



SOFTWARE QUALITY ASSURANCE

Lecture 5 (20-March-2013)

Instructor: Mr. Natash Ali Mian

Department of CS and IT
The University of Lahore



Switch off mobile phones during lectures, or put them into silent mode



CONTENTS

- Term Paper
- Quality Measurements





Please Obey Traffic Signals



Donate **blood**



to save lives



KEEP YOUR SURROUNDING CLEAN





Give Way to **Ambulance** because its destination is more important than yours

TERM PAPER



- Finalize Group Members 26-Feb-2013
- Finalize Topic 12-Mar-2013
- **Search Papers and Sort Selected (TODAY) 20-Mar-2013**
- **Go Through the Abstract and Introduction of Selected Papers 26-Mar-2013**
- Submit a Summary and Comments on related papers **TBD**
- Submit Initial Draft **TBD**
- Final Paper Submission **TBD**
- Final Presentation **TBD**

Please note that Every Phase has Marks



QUALITY MEASUREMENTS

QUALITY MEASUREMENT QUESTIONS

- What should be measured for quality?
- How often should quality measurement be taken and reported?



QUALITY MEASUREMENT CATEGORIES

- Measurement of defects or bugs in software
 - 100% of software projects
- Measurement of user-satisfaction levels
 - Only for software projects where clients can be queried



SOFTWARE DEFECT QUALITY MEASUREMENTS - 1

- Defect volumes (by product, by time period, by geographic region)
- Defect severity levels
- Special categories (invalid defects, duplicates, unduplicatable problems)
- Defect origins (i.e., requirements, design, code, documents, or bad fixes)



SOFTWARE DEFECT QUALITY MEASUREMENTS - 2

- Defect discovery points (i.e., inspections, tests, customer reports, etc.)
- Defect removal efficiency levels
- Normalized data (i.e., defects per function point or per KLOC)
- Other factors (i.e., complexity, creeping requirements, etc.)



SOFTWARE DEFECT QUALITY MEASUREMENTS - 3

- Defect repair speeds or intervals from the first report to the release of the fix



SOFTWARE USER-SATISFACTION QUALITY MEASUREMENTS - 1

- User perception of quality and reliability
- User perception of features in the software product
- User perception of ease of learning
- User perception of ease of use
- User perception of customer support
- User perception of speed of defect repairs



SOFTWARE USER-SATISFACTION QUALITY MEASUREMENTS - 2

- User perception of speed of adding new features
- User perception of virtues of competitive products
- User perception of the value versus the cost of the package



WHO MEASURES USER-SATISFACTION?

- Marketing or sales organization of the software company
- User associations
- Software magazines
- Direct competitors
- User groups on the internet, etc.
- Third-party survey groups



GATHERING USER-SATISFACTION DATA

- Focus groups of customers
- Formal usability laboratories
- External beta tests
- Requests from user associations for improvements in usability
- Imitation of usability features of competitive or similar products by other vendors



BARRIERS TO SOFTWARE QUALITY MEASUREMENT

- Lack of understanding of need to measure quality
- Often technical staff shies away from getting their work measured
- Historically, “lines of code” or LOC and “cost per defect” metrics have been used, which are a poor way of measuring software quality



OBJECT-ORIENTED QUALITY LEVELS

- OO technology is being adopted world-wide with a claim that it produces better quality software products
- OO technology has a steep learning curve, and as a result it may be difficult to achieve high quality software
- More data needs to be reported
- UML may play a significant role



OUTSOURCING AND SOFTWARE QUALITY

- Outsourcing in software industry is done in a variety of ways
- Every situation introduces new challenges for development of high quality software
- Software quality metrics must be mentioned in the outsourcing contract



QUALITY ESTIMATING TOOLS - 1

- Estimating defect potentials for bugs in five categories (requirements, design, coding, documentation, and bad fixes)
- Estimating defect severity levels into four categories, ranging from 1 (total or catastrophic failure) to severity 4 (minor or cosmetic problem)



QUALITY ESTIMATING TOOLS - 2

- Estimating the defect removal efficiency levels of various kinds of design reviews, inspections, and a dozen kinds of testing against each kind and severity of defects
- Estimating the number and severity of latent defects present in a software application when it is delivered to users



QUALITY ESTIMATING TOOLS - 3

- Estimating the number of user-reported defects on an annual basis for up to 20 years
- Estimating the reliability of software at various intervals using mean-time to failure (MTTF)
- mean-time between failures (MTBF) metrics



QUALITY ESTIMATING TOOLS - 4

- Estimating the “stabilization period” or number of calendar months of production before users can execute the application without encountering severe errors.
- Estimating the efforts and costs devoted to various kinds of quality and defect removal efforts such as inspections, test-case preparation, defect removal, etc.



QUALITY ESTIMATING TOOLS - 5

- Estimating the number of test cases and test runs for all testing stages
- Estimating maintenance costs for up to 20 years for fixing bugs (also for additions)
- Estimating special kinds of defect reports including duplicates and invalid reports which trigger investigative costs but no repair costs



QUALITY PROCESS METRICS

- Defect arrival rate
- Test effectiveness
- Defects by phase
- Defect removal effectiveness
- Defect backlog
- Backlog management index
- Fix response time
- Percent delinquent fixes
- Defective fixes



PRODUCT METRICS

- Defect density
- Defects by severity
- Mean time between failures
- Customer-reported problems
- Customer satisfaction



FUNCTION POINT METRIC - 1

- It was developed at IBM and reported to public in 1979
- It is a way of determining the size of a software application by enumerating and adjusting five visible aspects that are of significance to both users and developers



FUNCTION POINT METRIC - 2

- Inputs that enter the application (i.e., Input screens, forms, commands, etc.)
- Outputs that leave the application (i.e., Output screens, reports, etc.)
- Inquiries that can be made to the application (i.e., Queries for information)
- Logical files maintained by the application (i.e., Tables, text files, etc.)



FUNCTION POINT METRIC - 3

- Interfaces between the application and others (i.e., shared data, messages, etc.)
- Once the raw total of these five factors has been enumerated, then an additional set of 14 influential factors are evaluated for impact using a scale that runs from 0 (no impact) to 5 (major impact)



SCHEDULE PRESSURE AND QUALITY

- Healthy pressure
 - Motivates and keeps morale of the personnel high
- Excessive pressure
 - Has serious negative impact on the morale of personnel
 - Can lead to low quality software



WHO WILL MEASURE QUALITY AND WHEN

- The SQA planners for a project are required to determine
 - The list of quality assurance activities needed for a project
 - For each quality assurance activity
 - Timing
 - Who performs the activity and the resources needed
 - Resources required for removal of defects and introduction of changes



A WORD OF CAUTION

- Some development plans, QA activities are spread throughout the process, but without any time allocated for their performance or for the subsequent removal of defects. As nothing is achieved without time, the almost guaranteed result is delay, caused by “unexpectedly” long duration of the QA process
- Hence, the time allocated for QA activities and the defects corrections work that follow should be examined



PROJECT FACTORS

- Magnitude of the project
- Technical complexity and difficulty
- Extent of reusable software components
- Severity of failure outcomes if the project fails



TEAM FACTORS

- Professional qualification of the team members
- Team acquaintance with the project and its experience in the area
- Availability of staff members who can professionally support the team
- Familiarity with team members, in other words the percentage of new staff members in the team



WHY ERROR-PRONE MODULES?

- Excessive schedule pressure on the programmers
- Poor training or lack of experience in structured methods
- Rapidly creeping requirements which trigger late changes
- High complexity levels with cyclomatic ranges greater than 15



“GOOD ENOUGH” SOFTWARE QUALITY - 1

- Rather than striving for zero-defect levels or striving to exceed in 99% in defect removal efficiency, it is better to ship software with some defects still present in order to speed up or shorten time to market intervals
- Developed by the fact that major commercial software companies have latent software bugs in their released products



“GOOD ENOUGH” SOFTWARE QUALITY - 2

- Major commercial software companies have cumulative defect removal efficiency of 95% (and 99% on their best projects)
- This concept is very hazardous for ordinary companies, which usually have their defect removal efficiency level between 80%-85%
- Quality will be decrease for these companies



DATA QUALITY - 1

- Extremely important to understand issues of data quality
- Data results in (useful | useless) information
- Usually, governments are holders of largest data banks (are they consistent?)
- Companies are increasingly using data to their advantage over competitors



DATA QUALITY - 2

- Data warehouses present a unique challenge to keep data consistent
- Another problem is the interpretation of data



Thanks!

