NSF System Build & Alpha Deployment Process Document (Version 1)

Collaborative Research: Deep Insights Anytime, Anywhere (DIA2) – Central Resource for Characterizing the TUES Portfolio through Interactive Knowledge Mining and Visualizations

Objective

The purpose of this document is to ensure a smooth design, build, and alpha deployment process for DIA2 – NSF System. Our goal is to deploy an alpha version of the system should be deployed within NSF by the ***3rd week of July***.

Design Process

1. Data Schema Preparation

There are multiple elements that need to be in place before the DIA2 NSF System Build and Deployment can occur. Data is the most important piece. As of May 11, 2012 – the DIA2 has received the Entity Relationship Diagram (ERD) and a Data Dictionary for reading this ERD from NSF. This is not the actual data – but, descriptors of the data and structure thereof. Both documents are available on Dropbox under the folder *Data\_Related\_Items*. Additionally, NSF has also provided us with a data description of an internal monitoring system used for TUES. Based on these documents, the DIA2 will prepare a database schema to acquire data into the DIA2 system that will form the basis of all views. We will implement stubs that will allow us to expand the database as needed to accommodate new features that may emerge in future discussions. After the initial database ERD is completed, the technical team will review the ERD for accuracy, scalability, speed, and optimization. Looking at normalization of the database entities ensures data and referential integrity and also achieves the attributes outlined above. Our design will be close to 3rd normalized form (3NF). For members of the DIA2 team who would like a bit more background on the process of normalization – please refer to <http://en.wikipedia.org/wiki/Database_normalization>. We also emphasize that experience plays a significant role in determining and designing the 3NF form of a database. Therefore, members of the technical team will carefully review for 3NF from their experience and report back any concerns and issues.

1. Data Acquisition

Once the data schema has been finalized, we will order a set of machines that will be inserted into the NSF network. The parameters of the machine stack’s positioning within the NSF network and deployment are outlined in Appendix A of this document. The DIA2 team (Purdue portion of budget) will purchase 4 machines and 1 Gigabit Cisco Switch for network use. The machine allocation is as follows: 1 Data Server, 1 Middleware Computing Server, 1 UI User Facing Web Server, and 1 Backup Data Server. All machines will be brought to NSF for data acquisition and laid out in the network.

The DIA2 student team led by Hanjun Xian will travel to NSF to connect the machines and also to connect to the NSF database. ***Our team will have access to the data only when we are physically inside within NSF. Once we step outside*** Therefore, we will connect the student team will spend a maximum of a week at NSF to write adapters to the NSF data and oversee the transfer process. Once the data is in place – the DIA2 team will put in place automatic data refresh mechanisms that will occur on a pre-determined (probably daily – but, no worse than weekly) basis. Before leaving NSF – the DIA2 team will test out initial querying and also read of the database.

1. UI Design (UI Team – Please edit as you see fit to add more details)

Based on input from members of the UI team thus far, the DIA2 team will first finalize a list of features that will be included as part of the initial alpha build. These features will be based on negotiations between the user team and technical team taking into consideration existing user research and technical resources. One or two face-to-face meetings will be held (perhaps as part of the DIA2 All Hands Meeting and one prior or after) to identify and finalize the features for the alpha build. Once these features have been finalized, the interaction designer (Cheryl Qian) to create screens and storyboards of interactions. As these are created, there will be a need for continuous input from the user team and the technical team (what do the users want? what is possible technically?). This could be in the form of intensive daily AM design review meeting followed by design activity itself. The final product of this activity is high-level user views, screens, and interaction flow storyboards that will be used for the technical build (coding the system).

1. Technical Build or coding the system

Once the UI design recommendations and guidelines are completed, the technical builds will begin. The technical team will develop and distribute a Common Coding Standards document that will be used by all for developing code that will be eventually deployed. Also, the technical team will setup a code repository check-in/check-out mechanism to version control the code base. Details of how to check items in and out of the system will also be distributed to the build team. At this point, there will be a feature freeze implemented – meaning, any new features that are non-critical that are added will be rolled into future builds. The code will be developed in a development environment that resembles the deployment environment. Prior to the coding – the technical team will also develop a brief catalog of algorithms that will allow users to A Hackathon – where all graduate students working on the build are in the same place and are actively coding will be hosted at Purdue University (if necessary at NSF to work with the data elements) in the second week of July. Daily builds will be tested by the user team to ensure that the views are consistent and are indeed what the users expect. If working outside NSF, we will role the changes over to the NSF system and recruit friendly users to test the system.

1. Quality Assurance

In order to ensure that the quality of DIA2 deployed within NSF is indeed functioning as planned, the user team in collaboration with the evaluation team - using a series of usability studies, observations, and other methods will study and evaluate the use of the system. Further details on this activity are available within the proposal. The evaluation team will monitor communication, processes, and the deliverables to evaluate the efficacy of the processes and document their observations. A report about the process will help the team refine this process for the

Task Breakdown

KM – Krishna Madhavan, MV – Mihaela Vorvoreanu, NE – Niklas Elmqvist, CQ – Cheryl Qian, AG – Amanda Griffith, AJ – Aditya Johri, AW – Alan Wang, NR – Naren Ramakrishnan, AM – Ann McKenna, HC – Helen Chen, SdS – Sheri Sheppard, GL – Gary Lichtenstein, HX – Hanjun Xian

|  |  |  |
| --- | --- | --- |
| Task Details | Responsible Team Members | Target Completion Dates |
| Spec and order the machine | KM, AG | May 14, 2012 |
| Create initial ERD and associated data dictionary | HX (and students as needed) | May 21 ,2012 |
| Review of the ERD and data dictionary | KM, AJ, NE, AW, and NR | May 24, 2012 |
| Revise ERD and data dictionary as needed | HX, SS | May 26, 2012 |
| Final approval of ERD and data dictionary | KM, AJ | May 27, 2012 |
| Setup student travel details to NSF | KM | May 27, 2012 |
| Student team works on data acquisition at NSF | All students led by HX (AJ supervises in KM’s absence) | May 28 – June 1, 2012 |
| Machines deployed within NSF | Students working with NSF staff | June 1, 2012 |
| ASEE data collection on community aspects | AM, MV, AJ | June 10 – June 14, 2012 |
| *Discussions on UI design (design activities) – User Team please add details to this* | *MV, CQ* | *Week of June 18, 2012?* |
| All Hands Meeting | All | Week of June 25, 2012 |
| *Development of story boards and interaction flows begin* | *CQ, MV, AM* | *Week of July 2, 2012?* |
| *Initial set of story boards and screens delivered* | *CQ, MV, AM* | *Week of July 9, 2012?* |
| Common Coding Guidelines released | KM, HX | June 18, 2012 |
| Code Repository | KM, HX | June 18, 2012 |
| Hackathon - Coding begins | All students and PIs as needed | Week of July 16, 2012 |
| Initial rollout of code to NSF system | All | Week of July 23, 2012 |
| User testing begins | MV, AM, HC | Week of July 30, 2012 |

Appendix A

NSF IT posed the questions and we responded with the answers in italics.

1. How often will the servers be accessed by humans that need a monitor, keyboard and mouse
2. Ideally this answer is “only in emergency”.
3. If management by humans on keyboard, mouse and monitor is required often, the project will need to buy an appropriate KVM device. – Purdue will bring their own monitor and keyboard for physical access if required.
4. The servers will be placed in the cage, still a part of the data center, but with less limited physical access.

*Physical access*

*Physical access is only preventative/corrective – estimating .5/year*

*Code can be updated via internal network or NFS access by NSF or ITBSS contractors.*

1. What remote access protocol will the servers be managed with?
2. We provide an SSH gateway, secured with SecureID. A security review of SSH access would need to be completed, if the admin staff are not NSF employees. Our firewall accepts connections on port 80 and port 443, however we allow access through those ports to a limited number of servers. After determining need, this list might be modified to include DIA2.

*Remote access*

*SSH Standard 443 is preferred*

1. Is it possible to conglomerate all the network ports from the mini’s onto a “dumb” gigabit switch, or switches, and not suffer performance degradation?
2. We have a limited number of network ports. If we can implement a “star” Ethernet topology, with one or two trunk lines going back into our corporate switches, this would be beneficial to future projects to be implemented into the Arlington data center, by reducing resource use.
3. The project will need to supply the switches.

*Switch*

*Yes Purdue will provide a switch as specified by NSF. Currently Purdue uses generic Cisco gigabyte switch.*

1. How will DIA2 handle moving the servers, twice, if NSF renovates this building, or once, if NSF moves into a new building, both evolutions ending in Spring of 2015?

*NSF Move*

*The move is out of scope from this conversation.  It will be handled as part of the move.*

1. The combined power requirements for the mini servers is about 900 watts. Can we combine all ten power cords onto one power strip, and plug the strip into filtered, UPS power?
2. This would be beneficial to future projects to be implemented into the Arlington data center, by reducing resource use.
3. The project will need to supply the switches.

*Power*

*Standard, one power strip/two if requested.*

*Purchased Servers – DIS PM will coordinate*

Appendix B

|  |  |  |  |
| --- | --- | --- | --- |
| Required Feature | Implementation/Solution | Difficulty (1 – Easiest, 10 – Most difficult) | Priority - Higher the score – easier to implement and shorter term implementation |
| Expand Search Scope (Search the whole document, the project reports, the annual reports and the final report) | Indexing using 3rd party library such as Apache Lucerne. General search category. Full-text index Extract the data from PDF to database and perform database search Full text indexed using Lucene or related libraries; UI autocomplete using jQuery.autocomplete; A refined parser and extractor for getting specific data field from full text Provide structured data | 6.17 | 2.5 |
| Search within Specific Parts | General search category. Faceted search (see below). Categorized search/dashboard/time-stamped search snapshots Show them the structure of the database and allow structure base search For Full-Text Search we can using Apache Lucene. Advanced search; Page content can be filtered by a composition of conditions | 5.00 | 3 |
| Offer Flexible Search Criteria (Advanced) | General search category. Also allow for auto-complete of search strings (compare to Google Suggestions API) as well as faceted search (interactive drill-down into a multidimensional dataset). Saving searches or saving queries or the future. User specified correspondence/name disambiguation We have to brain storm what advanced search criteria are required! Onwards we can execute queries based on these parameters to get refined data from our database. Advanced search; Page content can be filtered by a composition of conditions | 5.40 | 3 |
| Search by Program (program element code) | General search category. Faceted search (see above). We have the solution implementation already We can search by program by firing queries with parameters related to specific programs. Make program element code searchable. This data field is stored in the db now. Have system recognize search patterns | 3.75 | 3 |
| Cover all Possible Aliases Automatically | Name disambiguation (work by VT and others). **Need more clarifications on proposal types** Deploy some simple aliases matching algorithm with the database We can use key word disambiguation algorithm.  Machine recommendation and human curation  Work with the VT team to extend the name disambiguation algorithm for resolving any entity; A long-term solution for involving novice Internet users for supervised name disambiguation and validation Provide possible aliases for users to check; Have system recognize search patterns | 6.00 | N/A |
| Export Data & Charts to Excel | At least as CSV output (can be imported into Excel). Maybe Purdue already has solution.  Percentages provided along with simple counts/summation Database related stuff We can use tools like BI publisher to make reports. But integration with these tools may require time and effort. Usually these tools are used in enterprise applications and are quite robust and flexible. Improve existing algorithm for supporting chart production Actually, it is not EXCEL specifically, but a structured and easy-to-use data export format | 4.20 | 3 |
| Understand Different Proposal Types | **Work with the NSF on this.** Continuing topic. Text categorization by topics (Wordonoi) Use different visualizations for generating understanding We can design schema tables that’ll carry proposal types and further these can be referenced in other tables. Flag different proposal types with different values in the db; Understand which data is valid and reliable on which document; Resolve data inconsistency across different documents  The current way to do categorization is by reading publications or by personal experience. To have a systematic categorization, an updated database and semantic analysis to mimic the manual process would be helpful. | 4.40 | 3 |
| Normalize Information | Need more clarification on other categorization requirements. **Depends on the complexity, more research may need to be done.** Assuming that it’ll require change of DB design Understand what measure is considered essential by the users | 5.50 | 1 |
| Smart Categorization | We consider this as part of the solution to the flexible search. May need to build a complete profile for each organization. However, the level of details in a profile is to be determined.  More specific database design Assuming it will require creation of new DB tables to store the categorization data. Creating a taxonomy/folksonomy based on past proposals and user input; Topic modeling algorithm; Possible to have the existing taxonomy used by NSF ( | 6.60 | 1 |
| Define Organization Properties for Search | General search category. Faceted search (see above). See the comments for “impact across all disciplines” We can come with attributes describing different organization (id, name, type etc) so as to make search easier. Not sure how we define organization properties | 6.50 | Priority to be determined |
| Evaluation by Programs | In the short term, simple productivity measures can be provided such as publication count, citation count, patent count, etc. Program specific search results We have to write complex queries and have to put forward as an input to GUI components. Need to define criteria for evaluation. More user input may be required | 5.75 | 2 |
| Impact across all disciplines | Very broad task, more of an overview task and requires more research. On the long term, need to do more research on how to understand the impact across all disciplines. Database related query considering different disciplines and using visualization Again we have to write complex queries and have to put forward as an input to GUI components. Basic scholarly impacts such as citations, publication production, and social network metrics can be easily measured. But more sophisticated measures may be needed. | 6.60 | Priority to be determined |
| Accurate Reports | Depending on the quality of the report data that we are going to get, it may  N/A Prior to generating these reports we can write stored procedure to filter the data prior to reporting it. But we have to brainstorm on the validation that needs to be done prior to reporting it. Validate numbers with the NSF internal database; Resolve data inconsistency across different documents Provide traceable data | 5.33 | N/A |
| Easy to Integrate Data | May define a universal document interface and creating a mapping interface when importing files in different formats.  To make a unified database design it’ll take tremendous amount of task to get it done. Provide structured and uniform data format | 5.50 | 3 |
| Historical Knowledge | **Not sure what the challenge is here. Is this data not freely available?** Display keyword/phrase/topic frequencies over time; text categorization/visualization tools can also be used here. Display field related knowledge, Wikipedia type information  Have to come-up with range queries to extract historical data to feed the GUI components. | 5.25 | 2 |
| Historical Portfolio Available | See above. Funding portfolio statistics over programs, divisions, geographical locations, and time. It’s not technically challenging. But the development takes time.  Not clear with this functionality (Need to discuss) Support archiving historical snapshots of data and data representation; Having a unique URL to link to a page even when the data on the page is no longer up-to-date and valid now | 4.67 | 2 |
| PDF searchable | May need to use 3rd party solutions for this. Again, we may have a solution to this. This may belong to “easy to integrate data”. We may convert pdf files into text files first and import them into the system. The pdf conversion process may not be perfect. But the efficiency after conversion would override the problem.  For Full-Text Search we can use Apache Lucene Full text indexed using Lucene or related libraries; A refined parser and extractor for getting specific data field from full text | 7.67 | 1 |
| Analysis chart (like EXCEL pivot table) | Excel PivotTable is essentially an OLAP datacube interaction. This would be an excellent way to interact with this data. We have existing work on ScatterDice/GraphDice for visualizing this type of data. Add more visualizations related to analysis We can write Specific Queries for it on-wards using visualization API’s to make analysis chart. Create the pivot table in Excel; Create a similar visualization on the UI | 4.33 | Priority to be determined |
| Takes all separate factors (PI transfers, supplements) into account | Need access to the StarMetrics database for this. It should not be technically challenging. But it may take time to figure out all different factors. Assuming that these are the factors we have to consider while querying up the data for reporting, this will lead us to create queries based on these parameters (Need more clarification for this functionality). Add additional fields to save such info | 5.75 | 1 |
| Intuitive buttons instead of weird strings of letters/numbers to define query | Simple interaction design changes (we even have an interaction designer for this?). Want a consistent and clean design. Should be part of the expanded search function Adding more button options on the web page We can write listener under these UI components, which will fire respective queries. Use both icons and text for creating buttons; Need user feedback for usability interaction design & data visualization | 2.80 | Priority to be determined |
| Provide search recommendations to the keywords that have aliases | See above on a suggestions/recommendation search mechanism. Can be extended to general search recommendations, not just for aliases. Based on name disambiguation, this should be simple. We have to dig our DB tables keeping in mind keyword disambiguation to get recommendations. Use the user-defined keyphrase-binding system; UI autocomplete using jQuery.autocomplete | 4.00 | 3 |
| Handle portfolio with broad goals and shifting categories | Need clarification Visualization to show trend in changing categories Creating a taxonomy/folksonomy based on past proposals and user input; Topic modeling algorithm; Associate one proposal with multiple labels | 5.50 | 1 |
| Use programs as nodes | Node-link visualization Provide comprehensive data to identify overall impact of the proposal | 3.00 | 1 |
| Live updating tree of topics | Extract a hierarchy of topics or ontology from the existing data, i.e. something of a folksonomy? New research needed here. It will take time to fully test the effectiveness and scalability of the algorithm Real time database queries This also seems to be related to GUI (I cannot comment on this area, need to discuss it) Update the taxonomy/folksonomy based on incoming data and user input; Timestamp and archive historical taxonomy/folksonomy | 6.20 | 1 |
| Don’t hide the transparency | Improve the provenance of information for all databases. Always provide information about source documents on which the analysis results are based Here probably we have to connect all the names together so that it’s clear that they have connection. It might require querying data from our database to know what is the exact relationship. Represent the underlying workflows for getting a specific data representation and data in a visual and interactive way; Allow users to adjust the workflow and conditions and get the new data | 5.00 | 3 |
| QuickBooks of portfolio – a clear high level portfolio analysis dashboard (like Google Analytics interface) | Dashboard functionality, perhaps using the Visualization Mosaics approach proposed by Stephen MacNeil) and Niklas Elmqvist for EuroVis 2012. Define role-based/personalized dashboard based on the user team’s persona studies. Add many different visualizations Again this is related to creating a GUI. Since this GUI is going to be highly interactive and could be the center/entry point for a user, this will require good effort to create. Create a landing page for each user based on their identities and allow to customize it by adding visual components that are useful to them | 7.00 | 3 |