EMSS 9

MEASUREMENT PLAN

Amol Shandilya, Yanyan Jiang, Minghui Jin

The Booch Group

1. Development team

Subject	Description
Customer	Developers
Goals	The code that the developers write shall be easy to read and understand, which would help the communication between different development teams. On the other hand, the complexity shall be as low as possible to reduce the emerge of errors.
Metric 1: Cyclomatic Complexity	Cyclomatic complexity could precisely measure path complexity. Various studies over the years have determined that methods having a cyclomatic complexity greater than 10 have a higher risk of defects. So that when any component with cyclomatic complexity over 10 should require a formal review. In terms of measurement tools, JavaNCSS is an open source project as a tool used for Java platform that can generate cyclomatic complexity report. And PMD and CheckStyle are also open source tool that could analyze Java source files with cyclomatic complexity rule. What's more, the PMD and CheckStyle also contain several other kinds of rules, such as code depth, and they could also use user-defined rules.
Metric 2: Code Coupling Degree	Code coupling degree concerns about the afferent coupling, efferent coupling, instability, abstractness, agility, etc. We could use multiple code analyzers to report these characters, such as JDepend, JarAnalyzer and Metrics plug-in for Eclipse. This report would help find out components' unnecessary reliability based on code coupling degree and system maintainability based on abstractness and instability.
Metric 3: Fog-Index	Fog-index can be used to measure the code readability. Based on the Gunning's Fog Index (FOG) Readability Formula, the Fog-index could be calculated. The ideal score for readability with the Fog-index is 7 or 8. Once the score is over 9, then the code should be formally reviewed.
Metric 4: Depth of Inheritance	Depth of inheritance is defined as "the maximum length from the node to the root of the tree". The deeper a class in the hierarchy, the

	greater the number of methods it will probably inherit which makes it harder to predict its behavior. And deeper trees involve greater design complexity since more classes and methods are involved. But deeper classes in the tree have a greater potential for reusing inherited methods. When using code analysis, the Extended Design Guideline rule set contains a maintainability area. A depth-of-inheritance around 5 or 6 should be an upper limit, if over 7, then the code needs to be formally reviewed.

2. QA team

Subject	Description
Customer	Testers
Goal	To test codes and anomalous components, find defects and problems to avoid system failure when delivering services to customers. And use the collected data for improving the whole test process.
Absolute Metrics	1. Total number of test cases. 2. Number of test cases passed, failed and blocked. 3. Number of defects found, accepted, rejected and deferred. 4. Number of critical defects. 5. Number of planned test hours and actual test hours. 6. Number of bugs found after shipping. 7. The life-time of the system. 8. The response time of the system.
Effectiveness Test Metrics	Test effectiveness metrics usually show a percentage value of the difference between the number of defects found by the test team, and the overall defects found for the software. The higher the test effectiveness percentage, the better the test set is and the lesser the test case maintenance effort will be in the long-term.

Coverage Test Metrics	Software quality metrics gauge the health of the application that are under test. Inevitably, the next core set of metrics that you want to analysis revolves around coverage. Test coverage metrics measure the test effort. It is usually presented as a percentage value.

3. Investors

Subject	Description
Customer	Project investors
Goals	Mainly concerned about the budget and the manufacturing period, including the planned budget, the actual cost and the overrunning budget, the planned release time and the actual release time, they would want these to be less and shorter.
Budget Metrics	1. Development team. Measure the person-days cost based on individual components. 2. Test team. Measure the person-days cost based on individual test cases. 3. Maintenance team. The maintenance team including two components: the bug-fix team and the support team (dealing with the help desk and failure reports through internet). The former one’s cost is calculated by the dollar amount of effort spent on a defect per developers. The latter one’s is measured by the number of persons involved. 4. Office resources cost & Computer resources cost. 5. Proprietary cost & Open source. This cost comes from the spending of buying patents, and the person-days spent on finding, re-engineering, integrating the open source. 6. User testing budget & Advertisement budget.
Manufacturing Period Metrics	Application development cycle & Testing cycle & Debug cycle. Use the Gantt chart to schedule the project, illustrating the start and finish dates of the terminal elements and summary elements of a project, including the work breakdown structure of the project. It also shows the dependency (i.e., precedence network) relationships between activities.

4. Support team

Subject	Description
Customer	Support team
Goal	To help maintain the system by systematically capturing failure information and categorizing them correctly and leading to quicker resolution times. Also, to help improve customer retention and brand image.
Metrics	Support team deals with help desk calls, the failure reports which are sent over the Internet from the user’s computer to servers run by the product developer. Use Ticketing System, which can offer a range of capture mechanisms and the ability to control, to deliver an integrated and seamless response.