Software Design and Evolution

Genesis: Timeline

Below we show our estimated timeline for the project.

Table Legend:
All Groups = A
Lemma & Zulian = LZ
Eberli & Ponzanelli = EP
Babazadeh & Minelli = BM

<table>
<tr><th colspan="2">Task</th><th>W1</th><th>W1</th><th>W2</th><th>W2</th><th>W3</th><th>W3</th><th>W4</th></tr>
<tr><td rowspan="3">Data Extraction</td><td>Bug Tracker</td><td rowspan="3">EP</td><td rowspan="3">EP</td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="10">INTEGRATION</td><td rowspan="10">REFINEMENT & TESTING</td><td rowspan="10">BACKUP & TESTING</td></tr>
<tr><td>Authors</td></tr>
<tr><td>Version Logs</td></tr>
<tr><td>Meta-Model Extension Integration</td><td></td><td>LZ</td><td>LZ</td><td></td><td></td></tr>
<tr><td rowspan="2">Code Model</td><td>Parser</td><td rowspan="2">LZ</td><td rowspan="2">LZ</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Tree Extension/ Extraction/Interpretation</td></tr>
<tr><td>Metric Extraction</td><td></td><td></td><td>A</td><td>A</td><td>A</td></tr>
<tr><td>DB</td><td></td><td>BM</td><td>BM</td><td></td><td></td></tr>
<tr><td>Browser Sandbox</td><td>Visualizations</td><td></td><td></td><td>A</td><td>A</td></tr>
<tr><td>Meta-Model Format Translation</td><td></td><td></td><td></td><td>BM</td><td>BM</td></tr>
<tr><td>Service</td><td></td><td></td><td></td><td>EP</td><td>EP</td></tr>
<tr><td>Query Engine</td><td></td><td></td><td></td><td>LZ</td><td>LZ</td></tr>
</table>

Each week is divided in two slots to allow each team to first plan the job and then implement it or refine the initial design.
More in detail, below there is a brief description of each task:

Data Extraction:
The goal is to crawl SVN repositories in order to be able to download all the revisions, extracting the needed information (i.e. commit comment). Then, Crawl bug trackers (i.e. Bugzilla and Jira) in order to identify and extract the required information that will be integrated in our extended, FAMIX-like, meta-model.

Meta-Model Extension Integration:
The task is to integrate in the actual code model the different extensions that all the different groups need.

Code Model:
The goal is to design and implement our own extended meta-model to be used by all the groups in the project. We also want to be able to import a given FAMIX meta-model (i.e. in the .mse format) and extract the necessary information. The meta-model should be flexible enough to allow new features to be added during the project. Also the capabilities to extract information from the model should be provided.

Metric Extraction:
This process is basically an activity in which each group is going to define which metrics are needed for the different typologies of analysis that they want to perform. This is a process that lasts for different weeks, as it is possible that different new analysis are planned and implemented in different weeks.

DB:
The goal is to devise a meaningful yet flexible DB schema in order to support the extensions to the meta-model decided by the other groups. We should choose the appropriate database model (either a relational database or a document-oriented database), prepare a means to store the data properly and make hypothesis on the metrics to extract.

Browser Sandbox:
The idea is to develop a common platform to do visualizations in the browser (this is the first client typology that we want to support). This will be done by all groups, as it is really important to have a platform able to support all the different visualizations that each group wants to create. The platform has also to be flexible enough to support the addition of new visualizations.

Meta-Model Format Translation:
The task here is to create a mapping between the meta-model and a format to be understood from clients. Most probably we will use the JSON format. By the end of the week we want to have already a simple flow from the service to the client.

Service:
The goal is to define and implement a service to extract information from the database and being able to provide these information to the query-engine. This will be done by being able to translate the original meta-model in the format designed by the other group.

Query-Engine:
The query-engine is divided in the server-side part which maps the requests of the clients doing some pre-processing and passes them to the service.
The majority of the query-engine part is located client-side and enables the clients to request data incrementally (basically for performance reasons), keeping track of the history of queries.

Integration:
We want to reserve one week for integrating all the parts together. We plan to have a working prototype before starting integration, but this step is necessary to correct all the possible incompatibilities.

Refinements & Testing
In this phase all the necessary refinements will be performed. Although we plan to start testing all the features from the beginning, we will do extensive tests sessions during this part of the project.

Backup & Testing
We will continue testing. We also use this time for eventual last-minute fixes.