Software Engineering Institute | Carnegie Mellon

PSP Advanced

Planning and Tracking Quality

Sponsored by the U.S. Department of Defense

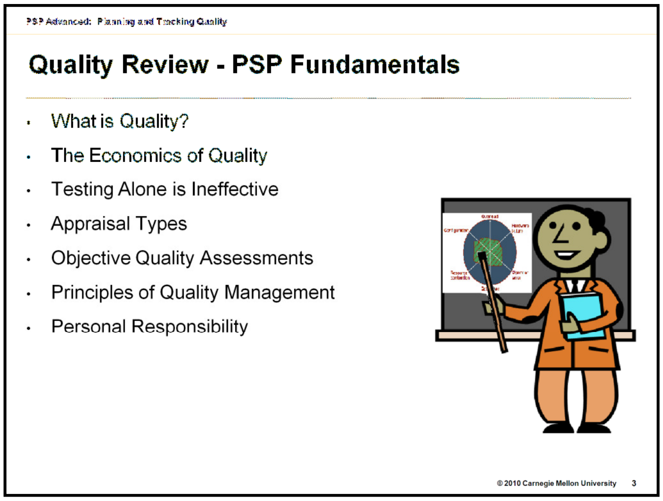
Lecture Topics

Quality Review

Leading vs. Lagging Indicators

Making a Quality Plan

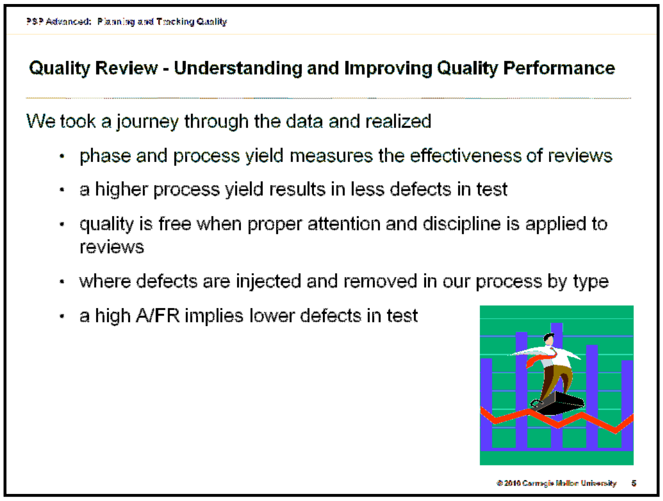
Applying Quality Principles Through the Quality Manager Role



Briefly refresh the students' memory about these topics

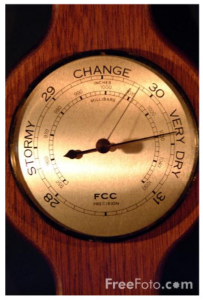
Quality Measures and Metrics discussed thus far

- to-date injected and removed
- to-date percent defects injected and removed
- phase defect removal rate
- process yield
- phase yield
- review yield
- percent appraisal COQ
- percent failure COQ
- COQ
- COQ A/FR
- defect density



Summarize the points they need to build a quality plan and track to it.

Leading vs. Lagging Indicators



Lagging Indicators

A Lagging Indicator is a measurable factor that changes value after the underlying conditions it measures have begun to exhibit a trend. Lagging indicators may confirm the existence of a condition or trend, but are not used for prediction.

- *Example: The primary lagging quality indicator is the actual number of injected and removed defects, because you don't know the total number of defects in the software until the development and testing is complete. It is too late to change the quality pattern or trend of the program then.*

The quality measures discussed to date have been Lagging Indicators, as they rely on one of the following measures:

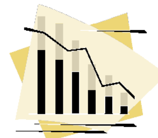
- total defects, and defects by phase, injected and removed
- actual size
- total actual time and actual time by phase

Leading Indicators

A leading indicator is a measurable factor that changes in advance of a new trend or condition and can be used in prediction.

- *Example: Leading indicators are statistics which are used to forecast how the economy will be performing in the future. Examples are unemployment rates, commodity prices, housing starts, inflation, bankruptcies, etc.*

A leading quality indicator is a measurable factor that is shown to relate to a lagging indicator which allows for corrective actions to be taken to achieve a desired level of quality.



Potential Control Parameters

To be useful, process control measures must be available during the process. Examples are

- size units reviewed per hour
- defects found per hour
- defects found per size unit

While no control parameter directly correlates with phase yield, review rate is the most useful control parameter.

Review rate is the parameter used in the PSP.

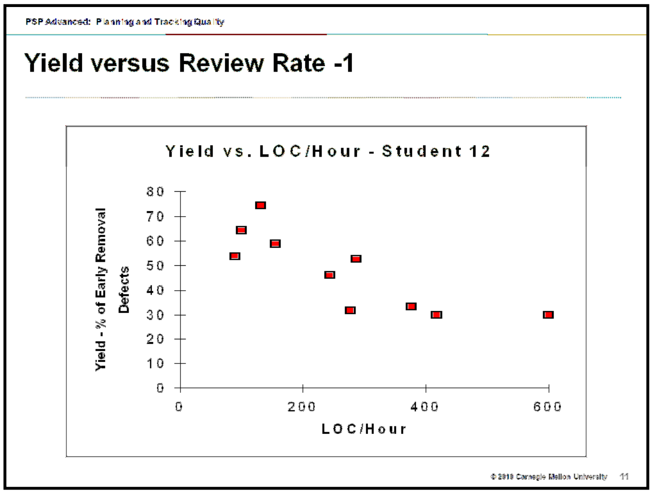
Yield Estimates

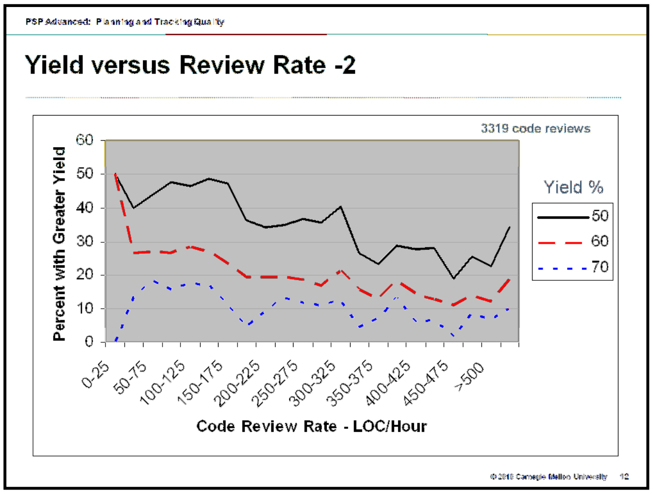
Yield can be estimated but not precisely calculated until all defects have been found through test and product use.

Yield measures are most useful when the developers and testers record all of the defects.

- design and code review defects
- compile defects
- test defects

By using process-control measures, you are more likely to do high-yield reviews.


PSP Advanced: Planning and Tracking Quality
Yield versus Review Rate -1
Yield vs. LOC/Hour - Student 12
Yield - % of Early Removal Defects
80
70
60
50
40
30
20
10
0
0
200
400
600
LOC/Hour
© 2010 Carnegie Mellon University 11


PSP Advanced: Planning and Tracking Quality
Yield versus Review Rate -2
3319 code reviews
Percent with Greater Yield
60
50
40
30
20
10
0
0-25
50-75
100-125
150-175
200-225
250-275
300-325
350-375
400-425
450-475
>500
Code Review Rate - LOC/Hour
Yield %
50
60
70
© 2010 Carnegie Mellon University 12

Yield versus Review Rate -3

While there is considerable variation, higher rates generally give lower-yield reviews.

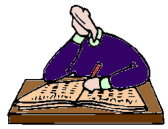
The PSP suggests the following upper limits for review rates:

- code (using the LOC measure): 200 LOC/hour
- documents: 4 pages/hour
- other measures: develop from your personal review data

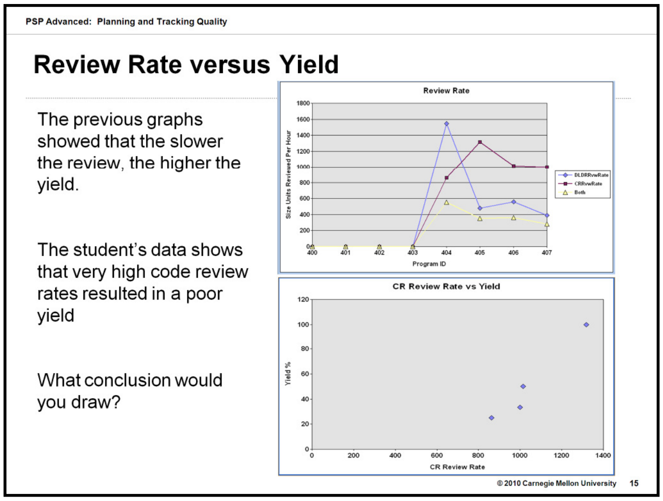
So lets revisit our student's data

Recall that the student's data indicated a low process yield of 50% and a goals was set to increase the process yield to 60%.

Also, recall the code review yield was only 37.5%.

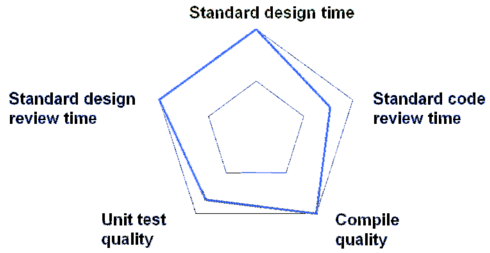


Phase	Current Phase Yield	Desired Phase Yield
Planning	0.00%	0.00%
Detailed Design	0.00%	0.00%
DLD Review	60.00%	60.00%
Code	5.88%	5.88%
Code Review	37.50%	50.00%
Compile	50.00%	50.00%
Unit Test	100.00%	100.00%
PM	0.00%	0.00%



A possible improvement for the student would be to plan for a slower review rate and compare the actual review rate to the plan review rate as a leading indicator to the potential quality of the code under review.

Component Quality Profile



Quality Profiles

The recommended quality profile values are determined as follows.

- design time > 100% of coding time
- design review time > 50 % of design time
- code review time > 50% of coding time
- compile defects < 10 defects/KLOC
- unit test defects < 5 defects/KLOC

The quality profile assigns a value for each of these measures. Values range from 0 to 1.

- If the measure obtained meets the criteria, the value for that item is 1.
- The value is assigned proportionately, based on the measure obtained.

Quality Profiles as leading indicators

- The quality profile values are good planning parameters
  - design time > 100% of coding time
  - design review time > 50 % of design time
  - code review time > 50% of coding time
  - compile defects < 10 defects/KLOC
  - unit test defects < 5 defects/KLOC

- For the first 3 parameters, as a phase is completed a comparison of the actual data can be performed between the completed phase and a previous phase as a leading indicator of quality.
- For the remaining parameters, the number of defects found and the current size of the product are know at phase completion. These can also be used as a leading indicator of quality.

The Process Quality Index

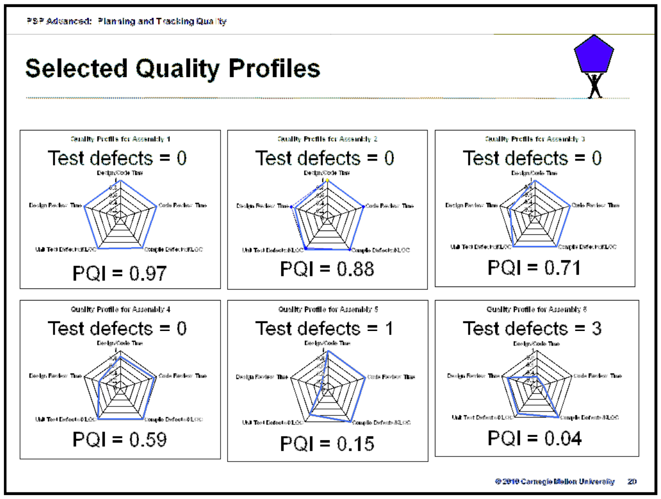
The process quality index (PQI) provides a quality figure or merit for every system element.

To calculate PQI, multiply the profile dimensions to produce a composite value that considers

- compile and unit test defect levels
- design and code review times
- time spent in design

Before test entry, PQI indicates the likelihood that a system element will have subsequent defects.

Values above 0.4 are considered to be good.


PSP Advanced: Planning and Tracking Quality
Selected Quality Profiles
Quality Profile for Assembly 1
Test defects = 0
Design/Code Time
Code Review Time
Compile Defects/KLOC
Unit Test Defects/KLOC
Design Review Time
PQI = 0.97
Quality Profile for Assembly 2
Test defects = 0
PQI = 0.88
Quality Profile for Assembly 3
Test defects = 0
PQI = 0.71
Quality Profile for Assembly 4
Test defects = 0
PQI = 0.59
Quality Profile for Assembly 5
Test defects = 1
PQI = 0.15
Quality Profile for Assembly 6
Test defects = 3
PQI = 0.04
© 2010 Carnegie Mellon University 20

Using the PQI

- If one assumes a desired defect density of ***x*** in program use and that each defect removal phase has a yield of 50%, then
  - *x* defects/size measure should be removed by system test
  - ***2x*** defects/size measure should be removed by integration test
  - ***4x*** defects/size measure should be removed by unit test
  - ***8x*** defects/size measure should be removed by compile
- For example, if the goal is less than 0.5 defects/size measure after system test, then the defect removal rate for the unit test phase should be 2 defects/size measure.
- Note: The defect density measures may be modified if desired. In particular, if there is no compile phase, then defect density for another defect removal phase such as personal reviews can be used instead of those for the compile phase.

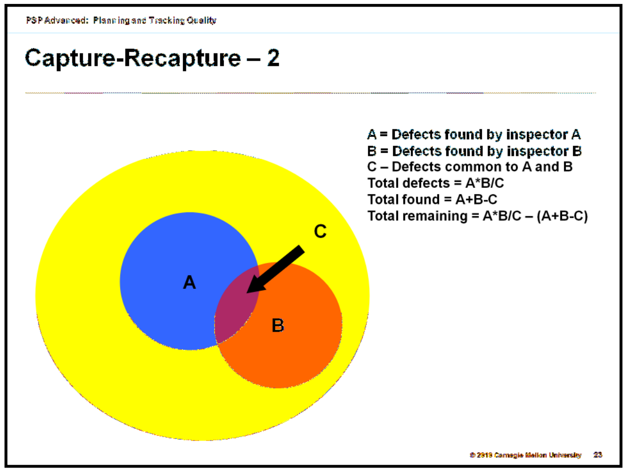
Capture-Recapture – 1

The capture-recapture method uses sampled data to estimate populations.

It can be used to estimate the defects in a product.

Data are required from two or more independent inspections, reviews, or tests of the defective product.

These data can then be used to estimate the total defects and the number of defects remaining.


PSP Advanced: Planning and Tracking Quality
Capture-Recapture – 2
A = Defects found by inspector A
B = Defects found by inspector B
C – Defects common to A and B
Total defects = A*B/C
Total found = A+B-C
Total remaining = A*B/C – (A+B-C)
C
A
B
© 2010 Carnegie Mellon University 23

Capture-Recapture – 3

The validity of capture-recapture estimates, like any other statistical procedures, critically depends upon the model assumptions. If the assumptions were not followed, the estimates are questionable.

- This defect estimate is only reliable when all the numbers A and B are greater than 4 and A-C, and B-C are both greater than 1.
- Even with these criteria, estimate error is likely 10% or more.
- Generally, larger numbers, give more reliable estimates.
- If A=B=C, you are likely to have found all the defects.

Lecture Topics

- Quality Review
- Leading vs. Lagging Indicators
- ➡ Making a quality plan
- Applying Quality Principles Through the Quality Manager Role

Our student is now ready to make a Quality Plan - 1

Recall that the student's quality goal is to increase the process yield to 60%. To accomplish this, the student plans to implement the following PIPs:

1. Process yield has been declining over the last 2 programs, which will lead to more defects being found in Test. To improve process yield, focus on improving the code review phase yield as it is only 37.5%. Try reviewing at a slower pace and take a break between the code and code review phase.
2. Review type 70 & 80 defects in the defect log and update review checklists to find and fix defects prior to test. They are the most costly and they are currently making it through to the failure phases.

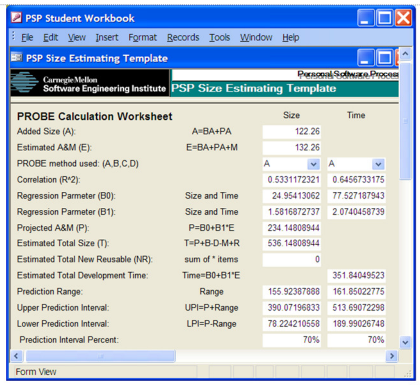
Our student is now ready to make a Quality Plan - 2

The student has:

- added a step in the Code Review process to take at least a 15 minute break between the Code and Code Review phases
- evaluated the defect logs to determine what needs to be added to the Design and Code Review checklists in order to remove type 70 and 80 defects earlier in the development process.

The student is now ready to estimate a new assignment using the updated processes and applying the new quality principles they have recently learned through their evaluation of their quality data.

Program 9 Size and Time Estimate



PROBE Calculation Worksheet		Size	Time
Added Size (A):	A=BA+PA	122.26	
Estimated A&M (E):	E=BA+PA+M	132.26	
PROBE method used: (A,B,C,D)		A	A
Correlation (R^2):		0.5331172321	0.6456733175
Regression Parmeter (B0):	Size and Time	24.95413062	77.527187943
Regression Parmeter (B1):	Size and Time	1.5816872737	2.0740458739
Projected A&M (P):	P=B0+B1*E	234.14808944	
Estimated Total Size (T):	T=P+B-D-M+R	536.14808944	
Estimated Total New Reusable (NR):	sum of * items	0	
Estimated Total Development Time:	Time=B0+B1*E		351.84049523
Prediction Range:	Range	155.92387888	161.85022775
Upper Prediction Interval:	UPI=P+Range	390.07196833	513.69072298
Lower Prediction Interval:	LPI=P-Range	78.224210558	189.99026748
Prediction Interval Percent:		70%	70%

Program 9 Plan Summary – Base on Historical Performance

Code Review Rate = 1050 LOC/Hr

Code Review Removal rate = 5.3 Defects/Hr

Process Yield = 50%

Phase	Plan Minutes	Defects Injected	Defects Removed	Phase Yield
Planning	46	0	0	0.00%
Detailed Design	121	1	0	0.00%
DLD Review	25	0	0.6	60.00%
Code	77	3	0.2	5.88%
Code Review	13	0	1.2	37.50%
Compile	2	0	1	50.00%
Unit Test	45	0	1	100.00%
Post Mortem	22	0	0	0.00%

Program 9 Plan Summary – Modified to meet Quality Goal

Using guideline of code review time > 50% of coding time

Code Review Rate = 363 LOC/Hr

Code Review Removal rate = 2.5 Defects/Hr

Process Yield = 60%

Phase	Plan Minutes	Defects Injected	Defects Removed	Phase Yield
Planning	46	0	0	0.00%
Detailed Design	121	1	0	0.00%
DLD Review	25	0	0.6	60.00%
Code	77	3	0.2	5.88%
Code Review	39	0	1.6	50.00%
Compile	2	0	1	62.50%
Unit Test	19	0	0.6	100.00%
Post Mortem	22	0	0	0.00%

Lecture Topics

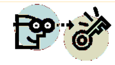
Quality Review

Leading vs. Lagging Indicators

Making a quality plan

→ Applying Quality Principles Through the Quality Manager Role

Quality Manager Team Role



Maintains a focus on product and process quality throughout the project

Leads the team in developing and following the quality plan

Ensures that a qualified moderator is available to lead team inspections or acts as inspection moderator

Tracks product and process quality measures

Ensures that team members provide their quality data, promptly.

Quality Manger Analysis and Reporting

For quality analysis the quality manager regularly:

- analyzes team quality data
- ensures that these analyses are available for team reference
- alerts the team whenever the defined process is not being followed
- recommends how to correct the problems
- alerts the team and management whenever quality problems require special attention

For reporting the quality manager:

- reports weekly to the team on quality measures and product quality status
- maintains the data to produce the defect, yield, ratio, rate, and component sections of the project report during the phase and project post mortem

Questions to ask as Quality Manager - 1

Are the team members properly recording their data?

Do they record the data as they do the work or after the fact?

Are the data complete and of sufficient quality to permit analysis? If not, what remedial actions do you recommend?

Are the team members using their data to assess the quality of their work?

Do the team members' data indicate that the work is of high quality? If not, what remedial actions do you recommend?

Are the team members holding team inspections of the requirements, design, and implementation products and are these inspections being done properly?

Are the team members conducting personal design and code reviews and are these reviews being done properly?

Questions to ask as Quality Manager - 2

Is component and/or module quality being reviewed before integration and system test?

Does the quality of all the components and modules meet the team's quality guidelines before integration and system test? If not, what is being done to fix the quality problems?

Do you need further support from management or the team leader in assuring quality work?

Are there any other quality issues that the team should be aware of?

Messages to Remember

To manage quality, you must plan and monitor quality.

You cannot know your quality performance without detailed tracking of actual data.

Consistent performance depends on the early removal of defects.

Quality does not happen by accident. It must be made to happen.

Software Engineering Institute | Carnegie Mellon