Introduction to System Dynamics Software Process Modelling

University of Sannio
2 May 2001

Juan F Ramil

Department of Computing
Imperial College
180 Queen's Gate
London SW7 2BZ
tel. +44 (0) 20 7594 8216
fax. +44 (0) 20 7594 8215

ramil@doc.ic.ac.uk
http://www.doc.ic.ac.uk/~ramil

Objectives

To introduce basic principles underlying construction, evaluation and use of system dynamics simulation models as a tool to plan, manage and control software processes

- To justify the use of such models to interpret and understand process behaviour and to address a variety of process issues, such as determination of policies for effort allocation

- To provide guidelines for construction of process models based on the system dynamics paradigm

- To provide examples of system dynamics models built using the *Vensim*® tool

Some Sources

Books

- Systems Thinking
  - Senge PM, *The Fifth Discipline*, Currency-Doubleday, New York, 1990

- System Dynamics - General
  - Forrester J, *Industrial Dynamics*, MIT Press, Cambridge, MA, 1961
  - Coyle RG*, System Dynamics Modelling - A Practical Approach,* Chapman & Hall, London, 1996*,* 413 p

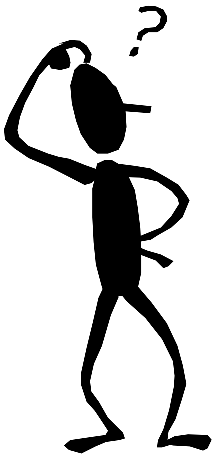
- Software Process Oriented:
  - Abdel-Hamid T and Madnick S, *Software Project Dynamics-An Integrated Approach*
  - Madachy R and Boehm BM*, Software Process Dynamics,* IEEE Computer Society, in preparation

Many Journal, Conference and Workshop Papers

- Survey Papers, covering other approaches in addition to SD :
  - Curtis B, Kellner MI and Over J, *Process Modeling*, Communications of the ACM, v. 35, n. 9, Sept. 1992, pp 75-90
  - Kellner MI, Madachy RJ and Raffo DM, *Software Process Simulation Modelling: Why? What? How?,* Journal of Systems and Software, v. 46, n. 2/3, April 1999, pp 91-106

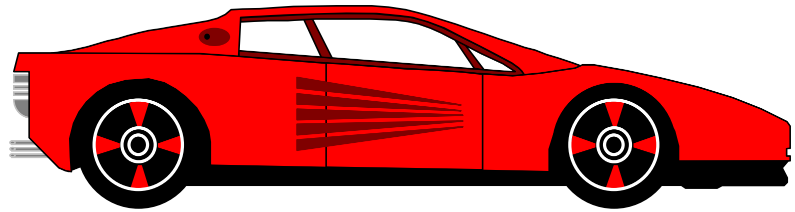
  - Some 40 papers in application of system dynamics modelling to software processes

Motivation



Can we estimate effort, work, schedule?

What is the performance of our software process?



Static Models

- Black box, also termed analytical models - e.g. cost and schedule estimation models - COCOMO, SLIM, for example:

$$\text{Estimated Effort} = a * \text{Size}^{b}$$

- a,b a function of various factors - experience, complexity

- Only a few factors considered

- Do not permit perturbation analysis

Role and Impact of Process and Project Dynamics

- Wide consensus that process and project dynamics plays an important role
  - hard to deny but not widely - explicitly - addressed
  - operates at different levels: team, project, organisation, short and long term, technical and global processes

- Boehm and Madachy mention some of the constituents of project and process dynamics:
  - schedule pressure
  - experience
  - work methods, such as reviews, inspections and Q & A
  - task underestimation
  - bureaucratic delays
  - etc...

- Need to approach process modelling with methods that promote understanding of the dynamics

System Thinking

- Focus on system structure rather than isolated events

- Global system view
  - e.g., local change does not guarantee impact at global system level

- Search for leverage points in process
  - small changes that are likely to provide large desired impact

- Behavioural process modelling, an answer
  - to address questions about evolution planning and management, performance, evolution policies and so on

Dynamic Models

- Simulation-based models - for example
  - discrete event
    - state of system represented by (finite) number of events
    - between events nothing happens
  - continuous time
    - state of system represented by continuous quantities that may change at every instant of time

- Software process exhibits behaviour of
  - discrete event - e.g. start and finish of activities, reception and delivery of an artefact
  - continuous time - e.g. consumption of resources, percentage of work achieved

- In principle different types of modelling paradigms can be useful

- Some workers have shown preference for continuous time simulation
  - it promotes “system thinking”, including a global view of process

System Dynamics Approach

- Developed in the late fifties and early sixties by Forrester in MIT
  - classical control systems insufficient to represent complexity of feedback interactions in many industrial and social systems

- Cycles of excessive and insufficient inventories - one of first problems studied
  - oscillations in inventories, a consequence of policies, a system property

- Example of Forrester's models:
  - urban dynamics
  - world model

- Applications in many fields

- Abdel-Hamid and Madnick - first widely available system dynamic software project model

System Dynamics Models

- Involve a set of coupled, nonlinear differential equations:

$$d\mathbf{x}(t)/dt = f(\mathbf{x}(t),\mathbf{p})$$

where **x**(t) is a vector of *levels* or state variables, f() is a vector-valued function and **p** is a vector of parameters

- solved numerically by simulation tool - Euler, Runge-Kutta, etc.

- models can be developed using a spreadsheet or writing a program a general purpose computer language

- sophisticated tools are available

Tools

- Dynamo
- iThink
- Stella
- Vensim
- See Coyle's book for comparisons across tools

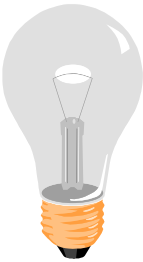
System Dynamics - Applications

- System Dynamics - a tool in software planning & management

  When adequately applied SD and other simulation models can help to:

  - explore counterintuitive behaviour
  - determine impact of existing policies, refine them, identify new improved policies
  - support resource and schedule estimation
  - identify potential process improvements and assess their impact
  - identify actions to mitigate process-related risks
  - build a common language amongst process owners and participants, share mental models, stimulate discussions
  - increase understanding of process dynamics

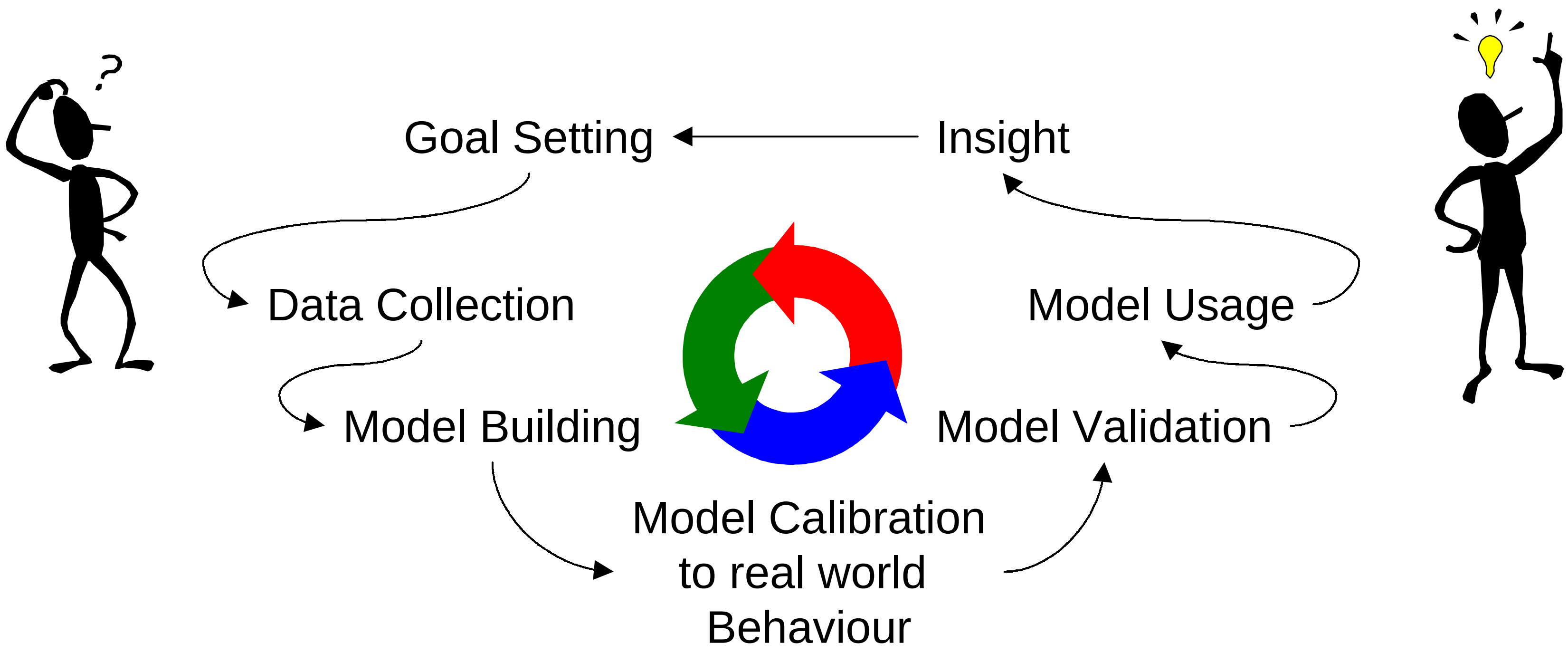
- Other tasks that can be supported by process simulations:



  - post-mortem analysis
  - personnel training - management flight simulators
  - analysis of management heuristics
  - integration with cost estimation systems
  - provision of insights to support decision making
  - support holistic, integrative view of software process

Modelling Process

- Not universally accepted process, but guidelines - art and science
- Displays certain similarities with software design and implementation
  - involves bounding, goal setting, abstraction and aggregation, etc.
- Intrinsically iterative



FEAST View

- Top - down approach
- Number of variables restricted to a minimun
- Modelling global feedback loops
- Step-wise refinement of model

Vensim System Dynamics Modelling Process

- Problem Statement - pencil and paper task

- Identify Variables

- Reference Modes - pencil and paper task

- Reality Check

- Dynamic Hypothesis

- Simulation Model: "...refinement and closure of a set of dynamic hypothesis to an explicit set of mathematical relationships..." *Vensim 4 Modelling Guide*

Simulation Model Representation

- Graphical views

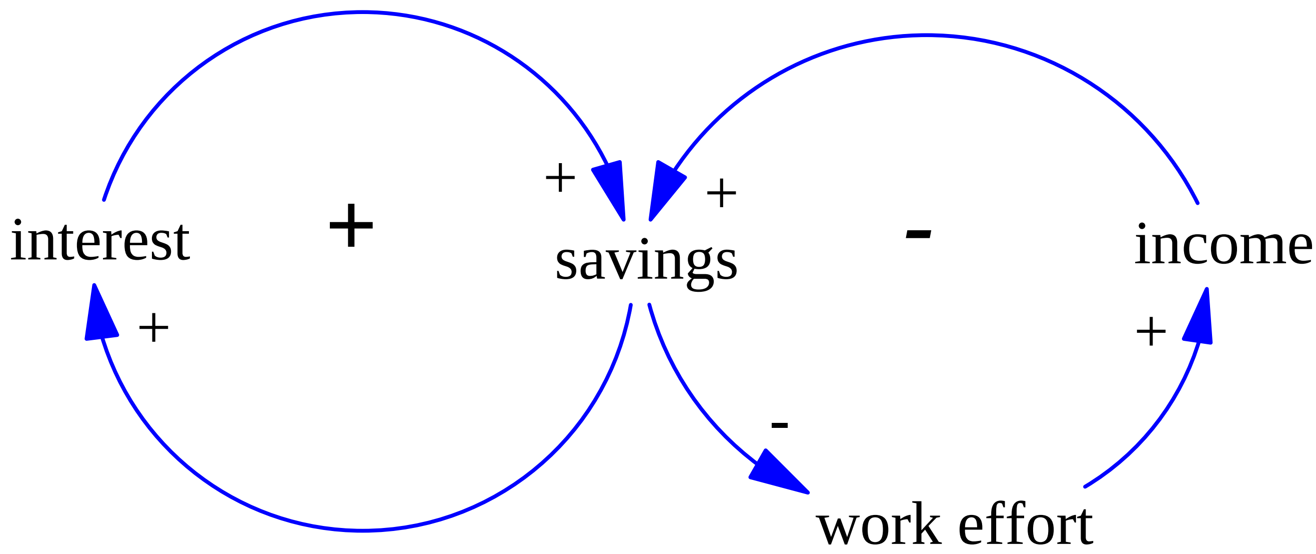
  - influence diagrams - also known as causal loops

  - level - rate diagrams

- Algebraic view
  Simulation languages - e.g. Vensim programming language

Example of an Influence Diagram

- A view of a savings and income problem

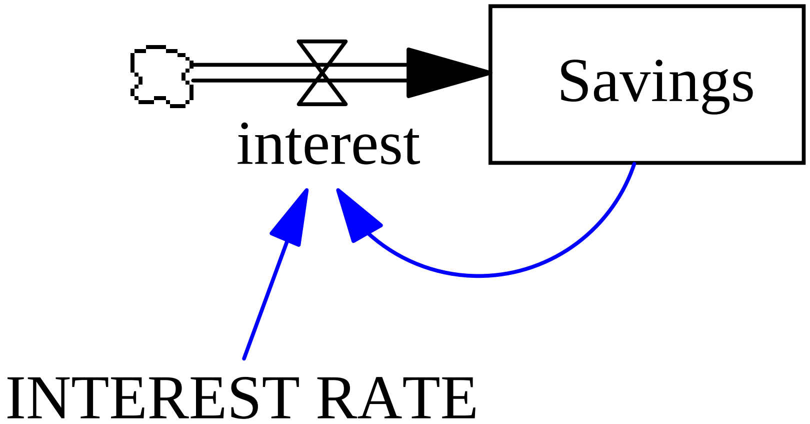


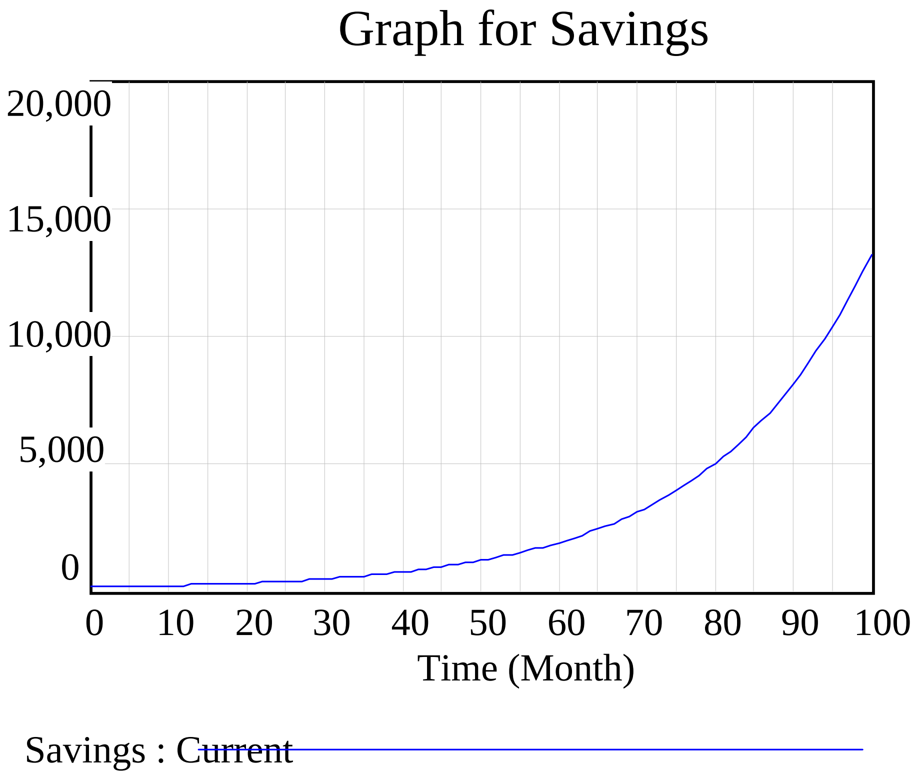
Variable Types in Level-Rate Diagrams

- Levels or Stocks
- Rates
- Auxiliary Variables
- Constants or Parameters

Level-rate Diagram Example

- A view of a savings and income problem



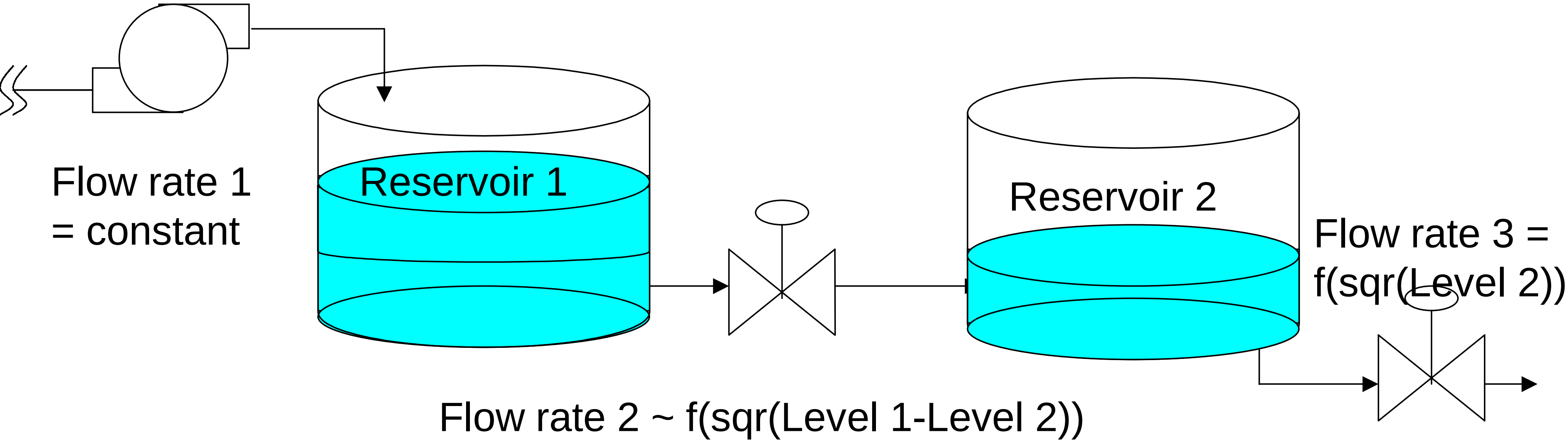


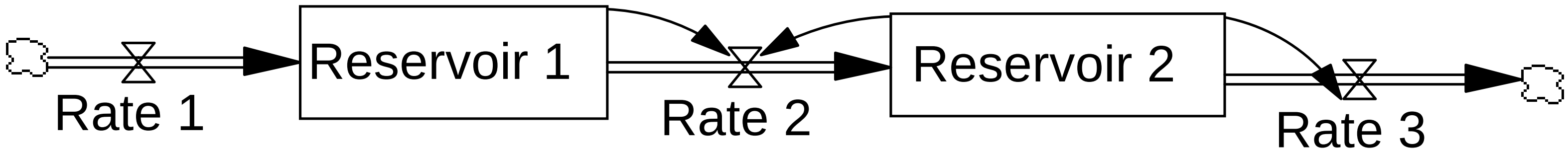
Algebraic View - Vensim language

```
interest=
     Savings*INTEREST RATE
     ~  Liras/Year
     ~  Represents the Cash Flow
              |

INTEREST RATE=
     0.05
     ~ Percent per Year
     ~ Interest Rate applied by Bank
              |

Savings= INTEG (
     interest,
              100)
     ~ Liras
     ~ Amount of Money in the Account
              |
```

```
********************************************************
     .Control
********************************************************~
          Simulation Control Parameters
     |

FINAL TIME  = 100
     ~        Year
     ~        The final time for the simulation.
     |
INITIAL TIME  = 0
     ~        Year
     ~        The initial time for the simulation.
     |
SAVEPER  =
      TIME STEP
     ~        Year
     ~        The frequency with which output is stored.
     |
TIME STEP  = 1
     ~        Year
     ~        The time step for the simulation.
     |
```

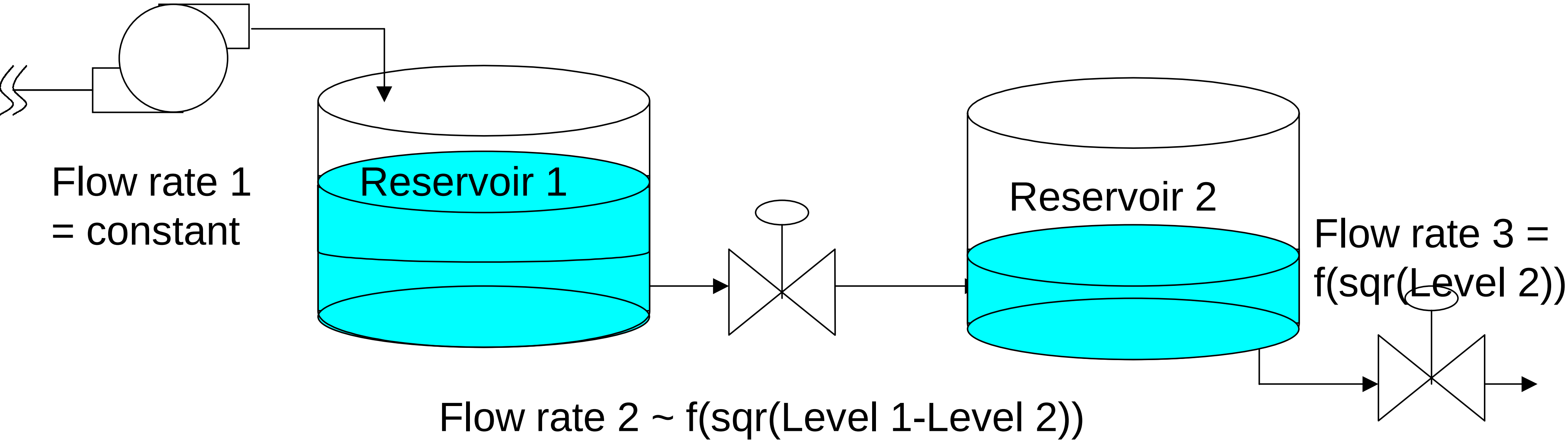
System Dynamics - Fluid System (Coupled Tank) Example (cont.)





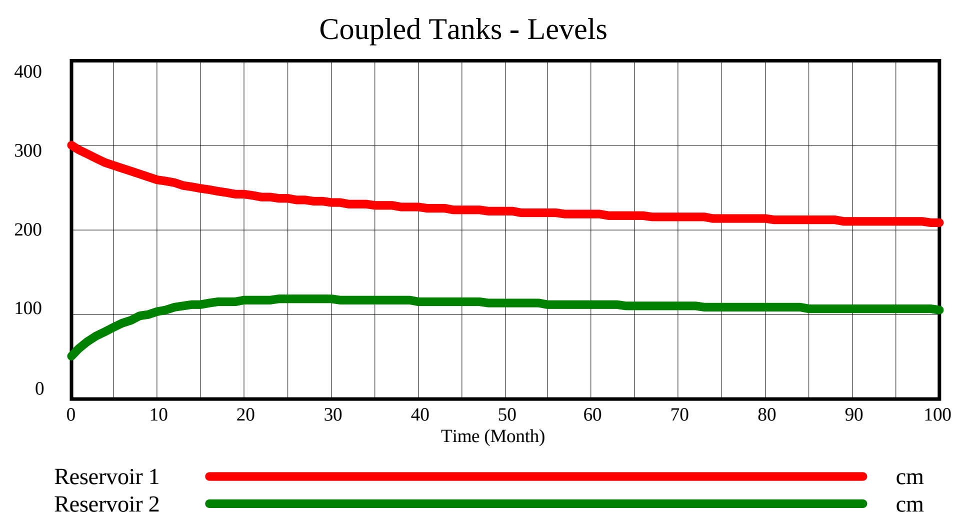
Level-rate diagram

System Dynamics - Fluid System (Coupled Tank) Example



Example in Vensim language

```
Rate 1 = 10
      ~
      ~      |
Rate 2 = IF THEN ELSE(Reservoir 1>Reservoir 2,
sqrt(abs(Reservoir 1-Reservoir 2)),
-1*sqrt(abs(Reservoir 2-Reservoir 1)))
      ~
      ~      |
Rate 3 =IF THEN ELSE(Reservoir 2>0,sqrt(Reservoir
2),0)
      ~
      ~      |
Reservoir 1 = INTEG(Rate 1-Rate 2,300)
      ~
      ~      |
Reservoir 2 = INTEG(Rate 2-Rate 3,50)
      ~
      ~      |
```



Brook's Law - Example

"Adding people to a late project makes it later"

Fred Brooks in *The Mythical Month-Month,* 1975, Anniversary Edition 1995

Why?

Brook’s Law - Why adding people to a late project, makes it later?

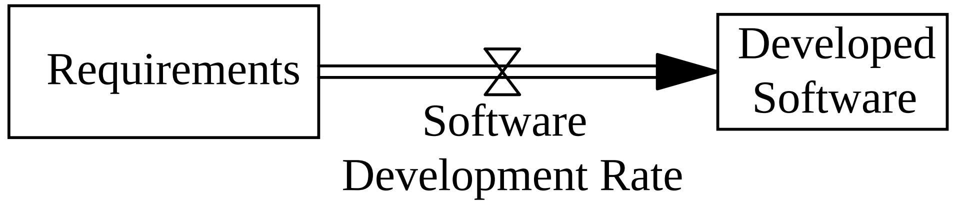
- Experienced developers must reduce their work rate to train new personnel

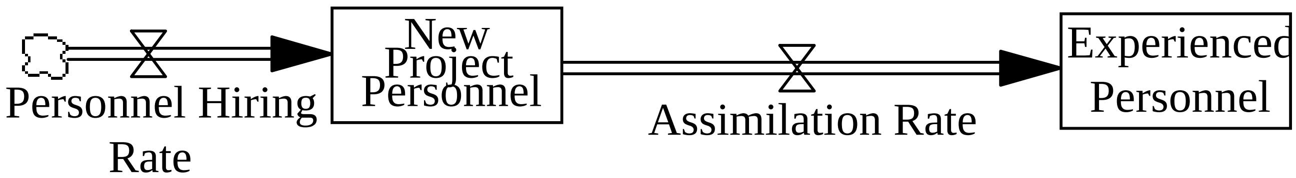
- As a team grows in size, effort wasted in communication increases
  - communication looses expected to be proportional to n(n-1)

  Abdel-Hamid and Madnick suggest expression: $0.06(n)^2$

Brook's Law - SD Model - follows closely model proposed by Madachy and Boehm

- 2 subsystems: work progress and human resource

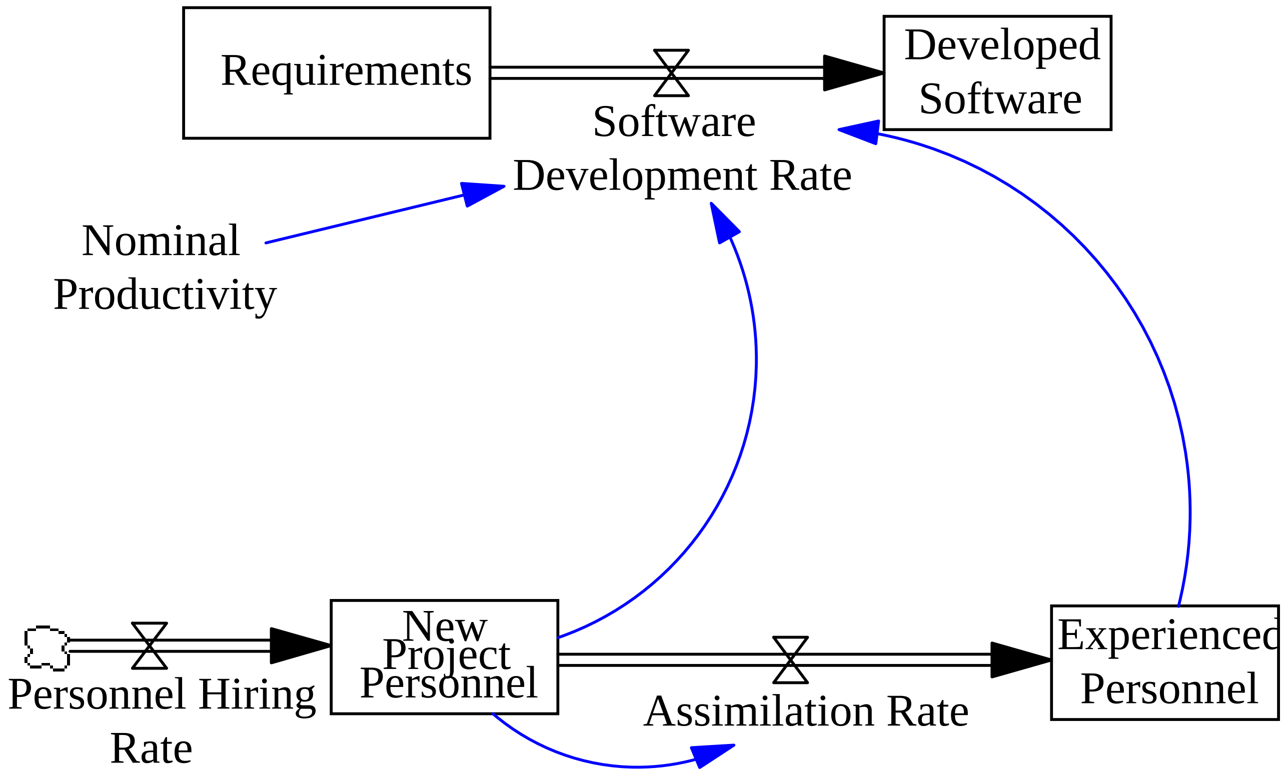




- 4 levels, 3 rate variables

Brook's Law - SD Model

- Software Development Rate as a function of resource

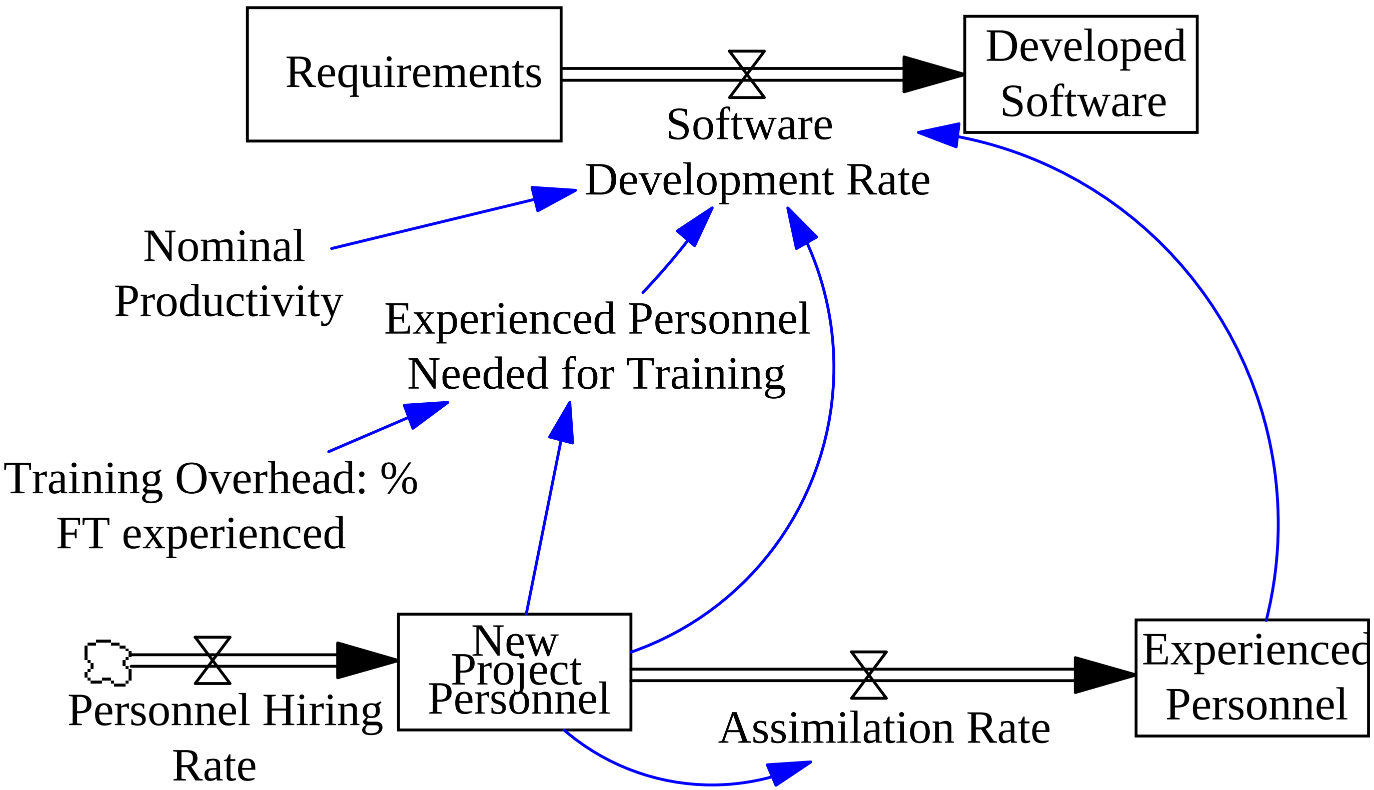


Software Development Rate = Nominal Productivity*(0.8*New Project Personnel+1.2*(Experienced Personnel-Experienced Personnel Needed for Training))

Assimilation Rate = New Project Personnel/20

Brook's Law - SD Model

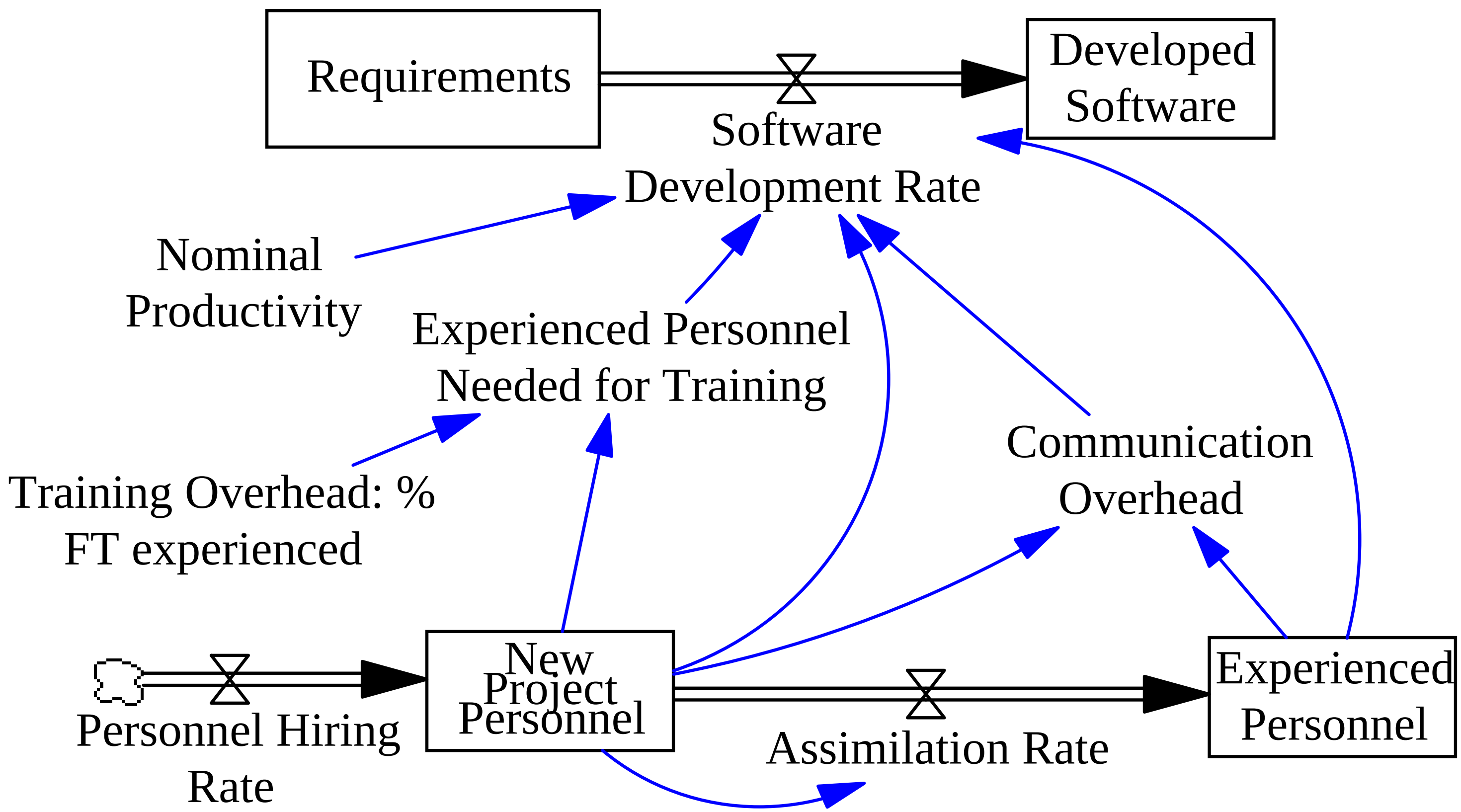
- A fraction of the experienced personnel effort gets diverted into training



Software Development Rate = Nominal Productivity*(0.8*New Project Personnel+ 1.2*(Experienced Personnel **-Experienced Personnel Needed for Training**))

Brook's Law - Full Model

- Communication losses as team size increases

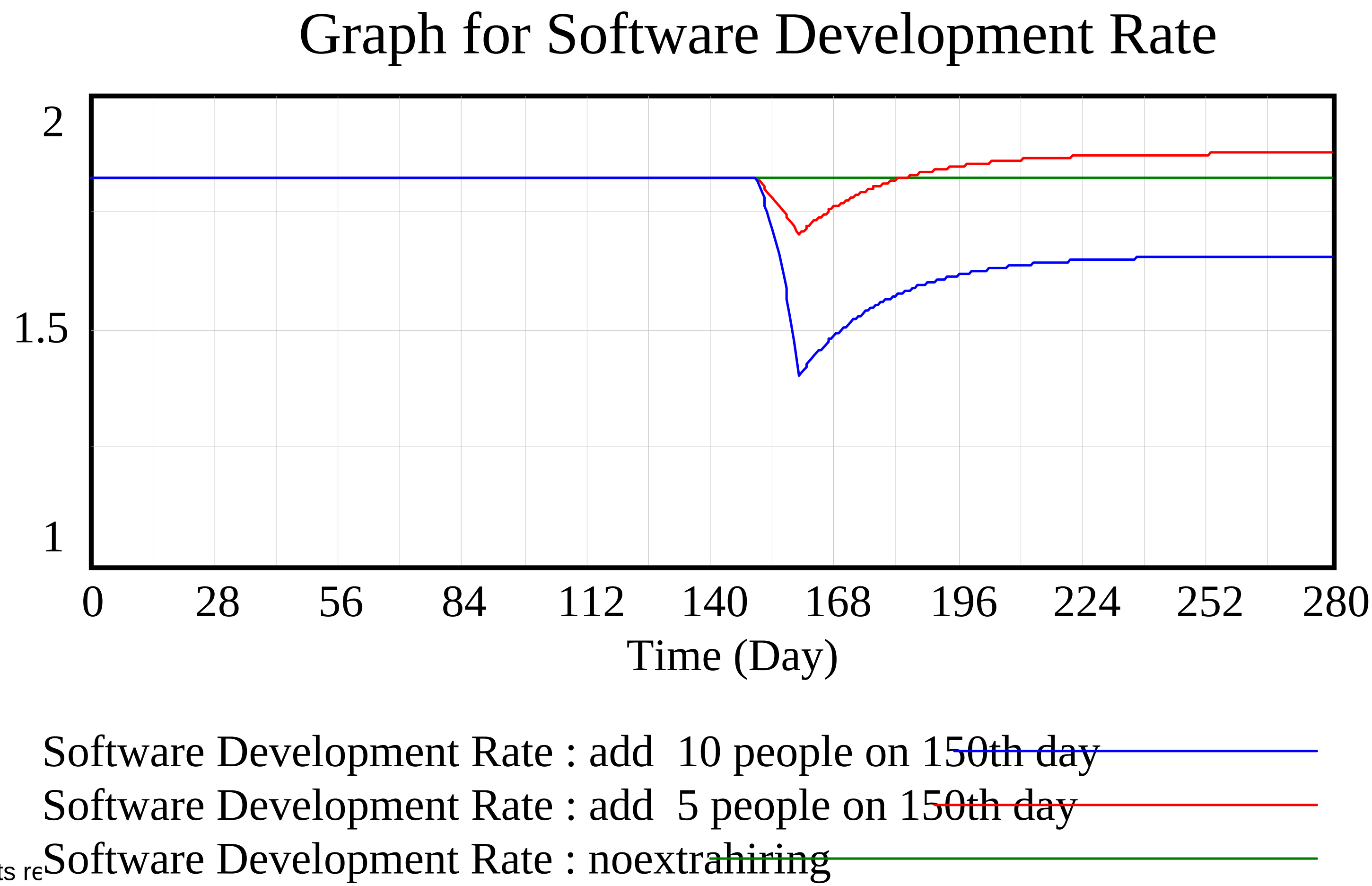


Software Development Rate = Nominal Productivity***(1-Communication Overhead/100)***
(0.8*New Project Personnel+1.2*(Experienced Personnel-Experienced Personnel Needed for Training))

Communication Overhead = 0.06*(Experienced Personnel+New Project Personnel)^2

Brook’s Law

- Output associated to different hiring policies - no hiring, add 5, add 10 people
- In model: adding people to the project brings a temporary slow down in work rate
- In model: adding people to the project beyond certain threshold reduces work rate



Related Topics

- Deterministic vs Stochastic simulations
- Monte Carlo simulation
- Optimisation

Some Tips

- Simulation step should be sufficiently small
  - time step - in Vensim convenience of use 2^{-n} = 1, 0.5, 0.25, 0.125, ... time units

- Be aware of possible effects of integration algorithm - Euler, Runge-Kutta

- Keep each model in its individual directory or folder

- Software process models are software too...

> 90 percent of what you learn from a model is during its construction - *M Lehman*

Advanced Topics in Simulation Process Modelling

- Hybrid simulations - combination of different simulation paradigms
- Modelling human behaviour
- Top-down vs bottom-up - emergent properties
- Generic process models, building blocks
- Model *evolution*
- Combined use of black-box and white-box models
- Validation, validation, validation...