Project proposal

Doug Richardson, Jon von Kampen, Chris Fibich, and Nick Watkins

**Project**

The purpose of the project we have taken is to make a web application allowing an easy browsing of the Common Weaknesses Enumeration (CWE) database. This application will prioritize being able to be easily searched and navigated. A means of showing the “path” between CWE entries will also be implemented. The priorities of implementation will be usability (in searching and navigation) and load time performance (when possible). The proposed project name is “Project Kintsugi.”

**Motivation**

The current CWE web site has very poor search functions. The only way to search for a CWE is by ID number. This renders it very difficult to search for a CWE if you don’t already know what you are looking for. It is also very difficult to see how CWEs are inter-related on the current system. This makes research and referencing a critical computer security database difficult and cumbersome.

**Functionalities**

The following are the required functionalities for the system to be deemed complete:  
  
The system must be able to store the XML data the CWEs are currently stored upon on a much more usable database to facilitate ease of searching. The system must, at a reasonable speed, be able to provide effective and understandable search results to find CWEs for the end user. Because CWEs are complex, the system will contain a full-text search mechanism and a ranking algorithm to prioritize certain searches with more readily relevant results. The system will have a means of “tracking” the user-searched path to show the user what “link-chain” they followed to get to the current result (assuming they stayed on the website the whole time).

The following are high-priority features that are not strictly required, but greatly preferred:

The system should be able to update the CWE database automatically, to keep the content fresh with the latest CWE data. Because the system is a public web application, security measures such as input validation should be implemented as well.

The following are medium priority features that are not required, but could be created if time permits

The system could have a means of showing CWE relationships between one another via graphing algorithms represented visually. The system could implement a shortest-path-algorithm between two given CWE nodes to calculate how far removed they are from one another. The system could have a means of altering the XML dumping algorithm to accommodate a significant change. The system could have an administrative mode to manually alter data as necessary (without having to directly access the database or associated files). The system could have a means to link or redirect the current CWE shown to the old CWE page. The system could show differences between CWE versions.

The following are low priority tasks that will only be implemented if all other higher priority tasks are finished or otherwise cannot be worked on:

The project would have a way of visually showing complete or partial graphical maps between the CWEs. The project would have a means of visually representing CWEs by category.

THE REQUIREMENTS AND THEIR PRIORITIES ARE SUBJECT TO CHANGE. This was created PRIOR to the confirmation of priorities by the client. Any changes made AFTER this confirmation will be confirmed upon agreement between the client and the team.

**Implementation strategy**

The first task we are hoping to accomplish is the research and understanding of a development language of choice to maximize its advantages and minimize unnecessary work. The languages we are currently researching are PHP and Python. We will also create diagrams of the physical and logical architecture. After that is finished, we will seek to understand the CWE XML schema and see how we can transfer that into a more usable database. We have decided on a client-server architecture.

The current implementation plan is to first establish a functional database and a means of populating it with the CWE data. Upon completion of that, the web application will be built. Once the “core system” is realized, the search algorithm will be refined and the ability to maximize its use will be available to the end user (i.e. search filters the user can choose). The ability to make “breadcrumbs” tracking the users path will be developed in parallel to this if possible. After this, we will focus on security and consistency of updates. Tertiary functions beyond those will be decided on a case-by-case basis.

The team has spoken of using a web-scripting language to create the web application, but has not committed to a single one (PHP and Python were the most commonly suggested). The database will be a linux-based SQL database (the specific format is currently unknown). Dr Gandhi has stated he will provide a Linux desktop terminal to run a dedicated database for the project on. This method was chosen to avoid unnecessary strain on UNO’s other dedicated systems (Loki, etc), as well as the load times that come with heavy access to such systems by the student body.  
  
The current implementation plan is a derivative of SCRUM. This was chosen because the project manager has had positive experience with SCRUM in a similar project, and is being adapted to suit the additional constraints of the projects. Because of extreme schedule constraints (the team members have other classes, classwork, and jobs), the group will be meeting at regular class time plus however long afterward is feasible (currently 1:30-3:30). Because of time constraints, the proposed length of “sprints” will be cut in half to two weeks. Acceptance testing meetings will double as retrospective meetings. The method of assignment and dedication of tasks has yet to be decided (I wish to get the entire team and the client involved, but schedule constraints make that difficult). Estimation of tasks will attempt to factor that in mind. We will attempt to maintain client feedback at regular intervals, while still giving the team sufficient time to make significant progress prior to the next meeting. Because this project is only has approximately three months on it, and potential meeting and work times are extremely constrained, the team will prioritize on completing the core systems to the highest quality possible above all else.

**Technical challenges**

This project will require reading and translating a foreign data set of semi-structured data to a more rigidly structured database for usability. This project will also require searching algorithm finesse (both in results and time optimization). This project will require several different platforms to be utilized as one coherent whole. This project will require the team to make sure the core system is “adaptable” for further features. If time permits, an additional challenge would be to visualize the data nodes and the relationship between them.

**Team**

The team consists of four members

Apart from Doug Richardson, the roles of the team members are largely undetermined at this point. Any allusions to a role that may be presented are simply suggestions by the project manager.

*Doug Richardson*: Doug will be primarily filling the role of project manager. This was chosen because he has completed the ISQA project management course with Doctor Duffner, as well as having prior experience on a similar project during his time abroad in Norway (the details are confidential, he signed an NDA). Because of time and schedule constraints, he will be working on small tasks and assisting others with theirs as needed. He will also be playing a role similar to the SCRUM master when appropriate.

Doug Richardson has chosen to primarily hone his skills in C. He also has a working knowledge of C#, Perl, Java, older HTML, and CSS. He is also experienced with a hex editor. He believes he could adapt to PHP or Python with relatively little difficulty.

*Jon von Kampen*: Among the group, Jon von Kampen has the most knowledge of security. He also has connections with the NUCIA faculty (whom our stakeholders are a part of). He also has administrative experience due to his indirect role in student government in the past.

Jon has experience with ColdFusion, SQL, HTML (prior to HTML5), basic CSS and Javascript. He also has a working knowledge of perl, and an elementary knowledge of PHP. He believes he could easily adapt to Python. He also has experience in C, Java, and Linux Bash scripts.

*Chris Fibich*: Chris has requested to focus his efforts on the design and development of the UI in the project.

He is most familiar with C++ and C#. He also has an understanding of C, Java (although it’s been a while) and Visual Basic. He is “very familiar” with .NET. He also has an understanding of Perl, HTML, CSS, Javascript, and PHP. He is less familiar with PHP, but has resources available to him to help catch up if need be.

*Nick Watkins*: Nick has not expressed an interest in any specific role yet. He is willing to learn and adapt to the tasks at hand, and willing to embrace the tasks given to him.

Nick has experience with Java, C, C++, C#, Perl, and some HTML (it’s been a while for the HTML). He also has experience with SAP proprietary languages.