What are reviews?   
-Feedback  
-The point of feedback is what is good already, and what still needs to be done or polished.  
  
Technical reviews  
Advantage: Early discovery of errors

Why do we want to remove errors?  
-**Reduce cost** as errors are expensive.

Reviews are basically each deliverable presentation.

There are many types of reviews:

Informal: eg Talking about project 5min before class.  
Formal: Presentations with customers demonstrating architecture, deliverable presentations.

Defects and faults mean the same thing: They both imply a quality problem discovered during review, and need to be fixed.

Before implementation of a system, around 75% of errors are caught. (Pure design, no code)

**Defect amplification removal process.** **NBNB**

**This** is a justification of why to spend time fixing errors and spending time on quality.  
**This** is justification to do/not to do the technical review… using the values in it.  
  
-Generation and detection of errors in a system, and shows the propagation of errors in the system.  
  
Amplification\*#Prev Errors + # New Errors – caught errors = # New Stage Errors

**Errors from previous step ->**

|  |  |
| --- | --- |
| Defects | Detection |
| Errors passed through | Percent efficiency for error detection |
| Amplified errors 1 : x |
| Newly generated errors |

* **Errors passed to the next step**

The result of this will give justification of what you must do. With cost analysis applied to this, what type of review must be done. Or if its cheaper to let errors go through?  
  
In ST1, need to o a cost/time evaluation.

**Review metrics (NBNB in st1)**

**preparation effort (Ep)**: hours/person. Effort (in person hours) required to review a work product **prior** to the actual review meeting

**Assessment effort (Ea) –** the effort (in person hours) that is expended during the actual review.

**Rework effort (Er) –** the effort (in person hours) that is dedicated to the correction of these errors uncovered during the review

**Work product size WPS –** A measure of the size of the work product that has been reviewed eg Lines of code, number of doc pages, number of UML models. This is used with table, to give a good idea if its worth your time and money to do a full/mini technical review, or risk the loss.

**Minor errors found, Err minor –** number of errs found categorised as minor (requiring less than some specified effort to correct)

**Major errors found, Err major -**  number of errs found categorised as major (requiring above than some specified effort to correct)

These formulas below given in test on sheet. Just need to know how to apply the 3 formulas (Analysing metrics)

Ereview = Ep + Ea + Er

Total newly generated errors = minor + major

Error density: Error total / WPS = how many errors in a work area

**NBNB Model in ST1**

Scale of formality: A combination of characteristics that determine level of formality: eg Life critical = full formal  
-Individual Roles:   
-Planning and preparation:  
-Distinct structure of review:   
-Follow up of reviewer:

Informal: Anything you can discuss, anywhere, at any time about project  
Formal: There is a meeting planned for, and there is a structure for that meeting.

Creation and verification: …

Formal Technical review:  
Guidelines must be established in advanced. Disclosed to everyone and agreed upon by all.

Work allocated to someone with no verification, formal documentation ect can cause more harm than doing nothing.

Some guidelines: in textbook  
-Review the product, not the producer  
-Limit debate and rebuttal  
-Set a meeting plan, follow it  
-Annunciate errors (highlight things that are important)  
-Prioritise worst errors. Don’t try fix everything.  
-Take written notes.  
-Limit feedback to person  
-Develop a checklist of things to be reviewed.  
-Allocate resources and schedule time for formal technical reviews.

-Identifying error, allocate error, verifying that’s its been corrected.  
-Don’t Lie