UNIT-V

Defects diagnosis and prevention defect study – Identification and analysis of defects – Correcting measure – Factors affecting reliability – MTTF – Calculation of reliability – Building reliability in the product – Evaluation of reliability – Interpretation of test results – Reliability control – Maintainability – Zero defects – Quality circle

Defects diagnosis and prevention defect study:

Defects:

A defect is associated with a quality characteristic that does not meet certain standards. Furthermore, the severity of one of more defects in a product or service may cause it to be unacceptable (or defective). The modern term for defect is nonconformity, and the term for defective is nonconforming item. The American National Standards Institute, the International Organization for Standardization, and the American Society for Quality provide a definition of a defect in ANSI/ISO/ASQ Standard A8402 (ASQ 1994).

Defects Diagnosis:

A casting defect is any characteristic which is not acceptable to the customer. Defect diagnosis is the procedure of identifying the type of defect and understanding its underlying cause. But some defects are very difficult to diagnose.

Defects Prevention:

It can be defined as a measure to ascertain that defects detected so far, must not reappear.

Often there is someone appointed as the lead to carry out the defect prevention activity. The coordinator is responsible for facilitating communication among the team members, planning and devising defect prevention guidelines etc.

Defect prevention can be understood in the light of the following activities:

Software Requirement Analysis: Requirement specifications form the integral part of a software development life cycle. If the requirements aren't well understood then it may cause an issue in proceeding further. If any flaw remains hidden at this phase, then it becomes really difficult to address the issue. Hence proper assessment and analysis of the requirements is very important.

Reviews: Whether it's the testing team that conducts the review process or the end-user, it is an effective way to scan a defect. Often it helps to seek assistance from the end-user because they have a fresh look at the system under review. It is a known fact that sometimes defects remain unnoticed when handled by the same team members, because of the monotony of work. So the review process can be self-review or peer review.

Defect Logging and Documentation: Now when the defects have been detected and analysed, it's time for recording those defects to enable - complete description of the defect to develop a better understanding of it, taking preventive measures so that the defect isn't carried forward to the next phases.

Root Cause Analysis and Determining Preventive measures: Identifying the root cause of the problem involves enhanced quality of the software application being developed, applying expertise of people to know what could be the best possible way to avoid such defects and finally targeting systematic errors.

Embedding Procedures in the root cause analysis: The defects analysed so far and the reasons that lead to such defects, must be tracked and used as a reference for further projects.

Identification and analysis of defects:

Problem verification is the first step of problem investigation. There are 3 main activities:

a) Verify the problem
b) Collect information
c) Describe the problem

To describe the problem specifically, (5W2H) terms (who, what, where, when, why, how, and how many) would help.

Defect analysis:

Defect analysis and prevention is an activity that impacts the entire development life cycle. Occurrence of defects also affects the budget of a project. A considerable amount of time and workforce is required to deal with the defects.

There are various metrics for analysis of defects. Let us understand each method in detail.

Defect Pareto Chart: This chart reflects the frequency of occurrence of various categories of problems. The defects that has a higher frequency of occurrence is observed and priority is assigned.

The above bar chart shows the various problems/defects encountered in the system. Like the 'vital few' shows that downloading problem is high among the rest. File not found and open as read-only file has subsequently higher defect occurrence. Accordingly, priority is assigned to address those defects.

Root Cause Analysis: This is the method of finding the reason that contributes to the defect. It is an essential part in the elimination of causes that leads to the defects. The key points that underlies the root cause analysis of a defect are-

- Reducing defects to improve quality
- Using expertise to catch the defects shall help in an efficient way

- A fish-bone diagram is used as an effective tool to carry out the process of root cause analysis.
- This is termed as the cause and effect diagram. The various arrows are the fish bones that represent the causes that lead to a defect and the resultant effect forms the face of the fish, which is to be dealt with.

Correcting measure:

After defects are logged and documented, the next step is to analyse them. Generally the designated defect prevention coordinator or development project leader facilitates a meeting to explore root causes.

The root cause analysis of a defect is driven by three key principles:

- **Reducing the defects to improve the quality:** The analysis should lead to implementing changes in processes that help prevent defects and ensure their early detection.
- **Applying local expertise:** The people who really understand what went wrong are the people present when the defects were inserted – members of the software engineering team. They can give the best suggestions for how to avoid such defects in the future.
- **Targeting the systematic errors:** There may be many errors or defects to be handled in such an analysis forum; however, some mistakes tend to be repeated. These systematic errors account for a large portion of the defects found in the typical software project. Identifying and preventing systematic errors can have a big impact on quality (in terms of defects) for a relatively small investment.

Factors affecting Reliability:

The reliability of a product (or system) can be defined as the probability that a product will perform a required function under specified conditions for a certain period of time. If we have a large number of items that we can test over time, then the Reliability of the items at time t is given by

$R(t) =$ Number of survivors at time t / Number of items put on test at time $t = 0$

At time $t = 0$, the number of survivors is equal to number of items put on test. Therefore, the reliability at $t = 0$ is

$$R(0) = 1 = 100\%$$

Mean time to Failure (MTTF)

MTTF applies to non-repairable items or devices and is defined as "the average time an item may be expected to function before failure". This can be estimated from a suitable sample of items which have been tested to the point of failure: the MTTF is simply the average of all the times to failure.

For example, if four items have lasted 3,000 hours, 4000, hours, 4000 hours and 5,000 hours, the

MTTF is 16,000/4 or 4,000 hours.

Calculation and evaluation of Reliability:

- Reliability is defined as the consistency of results from a test.
- Theoretically, each test contains some error – the portion of the score on the test that is not relevant to the construct that you hope to measure. – Error could be the result of poor test construction, distractions from when the participant took the measure, or how the results from the assessment were marked.
- Reliability indexes thus try to determine the proportion of the test score that is due to error.

There are four methods of evaluating the reliability of an instrument:

1. **Split-Half Reliability:** Determines how much error in a test score is due to poor test construction.

To calculate: Administer one test once and then calculate the reliability index by coefficient alpha, Kuder-Richardson formula 20 (KR-20) or the Spearman-Brown formula.

2. **Test-Retest Reliability:** Determines how much error in a test score is due to problems with test administration (e.g. too much noise distracted the participant).

To calculate: Administer the same test to the same participants on two different occasions. Correlate the test scores of the two administrations of the same test.

3. **Parallel Forms Reliability:** Determines how comparable two different versions of the same measure are.

To calculate: Administer the two tests to the same participants within a short period of time. Correlate the test scores of the two tests.

4. **Inter-Rater Reliability:** Determines how consistent two separate raters of the instrument are.

To calculate: Give the results from one test administration to two evaluators and correlate the two markings from the different raters.

Reliability Control:

- Reliability is defined as the consistency of results from a test.

Theoretically, each test contains some error

– The portion of the score on the test that is not relevant to the construct that you hope to measure.

– Error could be the result of poor test construction, distractions from when the participant took the measure, or how the results from the assessment were marked.

- Reliability indexes thus try to determine the proportion of the test score that is due to error.

Zero Defects:

"Zero Defects is a management tool aimed at the reduction of defects through prevention. It is directed at motivating people to prevent mistakes by developing a constant, conscious desire to do their job right the first time."

Zero Defects seeks to directly reverse the attitude that the number of mistakes a worker makes doesn't matter since inspectors will catch them before they reach the customer. This stands in contrast to activities that affect the worker directly, such as receiving a pay check in the correct amount. Zero Defects involves reconditioning the worker "to take a personal interest in everything he does, by convincing him that his job is just as important as the task of the doctor or the dentist."

Quality Circle:

Introduction

Dr. K. Ishikawa started Quality Control Circles (known as Quality Circles in India and in many nations) originally for the self and mutual development of the workmen. They are also a very logical outcome of the Japanese drive for training and accomplishment in quality control and quality improvement.

Quality Circle:

A group of employees who perform similar duties and meet at periodic intervals, often with management, to discuss work-related issues and to offer suggestions and ideas for improvements, as in production methods or quality control, called quality circle.

Key Characteristics of quality circle:

A circle, usually consisting of 6-8 members, from the same section.

- Membership of a Quality Circle is voluntary.
- Circle members should meet regularly, ideally once a week, in particular place also in particular time.
- Circle members select a name for their circle in the first meeting and elect a leader to conduct the meetings.
- Members are specially trained in problem solving and analysis techniques in order to play their role effectively.
- Circle works on a systematic basis to identify and solve work – related problems for improving quality and productivity not just discussing them.
- The management must ensure that solutions are implemented quickly once they have been accepted.

Objectives of QC:

- To improve quality and productivity.
- To reduce the cost of products or services by waste reduction, safety, effective utilization of resources, avoiding unnecessary errors and defects.
- To identify and solve work-related problems and interfere with production as a team.
- To tap the creative intelligence of people working in the org. and make full use of human resources.
- To improve communication within the organization.
- To improve employees loyalty and commitment to the organization and its goals. (Promoting Morale of employees)
- To build a happy, bright, meaningful work environment.
- To satisfy the human needs of recognition, achievement and self-development

QC may consist of the following elements:

1. **Steering committee**: Gen. Manager / works manager, rep. from top management, rep. of human resource development and a rep. of employees' union.

2. Coordinator: an administrative officer / personnel officer from middle level management.

3. Facilitator: senior supervisory officer / foreman. A facilitator may manage up to 10 circles. A facilitator is usually from one of the three departments – quality control, production or training.

4. Circle Leader: circle leaders may be from the lowest level of supervisors. A circle leader organises and conducts circle activities.

5. Circle members: line and / or staff workers (circle members should attend all meetings as far as possible, offer suggestions and ideas, participate actively in group processes, and attain training seriously.

How to implement quality circle:

- Firstly, the management is informed about the quality control circle process that is being planned.
- A committee is formed, and key persons such as a coordinator and in-house coach are selected.
- The scope is defined, and areas of application identified.

- First-line supervisors in the identified areas are given QCC presentations. It is important to make these impressive, and valuable tips on the subject are available.
- This is followed up with extensive training for coordinators and middle management on the process and their roles.
- Employees are invited to become members of a circle, and trained suitably once they sign up. Thus, a circle is formed and begins work. These may give rise to other circles.
- Problems are discussed and solved in a systematic manner in the QCCs. It is very important that solutions are implemented as quickly as possible, to maintain the momentum.

Maintainability:

Maintainability deals with duration of maintenance outages or how long it takes to achieve (ease and speed) the maintenance actions compared to a datum. The datum includes maintenance (all actions necessary for retaining an item in, or restoring an item to, a specified, good condition) is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance. Maintainability characteristics are usually determined by equipment design which set maintenance procedures and determine the length of repair times.

The key figure of merit for maintainability is often the mean time to repair (MTTR) and a limit for the maximum repair time. Qualitatively it refers to the ease with which hardware or software is restored to a functioning state. Quantitatively it has probabilities and is measured based on the total down time for maintenance including all time for: diagnosis, trouble shooting, tear-down, removal/replacement, active repair time, verification testing that the repair is adequate, delays for logistic movements, and administrative maintenance delays. It is often expressed as

$$M(t) = 1 - \exp(-t/MTTR) = 1 - \exp(-\mu t)$$

Where μ is constant maintenance rate and MTTR is mean time to repair. MTTR is an arithmetic average of how fast the system is repaired and is easier to visualize than the probability value.

Measuring reliability

Reliability is a measure of the consistency of a metric or a method.

Every metric or method we use, including things like methods for uncovering usability problems in an interface and expert judgment, must be assessed for reliability.

In fact, before you can establish validity, you need to establish reliability.

Here are the four most common ways of measuring reliability for any empirical method or metric:

Inter-rater reliability:

The extent to which raters or observers respond the same way to a given phenomenon is one measure of reliability. Where there’s judgment there’s disagreement. Some examples include:

- ✓ Evaluators identifying interface problems
- ✓ Experts rating the severity of a problem

Test-retest reliability:

Do customers provide the same set of responses when nothing about their experience or their attitudes has changed? You don't want your measurement system to fluctuate when all other things are static.

Parallel forms reliability:

Getting the same or very similar results from slight variations on the question or evaluation method also establishes reliability. One way to achieve this is to have, say, 20 items that measure one construct (satisfaction, loyalty, usability) and to administer 10 of the items to one group and the other 10 to another group, and then correlate the results. You're looking for high correlations and no systematic difference in scores between the groups.

Internal consistency reliability:

This is by far the most commonly used measure of reliability in applied settings. It's popular because it's the easiest to compute using software—it requires only one sample of data to estimate the internal consistency reliability.

Interpretation of Test Results:

Performance tests try to reduce the risks of downtime or outages on a multi-user systems by conducting experiments that use load to reveal limitations and errors in the system. Testing is usually assessing the performance and capacity of systems that were expensive and time-consuming to build.

All of the steps in global performance testing matter to successful projects and making good decisions. These steps include (but aren't limited to):

- Discovery,
- Modelling,
- Developing scripts, and
- Executing tests.

Data needs analysis to become information:

This is the place that my tutorial started. After running a performance test, there will be barrels full of numbers.

So what's next?

The answer is definitely not to generate and send a canned report from your testing tool. Results interpretation and reporting is where a performance tester earns their stripes.

Visualizing data with graphs is the most commonly used method for analyzing load testing results:

Most load testing tools have some graphing capability, but you should not mistake graphs for reports. Graphs are just a tool. The person operating the tool has to interpret the data that graphs help visualize, determine what matters and what doesn't, and present actionable information in a way that stakeholders can consume.

As an aside, here's an example of a graph showing how averages lie. Good visualizations help expose how data can be misleading.

he performance tester should form hypotheses, draw tentative conclusions, determine what information is needed to confirm or disprove them, and prepare key visualizations that both give insight on system performance and bottlenecks and support the narrative of the report.

Some of the skills necessary for doing this are foundational technical skills, understanding things like:

- ✓ Architecture,
- ✓ Hard and soft resources,
- ✓ Garbage collection algorithms,
- ✓ Database performance,
- ✓ Message bus characteristics, and
- ✓ Other components of complex systems.

Understanding that a system slows down at a certain load is of some value. Understanding the reason for the system slowing down: the limiting resource, the scalability characteristics of the system – this information is actionable. This knowledge and experience recognizing patterns can take years to acquire, and that learning is ongoing.

Other skills are socio-political in nature:

We need to know what stakeholders want to hear, because that reveals what information they are looking for:

- Who needs to know these results?
- What do they need to know?
- How do they want to be told?
- How can we form and share the narrative so that everyone on the team can make good decisions that will help us all succeed?

It is our job to be the headlights of a project, revealing information about what reality is. We want to tell the truth, but we can guide our team with actionable feedback to turn findings into a plan, not just a series of complaints.

It might seem daunting to imagine growing all of these skills:

The good news is that you don't have to do this all by yourself. The subject matter experts you are working with – Developers, Operations, DBAs, help desk techs, business stakeholders, and your other teammates — all have information that can help you unlock the full value of a performance test.

This is a complex process, full of tacit knowledge and difficult to teach. In describing how to do this, my former consulting partner and mentor Dan Downing came up with a six-step process called CAVIAR:

1. Collecting
2. Aggregating
3. Visualizing
4. Interpreting
5. Analysing
6. Reporting

1. Collecting is gathering all results from test that can help gain confidence in results validity:

Are there errors? What kind, and when? What are the patterns? Can you get error logs from the application?

One important component of collecting is granularity. Measurements from every few seconds can help you spot trends and transient conditions. One tutorial attendee shared how he asked for access to monitor servers during a test, and was instead sent resource data with five minute granularity.

2. Aggregating is summarizing measurements using various levels of granularity to provide tree and forest views, but using consistent granularities to enable accurate correlation:

Another component is meaningful statistics: scatter, min-max range, variance, percentiles, and other ways of examining the distribution of data. Use multiple metrics to "triangulate"" — that is, confirm (or invalidate) hypotheses

3. Visualizing is about graphing key indicators to help understand what occurred during the test:

Here are some key graphs to start with:

- Errors over load ("results valid?")
- Bandwidth throughput over load ("system bottleneck?")
- Response time over load ("how does system scale?")
  - ✓ Business process end-to-end
  - ✓ Page level (min-avg-max-SD-90th percentile)
- System resources ("how's the infrastructure capacity?")
  - ✓ Server CPU over load
  - ✓ JVM heap memory/GC
  - ✓ DB lock contention, I/O Latency

4. Interpreting is making sense of what you see, or to be scientific, drawing conclusions from observations and hypotheses:

Some of the steps here:

- Make objective, quantitative observations from graphs / data: "I observe that…" no evaluation at this point!
- Correlate / triangulate graphs / data: "Comparing graph A to graph B…" – relate observations to each other
- Develop hypotheses from correlated observations
- Test hypotheses and achieve consensus among tech teams: "It appears as though…" – test these with extended team; corroborate with other information (anecdotal observations, manual tests)
- Turn validated hypotheses into conclusions: "From observations a, b, c, corroborated by d, I conclude that…"

5. Assessing is checking where we met our objectives, and deciding what action we should take as a result:

Determine remediation options at appropriate level – business, middleware, application, infrastructure, and network. Perform agreed-to remediation, and retest.

Generate recommendations at this stage. Recommendations should be specific and actionable at a business or technical level. Discuss findings with technical team members: "What does this look like to you?" Your findings should be reviewed (and if possible, supported) by the teams that need to perform the actions. Nobody likes surprises.

Recommendations should quantify the benefit, if possible the cost, and the risk of not doing it. Remember that a tester illuminates and describes the situation. The final outcome is up to the judgment of your stakeholders, not you. If you provide good information and well-supported recommendations, you've done your job.

6. Reporting is last:

This includes the written report, presentation of results, email summaries, and even oral reports. The narrative, the short 30-second elevator summary, the three paragraph email – these are the report formats that the most people will consume, so it is worth spending time on getting these right, instead of trying to write a great treatise that no one will read. Author the narrative yourself, instead of letting others interpret your work for you.

Good reporting conveys recommendations in stakeholders' terms. You should identify the audience(s) for the report, and write and talk in their language. What are the three things you need to convey? What information is needed to support these three things?

Write a test report:

A written report is still usually the key deliverable, even if most people won't read it (and fewer will read the whole report).

One way to construct the written report might be like this:

Building reliability in the product – Evaluation of reliability:

Reliability:

Definitions:

1. The ability of an apparatus, machine, or system to consistently perform its intended or required function or mission on demand without degradation or failure.

2. Manufacturing: The probability of failure-free performance over an item's useful life, or a specific timeframe, under specified environmental and duty-cycle conditions. Often expressed as mean time between failures (MTBF) or reliability coefficient. Also called quality over time.

3. Consistency and validity of test results determined through statistical methods after repeated trials.

Manufacturers need to understand their current capabilities and continually improve their uptime and process availability. Traditionally, the capabilities of these facilities in their networks vary as a result of product mix, processing, packaging capabilities, shipping issues, access to markets, distribution networks, and the availability of their assets.

Some manufacturers take a strategic approach to improving their manufacturing network (site) availability by instituting and implementing corporate reliability guidance. This isn't the common approach, however. Many manufacturers let their sites develop their own practices; these may not reflect industry best practices and may compete with (and lose to) other corporate initiatives. Don't let this happen at your organization – here's a five-step process to developing a corporate reliability initiative.

5 Steps to build sustainable reliability:

Step 1: Develop the business case

The business case should focus on remediating what has plagued the business in the past and what the future focus should be in response to market conditions. The business case can focus on these issues:

- ✓ Uptime
- ✓ Operating cost
- ✓ Supply chain
- ✓ Demand
- ✓ Deferring capital
- ✓ Safety
- ✓ Product cost
- ✓ Breakdowns

Step 2: Leadership structure

The effectiveness of corporate reliability programs varies depending on:
Leadership – who drives the initiative both from a corporate and business position?

- ✓ Strategy – targeting what the business needs to survive
- ✓ Structure – including the "right" elements

- ✓ Culture – collaborating, focusing on defect elimination, and being proactive
- ✓ Talent – supporting its implementation
- ✓ Level of buy-in from participants – accepting the initiative from the boardroom to the production floor
- ✓ Competency – obtaining knowledge of practices

Step 3: Competency and baselines

Organizations need to have individuals with the right competencies to carry out a corporate reliability program. This starts with hiring people with the right skill and knowledge sets. Job descriptions need to be modified for use in the hiring process. Table 1 shows the distinction between reliability engineers and others involved in asset care.

Step 4: Program structure considerations

Following the "loose and tight" strategy, it is best to specify principles and high-level practice expectations and leave the rest to the sites to decide how to implement these practices.

Principle: Critical-equipment list

Critical equipment is equipment that merits more maintenance time and resources because of its value to the business. Identifying critical equipment within a production facility ensures that equipment receives the right priority level when determining allocation of resources.

Practice

- ✓ All equipment within the production facility (including rotating and fixed equipment, structures, systems and other components (electrical, mechanical, instrumentation) has been evaluated and processed through the critical equipment evaluation
- ✓ The critical-equipment list is available and is used in establishing priorities for preventive maintenance activities, condition monitoring activities, and spare-parts inventory decisions
- ✓ The critical-equipment list is used to prioritize maintenance tasks
- ✓ Failure of a piece of critical equipment triggers a detailed assessment of the failure, such as an RCA or RCM

Step 5: Business and site implementation

Metrics are vital to determining progress. As mentioned earlier, only a few metrics are needed from a corporate perspective. Each site should use about 5-10 metrics that tie directly to the corporate metrics. Each network should be responsible for metrics that represent their activity; these can then be leveraged across all sites.