Defect Removal Effectiveness

Defect Removal Effectiveness

- Effective defect removal can lead to reductions in the development cycle time and good product quality.
- Defect removal is one of the top expenses in any software project and it greatly affects schedules.
- For improvements in quality, productivity, and cost, as well as schedule, it is important to use better defect prevention and removal technologies to maximize the effectiveness of the project.

Defect Removal Effectiveness

- **Defect Removal Efficiency or Effectiveness?**
  - **Efficiency** is about applying a method within a a time period and see whether the results have been achieved.,
  - **Effectiveness** is about measuring whether the planned/expected results have been achieved by the method used

Different views of DRE definitions

- **In the 1960s, software projects were characterized by cost overruns and schedule delays, the only defect removal step was testing.**
- **In the 1970s, formal reviews and inspections were recognized as important to productivity and product quality. As a result, the value of defect removal as an element of the development process strengthened.**
- **Review vs. Inspection?**
- **How about walkthroughs?**

Different views of DRE definitions

- Fagan (1976) defined error detection efficiency as:

$$\frac{\text{Errors found by an inspection}}{\text{Total errors in the product before inspection}} \times 100\%$$

- Jones's definition (1986) is very similar to Fagan's

$$\text{Removal efficiency} = \frac{\text{Defects found by removal operation}}{\text{Defects present at removal operation}} \times 100\%$$

$$= \frac{\text{Defect found}}{\text{Defects found } + \underset{\text{(found later)}}{\text{Defects not found}}} \times 100\%$$

- In Jones's definition, defects found in the field are included in the denominator of the formula

Different views of DRE definitions

- **IBM Houston won many quality awards from NASA and for its outstanding quality in the space shuttle flight systems**

- **One of the metrics IBM used to manage quality is the early detection percentage, which is actually inspection defect removal effectiveness.:**

$$\text{Early detection percentage} = \frac{\text{Number of major inspection errors}}{\text{Total number of errors}} \times 100\%$$

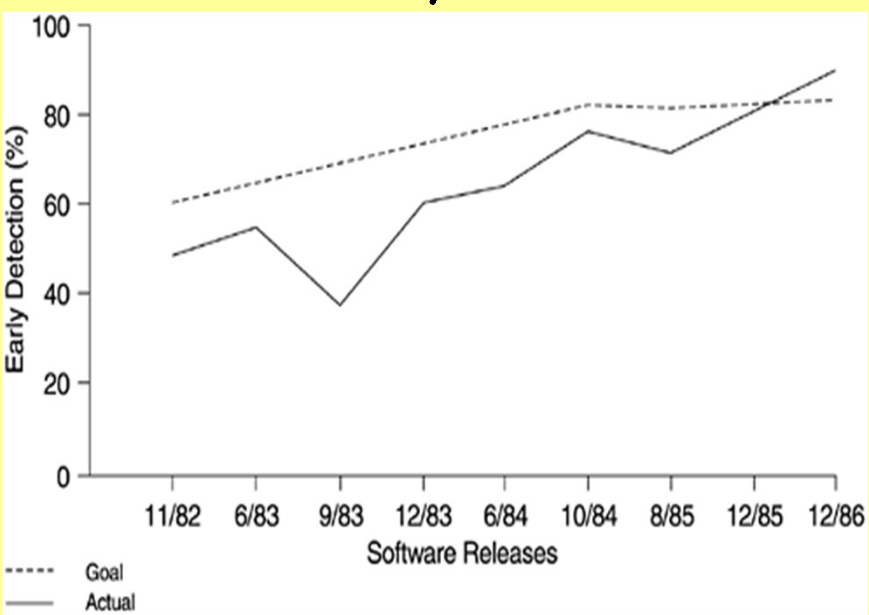
- where total number of errors is the sum of major inspection errors and valid discrepancy reports (discrepancy report is the mechanism for tracking test defects).

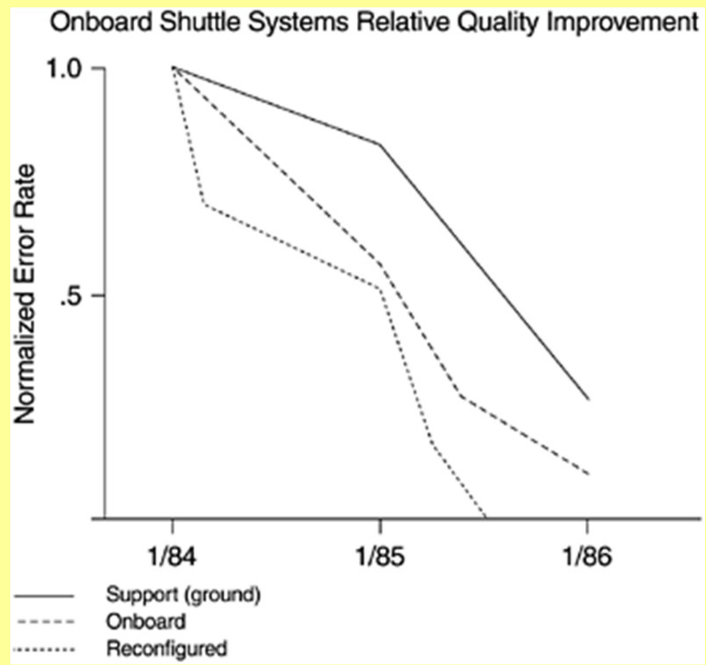
- According to IBM Houston's definitions, a major inspection error is any error found in a design or code inspection that would have resulted in a valid discrepancy report (DR) if the error had been incorporated into the **software**.

Different views of DRE definitions

- IBM Houston's data substantiated a strong correlation between inspection defect removal effectiveness and product quality

- For software releases from November 1982 to December 1986, the early detection percentages increased from about 50% to more than 85%. Correspondingly, the product defect rates decreased from 1984 to 1986 by about 70%.





Different views of DRE definitions

- **The effectiveness measure by Dunn (1987) differs little from Fagan's and from Jones's second definition. Dunn's definition is:**

$$E = \frac{N}{N+S} \times 100\%$$

  - where
    - E = Effectiveness of activity (development phase)
    - N = Number of faults (defects) found by activity (phase)
    - S = Number of faults (defects) found by subsequent activities (phases)

Defect Removal Effectiveness

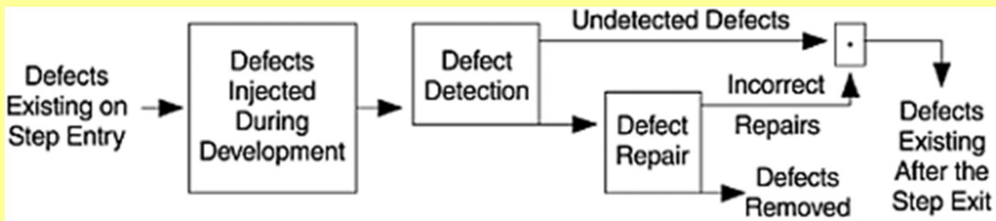
- To define defect removal effectiveness clearly, we must first understand the activities in the development process that are related to defect injections and to removals.
- Defects are injected into the product or intermediate deliverables of the product (e.g., design document) at various phases.
- It is wrong to assume that all defects of software are injected at the beginning of development.

Defect Removal Effectiveness

- For the development phases before testing, the development activities themselves are subject to defect injection, and the reviews or inspections at end-of-phase activities are the key vehicles for defect removal.
- For the testing phases, the testing itself is for defect removal.
- If the problems found by testing are fixed incorrectly, there is another chance to inject defects.

Defect Removal Effectiveness

- The **Following figure describes defect injection and removal at each step of the development process:**



- **The defect removal effectiveness for each development step, can be defined as :**

$$\frac{\text{Defects removed (at the step)}}{\text{Defects existing on step entry} + \text{Defects injected during development (of the step)}} \times 100\%$$

Defect Removal Effectiveness

- To derive an operational definition, we will use a matrix approach by cross-classifying defect data in terms of the development phase in which the defects are found (and removed) and the phases in which the defects are injected.

- Consider the following example:

Defect Removal Effectiveness

Where Found	Defect Origin: Requirements	High-Level Design	Low-Level Design	Code	Unit Test	Component Test	System Test	Field	Total
RQ	—			—					
I0	49	681							730
I1	6	42	681						729
I2	12	28	114	941					1095
UT	21	43	43	223	2				332
CT	20	41	61	261	—	4			387
ST	6	8	24	72	—	—	1		111
Field	8	16	16	40	—	—	—	1	81
Total	122	859	939	1537	2	4	1	1	3465

Defect Removal Effectiveness

- **Few remarks on the matrix data:**
  - The matrix is triangular because the origin of a defect is always at or prior to the current phase.
  - In this example, there were no formal requirements inspections so we are not able to assess the effectiveness of the requirements phase.
  - But in the requirements phase, defects can be injected that can be found later in the development cycle. Therefore, the requirements phase also appears in the matrix as one of the defect origins.
  - The diagonal values for the testing phases represent the number of bad fixes; bad fixes may go undetected until subsequent phases
- **Based on the conceptual definition given earlier, we calculate the various effectiveness metrics as follows**

Defect Removal Effectiveness

- **High-Level Design Inspection Effectiveness; IE (I0)**
  - Defects removed at I0: 730
  - Defects existing on step entry (escapes from requirements phase): 122
  - Defects injected in current phase: 859

$$IE\ (I0) = \frac{730}{122 + 859} \times 100\% = 74\%$$

Defect Removal Effectiveness

- **Low-Level Design Inspection Effectiveness; IE (I1)**
  - Defects removed at I1: 729
  - Defects existing on step entry (escapes from requirements phase and I0): $122 + 859 - 730 = 251$

- **Defects injected in current phase: 939**

$$\text{IE (I1)} = \frac{729}{251 + 939} \times 100\% = 61\%$$

Defect Removal Effectiveness

- **Code Inspection Effectiveness; IE (I2)**
  - Defects removed at I2: 1095
  - Defects existing on step entry (escapes from requirements phase, I0 and I1):

$$122 + 859 + 939 - 730 - 729 = 461$$

- **Defects injected in current phase: 1537**

$$\text{IE (I2)} = \frac{1095}{461 + 1537} \times 100\% = 55\%$$

Defect Removal Effectiveness

- **Unit Test Effectiveness; TE (UT)**
  - Defects removed at UT: 332
  - Defects existing on step entry (escapes from all previous phases):

$$122 + 859 + 939 + 1537 - 730 - 729 - 1095 = 903$$

- **Defects injected in current phase (bad fixes): 2**

$$\text{TE (UT)} = \frac{332}{903 + 2} \times 100\% = \frac{332}{905} \times 100\% = 37\%$$

Defect Removal Effectiveness

- For the testing phases, the defect injection (bad fixes) is usually a small number. In such cases, effectiveness can be calculated by an alternative method (Dunn's formula or Jones's second formula as discussed earlier).

- In cases with a high bad-fixes rate, the original method should be used

$$\text{Effectiveness} = \frac{\text{Defects removed at current phase}}{\text{Defects removed at current phase} + \text{Defects removed at subsequent phases}} \times 100\%$$

$$\text{TE (UT)} = \frac{332}{332 + 387 + 111 + 81} \times 100\% = \frac{332}{911} \times 100\% = 36\%$$

Defect Removal Effectiveness

- **Component Test Effectiveness; TE (CT)**

$$\text{TE (CT)} = \frac{387}{387 + 111 + 81} \times 100\% = 67\%$$

- **System Test Effectiveness; TE (ST)**

$$\text{TE (ST)} = \frac{111}{111 + 81} \times 100\% = 58\%$$

Defect Removal Effectiveness

- **Overall Inspection Effectiveness; IE**

$$IE = \frac{\text{Defects removed by inspections}}{\text{All defects}} \times 100\% = \frac{730 + 729 + 1095}{3465} \times 100\% = 74\%$$

- **Overall Test Effectiveness; TE**

$$TE = \frac{332 + 387 + 111}{332 + 387 + 111 + 81} \times 100\% = 91\%$$

Defect Removal Effectiveness

- **Overall Defect Removal Effectiveness of the Process; DRE**

$$\mathrm{DRE} = \left(1 - \frac{81}{3465}\right) \times 100\% = 97.7\%$$

- **To summarize, the values of defect removal effectiveness from this example are as follows:**
  - I0: 74%
  - I1: 61%
  - I2: 55%
  - Overall Inspection Defect Removal Effectiveness: 74%
    - UT: 36%
    - CT: 67%
    - ST: 58%
  - Overall Test Defect Removal Effectiveness: 91%
  - Overall Defect Removal Effectiveness of the Process: 97.7%

Defect Removal Effectiveness

- **Assume the data presented in previous figure are the defect data for a new project with 100,000 lines of source code (100 KLOC).**

- **There are few more interesting metrics such as**
    - product defect rate,
    - phase defect removal rates,
    - phase defect injection rates,
    - percent distribution of defect injection by phase,
    - phase-to-phase defect escapes.

- **For instance, the product defect rate is 81/100 KLOC = 0.81 defects per KLOC in the field (for four years of customer usage).**

Defect Removal Effectiveness

- **The phase defect removal and injection rates are shown in** below:

Phase Defect Removal and Injection Rates			
Phase	**Defects/KLOC (removal)**	**Defect Injection per KLOC**	***Total Defect Injection (%)***
Requirements	—	1.2	*3.5*
High-level design	7.3	8.6	*24.9*
Low-level design	7.3	9.4	*27.2*
Code	11.0	15.4	*44.5*
Unit test	3.3	—	
Component test	3.9	—	
System test	1.1	—	
Total	*33.9*	*34.6*	*100.1*

Defect Removal Effectiveness

- **Having gone through the numerical example, we can now formally state the operational definition of defect removal effectiveness. The definition requires information of all defect data (including field defects) in terms both of defect origin and at which stage the defect is found and removed.**
- **The definition is based on the defect origin/where found matrix.**
  - Let j = 1, 2, . . . , k denote the phases of software life cycle.
  - Let i = 1, 2, . . . , k denote the inspection or testing types associated with the life-cycle phases including the maintenance phase (phase k).

Defect Removal Effectiveness

- **Then matrix M defined in the following figure is the defect origin/where found matrix.**
  - In the matrix, only cells Nij,where i >= j (cells at the lower left triangle), contain data.
  - Cells on the diagonal (Nij where i = j) contain the numbers of defects that were injected and detected at the same phase
  - cells below the diagonal (Nij where i > j) contain the numbers of defects that originated in earlier development phases and were detected later.
  - Cells above the diagonal are empty because it is not possible for an earlier development phase to detect defects that are originated in a later phase.
  - The row marginals (Ni.) of the matrix are defects by removal activity, and the column marginals (N.j) are defects by origin.

Defect Removal Effectiveness

Defect Origin

Where Found	$j = 1$	$j = 2$	$j = 3$	·	·	·	·	$j = k$	
$i = 1$	N_{11}								$N_1.$
$i = 2$	·	N_{22}							$N_2.$
$i = 3$	·	·	N_{33}						$N_3.$
·	·	·	·	·					.
·	N_{ij} $(i >= j)$	·	·	·	·				$Ni.$
·	·	·	·	·	·	N_{ij} $(i = j)$			.
·	·	·	·	·	·	·	·		.
$i = k$	·	·	·	·	·	·	·	N_{kk}	$N_K.$
	$N \cdot 1$	$N \cdot 2$	$N \cdot 3$	·	·	$N \cdot j$	·	$N \cdot k$	N-Grand Total

Defect Removal Effectiveness

- **Phase defect removal effectiveness (PDREi) can be phase inspection effectiveness [IE(i)] or phase test effectiveness [TE(i)]**

$$PDE_i = \frac{N_{i.}}{\sum_{m=1}^{i} N_{.m} - \sum_{m=1}^{i-1} N_{m.}}$$

Defect Removal Effectiveness

- **Phase defect containment effectiveness (PDCEi)**

$$PDCE_i = \frac{N_{ii}}{N_{.i}}$$

Defect Removal Effectiveness

- **Overall inspection effectiveness (IE)**

$$PDEi = \frac{\sum_{i=1}^{I} Ni.}{\sum_{i=1}^{k} Ni. = N}$$

  - where I is the number of inspection phases.

Defect Removal Effectiveness

- **Overall test effectiveness (TE)**

$$PDEi = \frac{\sum_{i=I+1}^{k-1} Ni.}{\sum_{i=I+1}^{k} Ni.}$$

  - where **I** + 1, **I** + 2, . . . , **k** - 1 are the testing phases.

Defect Removal Effectiveness

- **Overall defect removal effectiveness (DRE) of the development process:**

$$DRE = \frac{\sum_{i=1}^{k-1} Ni.}{N}$$

Defect Removal Effectiveness and Quality Planning

- Phase defect removal effectiveness and related metrics associated with effectiveness analyses (such as defect removal and defect injection rates) are useful for quality planning and quality management.

- Effectiveness analyses can be done for the entire project as well as for local areas, such as at the component level and specific departments in an organization

- Release-to-release monitoring of these metrics can give a good feeling for the process capability of the development organization.

- Experiences from previous releases provide the basis for phase-specific target setting and for quality planning.

Phase-Based Defect Removal Model

- **The phase-based defect removal model (DRM) summarizes the relationships among three metrics**
  - defect injection,
  - defect removal, and
  - effectiveness.

- **The DRM takes a set of error-injection rates and a set of phase-effectiveness rates as input:**

$$\begin{aligned} \text{Defects at exit of a development step} = {} & \text{Defects escaped from previous step} \\ & + \text{Defects injected in current step} \\ & - \text{Defects removed in current step} \end{aligned}$$

Phase-Based Defect Removal Model

- The metrics derived from data from previous figure can be modeled step by step as shown below

Example of Phase-Based Def ect Removal Model						
Phase	**(A) Defect Escaped from Previous Phase (per KLOC)**	**(B) Defect Injection (per KLOC)**	**Subtotal (A+B)**	**Removal Effectiveness**	**Defect Removal (per KLOC)**	**Defects at Exit of Phase (per KLOC)**
Requirements	—	1.2	1.2	—	—	1.2
High-level design	1.2	8.6	9.82	x 74%	= 7.3	2.5
Low-level design	2.5	9.4	11.9	x 61%	= 7.3	4.6
Code	4.6	15.4	20.0	x 55%	= 11.0	9.0
Unit test	9.0	—	9.0	x 36%	= 3.2	5.8
Component test	5.8	—	5.8	x 67%	= 3.9	1.9
System test	1.9	—	1.9	x 58%	= 1.1	0.8
Field	0.8					

Phase-Based Defect Removal Model

- **Now if we are planning for the quality of a new release, we can modify the values of the parameters based on the set of improvement actions that we are going to take.**
  - If we plan to improve the effectiveness of I2 and unit tests by 5%, how much can we expect to gain in the final product quality?
  - What are the new targets for defect rates for each phase (before the development team exits the phase)?
  - If we invest in a defect prevention process and in an intensive program of technical education and plan to reduce the error injection rate by 10%, how much could we gain?

- **Approximate answers to questions like these could be obtained through the DRM.**

Cost Effectiveness of Phase Defect Removal

- **The cost of defect removal must be considered for efficient quality planning.**
- **Defect removal at earlier development phases is generally less expensive.**
- **The closer the defects are found relative to where and when they are injected, the less the removal and rework effort.**

Cost Effectiveness of Phase Defect Removal

- Fagan (1976) contends that rework done at the I0, I1, and I2 inspection levels can be 10 to 100 times less expensive than if work done in the last half of the process (formal testing phases after code integration).

- According to Freedman and Weinberg (1982, 1984), in large systems, reviews can reduce the number of errors that reach the testing phases by a factor of 10, and such reductions cut testing costs, including review costs, by 50% to 80%.

- Remus (1983) studied the cost of defect removal during the three phases of design and code inspection, testing, and customer use (maintenance phase) based on data from IBM's Santa Teresa (California) Laboratory. He found the cost ratio for the three phases to be 1 to 20 to 82.

Cost Effectiveness of Phase Defect Removal

- **Front-end defect removal activities in the form of reviews, walk-throughs, and inspections are less expensive than testing, in general practice, these methods are not rigorous enough.**

- **Fagan's inspection method is a combination of a formal review, an inspection, and a walkthrough. It consists of five steps:**
  - 1. **overview (for communications and education)**

    2. **preparation (for education)**

    3. **inspection (to find errors and to walk through every line of code)**

    4. **rework (to fix errors), and**

    5. **follow-up (to ensure all fixes are applied correctly)**

Defect Removal Effectiveness and Process Maturity Level

- **Based on a special study commissioned by the Department of Defense, Jones (Software Productivity Research, 1994; Jones, 2000) estimates the defect removal effectiveness for organizations at different levels of the development process capability maturity model (CMM):**
    - Level 1: 85%
    - Level 2: 89%
    - Level 3: 91%
    - Level 4: 93%
    - Level 5: 95%