HIT3311 - Software Deployment and Evolution

Assignment 2 - Team Assignment, Due: Week 12 (Friday)

Worth: 15%

Team Size: The size of the team can be 1, 2 or 3 students.

Aim

The aim of this assignment is to study the evolution history of software system in order to (a) understand the typical patterns of evolution, and (b) apply this understanding to infer architectural choices, key milestones as well as the development method used to build these applications.

The software system that needs to be studied is:

Azureus Bittorrent Client (https://sourceforge.net/projects/azureus/)

Assignment Task

You must complete the following tasks in this assignment and write a report capturing your findings as well as discuss your observations:

(a) Study the latest version of the software systems by reading user documentation as well as using the software in order to identify their key purpose and core features. You must present your findings indicating the key purpose and features of the software systems analysed.

(b) Change Log Audit: Analyze the change logs of Azureus and identify the key milestones for these products. The purpose of this analysis is to determine how the software has evolved and gained new features over time. Specifically - 'Did the software provide most of the key features identified in early versions or were these features added over time'. You will need to present the findings that will show if the evolution was focused on refining the features, or adding/removing new functionality.

(c) Change Log Audit: Audit the change log/Subversion log of your team project, Azureus and recommend improvements to the logging process. This task involves analyzing the logs using the Chapin model [2]. You will need to present an argument for [or] against the Chapin model [2] in helping improve the quality of the change logs.

i. If you analyze the team project, you must select 20-30 log entries for analysis.

ii. If you select the Azureus changes recorded for any two releases have to be identified within the change log and the logs for these versions have to be analyzed.

(d) Growth Analysis: Analyze the growth of Azureus at different levels of abstraction. You must perform a comparative analysis of growth as measured by the number of classes, number of methods, number of fields and the number of public methods. You must use calendar time in your growth analysis.

(e) Change Analysis: Analyze the distribution of software measures of both software systems and how these distributions change over time. Research by Vasa et. al. [1] has shown that popular classes tend to increase in popularity over time. You must verify if this claim holds for the two software systems that you are analyzing. The data set that you have been given does not include the systems that were used by researchers in [1].

(f) Infer information about the development method and architectural style based on the analysis performed in previous tasks. You may add additional analysis as discussed in the lectures, as well as your own methods in order to identify the architecture style/development strategy.

Note: All teams (independent of size) must attempt all tasks listed above. Single person teams need to investigate Azureus only.

Report

You must present your findings in a single report summarising the findings from the tasks. The report must use a readable font (12-point is ideal) and must be under 25 pages, excluding appendices.

The report must contain the following broad sections:

1. Report Overview: A brief summary of what this report contains, including a summary of your main findings . (max. 1 page)
2. Software Investigated: This section must provide a brief overview of the software systems under investigation for this assignment. The primary function of the software system and the main features must be presented. (max. 1 page)
3. Feature Evolution: This section should present how the features evolved in these software systems. You must answer the key question specified in Task (b). (max. 3 pages)
4. Change Log Audit: This section should present the Chapin model (including its strength compared to the simpler Swanson model as well as known limitations), the method and input data used for analysis, observations and a brief discussion. You must make an argument if the Chapin model can help improve the logs analyzed. You must state any limitations of your analysis approach in this section as well. (max. 3 pages)
5. Growth Analysis: This section must outline a short summary of Lehmans' laws and the expectations as set out by these laws (specifically, the growth rate). You must also present the analysis technique that you have used, followed by your findings and discussion of your observations. The technique that you use must allow comparative analysis. Your discussion should relate the observations to Lehman's laws initially and then cover other aspects to support your observations. (max. 4 pages)
6. Change Analysis: This section must present a short summary of the analysis technique that you have used followed by your findings and discussion of your observations. (max. 4 pages)
7. General Discussion: This section must present a discussion where you can put forward your inference of the development method and architectural style based on your analysis of the data provided. If you use any additional analysis you can create an additional sub-section to explain your approach and findings (max. 5 pages)
8. Limitations: This section must present a list of known limitations within the context of your study. (max. 1 page)
9. Appendix: Must include the sub-set of change logs that have been analysed, raw growth and change data used for analysis as well as a table with a short explanation of Lehman's laws.
10. References (References and citations must use either Harvard or IEEE style)

Final Report Submission

A printed copy of the report must be submitted into the Assignment Box located at E Level 1 (E201) in an A4 envelope with the subject code, convenors name, the team members names (as well as the Student IDs) on the cover sheet. The printed report should have a title page with the team members names and their Student IDs. All pages must be numbered. Penalties will apply for late submissions, refer to the Unit outline for details.

Progress Submission

All teams are encouraged to present their progress against all tasks in Week 12 during the tutorial. Teams should target to have the following information ready for the progress presentation:

1. Key features, Milestones and Feature evolution timeline for Azureus [Draft-- no discussion is needed at this point in time]
2. A print out of the sub-set of change logs that will be used for Task (b).
3. Charts of Growth, including the fit lines (quadratic or linear), equations as well as the R-squared values. Interesting outliers / versions that need further investigation in change logs.
4. Charts supporting change analysis. Versions that need further investigation in change logs.
5. Appendix

Progress submissions is optional, but have the advantage of providing teams with useful feedback.

Recommended Tools

- Excel (but GNU Plot will also work)
- Java Decompiler (one of the following):
    - DJ Java Decompiler (http://members.fortunecity.com/neshkov/dj.html)
    - Jad (http://www.varaneckas.com/jad)
- Eclipse, or Netbeans or any POPE [Plain Old Programmer's Editor]

The Java decompiler is helpful to understand additional information about the source code. This may be needed to confirm your hypothesis.

References

1. Rajesh Vasa, Jean-Guy Schneider and Oscar Nierstrasz, "The Inevitable Stability of Software Change," Proceedings of the 23rd International Conference on Software Maintenance (ICSM 2007), IEEE Computer Society Press, Paris, France, October 2007, pp. 4—13.
2. Chapin, N. and Hale, J.E. and Khan, K.M. and Ramil, J.F. and Tan, W.G., "Types of software evolution and software maintenance", Journal of Software Maintenance and Evolution: Research and Practice, 2001, John Wiley and Sons, Ltd. Chichester, UK.

