Topics covered

1. The production process
2. Engineering & Management perspective
3. Simple quantitative process models
4. Quantitative models & Economics concepts
5. Datasets and distributions
6. Explicit & implicit variables
7. Multi-dimensional vs multiple simpler models
2.1 The production (development) process
Concerns about the development process

• How can we figure out the performance of a software development process?

• How can we build quantitative models of development processes?
The development process can be modeled as a production process.

The production process

- Process order
- Inputs
- Process activities
- Outputs
A process order

Engineering plans

Requirements (Ex. UML Diagrams)
Inputs

Resources for construction

Human Resources


http://openexhibits.org/downloads/
Activities

Construction activities


Software development activities

http://www.atksolutions.com/articles/software_development_life_cycle.html
Outputs

A house

http://os-vecror.com/tag/colonial-house-clip-art

The software

http://www.clker.com/clipart-software.html
A production process of software
2.2 Engineering & Management perspective on a production process
Engineering & Management perspective

[Diagram showing the relationship between Organizational Objectives, Projects, Customers, Input resources, Product Requirements, Monitoring & Control process, Measures, Control Decisions, Constraints, and the Software Product.]
Monitoring & Control Process

• This process includes:
  • A collection of measurement results about the current and past performance of the process.
  • An analysis of the process performance against the project objectives and the goals of the organization.
  • Decisions to make adjustments (through evaluation and decision models)
Project objectives

- Specific to a project
- Generally multiple and concurrent:
  - Deliver a number of software functions,
  - within a specified time frame,
  - within a specified (limited) budget, and
  - with a level of quality (not always precisely specified)

In the Agile methodology, these correspond to Sprint objectives.
Organizational objectives

• Broader scope, not limited by the project objectives.

• Longer term view.

• Concerned with issues that extend beyond the life of the project.
  
  • Impact of the quality delivered by a development project over the many years of maintenance of the software being developed.

  • Usage of standards
2.3
Simple Quantitative Process Models
Productivity

Productivity = Outputs / Inputs

Number of functions delivered
Number of worked hours
Examples of productivity

• Web-based catalog application:
  • Average productivity of organization A = 30 Function Points per person-month
  • Average productivity of organization B = 33 Function Points per person-month.
The productivity ratio

Size of the Software Product

Outputs = Measured with ISO standards

Inputs = People-Hours

Sprint-1 -> Sprint-2 -> Sprint-3 -> Sprint-4 -> Sprint-5
Unit effort (unit cost)

\[ \text{Unit effort} = \frac{\text{Inputs}}{\text{Outputs}} \]
Examples of unit effort

- **Web-based catalog application:**
  - **Organization A:**
    - 210 work-hours in a person-month
    - 30 Function Points per month
    - Unit effort = 210 hours/30 Function Points = 7 hours/Function Point
  - **Organization B:**
    - 210 work-hours
    - 10 Function Points per month
    - Unit effort = 210 hours/10 Function Points = 21 hours/Function Point
Average

• The average productivity is built by:
  • calculating the productivity ratios of each individual project in a sample
  • adding them up, and
  • dividing the total by the number of projects in the sample
Characteristics related to the Average

- Minimum
- Maximum
- First quartile
- Last quartile
- 1 standard deviation
- 2 standard deviations
- Skewness
- Kurtosis
Standard deviation

• Shows how much variation (or dispersion) there is from the average.
  
  • Low standard deviation: the data points tend to be very close to the average.
  
  • High standard deviation: the data points are spread out over a large range of values.
Average

Box-plot: Average and Quartiles

A normal distribution and standards deviations
Skewness and Kurtosis

• Skewness is a measure of the asymmetry of the probability distribution of a random variable with a real value.
  • It can be positive or negative

• Kurtosis is a description of the ‘peakedness’ of a distribution.
Average

Skewness

Kurtosis

Copyright © 2015 Alain Abran
Linear & non-linear models

- Technique: statistical regression
- The Least-Squares Regression method is typically used to derive the regression equations.

Power models with exponents larger (blue line) or smaller than -1 (red line)
Nonlinear production models

• A model of a production process could be represented by any shape.

• Statistical techniques are available to model any shape.

• Example
  \[ Y \text{ (effort)} = A \times (\text{Size})^B \]
  \[ Y = A + BX + CX^2 \]
Production model with negative slope

Effort (Hours)

Size (Function Points)
2.4 Quantitative Models & Economics Concepts
Variable and Fixed costs

• Variable:
  • Expenses that depend directly on the number of outputs produced.
  • Ej: Number of work hours

• Fixed:
  • Expenses that not depend on the number of outputs.
  • Ej: Project management plans, change control procedures, quality controls, audits, etc.
Variable and fixed costs

Effort (Hours)

Size (Function Points)

a = Variable Costs
b = Fixed Costs
A linear model

\[ Y \text{ (effort in hours)} = f(x) = a \times \text{Size} + b \]

Size = number of Function Points (FP)
\(a = \text{variable cost} = \text{number of hours per Function Point (hours/FP)}\)
\(b = \text{fixed cost in hours}\)

Units:

\[ Y \text{ (hours)} = (\text{hours/FP}) \times \text{FP} + \text{hours} = \text{hours}\]
Production models and fixed costs

No fixed costs

Negative fixed costs (theoretical)
Economies and diseconomies of scale

• Economies of scales
  • The increase in output units requires a smaller increase in input units.

• Diseconomies of scales
  • The increase in output units requires a larger increase in input units.
Economies and diseconomies of scale

Diagram showing the relationship between effort (in hours) and size (in function points) with curves indicating economies of scale and diseconomies of scale. The graph illustrates how as size increases, effort may decrease (economies of scale) or increase (diseconomies of scale).
2.5 Software Engineering Datasets & their Distribution
Wedge-shaped data sets

• Behaviour: Increasing dispersion of effort as size increases.

• Consequence (when all the projects are combined into a single set)
  • Size alone does not adequately describe the relationship with effort.
  • Additional independent variables are necessary.
Wedge-shaped data sets
Wedge-shaped data sets

- Behaviour: Increasing dispersion of effort as size increases.
- Consequence (when all the projects are combined into a single set)
  - Size alone does not adequately describe the relationship with effort.
  - Additional independent variables are necessary.
Wedge-shaped data sets: Causes of high dispersion in productivity

- The project data come from organizations with distinct production processes (distinct productivity behavior).

- The project data represent the development of software products with major differences (software domains, non-functional requirements, etc.).

- The development process is out of control (unpredictable productivity performance from projects developed in an ad hoc manner).

- Data collected is based on post-event opinions, outside a sound measurement program.
Homogeneous data sets

• The dispersion of the effort as size increases is highly consistent.

• The increases in functional size explain 80% to 90% of the increases in effort.
  • All the other factors together explain only 10% to 20% of those increases.
Homogeneous data sets

![Graph showing the relationship between Effort (Hours) and Size (Function Points).]
Homogeneous data sets: Causes of low dispersion in productivity

- The project data come from a single organization with well implemented development standards.
- The project data represent the development of software products with very similar characteristics.
- The development process is under control, with predictable productivity performance (levels 4 or 5 of the CMMI model).
- The data were collected in an organization based on a sound in-process measurement program.
2.6
Productivity Models: Explicit & Implicit variables
Explicit variables

- The output (size) → independent variable
- The input (effort) → dependent variable
Implicit variables

- Team experience
- Project manager experience
- Software engineering environment platform
- Design methodology
- Quality controls
Size: Dominant variable in datasets

- Size is dominant if:
  - Most of the implicit variables in a set of projects are similar for all the projects in the sample.
    - These variables should have minimal impact on the unit cost.
    - The functional size will be the dominant independent variable impacting size.
Example 1: Size as a dominant variable in a dataset

Telon (70, 650)

Effort (Hours)

\[ Y = 5.5 \times Fp + 1046 \text{ hours} \]

\[ R^2 = 0.75 \]

The TELON dataset in the ISBSG 1999 Release
(Abran, Ndiaye, Bourque, 2007)
Example 2: Size as a dominant variable in a dataset

Homogeneous dataset of 21 projects (Abran 1994)
2.7 Multi-Dimensional vs Simpler Models
A classical approach

To build a single multi-variable estimation model and include in it as many cost drivers (i.e. independent variables) as possible (a ‘catch-all’ model).
Building models ...........

• From available data.
• On opinions on cost-drivers.
• Multiple models with coexisting economies and diseconomies of scale.
Models from available data

• The builders have access to a reasonable set of data points on completed projects.
  • The largest possible number of variables included in the available dataset
  • Cost drivers (the authors’ own definition)
  • Measurement rules for these costs drivers (authors’ own definition)
  • Impact factors (authors’ own assignment).

• Complex models with a large number of variables ‘n’
Models built on opinions on cost-drivers

• Feel good estimation models

• Based on practitioners’ opinions
  • about various variables and the corresponding estimated impact.

• This ‘expert judgment approach’
  • It is used when an organization does not collect data.
A more realistic approach

A single model cannot be the best in all circumstances.

There is a very large diversity of development processes, different mixes of costs drivers, and most probably different impacts of these cost drivers, depending on the contexts.
A more realistic approach

The classical concepts of economies and diseconomies of scale are applicable to software development processes.
Wedge shape with 3 data subsets with economies/diseconomies of scale.
Economies and diseconomies of scale

• Zone 1: The lower part of the wedge-shaped dataset represents the set of projects demonstrating large economies of scale.

• Zone 2: Middle of the wedge-shaped dataset.

• Zone 3: The upper part of the wedge-shaped dataset represents the set of projects demonstrating diseconomies of scale.
Economies and diseconomies of scale

• Three distinct productivity models within this single dataset:

\[ f_1(x) = a_1 x + b_1 \quad \text{zone 1} \]
\[ f_2(x) = a_2 x + b_2 \quad \text{zone 2} \]
\[ f_3(x) = a_3 x + b_3 \quad \text{zone 3} \]
Discussion

Group discussion on Figure 2.19 with multiple candidate models from a wedge-shape dataset.
Exercises 1 to 5
Term Assignments 1 to 3