Abstract— Component software quality has a major influence in development project performances such as lead-time, time to market and cost. It also affects the projects within the organization, the people assigned into the projects and the organization in general. Organization must have indicators which are monitored on monthly basis and which are giving an indication about project performances. It helps the organization in the resource planning and replanning for all current running projects but also in the planning for the future projects. This paper presents content and process for fault reporting in the telecommunication software with the emphasis on the customer reports.

Keywords: Software quality, Key performance Indicators, software engineering, software component.

1. INTRODUCTION

Software quality is an increasingly important topic as software technology is applied frequently in diverse technical systems. Software measurement processes supporting the evaluation of software quality is therefore an increasingly integral part in any software development process. The motivation for software measurement is to understand, control and improve the various quality attributes of the software and the underlying development process. A prerequisite is the definition of the quality attributes and the related measures / metrics [12].

The number of faults in a large software project has a significant impact on project performances and hence is an input to project planning [1], [2].

The ANSI/IEEE definitions for quality control is measuring the quality of a product and the Quality Assurance measures the quality of processes used to create a quality product.

IEEE Standard 12207 defines QA this way: "The quality assurance process is a process for providing adequate assurance that the software products and processes in the product life cycle conform to their specific requirements and adhere to their established plans." [3].

The ISO/IEC 9126 product quality standard [4] is used to categorize the metrics.

The Software Capability Maturity Model version 1.1 states it this way: "The purpose of Software Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and of the products being built".

IEEE Standard 1061 defines software quality this way: "Software quality is the degree to which software possesses a desired combination of attributes (e.g., reliability, interoperability)" [11].

As the quality level of the final product is set at the beginning of the project, a large number of faults can result in project delays and cost overruns [5]. Planning precision and predictability is crucial for the any projects in operations Error! Reference source not found. All stakeholders are dependant on that the project deliver timely according to projects plans.

This target aims to put focus on the planning precision of the different projects phases. The focus area is planning precision in all project phases – keeping projects committed delivery dates. It is also about keeping the balance in the project (alignment between Frame size, requirement volume, project schedules, a projects capacity and financial forecasts).

Software components can be measured and compared using the amount of faults (Trouble Reports) reported on them. These data are essential in deciding process about software quality and will be used in decision about preventive and corrective actions for software components and application in general. Using and combining those data and finally defining key performance indicators (KPI) for telecommunication software, organization can monitor the performance of their own software and decide about necessary actions on time and with the customer satisfaction.

2. TROUBLE REPORT HANDLING

Large data base is needed to collect and store different kind of faults received in different projects, applications or live exchanges all over the world for Ericsson projects or exchanges. Different types of data are collected for every single applications and every single software component.

MHWeb is an integrated system on the web with components which are used in Design, Design Maintenance (DM) and Support organizations for Maintenance and Customer Support purposes [7].

MHWeb offers an easy to use interface for Trouble Report (TR) handling, solution handling, analysis and measurement. MHWeb offers functionality to search and browse of Automatic Ericsson Exchange (AXE) corrections, TRs and Correction Note Information (CNIs). It also offers analyze possibility of Software Records (SWR), and Restart Errors printouts. MHWeb also has a TR tracking component to make the tracking process of TRs easier for projects. The open architecture of MHWeb allows rapid development of new components.

There is a general process on how the TR must be written for providing information when writing a TR and must be used and help to enable customers, Ericsson Local Support (1st line support office ELS) staff (on site), design offices and verification departments, to issue comprehensive and fully documented TR [8].

TRs should be used to report suspected or established troubles in software, documentation, parameters and hardware (if systematic/repetitive troubles are suspected). The recommendations stated in the node specific part should be considered as the minimum requirements for meaningful TR analysis.
If required data, descriptions or enclosures are missing the TR might be rejected due to lack of information [9].

The figure below describes the flow of a TR and a solution from 2nd line support via the PLM and Design Maintenance (DM) organizations back to 2nd line.

Special attention should be paid to the Network level, Node level, Node component level and Faulty Product fields since this information is used for automatic routing of the TR to the appropriate support/design organization. The information in for example the Node level product field routes the TR to the correct office if the Node level-owner is connected to the appropriate Modification Handling Office (MHO). This is done when defining a product as a Node level product. The information in the Faulty product level field routes the TR to the responsible Design Maintenance office.

For development TR, attention should be paid to in which configuration/LSV (increment/build/correction package) the fault was found in. That is when a fault is found and the faulty product level is not yet registered in PRIM.

Very important information that is mandatory to be included in the TR is the Customer target date. This target date specifies a date when the TR originating organization expects the reception of the TR answer according to contractual commitments with end customer and Ericsson Global Support (2nd line support office EGS) and Product Line Maintenance (PLM) support.

Document fault shall always be written towards the documents belonging product. The concerned document number and revision shall be stated in the TR.

TRs are initially given one of three priority levels: A, B or C priority, where a priority represents the highest priority level. The priority of a TR is set based on the effects of a problem (or its symptoms) and not the actions taken to resolve the problem. The answering time requirements for Ericsson TRs and lead-time goals are defined in an operational agreement between Business Unit Global Services (BUGS). Also, the same is valid for the various Node Development Center (BUGS). Also, the same is valid for the various Node Development Center (NDC) and the design organization.

For C prioritized external TRs, normally no correction will be provided with correction package prior to the next system upgrade. TR priority is divided in different software dependent categories to reflect some of the different type of systems in use: AXE Control System (AFZ), Cello Package Platform (CPP), The Ericsson Server Platform (TSP), Wireless Package Platform (WPP), Ericsson ATM Data Communication Switch (AXD) and Engine Integral Solution (ENGINE).

The following troubles are classified as Priority A troubles:

- TRs related to node Complete Exchange/System Failures (CEF/CSF)
- Complete loss of connection processing requiring manual intervention.
- Cyclic Restarts / Cyclic Reloads.
- Severe restrictions in traffic handling capability on switch level
- No terminating connections.
- No originating connections.
- Connection processing affected in more than 30% of the traffic handling.
- Automatic System Reload / Large Restart.
- Automatic Small Restarts.
- Frequently occurring Automatic Small Restarts.

Attention should be paid in order to fill in all elements of the TR correctly:

- Administrative data (references, trouble)
-Trouble description (standardized headings)
- Observations (events observed and actions taken)
- Enclosures – Mandatory

2.1. Administrative data

Administrative data within the TR form includes:

- Heading
- Submitter Priority
- Customer/Project
- Market Reference
- Site, Country and Activity
- Network level product
- Node level product
- Node component level product
- Designation
- Faulty Product
- Configuration/LSV
- 1st and 2nd Technical contact person
- Customer Target date

The Customer Service Request (CSR) identity is mandatory when the TR is issued as a result of a customer service request. In this situation, the customer specific data (i.e., customer, country code...) must reflect the information contained in the CSR.

Specific information related to the Ericsson Technical Contact Person involved in the problem is mandatory. This includes the person’s corporate userid, email address, and phone number. 1st Technical contact person shall be the primary contact person for additional information. 2nd Technical contact person shall be the person to contact as a backup for the 1st Technical contact person.
• AXE problems, which require (i.e.: no other solutions could be found) a restart to temporarily, solve a problem, causing traffic disturbances.
• Loss of charging or accounting data.
• Complete loss of Operation and Maintenance (O&M) requiring manual intervention.
• Documentation faults leading to one of the situations mentioned above (including Data Conversion Document faults).

The following troubles are classified as Priority B troubles:
• Non-AXE, traffic disturbance that requires a restart.
• Negative impact on subscriber services (e.g. no call forwarding for a limited amount of subscribers).
• Limited O&M functionality, Operational Instruction (OPI) does not give expected results.
• Input/Output Group (IOG) related problems.
• Failure of alarms.

The priority C faults will normally not be corrected until next release of the product. The following troubles are classified as Priority C troubles:
• Documentation faults.
• Incorrect system printouts (limited to lay-out problems)
• Other minor faults with limited impact on services.
• Timing problem.
• Incorrect printout produced by script.
• Violation of Design-rules.

2.2. Trouble description

Standard headings must be used. Below is an example of a standard heading format including keywords such as FORLOPP, fault code or block name. This makes it easier to search through the TR list.

FORELOPP FC <faultcode>[ in <blockname>] <inf1> <inf2> <inf3> <inf4>

Where <fault code> is the fault CODE,
<block name> is the block name (if available),
<inf1> is parameter INF1,
<inf2> is parameter INF2,
<inf3> is parameter INF3,
<inf4> is parameter INF4 from the system SOFTWARE ERROR INFORMATION printout.

The exact format shown as this makes it easier to search through the TR list by using key words such as FORLOPP, fault code or block name. Example:
FORELOPP FC 0008 in MPAG 0266 0179 0000 0000

2.3. Observation

The TR Observation sections must be filled with accurate information clearly describing the problem, its effect on the system and how the problem was resolved.

The Trouble Effect field must indicate what network impact was caused by the problem and with what frequency the problem occurs/occurred (i.e. was a restart experienced, were there any traffic limitations as result of this problem). In this field the TR issuer might indicate possible impact(s) for Ericsson or for the Customer or both, caused by the problem.

The Trouble Description field must include a clear description of events that took place before and after the problem occurred. If the fault has been isolated, a technical description of the fault must also be given. The actions taken to overcome the problem (if any) must be stated in the Measures. If a work around has been developed (temporary solution) or a remedy that limits the effects of the problem is available it must also be included.

2.4. Enclosures

The attached enclosures must be relevant for reported problems/events (timing at log search, list of software etc.). The following items should be considered, as they are useful for the further handling of the TR:
• Always report only one problem per TR.
• Do not send secondary TRs unless they add valuable information.
• List names and telephone numbers for the technical contact persons.
• If an Approved Correction (AC) is expected, the TR’s priority must be A or B.
• Was the belonging Emergency Correction (EC) verified and/or loaded?
• Was any specific activity ongoing while the fault occurred? If yes, please refer to the used OPI!
• Is the problem site specific?
• Can the problem be reproduced?
• Is it possible to assist design with correction verification?
• Include information about the frequency of the problem
• Always specify in the notebook which WI (document number + revision) that was used when writing the TR.
• Always investigate and fill in the experienced In Service Performance (ISP) information.

To enable identification of all ECs (loaded patches), the Correction Identity must contain the same identity as the corresponding Ericsson references. This means that one TR must be written for every individual EC irrelevant if the EC is for emergency solutions, parameter setting or for any other reason.

The following items require special attention:
• Always fill in the technical contact person in the 1st Tech. Contact field.
• Fill in the Customer Target Date field.
• State which node specific TR Guidelines (TRG) has been followed in the note book
• For Plex only, always check SWR beautifier in MHWeb for known faults (new or cancelled ACs) and already reported faults
2.5. TR answer code distribution

The TR Answer Code Distribution (TRACD) measurement shows the distribution of answer codes over a specified date range. The purpose of the TRACD measurement is to assess the types of answers to TRs an organization gives over time.

The TRACD measurement shows the distribution of A, B, D, M and other answer code categories over a specified date range. A means already corrected, B means fault, D means market requirements… Only TRs that are considered answered are counted. The TR must also have an answer code. TRs without answer code are not counted.

The answer code distribution is determined by dividing the count of TRs with a particular answer code by the count of all the answered TRs for that period. The distribution is displayed as a percentage. The sum of all distributions is 100%. If there are no TRs counted during the period then the distribution for all answer codes is 0%.

The TRACD measurement presents the moving average of answer code distribution by time increment.

The following formula is used for calculating the moving average for a time increment:

Moving average = (sum of answer code distributions for the time increments to include) / (number of time increments to include)

The result from the TR Answer Code Distribution measurement is presented as a graph.

The TRACD measurement generates a graph with bar charts of answer code distribution (A, B, D, M, Other) by time increment. The horizontal axis represents a date range made up of time increments. The user selects the date range and the size of the time increments. The typical display is monthly increments over a period of one year, although weekly, and quarterly, are also selectable increments. The vertical axis represents the answer code distribution.

Overlying the bar chart are lines showing the moving average of TRACD for each time increment and answer code (A, B, D, M, Other).

3. APPROVED CORRECTION HANDLING

The purpose with an Approved Correction (AC) is to correct a fault in an identified product [10]. To provide customers with prompt solutions to problems caused by faults, it may be required to deliver and install AC on the most granular level possible.

Below is a flow that shows the different steps of the total correction handling procedure at a very high level.

Figure 3. AC workflow

For any platform, an AC is defined as a change in a product baseline due to a TR. However, the platforms use different technologies that affect the way an AC is handled, delivered and installed.

Within Ericsson technology there are products that can be corrected directly in the assembler code AC area like Plex products without having to recompile the source SW. However other technologies requires re-compilation of the original SW or load modules including the AC to the software, and the re-generated load modules then replace the existing modules in the system.

For patchable products, ACs are software items that are added to an existing product baseline. For non-patchable products, ACs are software items recompiled with the original software.

An AC is always written on one product or one configuration/LSV on the responsible Design MHO. One TR can give rise to one AC. An AC for a non-patchable product can in its’ turn have several Components connected. The AC and the Component are on different product levels.

An AC does not require a complete validation of the product.

Related ACs are connected to each other via their TR's. The ACs connected to Parent TR and its’ different Child TR's are named Related ACs and will be visible in each of these ACs. Since AC verification has to be coordinated, the AC connected to the Parent TR may not be set to...
Tested before the all ACs connected to the Child TR's are in status Tested.

3.1. Related ACs

Relationship between TR and Corrections (example):
- Parent TR => Correction A
- Child TR1 => Correction B
- Child TR2 => Correction C

The connection between Parent TR and Child TR1&2 can also be seen as a connection between Corrections A, B&C. Corrections A, B&C are related corrections. The relations will be visible in Correction A, B&C.

![Figure 4. TR and AC relationship](image)

However, the Designer may decide if this recommendation shall be followed or not, since it may be possible to verify the corrections separately.

3.2. Mapping corrections

A Mapping Correction is a correction based on a Mapping TR. Mapping TR's are used when a TR is written on a product and needs to be solved for other versions of the product. In the same way a Mapping Correction will solve the same fault as the original Correction, but in another version of the product. There will be an indication in the correction showing that this is a Mapping Correction.

3.3. Approved corrections

The term Approved Correction (AC) defines what is common to all different kinds of ACs, that is PACs, SPACs, MACs and AMCs as described below. The AC itself does not exist as a document.

An AC is a software modification that does not require recompilation of the code. The final solution is implemented when the source product is revised and then introduced into the customer's exchange at a suitable point in time.

A Primary Approved Correction (PAC) is an AC designed by and tested under supervision of the designer of the affected product. PACs are only written for released products. PACs for APT products may never be used in an exchange in service. They must first be adapted. PACs must never be released unless properly verified.

A Test Correction (TC) is a correction written for products not yet released. A TC is written and used during a test period when a new product is being verified. The TC is prepared according to the same procedure as the correction. After the test period, all TCs should be implemented in source code or transferred as corrections before cancellation by the design project.

An Approved Market Correction (AMC) is created to change the SW unit functionality and can be used for several customers. The design responsible organization is responsible for the AMC. An AMC is applicable for several customers.

A Semi-Primary Approved Correction (SPAC) is an APT correction based on a PAC and adapted for a certain target code version (APZ). The SPAC is created by copying the information from the PAC and then changes are made in the assembler (ASA), if required. SPACs are never made for APZ products. SPACs are only made for products produced by APS3. For products produced by APS2, MACs are used instead.

A Modified Approved Correction (MAC) is a SPAC that has been adapted for a specific Global Application System (GAS), Market Application System (MAS) built on a GAS, or a traditional Application System (AS).

An Emergency Correction (EC) is written to provide a remedy or restoration to a fault. The modification is implemented for a specific application system with or without the approval and verification of the designer of the product concerned. ECs are used when the delivery time is very crucial and the quality is less crucial for a correction. ECs are used as temporary measures until the fully verified solution (AC for patchable products) is available. ECs can also be used as solutions to problems in old products with ML4, for which active support is no longer provided. ECs are also prepared as temporary solutions to application-specific faults such as parameter or marshal errors. An EC can never be used as an official measure in a TR answer.

A Market Correction (MC) is connected to a trouble report, and provides new functionality that is outside normal design or has not been released as a final product. The modification is implemented for a specific application system with or without the approval and verification of the designer of the product concerned.

A Parameter Correction (PC) is created to change application-specific parameter values. The modification is implemented for a specific application system with or without the approval and verification of the designer of the product concerned.

3.4. Approved correction cancellation

When a user suspects a correction to be faulty, the correction is marked with status CSF (Correction Suspected Faulty) to inform other users. When the status is set to CSF, the user must write a TR on the affected product to inform the designer so that the necessary measures can be taken. It is mandatory to enter the TR ID in the correction. The correction, that has the status CSF, is still released and it is possible to fetch the correction from the support system.

If the TR leads to that a new correction is created, the CSF marked correction will be cancelled when the successor correction is released. If the CSF marked correction turns out to be OK the CSF mark is removed.

When CSF status is set on a PAC, all existing SPACs and MACs will also get the suspected faulty status. If the CSF status is set on a single SPAC, no other SPACs, or the PAC, will get this status, but all existing MACs will also get the suspected faulty status.
4. KEY PERFORMANCE INDICATORS

Key performance indicators (KPI) are introduced for monitoring purpose. KPIs are tools for monitoring performance in supplier’s organizations.

This enables visibility of major performance elements (Cost, Time, and Quality), and performance improvements for any organizations within the Ericsson group.

The aim is to secure that targets are met, and through evaluation of progress and benchmarking we could see how efficient supplier’s organization is, and on what level of Operational Excellence.

4.1. TR closure rate

TR closure rate (TRCR) measures supplier’s ability to answer TRs within the specified goals.

It is based on deviation between the actual TR answering times and TR goals, set by the Ericsson Assignment Owner (the measurement is also known as TR Overdue Time). TRCR is reported as lost days, averaged across priority A, B and C TRs. 0 is the lowest result, indicating that the TRs are answered within the goals.

Formula for TRCR (days) is:

$$\text{TRCR} = \frac{\text{(# of lost days within the time increment for all open and new TRs)} \times \text{# of TRs}}{\text{(# of open TRs at beginning of time increment + # of new TRs during time increment)}}$$

The TR handling time starts at the point at which the TR enters the supplier organization, and ends at the point at which the TR is answered (time spent before TR is forwarded from other organization to Supplier should be excluded from measurement). Time increment is typically 12 months in the past from reporting date. Measurement is done on a monthly basis.

4.2. Cost per TR

Cost per TR (CTR) measures supplier’s efficiency in fixing TRs (answer plus solution), i.e. maintenance costs relative to TRs resolved in man-hours. CTR is based on total hours consumed relative to TRs resolved. It is reported as man-hours. CTR target is defined separately for each product area/assignment.

Formula for CTR:

$$\text{CTR} = \frac{\text{(Cost of Maintenance activities)}}{\text{(TRs Resolved)}}$$

Cost of Maintenance activities is total hours spent on TR Handling activities (TR Administration, TR Analysis, TR Design/Fix, TR Testing, Project Management Overhead, Other TR related costs, Packaging on CP/EC, Load files).

Number of TRs resolved are TRs that include a fix/solution (Finished TRs, TRs with a fix/solution (closed), Cancelled TRs).

Measurement is done on a monthly basis.

4.3. AC cancellation rate

AC cancellation rate (ACR) measures supplier’s efficiency in providing first time right solution for the TRs.
ACR is based on total number of released ACs during specified time period (it is usually rolling year, or 12 last months). It is reported as a percentage. ACR target is defined separately for each product area/assignment. Formula for ACR:

\[
\text{ACR} = \frac{\text{# of cancelled ACs}}{\text{# of all released ACs for the measurement period}}.
\]

Measurement is done on a monthly basis.

4.4. Relationship between KPIs

All previously mentioned indicators are some hove in connected with each others. It can be concluded that:

- The higher number for TRCR means lots of pressure on the people in DM&DFU projects (project is under estimated and the number of the TR is more than expected). Also, it means that the quality of the released software products is also not good enough. It is characteristics of the very first time products (first release of the application).

- The higher number of CTR means that project is over estimated (people are reporting less TRs in their monthly work). But, on the other hand it means that the quality is probably good enough. It is characteristics of the 3rd or higher generation of the applications.

- The higher number of the ACR means that the project is staffed with the inexperienced people (the first time right ACs are not good enough and the TRs are coming on the faulty solutions). It is also very hard to conclude on the overall quality but usually products with the higher ACR are more mature products and with the better quality. It is characteristics of the outfacing applications.

On the other hand if we have higher CTR, and we have to decrease it (removing resources from the project) with the same number for ACR, TRCR probably will increase.

5. CONCLUSION

Quality is something very hard to define, but it is a measure about how confident is user of the services in the operations. However, fault-free product most likely will not be affordable. Without some balance to the interests of the Quality assurance function, it can become too large. These are the influences of the classic market-share dilemma. There is no perfect quality; only good enough.

It is always about balancing and moving the key performance indicators numbers over the major performance elements.

Extensive information collected in the different trouble report fields from all over the world for different applications are necessary to point out the root cause for every single faults. Collecting process must be reassessed timely and constantly improved and as such will help developing organization to collect all relevant data. Those data, key performance indicators, will help organization to make conclusion about software component quality and applications in general on time without unnecessary rework which is time and resource consuming.

Quality indicators presented in this paper will help in early detection of the quality of the products/projects and will propose to do some corrective actions. On such way, organization can take appropriate actions to gain desired performances for the software component and applications in general.

REFERENCES


Figure 8. Key performance indicators in DM and DFU projects

In the figure above, all KPI indicators are put in relation.

For example, if the TRCR is higher than expected, and if we have to decrease it with the same number of the resources in the project, probably the ACR with increased with the time (more TR will be answered in the same period but there will be no time for additional AC quality checks).