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

Project Kintsugi: Work Breakdown Structure and Strategy

# Strategy

In order to understand the work breakdown structure, one must understand the overall strategic direction of the project.

The first step is general conceptualization. This step involves understanding what the project is, as well as the architecture and planning thereof. During this step, we have chosen the languages we choose to use, and the formatting of the applications. We have also constructed a rough estimate of the physical and logical architecture, as well as a mockup of the User Interface. These documents are of course subject to change as the project progresses. Researching and understanding anything we don’t know (such as the frameworks or version control platform used) is also done during this step.

The second step involves the conception and creation of the database. This is focused upon because the database is the core of the system, and all other applications depend on its successful research and implementation. Should we decide to use a database platform instead of work with the raw XML data given (and we have), we would need to create a database schema that is compatible with the schema that CWE data XML files use. Once this is complete, we would need to code this database and save it on a central platform to build the application around. At this time, we can also create some of the early framework for the web application if any of the team members has any time to do so.

The third step involves three sub-tasks that can be completed in semi-parallel.

The first sub-task is the finding or creation of a means to populate the new database with the XML data. If at all possible, we will try to use existing methods or software to transfer the XML data over to the database. Should we not find any that fit our needs that can be feasibly obtained; we will create our own application to do so.

The second sub-task is the construction of the web-application. This will involve the visual representation of the page and how the data is presented. The main priority is how the CWE data is displayed. The search engine page will be delayed until at least a rudimentary search engine is complete, but starting progress on it is acceptable should we get ahead of schedule.

The final sub-task is the creation of a rudimentary search engine. This will not be the final search engine. However, if schedules and workload gets too intense, this will be a ‘fall back point.’ This engine will allow searches based on more potential criteria then just the ID number. For this task, we will rely heavily on the search tools provided by the chosen database framework.

The final step involves the creation of the preferred search algorithm as well as the “breadcrumb” sub-system to navigate the web application. The search algorithm will have a full-text parser and ranking algorithm within it. Filtering mechanisms will also be provided (for example “CWEs coded in C”). The search page will be refined to allow for these things. If possible, the overhead of the program will be minimized for search results to appear in a timely manner assuming a reasonable internet connection.

All other tasks will be considered on an “as needed” basis. Because we are focusing on the “M” and “S” tasks in the MoSCoW document, those will be the ones reflected in the work breakdown structure. Other, lower priority tasks will be taken on ‘as needed.’ These will be added to the WBS accordingly.

# Work Breakdown Structure

This model of the WBS is based around the dependencies of the architecture (such as the need for the database) and the overarching strategy invoked. If tasks are in red, then they can be carried over to the next phase and the phase can still be considered complete. The work breakdown structure does not include the writing of the report, the preparation of the presentation, or the security sub-project.

THIS IS SUBJECT TO CHANGE AS WE GAIN A GREATER UNDERSTANDING OF THE TASKS.

Phase 1 (08/26 – 09/30)

Initial Conceptualization  
-Pick a language  
--Research language as appropriate  
--Learn language  
--Learn specific frameworks of sub-languages as appropriate  
--Learn other tools though to be useful as appropriate  
-Physical architecture conceptualization  
-Logical architecture conceptualization  
-Draft functional requirements  
-Draft nonfunctional requirements  
-Draft system assumptions  
  
  
Database Creation  
-Learn and understand XML schema for incoming data  
-Choose and learn database language  
-Resolve database storage issue (where are we going to put it)  
-Draft schema for database  
--Map out data relations between old and new schema  
--Map out foreign keys  
--Refine and normalize schema if necessary (recommend 3NF)  
-Code and construct database  
  
Website creation  
-Make basic website framework with chosen language  
-OPTIONAL: Make a display mockup of what the web site will look like with some basic test data

Phase 2 (10/01 – 10/31)

Data dumper  
-Research XML converters for our purposes  
--If none found, create one  
-Create data importing mechanism to populate database

Web application  
-Finalization of web-application architecture  
-Linking the database to the web application   
--Choosing an insulated means to do so within our chosen framework (if possible)  
--Developing and implementing the link between the database and the web application  
-Creation of CWE display page  
-Population of display page (with 'home made' test data if database is not populated yet)  
-Creation of CWE search page (may only be a mockup if prior to search the search engine).

Rudimentary search mechanism  
-Creation of rudimentary search mechanism superior to that of the current CWE site.  
-Implementation of rudimentary search mechanism.  
[KEEPING IN MIND that the search mechanism will be greatly revised in phase 3, this is just a stopgap "Minimum requirements" sort of thing].  
  
  
Phase 3 (11/01 – 11/30)  
-Full text search algorithm  
--Conception  
--Implementation  
-Ranking algorithm  
--Conception  
--Implementation  
-Filtering mechanism  
--Conception  
--Implementation  
--Updating UI to accommodate its use  
-Text mining sub-algorithm  
--Conception  
--Implementation  
-Holistically combining the search mechanisms into one amalgamated whole  
-Conception of back-trace (BT) log subsystem  
-Implementation of BT log  
--Coding of BT log mechanisms  
--Usage of BT log on UI  
-Input Validation (may already be implemented as part of security sub-project)  
--user input validation  
--defense against SQL injection  
--insulating users from internal system knowledge

As these tasks are approached, they will be discussed and broken down further. This is the rough WBS as per the system architecture and dependencies, as well as the priorities of the project requirements.