

Remimeo

C/S Series 31

PROGRAMMING AND MISPROGRAMMING

There are three important areas of technical application:

1. Auditing Cases.
2. Case Supervising Cases.
3. Programming Cases.

Auditing generally should be gotten into an org on the routine basis of

1. Get Auditing Volume UP.
2. Get Auditing Quality UP.
3. Get C/Sing Volume UP.
4. Get C/Sing Quality UP.
5. Get Programming Volume UP.
6. Get Programming Quality UP.

To do it in any other sequence is to organize before producing or to inhibit production.

Auditing quality is raised by getting in Cramming and getting Cramming done.

C/S Quality is raised by C/S study of cases and the Qual Sec Cramming the C/S.

Programming quality is raised by getting FESes done so that the action does not block production and Cramming or Programming and then studying the case to make the Programming more real and effective.

MISPROGRAMMING

1. Programming a case without data is risky. Dropping out the FES step, not getting white forms done, etc, short cutting on data in general can cost tremendous amounts of lost auditing.
2. Doing a vague general hopeful programme of Repair (Progress) trusting something will come up is ineffective. With data on the person's life even on a pc never before audited, one can hit the key points even if only with 2 way comm on them. Cases that have been audited and are boggy are so for a reason. Programming without finding that reason can be very ineffective and result in few wins.
3. Running a new major program into an incomplete major program can be as deadly as failing to flatten a process before starting another process only more so.
4. Failing to end off a program when its End Phenomena is achieved is another gross error.
5. Being too ignorant of the basic bank and the tech *theory* (as different than processes) is another barrier to programming.
6. Not Programming at all.

The above six are the principal gross errors in programming.

L. RON HUBBARD
Founder