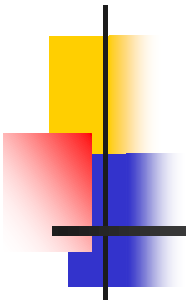
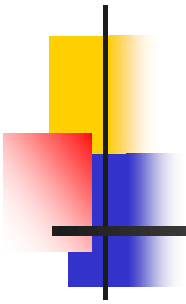


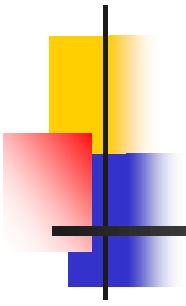
REUSE and METRICS

- Reuse libraries
- Component based S/W Engineering:CBSE
- Object oriented techniques promote reuse



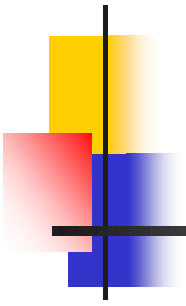
Commercial Reuse

- Microsoft MFC library
- Visual Basic Controls
- Component object model COM objects
- Java beans
- API`s
- CORBA
- STL : standard template library
- COTS



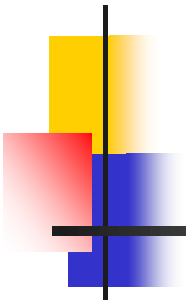
Reuse and Quality

- You could reuse any deliverable item, eg project plan, user manual
- For reuse of code there are models that concern residual bugs



Model basis

- If a system consists in part of reused code its quality will probably be higher
- The reused components have passed through integration and system test again
- On subsequent reuse this happens again



Model mathematics

- Dvr – residual bugs in reused code
- Dvn – residual bugs in the new code
- Dr1 = Dvn(1-R) + Dvr(R) where R is the proportion of reused code

Code reused is debugged again

PHASE	% of lifetime errors
Inspect spec	7.69
Inspect design	19.7
Inspect code	23.93
Test modules	20.88
*Integration test	14.27
*System test	7.92
Residual bugs	5.61
* *Repeated on reuse*	

So if 22.19% of bugs are found on reuse

- Let p = 1 - 0.2219
- Dvn = 4 Dvr = 2 errors/KLOC , 0.6 is reused

- Dr1 = 2.8
- Dr2 = 4 x 0.4 + 2 x 0.6 x p
- Dr2 = 2.53372

- In general Dri =

Dvn x (1-R) + Dvr x R x p

REUSE LEVERAGE

- $R_{lev} = OBJ_{reused} / OBJ_{built}$

- As R_{lev} increases then benefit increases: R_b

- R_b of system $S = \frac{[C_{noreuse} - C_{reuse}]}{C_{noreuse}}$