

```

1  -- vim: ft=lua ts=2 sw=2 et:
2
3  local o,oo,obj,from,within,COCOMO
4
5  fmt = string.format
6  function oo(t) print(o(t)) end
7  function o(t, u)
8    if t<0 then return "["..table.concat(map(t,tostring)," ").".".."]" else
9    u={}; for k,v in pairs(t) do u[1+#u] = fmt("%.5s%s",k,v) end
10   return (t.is or "").."["..table.concat(sort(u)," ").".".."]" end end
11
12 function obj(name, t,new)
13   function new(kl,...)
14     local x=setmetatable({},kl); kl.new(x,...); return x end
15   t = {__tostring=o, is=name or ""}; t.__index=t
16   _ = t
17   return setmetatable(t, {__call=new}) end
18
19 function _cocomo()
20   local p,n,s,w="*",_,"",**
21   return { loc = {"I",2,200},
22     acap= {n,1,5}, cplx={p,1,6}, prec={s,1,6},
23     aexp= {n,1,5}, data={p,2,5}, flex={s,1,6},
24     ltex= {n,1,5}, docu={p,1,5}, arch={s,1,6},
25     pcap= {n,1,5}, pvols={p,2,5}, team={s,1,6},
26     pcon= {n,1,5}, rely={p,1,5}, pmat={s,1,6},
27     plex= {n,1,5}, ruse={p,2,6},
28     sced= {n,1,5}, stor={p,3,6},
29     site= {n,1,5}, time={p,3,6},
30     tool= {n,1,5} } end
31
32 function _risk()
33   local _r,ne,nw,sw4,sw,sw4,ne46,w26,sw46
34   o = 0
35   ne={ {o,o,o,1,2,o}, -- bad if lohi
36     {o,o,o,o,1,o},
37     {o,o,o,o,o,o},
38     {o,o,o,o,o,o},
39     {o,o,o,o,o,o},
40     {o,o,o,o,o,o} }
41   nw={ {2,1,o,o,o,o}, -- bad if lolo
42     {1,o,o,o,o,o},
43     {o,o,o,o,o,o},
44     {o,o,o,o,o,o},
45     {o,o,o,o,o,o},
46     {o,o,o,o,o,o} }
47   nw4={ {4,2,1,o,o,o}, -- very bad if lolo
48     {2,1,o,o,o,o},
49     {1,o,o,o,o,o},
50     {o,o,o,o,o,o},
51     {o,o,o,o,o,o},
52     {o,o,o,o,o,o} }
53   sw={ {o,o,o,o,o,o}, -- bad if hilo
54     {o,o,o,o,o,o},
55     {o,o,o,o,o,o},
56     {1,o,o,o,o,o},
57     {2,1,o,o,o,o},
58     {o,o,o,o,o,o} }
59   sw4={ {o,o,o,o,o,o}, -- very bad if hilo
60     {o,o,o,o,o,o},
61     {1,o,o,o,o,o},
62     {2,1,o,o,o,o},
63     {4,2,1,o,o,o},
64     {o,o,o,o,o,o} }
65   -- bounded by 1..6
66   ne46={ {o,o,o,1,2,4}, -- very bad if lohi
67     {o,o,o,o,1,2},
68     {o,o,o,o,1,1},
69     {o,o,o,o,o,o},
70     {o,o,o,o,o,o},
71     {o,o,o,o,o,o} }
72   sw26={ {o,o,o,o,o,o}, -- bad if hilo
73     {o,o,o,o,o,o},
74     {o,o,o,o,o,o},
75     {o,o,o,o,o,o},
76     {1,o,o,o,o,o},
77     {2,1,o,o,o,o} }
78   sw46={ {o,o,o,o,o,o}, -- very bad if hilo
79     {o,o,o,o,o,o},
80     {o,o,o,o,o,o},
81     {1,o,o,o,o,o},
82     {2,1,o,o,o,o},
83     {4,2,1,o,o,o} }
84   return {
85     cplx= {acap=sw46, pcap=sw46, tool=sw46}, --12
86     ltex= {pcap=nw4}, -- 4
87     pmat= {acap=nw, pcap=sw46}, -- 6
88     pvols= {plex=sw}, --2
89     rely= {acap=sw4, pcap=sw4, pmat=sw4}, -- 12
90     ruse= {aexp=sw46, ltex=sw46}, --8
91     sced= {cplx=ne46, time=ne46, pcap=nw4, aexp=nw4, acap=nw4,
92       plex=nw4, ltex=nw, pmat=nw, rely=ne, pvols=ne, tool=nw}, -- 34
93     stor= {acap=sw46, pcap=sw46}, --8
94     team= {aexp=nw, sced=nw, site=nw}, --6
95     time= {acap=sw46, pcap=sw46, tool=sw26}, --10
96     tool= {acap=nw, pcap=nw, pmat=nw} } end -- 6
97
98 function from(lo,hi) return lo+(hi-lo)*math.random() end
99 function within(t) return t[math.random(#t)] end
100
101 COCOMO=obj"COCOMO"
102
103 function _:NEW(coc,risk)
104   self.x={}; self.y={}
105
106 function _:set(t)
107   self.y = {}
108   for k,v in pairs(t) do self.x[k] = v end end
109
110 function _:effort()
111   local em,sf=1,0
112   for k,t in pairs(self.coc) do
113     if t[1] == "*" then em = em * self.y[k]
114     elseif t[1] == "-" then em = em * self.y[k]
115     elseif t[1] == "+" then sf = sf + self.y[k] end end
116   return self.y.a*self.x.loc^(self.y.b + 0.01*sf) * em end
117
118 function _:risks()
119   local n=0
120   for al,t in pairs(self.risk) do

```

```

121     for a2,m in pairs(t) do
122       n = n + m[self.x[al]][self.x[a2]] end end
123   return n end
124
125 function _:y(w,x)
126   if w=="I" then return x end
127   if w=="4" then return (x-3)*from( 0.073, 0.21 ) + 1 end
128   if w=="-" then return (x-3)*from(-0.187, -0.078) + 1 end
129   return (x-6)*from(-1.56, -1.014) end
130
131 function _:ready(coc,risk)
132   local y,effort,ready,lo,hi
133   coc0, risk0 = cocomo.defaults()
134   coc = coc or coc0
135   risk = risk or risk0
136   for k,t in pairs(coc) do
137     lo,hi = t[2],t[3]
138     self.x[k] = int(self.x[k] and within(self.x[k],lo,hi) or
139       from(lo,hi))
140     self.y[k] = self.y[k] or self:y(t[1], self.x[k])
141   end
142   self.y.a = self.y.a or from(2.3, 9.18)
143   self.y.b = self.y.b or (.85-1.1)/(9.18-2.2)*self.y.a+.9+(1.2-.8)/2
144   self.y.effort = self.y.effort or cocomo:effort()
145   self.y.risk = self.y.risk or cocomo.risks()
146   return self end
147
148 Eg.all {
149   one = function(self)
150     local function say()
151       print("")
152       --lib.o(i.x)
153       lib.oo (effort= self.y.effort,
154         loc = self.x.loc,
155         risk = self.y.risk,
156         pcap = self.x.pcap)
157     end
158     self = cocomo.ready()
159     cocomo.new(self, {pcap=4})
160     self = cocomo.ready(self)
161     say()
162   cocomo.set(self, {pcap=1})
163   self = cocomo.ready(self)
164   say()
165   end}
166
167
168

```