

GABINETE DE COMUNICAÇÕES HUBBRD
Solar de St. Hill, Grinstead Oriental, Sussex,
HCOB DE 31 DE MARÇO DE 1971

C/S Série 31

PROGRAMAR E PROGRAMAR MAL

Existem três áreas importantes de aplicação técnica:

1. Audição de casos.
2. Supervisão de Casos.
3. Programar casos.

A audição deve geralmente ser introduzida numa org na base de rotina de:

1. Obter um ALTO Volume de Audição.
2. Obter uma ALTA Qualidade de Audição.
3. Obter um ALTO Volume de C/S.
4. Obter uma ALTA Qualidade de C/S.
5. Obter um ALTO Volume de Programação.
6. Obter uma ALTA Qualidade de Programação.

Fazê-lo em qualquer outra sequência é organizar antes de produzir ou inibir a produção.

A qualidade da *Audição* é aumentada entrando em Cramming e produzindo Cramming.

A qualidade do *C/S* é aumentada pelo estudo do *C/S* dos casos e do Cramming do *C/S* pelo Sec de Qual.

A qualidade da *programação* é aumentada produzindo FES para que a acção não bloqueie a produção, Cramming e Programação estudando então o caso a fim de tornar a Programação mais real e eficaz.

PROGRAMAR MAL

1. Programar um caso sem dados é arriscado. Descurar o passo de FES, não ter Forms Brancos, etc., atalhar dados em geral, pode custar a perda de tremendas quantidades de audição.
2. Fazer um vago, geral, esperançoso Programa de Reparação (Progresso) acreditando que algo surgirá, é ineficaz. Com dados sobre a vida das pessoas e até dum Pc nunca dantes auditado, podemos tocar nos pontos chave mesmo que apenas com 2WC. Casos que foram auditados e estão atoladiços, estão assim por alguma razão. Programar sem encontrar essa razão, pode ser muito ineficaz e resulta em poucos ganhos.
3. Correr um novo programa maior dentro de um programa maior incompleto, pode ser tão mortal como deixar de esgotar um processo antes de iniciar outro, ou mais.
4. Deixar de terminar um programa quando o seu EP é atingido, é outro erro grosseiro.
5. Ser muito ignorante do banco básico e da *teoria* da tech, (diferentemente dos processos) é outra barreira à programação.
6. Não programar nada.

Os cinco erros acima são os principais erros grosseiros em programação.

L. RON HUBBARD
Fundador