D7.6.1-4 Model Library: project and operations management

D7.6.1 – Public project management library
D7.6.2 – Company specific project management library
D7.6.3 – Public O&M library
D7.6.4 – Company specific O&M library

MODRIO (11004)

Version 1.1

Date 30/11/2015

Authors

Joona Tuovinen VTT

Executive summary

This is a collection of system dynamics structures often used in project and operations management models. These common structures are useful as a reference to what type of complex problems have already been identified and how they are usually modeled. The collection summarizes the MODRIO model library deliverables relating to project and operations management modeling and serves as the public deliverable for the following:

D7.6.1– Public project management library
D7.6.2 – Company specific project management library
D7.6.3 – Public O&M library
D7.6.4 – Company specific O&M library

Summary

1. Reoccurring Structures

This is a collection of system dynamics structures often used in project and operations management models. It is useful as a reference to what type of decision making structures have already been identified and how they have been formulated.

The purpose is not to give parts that assemble into a comprehensive working model. Rather, the purpose is to show structures, small parts or sub-models that reoccur in project and operations management models. This is to give ideas to how certain aspects or dynamics can and often are formulated in the system dynamics framework.

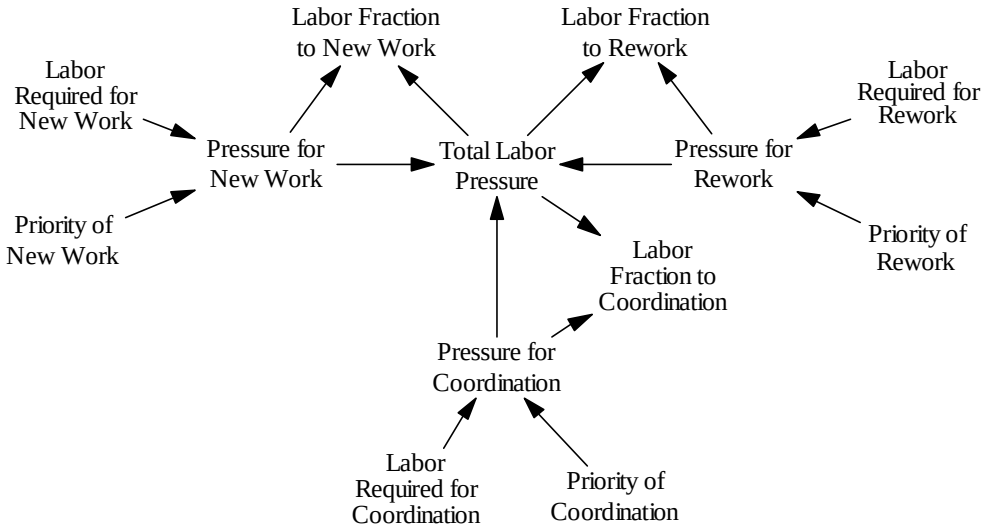
These reoccurring structures should not be used as invariant. They should be adopted and changed to address the specific questions or problem hypothesis that each individual models is built for. One should never feel obligated to use the structures as is. Instead, let them just be a starting point to check that you have addressed some of the relevant dynamics that many other researchers and practitioners have already discovered.

1.1. Allocation

Description:

The allocation structure describes how a limited resource is distributed between different purposes or users. In project management models (see e.g. [1][2][3]) labor is usually allocated to different tasks (e.g. new initial work or rework) depending on task needs and priorities.

Structure:



Equations:

```
Labor Fraction to Coordination=Pressure for Coordination/Total Labor Pressure
        Units: fraction
Labor Fraction to New Work=Pressure for New Work/Total Labor Pressure
        Units: fraction
Labor Fraction to Rework=Pressure for Rework/Total Labor Pressure
        Units: fraction
Labor Required for Coordination
        Units: man*hours
```

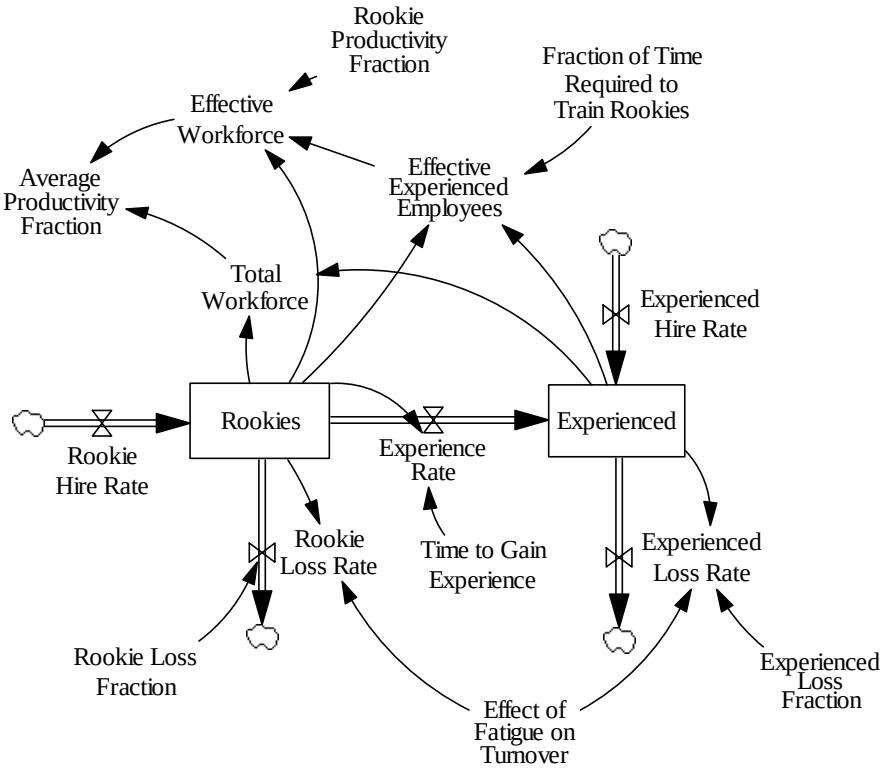
Labor Required for New Work
Units: man*hours
Labor Required for Rework
Units: man*hours
Pressure for Coordination=Priority of Coordination*Labor Required for Coordination
Units: man*hours
Pressure for New Work=Priority of New Work*Labor Required for New Work
Units: man*hours
Pressure for Rework=Priority of Rework*Labor Required for Rework
Units: man*hours
Priority of Coordination
Units: dimensionless
Priority of New Work
Units: dimensionless
Priority of Rework
Units: dimensionless
Total Labor Pressure=Pressure for New Work+Pressure for Rework+Pressure for Coordination
Units: man*hours

1.2. Employee Mix

Description:

The employee mix is a common way to model the learning and experience of employees. (see e.g. [2][4][5][11] or for more disaggregate mixes [6][7]) Usually it is a way to model how experience affects productivity, quality, and the delay in turning a new hire into a fully productive worker. Many models utilize this type of employee mix aging chains instead of a co-flow experience structure. For the co-flow experience formulation see e.g. [1][2]

Structure:



Equations:

Average Productivity Fraction=Effective Workforce/Total Workforce
Units: dimensionless
Effect of Fatigue on Turnover
Units: dimensionless
Effective Experienced Employees= MAX(0,Experienced-Fraction of Time Required to Train Rookies*Rookies)
Units: employees
Effective Workforce=Effective Experienced Employees+Rookie Productivity Fraction*Rookies
Units: employees
Experience Rate=Rookies/Time to Gain Experience
Units: employees/Month
Experienced= INTEG (Experience Rate+Experienced Hire Rate-Experienced Loss Rate, Initial Experienced)
Units: employees
Experienced Hire Rate
Units: employees/Month
Experienced Loss Fraction
Units: dimensionless/Month
Experienced Loss Rate=Experienced Loss Fraction*Experienced*Effect of Fatigue on Turnover
Units: employees/Month
Fraction of Time Required to Train Rookies
Units: dimensionless

Rookie Hire Rate
Units: employees/Month
Rookie Loss Fraction
Units: dimensionless/Month
Rookie Loss Rate=Rookie Loss Fraction*Rookies*Effect of Fatigue on Turnover
Units: employees/Month
Rookie Productivity Fraction

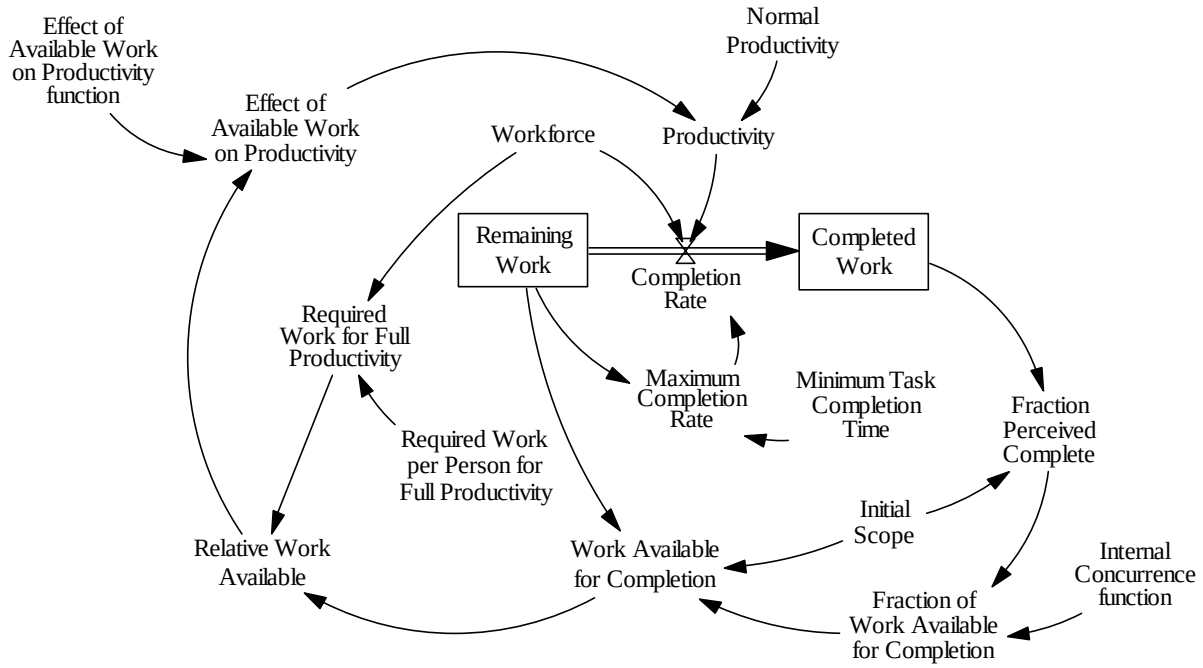
Units: dimensionless
Rookies= INTEG (Rookie Hire Rate-Experience Rate-Rookie Loss Rate, Initial Rookies)
Units: employees
Time to Gain Experience
Units: months
Total Workforce=Rookies+Experienced
Units: employees

1.3. Limited Completion Rate

Description:

The limited completion rate structure shows commons ways how work completion might be limited or how to keep a stock of work from being drained to negative. The simplest being formulated through a *Minimum Task Completion Time* that limits the completion to the fastest possible time to finish all the remaining work. (see e.g. [4]) Another limiting factor might be that by adding workers you aren't able to increase the completion rate indefinitely. (see e.g. [1][3]) A hundred men might take the same time as one man to finish changing a light bulb. A third limiting factor might be internal or external work concurrence meaning that all the work might not be available instantly. One cannot work building the tenth floor of a high-rise before the ninth floor is built. (see e.g. [8] for external concurrence)

Structure:



Equations:

Completed Work= INTEG (Completion Rate,0)
Units: tasks
Completion Rate=MIN(Maximum Completion Rate, Workforce*Productivity)
Units: tasks/Month
Effect of Available Work on Productivity=Effect of Available Work on Productivity function(Relative Work Available)
Units: dimensionless

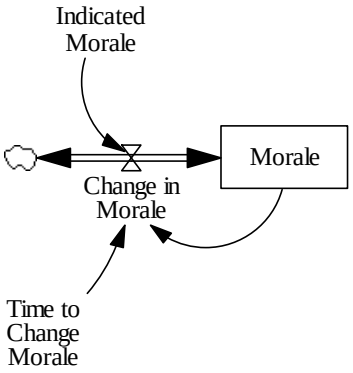
Effect of Available Work on Productivity function= *user defined function*
Units: dimensionless
Fraction of Work Available for Completion= Internal Concurrence function(Fraction Perceived Complete)
Units: dimensionless
Fraction Perceived Complete=Completed Work/Initial Scope
Units: dimensionless
Initial Scope
Units: tasks
Internal Concurrence function= *user defined function*
Units: dimensionless
Maximum Completion Rate=Remaining Work/Minimum Task Completion Time
Units: tasks/Month
Minimum Task Completion Time
Units: months
Normal Productivity
Units: tasks/person/Month
Productivity=Effect of Available Work on Productivity*Normal Productivity
Units: tasks/person/Month
Relative Work Available= Work Available for Completion/Required Work for Full Productivity
Units: dimensionless
Remaining Work= INTEG (-Completion Rate, Initial Scope)
Units: tasks
Required Work for Full Productivity=Required Work per Person for Full Productivity*Workforce
Units: tasks
Required Work per Person for Full Productivity
Units: tasks/person
Work Available for Completion=Initial Scope*Fraction of Work Available for Completion
-(Initial Scope-Remaining Work)
Units: tasks
Workforce
Units: people

1.4. Morale

Description:

Morale is often mentioned as a ripple effect to fatigue or pressure. (see e.g. [9][10][12]) It is described as a sense of hopelessness to difficult situations lowering productivity and or quality even further. It can develop through e.g. long-term overtime, schedule pressure, or layoffs.

Structure:



Equations:

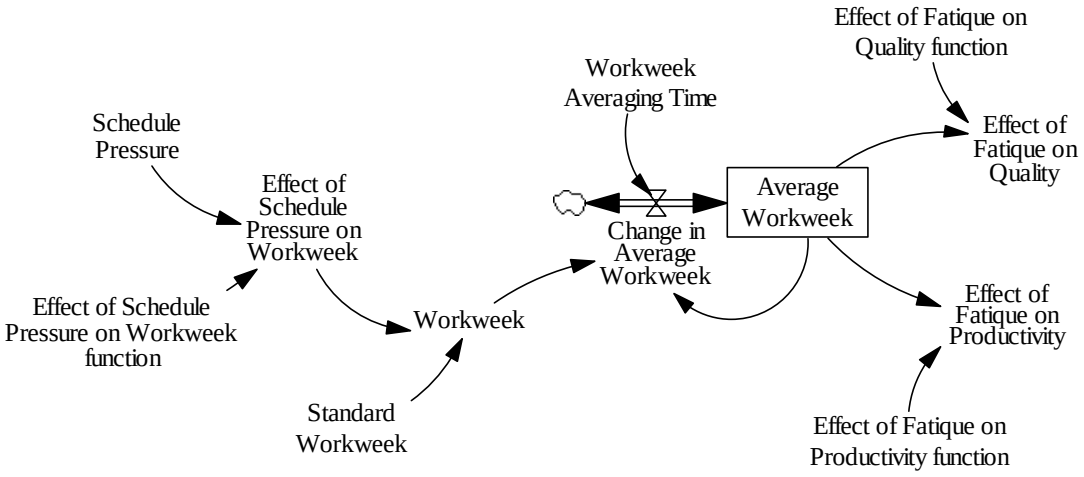
Change in Morale=(Morale-Indicated Morale)/Time to Change Morale Units: dimensionless/Month Indicated Morale Units: dimensionless Morale= INTEG (Change in Morale, Initial Morale) Units: dimensionless Time to Change Morale Units: months

1.5. Overtime

Description:

The overtime structure shows a common way to model the changes in working hours due to heavy workloads. It is one way for managers to try to get back on track in tasks or projects that are behind or to cope with increased demand. In the short term if increases productivity through more hours worked in a week but it often has ripple effects in the long term. The long term effects have been modeled with e.g. fatigue that lowers productivity and quality, and burnout creating more employee turnover. (see e.g. [1][2][4][13])

Structure:



Equations:

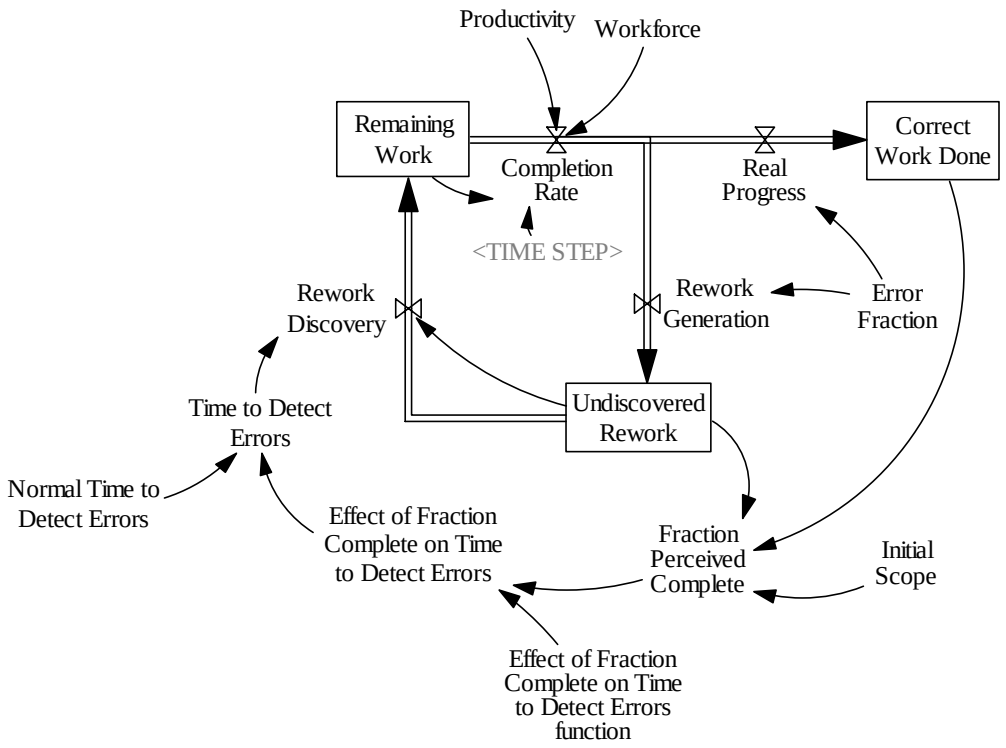
Average Workweek= INTEG (Change in Average Workweek, Standard Workweek)
Units: Hours/Week
Change in Average Workweek=(Workweek-Average Workweek)/Workweek Averaging Time
Units: Hours/Week/Month
Effect of Fatique on Productivity=Effect of Fatique on Productivity function(Average Workweek)
Units: Dimensionless
Effect of Fatique on Productivity function= *user defined function*
Units: Dimensionless
Effect of Fatique on Quality=Effect of Fatique on Quality function(Average Workweek)
Units: Dimensionless
Effect of Fatique on Quality function= *user defined function*
Units: Dimensionless
Effect of Schedule Pressure on Workweek= Effect of Schedule Pressure on Workweek function(Schedule Pressure)
Units: Dimensionless
Effect of Schedule Pressure on Workweek function= *user defined function*
Units: Dimensionless
Schedule Pressure
Units: Dimensionless
Standard Workweek
Units: Hours/Week
Workweek=Effect of Schedule Pressure on Workweek*Standard Workweek
Units: Hours/Week
Workweek Averaging Time
Units: Month

1.6. Rework Cycle

Description:

The rework cycle is one of the most utilised and core structures of project management models [9]. It describes how the perceived completion of a project is based on faulty information. That is, the work that is seen completed contains work that still has to be redone (i.e. *undiscovered rework*). Now the troublesome part is that most management decision including transferring people on and off the project is based on the faulty information about the fraction of work that actually is done correctly.(see e.g. [14][16][17]) Common mechanism that affect the discovery of error and rework come through the completion of subsequent work (when having to build on top of faulty work the errors become evident) and or by quality assurance work.

Structure:



Equations:

Completion Rate=MIN(Productivity*Workforce, Remaining Work/TIME STEP)
Units: tasks/Month
Correct Work Done= INTEG (Real Progress, 0)
Units: tasks
Effect of Fraction Complete on Time to Detect Errors=Effect of Fraction Complete on Time to Detect Errors function(Fraction Perceived Complete)
Units: dimensionless
Effect of Fraction Complete on Time to Detect Errors function= *user defined function*
Units: dimensionless
Error Fraction
Units: dimensionless
Fraction Perceived Complete=(Correct Work Done+Undiscovered Rework)/Initial Scope
Units: dimensionless
Initial Scope
Units: tasks
Normal Time to Detect Errors
Units: months
Productivity
Units: tasks/Month/person
Real Progress=(1-Error Fraction)*Completion Rate
Units: tasks/Month
Remaining Work= INTEG (Rework Discovery-Completion Rate, Initial Scope)
Units: tasks
Rework Discovery=Undiscovered Rework/Time to Detect Errors
Units: tasks/Month
Rework Generation=Error Fraction*Completion Rate
Units: tasks/Month
Time to Detect Errors=Effect of Fraction Complete on Time to Detect Errors*Normal Time to Detect Errors
Units: months

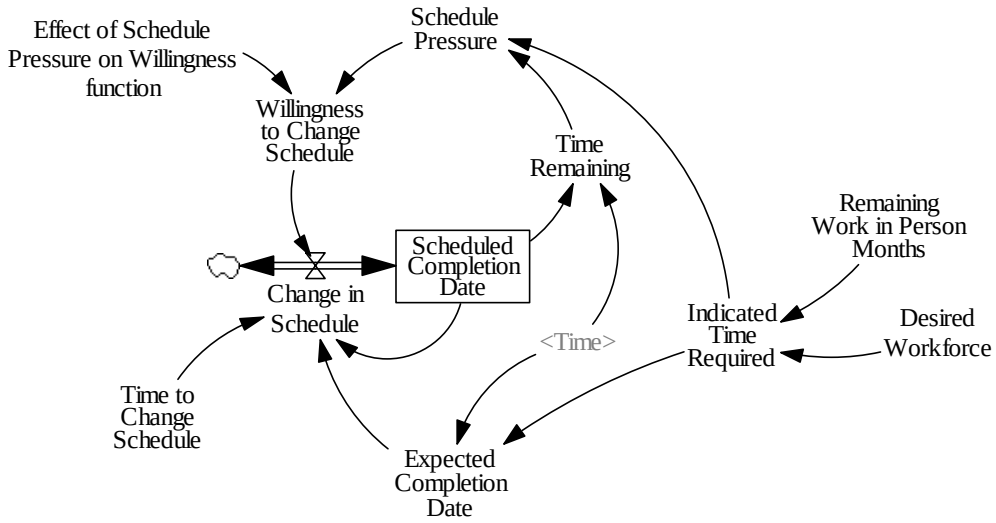
Undiscovered Rework= INTEG (Rework Generation-Rework Discovery, 0)
Units: tasks
Workforce
Units: people

1.7. Schedule

Description:

The schedule structure shows how much time there is to complete a work or project that has a deadline. It also shows how the schedule can change. Often especially in project that go wrong the schedule starts to slip but it can also be used as a management policy to alleviate pressure when hiring more or working overtime are not as desirable. (see e.g. [1][3][14])

Structure:



Equations:

Change in Schedule=Willingness to Change Schedule*(Expected Completion Date-Scheduled Completion Date)/Time to Change Schedule
Units: Months/Month
Desired Workforce
Units: People
Effect of Schedule Pressure on Willingness function= *user defined function*
Units: Dimensionless
Expected Completion Date=Indicated Time Required+Time
Units: Months
Indicated Time Required= Remaining Work in Person Months/Desired Workforce
Units: Months
Remaining Work in Person Months
Units: Person*Month
Schedule Pressure=Indicated Time Required/Time Remaining
Units: Dimensionless
Scheduled Completion Date= INTEG (Change in Schedule, Expected Completion Date)
Units: Months
Time Remaining=Scheduled Completion Date-Time
Units: Months

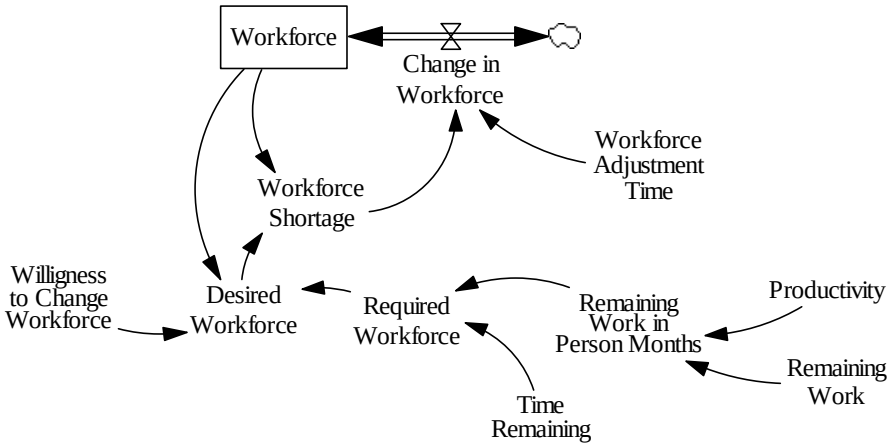
Time to Change Schedule
Units: Months
Willingness to Change Schedule=Effect of Schedule Pressure on Willingness f(Schedule Pressure)
Units: Dimensionless

1.8. Workforce

Description:

The workforce structure shows a simplistic way to model the changes in headcount. One relevant factor often is the willingness to change the workforce. In operations management it might be the budget, and in projects it might be the perceived completion fraction that makes ones willingness to change the headcount (see e.g. [1][3][13][14][15])

Structure:



Equations:

Change in Workforce=Workforce Shortage/Workforce Adjustment Time
Units: People/Month
Desired Workforce=Willigness to Change Workforce*Workforce+
(1-Willigness to Change Workforce)*Required Workforce
Units: People
Productivity
Units: Tasks/Person/Month
Remaining Work
Units: Tasks
Remaining Work in Person Months=Remaining Work/Productivity
Units: Person*Month
Required Workforce=Remaining Work in Person Months/Time Remaining
Units: People
Time Remaining
Units: Month
Willigness to Change Workforce
Units: dimensionless

Workforce= INTEG (Change in Workforce, Required Workforce)
Units: People
Workforce Adjustment Time

Units: Month
Workforce Shortage=Desired Workforce-Workforce
Units: People

2. References

[1] Ford, D. (1995). The Dynamics of Project Management: An Investigation of the Impacts of Project Process and Coordination on Performance. MIT Thesis

[2] MacInnis, D. (2004) Development of a System Dynamics Based Management Flight Simulator for New Product Development, MIT Thesis

[3] Hines, J. (2005) Molecules of Structure: Building Block for System Dynamics Models. Version 2.02.

[4] Oliva, R. and Sterman, J. (2010). Death spirals and virtuous cycles. In P.P. Maglio, C.A. Kieliszewski and J.C. Spohrer (eds), Handbook of Service Science.

[5] Oliva, R. (1996) A Dynamic Theory of Service Delivery: Implications for Managing Service Quality. MIT Thesis.

[6] Trost, C. (2002), A dynamic model of work quality in a government oversight organization. System Dynamics Review, 18: 473–495.

[7] Homer, J. (1999), Macro- and micro-modeling of field service dynamics. System Dynamics Review. 15: 139–162.

[8] Ford, D.. and Sterman, J. (1998), Dynamic modeling of product development processes. System Dynamics Review. 14: 31–68

[9] Lyneis, J. and Ford, D. (2007) System dynamics applied to project management: A survey, assessment, and directions for future research. System Dynamics Review, 23: 157–189.

[10] Thomson, D. (2002) A System Dynamics Based Analysis of "Rank and Fire" Management Policy. MIT Thesis

[11] Sterman, J. (2000) Business Dynamics: System Thinking and Modeling for a Complex World,McGraw-Hill.

[12] Morecroft, J. (2007) Strategic Modelling and Business Dynamics. A Feedback Systems Approach. John Wiley & Sons.

[13] Nasirzadeh, F. and Nojedehi, P. (2013). "Dynamic Modeling of labor productivity in construction projects", International Journal of project management, 31.

[14] Richardson, G. (2014), "Model" teaching III: Examples for the later stages. System Dynamics Review. 30: 291–299.

[15] McKenna N. (2005) Executing Major Projects through Contractors, MIT Thesis.

[16] Roberts, E. (1974) A simple model of R & D project dynamics. R&D Management, 5: 1–15

[17] Vensim Modeling Guide – www.vensim.com