GEANT-4 GPU Port:

Software Requirements Specification
Volere Template, Edition 16

Stuart Douglas – 1214422
Matthew Pagnan – 1208693
Rob Gorrie – 1222547
Victor Reginato – 1209975

Version 0
October 12, 2015

Contents

1 Revision History 1

2 Project Drivers 1

2.1 Purpose of Project 1

2.1.a Project Background 1

2.1.b Goal of the project 1

2.2 Stakeholders 1

2.2.a The Client 2

2.2.b The Customer 2

2.2.c Other Stakeholders 2

2.2.d The Hands-On Users of the Product 2

2.2.e Personas 2

2.2.f Priorities Assigned to Users 3

3 Project Constraints 3

3.1 Mandated Constraints 3

3.2 Naming Conventions & Terminology 3

3.3 Relevant Facts and Assumptions 4

3.3.a Facts 4

3.3.b Assumptions 4

4 Functional Requirements 4

4.1 The Scope of the Work 4

4.1.a The Current Situation 4

4.1.b The Context of The Work 5

4.1.c Work Partitioning 5

4.2 Business Data Model & Data Dictionary 5

4.3 The Scope of the Product 5

4.4 Functional Requirements 6

4.5 Look and Feel Requirements 7

4.5.a Appearance Requirements 7

4.5.b Style Requirements 8

4.6 Usability and Humanity Requirements 8

4.6.a Ease of Use Requirements 8

4.6.b Personalization and Internationalization Requirements 9

4.6.c Learning Requirements 9

4.6.d Understandability and Politeness Requirements 9

4.6.e Accessibility Requirements 10

4.7 Performance Requirements 10

4.7.a Speed and Latency Requirements 10

4.7.b Safety Critical Requirements 10

4.7.c Precision of Accuracy Requirements 11

4.7.d Reliability and Availability Requirements 11
4.7.e Robustness or Fault-Tolerance Requirements 11
4.7.f Capacity Requirements 12
4.7.g Scalability Requirements 12
4.7.h Longevity Requirements 12
4.8 Operational and Environmental Requirements 13
4.8.a Expected Physical Environment 13
4.8.b Requirements for interfacing with adjacent Systems 13
4.8.c Productization Requirements 14
4.8.d Release Requirements 14
4.9 Maintainability and Support Requirements 14
4.9.a Maintenance Requirements 14
4.9.b Supportability Requirements 14
4.9.c Adaptability Requirements 14
4.10 Security Requirements 15
4.10.a Access Requirements 15
4.10.b Integrity Requirements 15
4.10.c Privacy Requirements 15
4.10.d Audit Requirements 16
4.10.e Immunity Requirements 16
4.11 Cultural Requirements 16
4.12 Legal Requirements 16
4.12.a Compliance Requirements 16
4.12.b Standards Requirements 16

5 Project Issues 16
5.1 Open Issues 16
5.2 Off-the-Shelf Solutions 17
5.2.a Ready-Made Products 17
5.2.b Reusable Components 17
5.2.c Products That Can Be Copied 17
5.3 New Problems 17
5.3.a Effects on the Current Environment 17
5.3.b Effects on the Installed Systems 17
5.3.c Potential User Problems 18
5.3.d Limitations of the Anticipated Implementation Environment That May Inhibit the New Product 18
5.3.e Follow-Up Problems 18
5.4 Tasks 18
5.5 Migration to the New Product 19
5.5.a Requirements for Migration to the New Product 19
5.5.b Data That Has to Be Modified or Translated for the New System 19
5.6 Risks 19
5.7 Costs 19

5.8 User Documentation and Training 20
5.9 Waiting Room 20
5.10 Ideas for Solutions 20

1 Revision History

All major edits to this document will be recorded in the table below.

Table 1: Revision History

Description of Changes	**Author**	**Date**
Initial draft of document	Stuart, Matthew, Rob, Victor	2015-10-07

2 Project Drivers

2.1 Purpose of Project

2.1.a Project Background

Physics researchers use software simulations to model how particles interact with environments, and to determine the effects of these interactions. Members of McMaster's Engineering Physics department use GEANT-4 – a simulation toolkit developed by CERN. McMaster's researchers have developed their own fork of the software, G4-STORK, designed to study McMaster's nuclear reactor. Currently, running G4-STORK simulations (as well as other GEANT-4 simulations) that require many particles takes a long time to compute when run on the CPU. This limits researchers to smaller number of particles than are realistic, and prevents them from seeing the effects of longer time periods.

2.1.b Goal of the project

The goal of this project is to significantly increase the computation speed of G4-STORK simulations, which should transfer well to other GEANT-4 projects. Increasing the speed will allow researchers to use more accurate models, to see the effects of longer time periods on the particles, and to generally increase their productivity. This will be achieved by porting the algorithms that currently run on the CPU to the GPU, taking advantage of the parallel power of the GPU's many cores.

2.2 Stakeholders

The stakeholders that are currently involved with the project are:

- the project group
- the supervisors of the project (Dr. Adriaan Buijs and Dr. Emil Sekerenski)
- McMaster Engineering Physics Department, specifically grad student Wesley Ford

2.2.a The Client

The clients for the project are [Unnecessary "the" —DS] Dr. Buijs and his grad student Wesley Ford, representing the McMaster Engineering Physics Department. The clients proposed the project because they have invested interest in running G4-STORK simulations more efficiently. They will be using the parallelized code to run and study nuclear simulations, and require that the code executes much more quickly to obtain useful data.

2.2.b The Customer

The customer in this case also includes the clients. As such they will be the part of the end-user group that we will cater to. The customer is also other members of the Engineering Physics department who wish to run simulations using GEANT-4 as they also have use for the end product. The users will want to run simulations with many particles and particle collisions, the optimization of the code will allow for them to do this in a timely fashion.

2.2.c Other Stakeholders

The project group members hold stake in the successful completion of the project. If the project is done well, and the documentation is error free and concise, they are bound for high grades in their course.

Collaborators and GEANT-4 users external to McMaster could be potential stakeholders. If the software is successfully completed and the resulting performance improvements are significant, the changes will be submitted to the GEANT-4 repository. It is then up to the GEANT-4 group to decide if those contributions are accepted or not, and if they are then all users of GEANT-4 stand to benefit from them. However, they are not considered stakeholders throughout the development of the project, the focus is instead on those mentioned in 2.2.

2.2.d The Hands-On Users of the Product

Graduate students and professors from the McMaster Engineering Physics Department will be hands-on users of the product. They will all have familiarity with the existing product.

2.2.e Personas

Consider Matt Douglas, a graduate student in the Engineering Physics Department. Matt wants to study how particles in McMaster's nuclear reactor affect each other, given some specific starting conditions, and a very large number of particles. Matt knows how to use the existing G4-STORK program, but it will take weeks to run the computations. Matt inputs his desired number of particles and specific starting conditions into a G4-STORK simulation. He enables the GPU features of the new product, and successfully runs the simulation in a reasonable amount of time. He observes interesting results that would not have been feasible to obtain with the old product.

2.2.f Priorities Assigned to Users

Priority will be given to Wesley Ford, the grad [“graduate” —DS] student that proposed the project to Dr. Buijs. He will be the primary user of G4-STORK as it is integral to his current research.

3 Project Constraints

3.1 Mandated Constraints

There are global constraints put in place by the existing software, the stakeholders, and the structure of 4ZP6. The project must be built upon the existing GEANT-4 code, which is used by G4-STORK to run the simulation. The final product must be able to run any code/simulation that ran on the existing software. The software must run in parallel on a NVIDIA GPU. Additionally, the final product needs to be completed by the end of April, 2016. If these global constraints are not met the final product is not acceptable.

[Why an NVIDIA GPU? Was that constrained by your client? —DS]

3.2 Naming Conventions & Terminology

Throughout the document, “the project”, “the product”, and/or “the software” all refer to the modified GEANT-4 code that will include the capability to be run on a GPU. The “existing software” refers to the current GEANT-4 simulation toolkit.

Table 2: Glossary

Term	Description
GEANT-4	open-source software toolkit used to simulate the passage of particles through matter
G4-STORK	(Geant-4 STOchastic Reactor Kinetics), fork of GEANT-4 developed by McMaster's Engineering Physics department to simulate McMaster's nuclear reactor
GPU	graphics processing unit, well-suited to parallel computing tasks
CPU	computer processing unit, general computer processor well-suited to serial tasks
GP-GPU	concept of running "general-purpose" computations on the GPU
CUDA	parallel computing architecture for general purpose programming on GPU, developed by NVIDIA

3.3 Relevant Facts and Assumptions

3.3.a Facts

- GEANT-4/G4-STORK are programmed in C++
- GEANT-4/G4-STORK run simulations on the CPU
- Simulations run calculations on each particle independently
- Calculations on each particle are relatively simple probabilities

3.3.b Assumptions

- The user will have a strong understanding of particle physics
- The user will know how to use G4-STORK

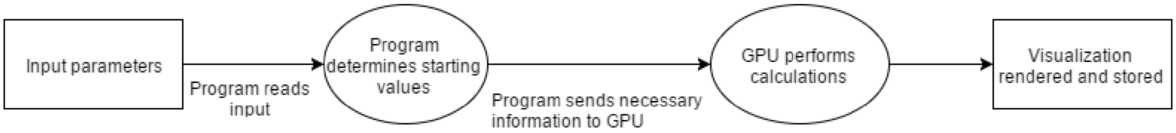
4 Functional Requirements

4.1 The Scope of the Work

4.1.a The Current Situation

The project, G4-STORK, is currently at a point where it is able to run all necessary simulations, on the CPU. Currently, when the G4-STORK simulation is run with a large number of particles or over a period of several seconds it takes a lot of computation time.

4.1.b The Context of The Work


Input parameters
Program reads input
Program determines starting values
Program sends necessary information to GPU
GPU performs calculations
Visualization rendered and stored

[Figure caption? —DS]

4.1.c Work Partitioning

All tasks will be divided equally among all members of the development team. The team will work together to initially define the interface for using the GPU to run calculations, and once that is completed each member will independently work on porting specific classes to the GPU. Testing will be done throughout the development by the developers. This ensures that all developers are able to easily debug their contributions. [This section is supposed to show a breakdown of the business use cases —DS]

4.2 Business Data Model & Data Dictionary

NA – The product does not store any data, and requires no data model.

4.3 The Scope of the Product

The following table outlines the use cases for the product.

Table 3: Product Use Cases Summary

PUC #	**PUC Name**	Actor(s)	**Input/Output**
1	Simulating Particles	Researcher	Simulation parameters (in), Distribution of particle's locations (out)

Descriptions of each PUC, referenced by PUC # are as follows. [PUC is not in your glossary —DS].

1. The software will be used by researchers wishing to simulate large numbers of particles interactions with materials. The researcher sets simulation parameters, including the number of particles, their lifetime, and the material properties before running the simulation. On completion, the program gives back a map of where each particle traveled, so researchers can study where the particles are most probably to end up.

4.4 Functional Requirements

Req. #: 1 **Req. Type**: 4.4 **Use Case #**: 1

Description: Particle computations run on the GPU

Rationale: Design requirement, will allow particle simulations to run faster (requirement 10)

Fit Criterion: Running the product with GPU computation enabled will result in all computations on particles being offloaded from the CPU (existing product) to the GPU (new product)

Priority: Very High **History**: Created September 29, 2015

Req. #: 2 **Req. Type**: 4.4 **Use Case #**: 1

Description: Changing existing projects to run with new GPU functions should be easy

Rationale: Design requirement, the user is able to easily choose to run old or new projects on the GPU.

Fit Criterion: User should be able to quickly enable GPU computations after referencing the documentation. If their hardware is compatible, their project should run with no errors.

Priority: High **History**: Created September 29, 2015

[Your description shouldn’t include “should be easy”. It should be detailing *what* the product is supposed to do. Ease of use is a non-functional requirement. —DS]

Req. #: 3 **Req. Type**: 4.4 **Use Case #**: 1

Description: Existing projects should not be affected by the new code. By default, they will continue to run on the CPU.

Rationale: Design Requirement, need to ensure that users can continue to use GEANT-4 as before.

Fit Criterion: Running an existing simulation should execute on the CPU by default as before with no performance regressions and identical results.

Priority: High **History**: Created September 29, 2015

[This requirement should be split. The default case should have its own requirement. —DS]

Req. #: 4 **Req. Type**: 4.4 **Use Case #**: 1

Description: Trying to run the simulation on the GPU with a computer that does not have a compatible graphics card should be detected and cause it to run on CPU like before.

Rationale: New product should not limit the amount of users who can use GEANT-4

Fit Criterion: Any computer that can currently run the existing product should be able to run the new product.

Priority: Medium **History**: Created September 29, 2015

[This requirement should be split. There should be one for detecting hardware and one for what to do based on hardware detected. —DS]

[Will users be aware of whether or not their machine can run the code on the GPU? —DS]

4.5 Look and Feel Requirements

4.5.a Appearance Requirements

NA – The scope of the project does not include any visual aspects. Simulations are run through the command line. [Then why does your 4.1.b diagram show “Visualizations rendered and stored?” —DS]

4.5.b Style Requirements

Req. #: 5 **Req. Type**: 4.5 **Use Case #**: 1

Description: The code should be well organized and appear readable to any who might want to edit it.

Rationale: Revised code should not inhibit the modifying or understanding of G4-STORK

Fit Criterion: Any persons with an understanding of the existing project should be able to clearly understand the modified code.

Priority: Medium **History**: Created October 6, 2015

[This is not a requirement for your end-product. —DS]

4.6 Usability and Humanity Requirements

4.6.a Ease of Use Requirements

Req. #: 6 **Req. Type**: 4.6 **Use Case #**: 1

Description: The functionality and procedures involved should be manageable for a university undergraduate (or higher) with experience in quantum mechanics, linear algebra, and computer programming.

Rationale: The software is extremely technical, and requires exposure and understanding to quantum physics and computer programming.

Fit Criterion: Any given McMaster undergrad with experience in quantum mechanics, linear algebra, and computer programming should be able to learn to use the software.

Priority: Medium **History**: Created October 6, 2015

4.6.b Personalization and Internationalization Requirements

Req. #: 7 **Req. Type**: 4.6 **Use Case #**: 1

Description: There will be no adjustments for personal and international preferences offered.

Rationale: The software follows a rigerous scientific standard.

Fit Criterion: A given user with personal (non-technical) barriers to achieving complete satisfaction with the software will be responsible for overcoming such barriers.

Priority: Medium **History**: Created October 6, 2015

[“Rigorous” —DS]

4.6.c Learning Requirements

Req. #: 8 **Req. Type**: 4.6 **Use Case #**: 1

Description: Utilization of the software should be learnable given 2-3 weeks of training.

Rationale: Although the prerequisite knowledge assumed of the user is massive, the learning curve for such a person is not.

Fit Criterion: Any university undergraduate student with solid foundations in quantum mechanics and computer programming should be able to learn to operate the software in roughly 2 or 3 weeks.

Priority: Medium **History**: Created October 6, 2015

4.6.d Understandability and Politeness Requirements

NA

4.6.e Accessibility Requirements

Req. #: 9 **Req. Type**: 4.6 **Use Case #**: 1

Description: The software should be accessible to any person who can learn and use a shell based operating system.

Rationale: Vision/reading/typing is required for the operation of the software, and there will be no accessibility alternatives offered.

Fit Criterion: A blind university student who potentially has trouble using a shell based operating system may have trouble with the software.

Priority: Medium **History**: Created October 6, 2015

[Bad fit criterion —DS]

4.7 Performance Requirements

4.7.a Speed and Latency Requirements

Req. #: 10 **Req. Type**: 4.7 **Use Case #**: 1

Description: Decrease the time it takes to run a particle simulation while maintaining the same output.

Rationale: The entire purpose of the project is to improve the speed of the simulation.

Fit Criterion: Running a simulation with a given set of input parameters should complete significantly faster on the product as compared to the existing software. Both should have identical outputs.

Priority: Very High **History**: Created September 27, 2015

4.7.b Safety Critical Requirements

NA – The software is not used in safety critical environments. It is a tool for researchers to study particle interactions.

4.7.c Precision of Accuracy Requirements

Req. #: 11 **Req. Type**: 4.7 **Use Case #**: 1

Description: Results should have same accuracy whether simulation is run on CPU or GPU.

Rationale: If results are not as accurate as with the existing product then researchers will not be able to draw as strong conclusions.

Fit Criterion: The results of a simulation should be identical on the new product as the existing one, given that the inputs are the same.

Priority: High **History**: Created September 27, 2015

4.7.d Reliability and Availability Requirements

Req. #: 12 **Req. Type**: 4.7 **Use Case #**: 1

Description: The product should be at least as stable as the existing product.

Rationale: Researchers require an extremely stable product, we do not want to introduce any new crashes or bugs.

Fit Criterion: Testing the product with a variety of simulations should never result in a crash.

Priority: High **History**: Created September 27, 2015

4.7.e Robustness or Fault-Tolerance Requirements

NA – The product does not need to maintain standards during an unforseen event. It does not handle any sensitive data, and is not constantly required to be running by its users.
[Shouldn't there be some safeguards in case of an error? —DS]

4.7.f Capacity Requirements

Req. #: 13	**Req. Type**: 4.7	**Use Case #**: 1

Description: Due to speed improvements (requirement 10) a larger number of particles should be able to be simulated in the same period of time.

Rationale: Increasing the number of particles in the simulation will allow researchers to better simulate real-world interactions.

Fit Criterion: A simulation running on the new product will be able to simulate a significantly larger number of particles in the same time as the same simulation on the existing product.

Priority: High	**History**: Created September 27, 2015

4.7.g Scalability Requirements

NA – The product will be used by single researchers, and is run locally on their machines.
[What if the researchers wanted to run it on a server with multiple GPUs (or hundreds of CPU cores?) —DS]

4.7.h Longevity Requirements

NA – The product is to be completed by April 2016, and there are no plans to extend that lifetime.

4.8 Operational and Environmental Requirements

4.8.a Expected Physical Environment

Req. #: 14 **Req. Type**: 4.8 **Use Case #**: 1

Description: The product shall be used by an Engineering Physics professor, researcher or student who will be sitting down in a temperature controlled environment.

Rationale: The product should typically only be used in an office environment.

Fit Criterion: 95% of all uses of the product will be used by Engineering Physics professors, researchers or students in sitting down in temperature controlled environments.

Priority: Low **History**: Created October 4, 2015

4.8.b Requirements for interfacing with adjacent Systems

Req. #: 15 **Req. Type**: 4.8 **Use Case #**: 1

Description: The product shall work with the last four versions of GEANT-4

Rationale: Backwards compatibility is a nice thing to have.

Fit Criterion: At least the last four versions of GEANT-4 will be able to run this product.

Priority: Low **History**: Created October 4, 2015

4.8.c Productization Requirements

Req. #: 16 **Req. Type**: 4.8 **Use Case #**: 1

Description: The product will available on a public repository along with clear instructions for installation

Rationale: Want to make the product easily available

Fit Criterion: 90% of users should be able to acquire the product with out much trouble

Priority: Low **History**: Created October 4, 2015

4.8.d Release Requirements

Req. #: 17 **Req. Type**: 4.8 **Use Case #**: 1

Description: New versions of the product that have been patched will be available on the public repository and shall not cause previous features to fail.

Rationale: Users should be encouraged to use the latest patch, and should be confident that updating won't negatively impact their work

Fit Criterion: Each new patch of the product will not cause any previous features to fail.

Priority: Medium **History**: Created October 4, 2015

4.9 Maintainability and Support Requirements

4.9.a Maintenance Requirements

Baeause G4-STORK is open source, the code written can be maintained by anyone. The original developers will not be maintaining the code, it will have to be done by end users.

4.9.b Supportability Requirements

There will be a setup.txt file describing briefly how to add a function and set inital parameters.

4.9.c Adaptability Requirements

The completed project is expected to run on Fedora and OSx. [Why only those? —DS]

4.10 Security Requirements

4.10.a Access Requirements

Req. #: 18 **Req. Type**: 4.10 **Use Case #**: 1

Description: All users have access to all aspects of the product.

Rationale: The product will be open-source.

Fit Criterion: Any user should never fail to access some functionality of the product.

Priority: High **History**: Created October 6, 2015

4.10.b Integrity Requirements

Req. #: 19 **Req. Type**: 4.10 **Use Case #**: 1

Description: There will be no compromise on the integrity of data outside of physical hardware malfunctions.

Rationale: All data is stored locally and unaltered.

Fit Criterion: A user upon gathering input or output data should never find that their data has been compromised.

Priority: High **History**: Created October 6, 2015

4.10.c Privacy Requirements

Req. #: 20 **Req. Type**: 4.10 **Use Case #**: 1

Description: The privacy of the data will be managed entirely by and at the discretion of the user.

Rationale: The data is stored and manipulated entirely locally.

Fit Criterion: No user will ever find that their data has been shared without their discretion.

Priority: High **History**: Created October 6, 2015

4.10.d Audit Requirements

4.10.e Immunity Requirements

Req. #: 21 **Req. Type**: 4.10 **Use Case #**: 1

Description: There will be no specific features to ensure immunity.

Rationale: The only impact external software can have on the program is a cut in performance.

Fit Criterion: The user will be held responsible for ensuring that no external software is hogging the resources required for the project to run at optimal performance.

Priority: Medium **History**: Created October 6, 2015

4.11 Cultural Requirements

NA – The product is intended for a small number of stakeholders. All content is objective and scientific.

4.12 Legal Requirements

4.12.a Compliance Requirements

NA – The product does not store or access any user information. It does not need to comply with any legal standards.

4.12.b Standards Requirements

NA – There are no internal standards that are required to be met by the product.

5 Project Issues

5.1 Open Issues

- The amount of time required to speedup all classes and libraries involved in G4-STORK is unknown, feasibility of project is in question

- The actual speedup of G4-STORK achievable is unknown it could be far less than desired

5.2 Off-the-Shelf Solutions

5.2.a Ready-Made Products

No fully-fledged solution to the given problem has been implemented and distributed. There have been attempts, however, to solving certain portions of the problem. At the Helsinki Institute of Physics (HIP) a team of three attempted to test the potential efficiency increase of porting GEANT-4 code to a GPU. They ported some small portions of GEANT-4, but determined that taking on the entire project would be difficult. Their conclusions supported the conjecture that running the code on a GPU would offer significant increases in performance.

5.2.b Reusable Components

Although there is not any hard-copy code that can be reused, the team of three from HIP have published a research paper that may be used as a starting point for the current project.

5.2.c Products That Can Be Copied

NA – There are no existing products that can be copied.

5.3 New Problems

5.3.a Effects on the Current Environment

The new product will be designed as an opt-in addition to the existing product. That is, unless manually changed by the user, the program will execute identically as before, running on the CPU. The motivation for this is to ensure compatibility, as the programming environment for GP-GPU programming is restricted to certain hardware. The current environment will not be affected by the changes unless the user specifically decides to use them. [You are describing design decisions, not effects on the current environment —DS]

5.3.b Effects on the Installed Systems

Requirement 2 specifies the importance of creating a simple interface for the user to enable the changes in the new system, or revert to the previous one. Changes to the code will be isolated, and only used when they are manually enabled. [Again you are talking about design. Here you should be detailing what your how your product interfaces with the existing system. Are there any potential conflicts that would need to be solved (i.e. by your design decision)? —DS]

5.3.c Potential User Problems

Due to the separation of the changes and the existing product, users will not negatively respond to the changes, indeed they won't even notice them unless actively looking. To use the new product's features, users will enable them and then execute the program in an identical manner to how the existing software works (requirement 2). [You are again making design decisions instead of detailing potential problems to be solved by those decisions —DS]

5.3.d Limitations of the Anticipated Implementation Environment That May Inhibit the New Product

To run the GP-GPU computations, specific hardware is required [You should stop here, unless you explained (in the constraints section) why you need an NVIDIA card. Also, what is "recent"? —DS] (recent NVIDIA graphics card).

5.3.e Follow-Up Problems

There are a number of potential situations that could lead to the product failing. We are confident that we will be able to succeed, however we realize that there is a possibility of failure, and have outlined the potential causes below.

- Learning curve for existing G4-STORK codebase is too steep, cannot gain adequate understanding to implement changes in time constraints
- Porting existing algorithms to CUDA requires too much work, and we are not able to run the algorithms on a GPU within time constraints
- The current product's interface for the specific algorithms that will be ported is not well-enough defined
- The large models used in the simulation exceed memory limitations on the GPU, and cannot be run on existing hardware
- Performance gains from the GPU are negligible, due to the structure of the computations
- Numerical accuracy problems lead to different results from simulations run on the existing product vs. the new product

5.4 Tasks

Record of Proposed Project	September 18
Problem Statement	September 25
Requirements Document Revision 0	October 9
Proof of Concept Plan	October 23

Test Plan Revision 0	October 30
Proof of Concept Demonstration	November 16 - 27
Design Document Revision 0	January 1
Revision 0 Demonstration	February 1 - 27
User's Guide Revision 0	February 29
Test Report Revision 0	March 21
Final Demonstration (Revision 1)	Exam period
Final Documentation (Revision 1)	April 1

5.5 Migration to the New Product

5.5.a Requirements for Migration to the New Product

Once the end product has been implemented, all new projects using G4-STORK running on GPU's will use the new code.

5.5.b Data That Has to Be Modified or Translated for the New System

In order to take advantage of the GPU speedup, old projects will need to be modified to use the new classes written for GPU acceleration.

5.6 Risks

Because our software has research based aspirations and goals, some traditional risks are less catastrophic to the success of the project, but are still risks nonetheless. The following may pose to be potential risks:

- Excessive schedule pressure
- Technical/Resource limitations
- Inadequate measurements [What do you mean by this? —DS]
- Optimistic time constraints

5.7 Costs

All software used in the project is open-source and/or available for free. Existing hardware will be used for development, so there are no associated monetary costs.

We have very clear and well-defined deadlines for each deliverable, and are in full confidence that we will meet each one. The time it takes for each deliverable will be variable, but the date of completion for each is concrete, as outlined in 5.4.

5.8 User Documentation and Training

- Function descriptions shall be provided for every new function in the code
- There shall be a thorough Readme file accompanying the project that will explain to the user the changes as well as how to enable GPU computation
- Users who know how to use G4-STORK should be able to easily run it on the GPU

5.9 Waiting Room

There are currently no plans for extended requirements for future releases.

5.10 Ideas for Solutions

Some solutions to a few potential problems have been breifly mentioned in the above section (see 5.3). Outside of those, we can resolve usability issues with a well-written user manual, which will explain how to download the software (or for previous GEANT4 users, how to upgrade). We can also provide documentation which can distinguish if a certain user's machine is eligible for the software.

Additionally, in terms of organizational and production problems that may arise, we will schedule meetings with both our group and our supervisors. We can utilize well known software development procedures (such as an Agile strategy) to ease the managerial work required for development of the product to run smoothly.