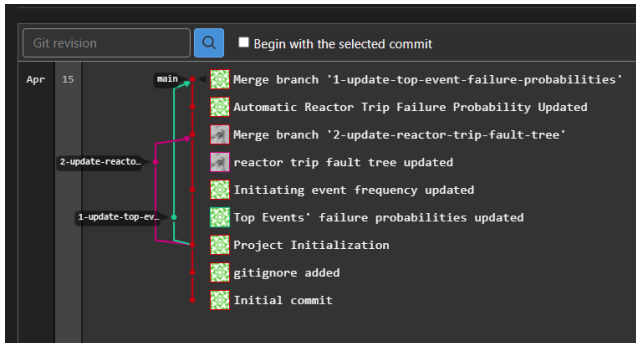


## Models and Results Database (MAR-D)

- The NRC funded the development of a way to convert PRA models into manageable text-based format.
- Being text-based, MAR-D gives the ability to store, modify, update, compare, and collaborate in PRA models using Git.
- Most PRA tools, like SAPHIRE and CAFTA, read and export in MAR-D format.
- Even though MAR-D is used in this demonstration, any text-based format, like json, is compatible with the same approach.

# Demonstration

- Repo: <https://github.com/mostafa-hamza/PSAM16>



## Final Remarks

- The methodology presented is a patch to provide version control and collaboration capabilities for PRA tools.
- Version control maintains the traceability and credibility of the PRA model, allowing for easier peer review and quicker assessment of the impact of model changes.
- Collaboration facilitates parallel development of different model branches; this reduces the time required to build the PRA model.
- Better implementation of the proposed approach can be incorporated directly into the PRA tools which will provide the user with the capability to directly pull, modify, push, and merge any changes directly from the PRA tool.

**Thank you  
Questions?**

