Script for Product Overview Presentation

**[SCRIPT FOR PROJECT OVERVIEW PRESENTATION]**

**[Title slide]**

**Hello, my name is Justin Campbell; a little bit about myself before I dive into the presentation; I am a research engineering/scientist assistant employed by The University of Texas at Austin’s Computational Research in Ice and Ocean Systems or CRIOS group. The group is part of the university’s Oden Institute for Computational Engineering and Sciences and their mission is to apply advanced computational methods to improve the understanding of the role of the global ocean, sea ice, and ice sheets in the climate system. In particular, Bayesian inverse methods, scientific machine learning and other techniques are used to develop numerical models that can improve the group’s mechanistic understanding of current, and predict future, ocean phenomena. One such climate model is the Massachusetts Institute of Technology general circulation model (or MITgcm for short). As the name suggests, the model is used to study oceanic and atmospheric phenomena using finite volume methods and is a community model with open-source code where hundreds of researchers and scientists have contributed to the development of the model. As part of the project entitled Estimating the Circulation and Climate of the Ocean (ECCO for short), the model output is assimilated with remote observational data from satellites and in-situ observational data from cruise ships and argo floats to produce a time-evolving estimate of atmospheric and oceanic circulation properties.**

**This project involves the development of post-processing and data analysis routines that are geared specifically towards operating on model output data of the format that MITgcm produces. My responsibilities as a research engineering scientist assistant involve contributing to the development of data analysis and climate model simulation workflows, and as part of the latter, bottlenecks in MITgcm model user’s workflows both within the group and in the larger MITgcm community have motivated the undertaking of this project to develop portable units of software that can automate much the workflow of setting model problem specifications, building and running model experiments using MITgcm software, and ultimately, interpreting the results using ECCO software. Ideally, these products will reduce the time and energy needed for an end user to execute this workflow for their research using this model (which has been in development for a significant amount of time).**

**[Problem Statement Slide]**

**The underlying problem statement that motivates this project and that was in part, inspired by conversations with research engineering associates and graduate student researchers my organization, namely, with Ivana, Helen, and Shreyas is that many scientists, researchers, and other members of the MITgcm community experience complexity with, and time-intensive involvement in, configuring access to computing resources, and establishing environments for building, running, and visualizing MITgcm model problems.**

**Additionally, many users find it inconvenient and time-consuming to read through numerous user guides and tutorials to become familiarized with learning new skill sets to perform their research such as determining what software modules and libraries to install on their host machine, learning new skills such as shell scripting, navigating Linux environments, and others**

**It can also be meddlesome for users to perform the post-processing/data science tasks of writing out, importing, wrangling, and visualizing model output data using numerous analysis routines that exist within the MITgcm-ECCO community that are always changing**

**Lastly, many problems developed using MITgcm can take an unreasonable amount of time to build and run on host architecture and could benefit from a more lightweight execution environment**

**[Need Statement Slide]**

**Given these limitations and inconveniences with building, running, and interpreting MITgcm model problems, there exists a need for the development of portable units of software that can be used to build and run models in an isolated environment where minimal prior knowledge of software module, library, and other dependency installation are required.**

**These portable units of software would ideally come equipped with a pre-installed operating system, along with necessary software modules, libraries, and dependencies needed for the end user to perform all tasks in the MITgcm-ECCO workflow. This would effectively reduce the need for the end user to determine what software tools to install, and certain skill sets in computing (like shell scripting) to acquire, along with the need to comprehend analysis routines for post-processing before building, running, and interpreting results from models.**

**These isolated environments would ideally function as a one-stop-shop for all MITgcm-ECCO workflow tasks and as lightweight virtual machines that can run any host machine (with some fundamental limitations) regardless of that machine’s architecture, operating system, etc.**

**[Project Objectives 1 Slide]**

**To address this project’s need statement, project objectives have been documented in two categories, namely, SMART Goals (concrete goals that are smaller in scale and more achievable) and Stretch Goals, namely, ambitious goals that expand on or provide an enhanced functionality to, the concrete goals**

**[Project Objectives 2 Slide]**

**The Smart goals are as follows:**

1. **Develop portable units of software (ideally containers) that can be leveraged to build, and run computationally inexpensive MITgcm model problems with installation of MITgcm, ECCO libraries and their dependencies and without mpi libraries**

1. **Configure support for these units of software on the most frequently used computing platforms, architectures, and operating systems according to CRIOS group users**

1. **Document the software products in code repository and image registry and present results to MITgcm-ECCO master repository managers**

**[Project Objectives 3 Slide]**

**The Stretch Goals that I plan on implementing time-permitting and assuming technical compatibility are as follows:**

1. **Implement a graphical-user-interface for end users to input MITgcm model problem parameters and specifications ( such as step size, spatial and temporal domain boundaries, etc) along with desired post-processing and data analysis tasks such as invoking ECCO to render data visualizations and render output on a GUI in the portable unit of software.**
2. **Scale up support for MITgcm model problems and ECCO visualization tasks to enable more computationally expensive configurations**
3. **Add support for ASTE regional domain model problems**

**4. Scale up support for product to run on most-widely adopted computing platforms, architectures, and OS according to general purpose MITgcm community use.**

**[Design Process Slide]**

**On this slide is a schematic that will overview the design process framework that I will follow in working on this project. In particular, the framework follows a conventional iterative product design process that begins with definition of the problem statement and associated need statement that have been discussed earlier in the presentation.**

**The next phase of the project to be completed, and that is currently in progress, will be the phase documented as ‘Understand’ which will involve the definition of design requirements (both subjective technical requirements and quantitative requirements), performing background research on software tools that the products in this project will utilize through user guides and literature reviews and documenting the outcomes of projects that are similar in scope, and lastly finalizing customer needs (which will likely involve communication with other senior research associates and graduate students to get their input on how this product can be optimized for their collective benefits).**

**Following the ‘Understand’ phase, we have the ‘Ideate’ phase which will involve the development of sketches and schematics to communicate the flow of information in the product in a hardware/software-independent way by defining the product as a black-box with inputs and outputs and identifying each input/output in the product and the systems they communicate with. Then, the black-box definition of the product will be translated into hardware-software specific product designs deemed as preliminary designs and documented.**

**Transitioning to the ‘Evaluate’ phase, the preliminary designs will be iteratively tested, troubleshooted, and working designs will then be checked against requirements to determine whether or not they fit with their strengths and weaknesses enumerated. Requirement compromise and decision matrix schematics will then be developed to document trade-offs in the preliminary designs. From here, a decision tool will be incorporated to rate down-selected designs that meet all necessary requirements in a feasibility study and the design or designs that are optimal according to the results of the feasibility study will be selected as detailed designs to carry over into the ‘Prototype and Test’ Phase as shown.**

**Here, detailed technical drawings and schematics of the detailed designs will be documented and mathematical and/or computed models for the designs will be implemented in the applicable system architecture. Using this information, the physical model of the detailed designs will be built. From here, an analysis plan will be developed to specify approaches to testing and analyzing the performance of detailed designs according to properties such as compile time, runtime, compute resource allocation, etc. and test results of detailed design prototype performances will be documented according to this analysis plan. From here, the last step before communication of the final product will be to perform a requirements verification to check the working product against all critical project requirements according to end user and stakeholder expectations.**

**After it is determined that the final product is optimized according to critical project requirements in the verification stage, the project work will be committed to a public code repository and MITgcm and ECCO researchers will be granted privileges to view the content. To wrap up the project, meetings will be scheduled with senior research scientists to communicate the results of the project and address plans for incorporating the final product into the MITgcm-ECCO workflow.**

**[Project Workflow Slide]**

**Presented on this slide is a high-level overview of the workflow that I am undertaking in working on this project. This workflow can be thought of as a working methodology for performing tasks from the ‘Understand’ to the ‘Prototype and Test’ phase of the product design process schematic mentioned earlier. As an iterative process, the first task involves gathering and verifying requirements in communicating with end users on the project. Since the product’s first release will be intended for CRIOS group MITgcm users specifically, high-level communication regarding project objectives, design requirements, etc. will be established with Patrick, researchers, and other graduate students who will benefit most from portability and improved efficiency of compiling and running MITgcm model problems. The next step in the workflow is the system design phase which will involve developing flowcharts for high-level layers of communication and procedures between product elements, and within these product elements, class diagrams and pseudocode in an architecture-agnostic fashion. From here, the diagrams will be implemented in code, the code incorporated into a containerized environment, and ultimately tested all on a Centos7 Linux OS running on an Oracle VirtualBox Hypervisor as the development environment.**

**This development environment was chosen because of lack of permissions to run containers directly on sverdrup or TACC login or compute nodes. This brings up the next stage in the workflow which involves using another containerization software called ‘singularity’ to build and test scripts from runnable containers. Singularity is installed on TACC and sverdrup login nodes, and therefore, the development environment for building and testing singularity scripts from functional containers will be a Centos7 OS on sverdrup’s and stampede2’s compute nodes respectively. Now, it should be mentioned that the initial goal for the end-user environment of the product is to have it running on Centos7 on sverdrup for immediate access by the group. However, stretch goals involve configuring the product to run on stampede2 compute nodes, and perhaps other Linux distributions, and computing architectures used by MITgcm users. In step 8, after the output of the singularity scripts are checked against their container counterparts in the virtual machine environment, the final product will be pushed up to code and image repositories and then ultimately tagged for release to the public. Any recommendations for feature improvements, bug fixes, performance enhancements, etc. will be continously incorporated in the workflow bringing us back to step 1 in this iterative design.**

**[Communication Hierarchy Slide]**

**With respect to project collaboration, I’ve established this high-level top-down flowchart to characterize the communication hierarchy and scope of communication with different collaborators on this project. At the beginning of the project, I have briefed my manager, Patrick of the problem and need statements, along with objectives of this project, and my chosen design process. I will establish recurring meetings with him throughout the project lifecycle to brief him on updates as I progress through designing, testing, and implementing my products. Below, I’ve documented the contact information of two of the group’s senior research associates, An and Helen with whom I plan on setting up periodic meetings to have high-level discussions about gathering project requirements and product end goals to address CRIOS group needs. These high-level discussions will be incorporated into ad-hoc meetings with graduate research assistants Ivana, Shreyas, and Matthew (around their research schedules) to troubleshoot system design errors, validate requirements, and gauge feasibility of system designs. Lastly, time permitting, in addressing the stretch goals to release a product that is compatible with computing hardware and software in the general MITgcm-community and thus not limited to the CRIOS group, I will establish periodic meetings with senior research scientists with NASA’s JPL in Ou and Ian.**

**[Next Steps Slide]**

**The next steps in the project design process will involve performing a literature review on the broad subject area of using portable units of software to run climate models in a platform agnostic way, reading through relevant user guides for installing and operating software tools applicable to this project, beginning the ‘Ideate’ phase of the design process by developing sketches and diagrams that communicate layers of information in system design, and lastly, scheduling and incorporating feedback from, periodic meetings with project collaborators to gather and check design requirements.**

**[References Slide]**

**And here are the references. Are there any questions?**