Object Points

4th Generation Languages

- A set of high level languages
  (each “generation” moves away from machine code)
- They can be distinguished between:
  - Data management tools: that allows sophisticated functions for data manipulation (examples: SAS, SPSS, Mathematica, Matlab, ABAP)
  - Report and form generation languages, that specify the data format and the output to be generated (e.g. SQL, Oracle Forms)
  - Automatic code generators from CASE tools (e.g Rails)

Object Points

- Function-related metric for 4th Generation Languages
- Computation is simpler than FP

- Some references:
  - Kauffman, Wright, Zweig, Automating Output Size and Reuse Metrics in a Repository-Based Computer Aided Software Engineering (CASE) Environment IEEE Trans. Software Engineering, 20(3), p. 169-186, 1994

Object Points Computation

- Similar to FP
- Compute the number of screens and classify them as simple, medium, complex
- Compute the number of reports and classify them as simple, medium, complex
- Count the number of modules that have to be developed
- Use weight matrices to sum the values above, taking into account reused code
- A formula translates OPs into productivity measures

Screen and Report Classification

Object point complexity levels for screens

	Number and sources of data tables		
Number of Views Contained	**Total < 4**	**Total < 8**	**Total 8+**
<3	simple	simple	medium
3-7	simple	medium	difficult
8+	medium	difficult	difficult

Object point complexity levels for reports

	Number and source of data tables		
Number of Sections Contained	**Total < 4**	**Total < 8**	**Total 8+**
0-1	simple	simple	medium
2-3	simple	medium	difficult
4+	medium	difficult	difficult

Source: http://yunus.hacettepe.edu.tr/~sencer/objectp.html

Object Points Computation

	Weight		
Type	**Simple**	**Medium**	**Difficult**
Screen	1	2	3
Report	2	5	7
Modules	10	10	10

$$NOP = \sum_{i=1}^{3} \left[\begin{array}{ccc} k_i^s & k_i^m & k_i^d \end{array} \right] \cdot \left[\begin{array}{c} n_i^s \\ n_i^m \\ n_i^d \end{array} \right] \cdot \frac{(100 - r)}{100}$$

$$EFFORT = \frac{NOP}{PROD}$$

where:

r = percentage of components reused

PROD = productivity

Productivity estimates

- Productivity **between 4 and 50 object points/month**, depending on tool support and developer capability:

Developer Experience	Very Low	Low	Nominal	High	Very High
PROD	4	7	13	25	50

Adapted from: http://yunus.hacettepe.edu.tr/~sencer/objectp.html

- Other works indicates up to **75 OP/MM**

Remarks

- Simple to estimate
  (remember Joel Henry's approach to FP computation)
- OP takes into account reused code
  (something which will also appear in some elaborations of the COCOMO model)