Project management through the lens of the system dynamics approach

Eduardo Ferreira Franco (eduardo.franco@alumni.usp.br)

Stefano Armenia (SYDIC – System Dynamics Italian Chapter – armenias@mac.com)

1 Introduction

Managing complexity is becoming an important issue in project management literature because it has become an inseparable aspect of projects and one of the critical factors when a project fails. Besides, the dynamic complexity of projects appears to be increasing, and the PMI (2013) acknowledged that properly managing it requires the application of critical thinking approaches. System dynamics is an approach particularly suited for modeling and analyzing dynamic complexity, and Sterman (2000) described it as:

> *[A] perspective and set of conceptual tools that enable us to understand the structure and dynamics of complex systems. System dynamics is also a rigorous modeling method that enables us to build formal computer simulations of complex systems and use them to design more effective policies and organizations.*

When talking about complex systems, we need first to understand what characterizes them. We need to identify common properties that could describe these complex systems. Several authors have studied and proposed a set of intriguing properties that are shared by most of them. To give a definition of a complex system, we quote one provided by Mitchell (2011):

> *[A] system in which large networks of components with no central control and simple rules of operation give rise to complex collective behavior, sophisticated information processing, and adaptation via learning or evolution.*

Adaptation plays an essential aspect in complex systems, and it is not different in complex systems where project management is embedded. Through learning or evolutionary processes, complex systems must adapt to change their current behaviors to improve their chance of survival (or success).

The discussion of project success is broad and beyond our scope. We will adopt the iron triangle model for simplicity, consisting of time, quality, and cost dimensions (Atkinson 1999). It is still widely used as the evaluation criterion for project success. For a project to succeed, it must conform to planned goals related to the iron triangle's dimensions.

In the following sections, we will demonstrate how system dynamics can be used in the project management field to understand better the intrinsic complexities that prevail in this domain. For this, we briefly present a basic introduction to the system dynamic approach and then give a practical example by showing how to develop a simple but insightful project management simulation model that can demonstrate the emergence of insightful behavioral patterns.

2 System dynamics

What is the system dynamics approach? System dynamics was developed by Forrester (1961) in the '50s to study business complex problems related to industrial processes, and it is founded on the scientific method. The System Dynamics Society's website[1] describes it as follow:

> *System dynamics is a computer-aided approach for strategy and policy design. It uses simulation modeling based on feedback systems theory and is an analytical approach that complements systems thinking. It applies to dynamic problems arising in complex social, managerial, economic, or ecological systems — literally any dynamic systems characterized by interdependence, mutual interaction, information feedback, and circular causality.*

It can be described as an iterative approach composed of five steps: 1) properly articulate a problem, 2) define a dynamic hypothesis that accounts for a developing theory that explains the problem under investigation, 3) develop a complete simulation model with equations, parameters, and initial conditions, 4) test the model to build confidence in the proposed formulation, and then 5) design and evaluate candidate policies.

The behavior of a complex system arises from its structure, i.e., the interplay of positive and negative feedbacks generates several dynamic behavioral patterns. Most dynamics are instances of a small number of basic patterns. Among these patterns' building blocks, there is the exponential growth, which is produced by positive feedback; the goal-seeking, created by negative feedback; oscillations, which are caused by negative feedback with time delays; and

---

[1] https://systemdynamics.org/what-is-system-dynamics/

other more elaborated patterns such as S-shaped growth, overshoot, and collapse, which arise from nonlinear interactions of these previous basic patterns.

The modeling of feedback structures is an essential activity, and in doing so, promotes a better understanding and thus, improves the decision-making process awareness. Usually, the most complex behaviors arise from the interaction of two basic feedback loops: balancing (B) and reinforcing (R), shown in Figure 1.

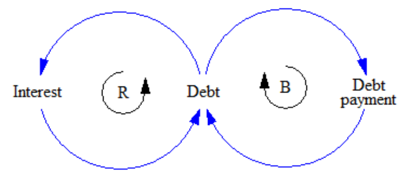

Figure 1. Feedback loop simple structure.

Behind the graphical representations of these two types of polarities, there is a simple mathematical formulation. Equations 1 and 2 show the mathematics associated with positive and negative link polarity, respectively, in which the independent variable is denoted by "*X*" and the dependent by "*Y*."

$$\frac{\partial Y}{\partial X} > 0 \ or \ Y = \int_{t_0}^{t} (X + \cdots) ds + Y_{t_0} \qquad (1)$$

$$\frac{\partial Y}{\partial X} < 0 \ or \ Y = \int_{t_0}^{t} (-x + \cdots) ds + Y_{t_0} \qquad (2)$$

Stocks and flows are, together with feedbacks, central concepts in the system dynamics approach. Stocks represent accumulations that describe the state of the system. They provide inertia, memory, and delays that arise as the difference between inflows and outflows adds to the stocks. They also create dynamic disequilibrium in a system. Conversely, flows are the mechanisms responsible for altering the state of a system. Their inflows and outflows, respectively, increase or decrease the quantity accumulated in stocks.

Figure 2 shows how these two basics elements are used to formulate a simple system dynamics model. In short, *stocks* represent state variables, *flows* denote the stocks' rate of change, *variables* can be used for increasing the model's overall understandability and computations, and *links*, the blue arrows, connect several of the model's elements.

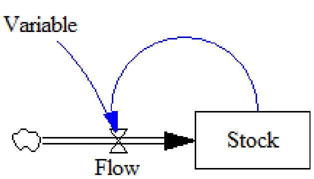

Figure 2. System dynamics' basic modeling elements

The mathematical structure of a system dynamics simulation model corresponds to a set of coupled, nonlinear, first-order differential (or integral) equations. This structure can be described by a vector of levels (*x*), a set of parameters (*p*), and a nonlinear vector function (*f*), according to Equation 3.

$$\dot{x}(t) = f(x, p) \qquad (3)$$

Most system dynamics modeling tools allow to visually describe the formulation without manually specifying and calculating most of the equations associated with a simulation model. These modeling tools themselves make these calculations, and it is up to the modeler to define the rate equations and auxiliary variables.

3 A practical example

A straightforward simulation model is developed in the following subsections to demonstrate how the system dynamics approach can be used to explore and investigate dynamic behavioral patterns that could emerge in a hypothetical project management context. We describe the example incrementally, where each step builds up with the results from the previous step to add more details to the formulation.

We used the Vensim[2] modeling software to develop these examples, which has a free version for educational purposes (i.e., Personal Learning Edition – PLE) suitable for practicing the following steps.

3.1 Task accomplishment

Usually, a project consists of a set of tasks that must be accomplished to achieve the desired goal within predefined constraints related to time, quality, and cost (i.e., the iron triangle evaluation model).

The first example depicts the project's tasks accomplishment mechanism, which is shown in Figure 1. It consists of two stocks (one representing the "*Work to do*" and the other the "*Work done*") and a rate between them (i.e., "*Work rate*"). This simple model shows how tasks in "*Work to do*" stock are performed according to the "*Work rate*" and then added to the "*Work done*" stock as they get completed.

The flow of tasks between the two stocks halts when the "*Work done*" reaches the project's scope (i.e., the "*Project is done*" condition), which is defined by the auxiliary variable "*Initial project definition*". This dynamic is also graphically depicted on the right side of Figure 1. The "*Work to do*" stock begins with 1,000 tasks and then decreases according to a "*Work rate*" of 100 tasks per month (red line). After ten months, all the 1,000 tasks moved from the "*Work to do*" to the "*Work done*" stock (blue line), and the project is completed.

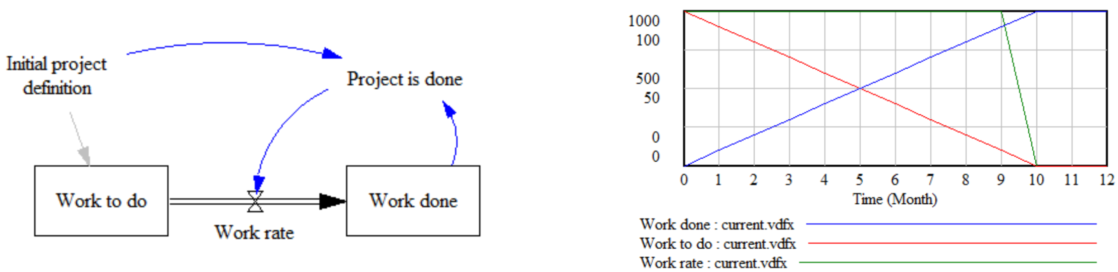

Figure 3. Task accomplishment model and outputs.

From the definitions given above, we can see that the two stocks (i.e., "*Work to do*" and "*Work done*") correspond to the integrative of the associated flow (i.e., "*Work rate*"), and both take into consideration their initial state. The "*Work to do*" stock has a negative relationship with the "*Work rate*" flow, which can be read as "Work rate subtracts to Work to do." Yet, "*Work done*" stock has a positive relationship with the "*Work rate*" flow, which can be read as "Work rate adds to Work to do."

This first example described how a set of tasks is accomplished within a period of time (i.e., ten months), corresponding to one of the three iron triangle model's dimensions previously discussed, i.e., **time**.

3.2 Errors and rework

Although straightforward and intuitive, the model described in the previous section illustrated some relevant concepts that can be further extended to develop a more realistic formulation. One of its drawbacks, or simplifications, is that it assumed that tasks are performed without errors, thus not implying the burden of extra rework due to quality issues and completing the project in the ideal planned schedule.

Figure 4 shows a slightly modified version of the model previously shown in Figure 3. In Figure 4, we can quickly identify the presence of a third stock, which is denoted "*Quality assurance*." After completing the tasks, they go through the quality assurance process to be inspected, thus guaranteeing that they were performed according to predefined quality specifications. If tasks are completed appropriately, they move on to the "*Work done*" stock and advance the project progress. Otherwise, they are sent back to the "*Work to do*" stock to be reworked.

Besides the newly introduced stock, two rates and two auxiliary variables were added to the model formulation. The two rates account for the "*Approved work rate*" and "*Discover rework rate*," representing the completed work and the low work quality sent for rework. These two rates are computed taking into consideration the "*Quality assurance*" stock level, the "*QA duration*," and the "*Work quality*."

"*Discover rework rate*" and "*Approve work rate*" are fractions of the "*Quality assurance*" available tasks, and they are proportional to the "*Work quality*" definition. Another interesting observation from the plot shown on the left side of Figure 2 is that the project took longer to be completed (i.e., approximately twenty months) when compared to the first model (i.e., ten months). Why did this happen? Due to quality issues and the burden of rework.

[2] https://vensim.com/

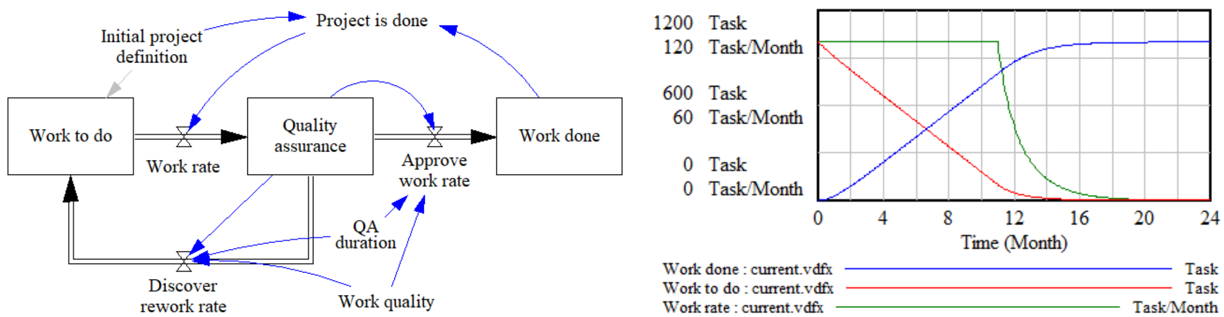

Figure 4. Quality assurance and rework model and outputs.

This second example enabled us to explore another dimension of the iron triangle model, i.e., **<u>quality</u>**. More interesting was to investigate how quality issues could compromise the necessary time to complete the hypothetical project.

3.3 *Schedule pressure & workforce*

In the first example, we had an ideal scenario with no rework activity where the project was completed in the planned schedule. Next, in the second example, we saw a more realistic formulation where we considered quality issues that caused rework activities and the project being completed almost twice the time it was initially expected (twenty vs. ten months).

Now, in this third and last example, we will extend the examples shown in previous sections. We will consider that decision-makers take corrective measures when they perceive the project is falling behind the planned schedule. The action taken is to bring more people to the project and thus increase the "*Work rate*" to speed up the project progress.

This new modified version is shown in Figure 5. The first change was to decouple the "*Work rate*" equation. Instead of a fixed value, it is now computed as the product of "*Productivity*" and the project's "*Workforce*." Then, a couple of intermediate steps are evaluated to assess the allocated workforce's shortage or excess: 1) calculate the "*Required workforce*" to complete the remaining "*Work to do*" within the "*Scheduled time remaining*"; 2) based on the difference between the "*Required workforce*" and the "*Workforce*" on the project, and taking into consideration the "*Time to adjust workforce*," the "*Workforce adjust rate*" moves human resources in and out of the project.

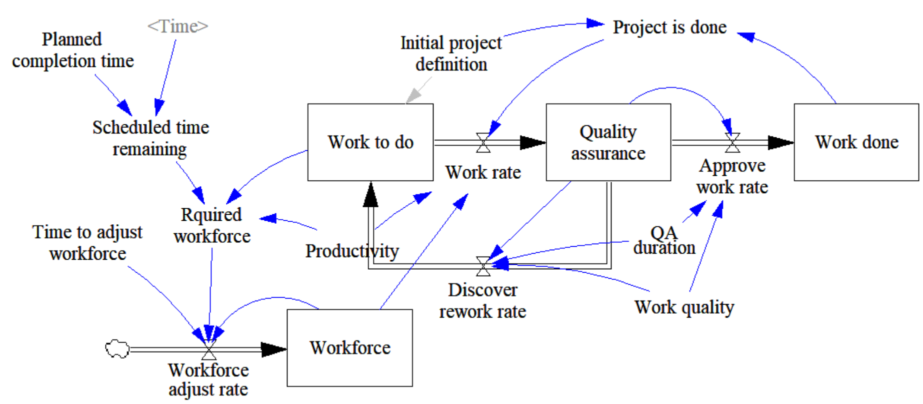

Figure 5. Schedule pressure & workforce model.

After these modifications, what are the expected changes to the dynamic behaviors previously observed? The same variables shown in the previous examples are plotted again in Figure 6 to analyze the newly introduced changes. As expected, the "*Work rate*" now varies over time as "*Workforce*" is perceived to be missing or exceeding when comparing the available and the necessary workforce. The mobilized workforce peaked near month nine and declined as the remaining "*Work to do*" diminished below the "*Work rate*" throughput.

Still, from Figure 6, we note that the project ended earlier than in the second example. The project now finished near month #15, and, in the previous example, the project finished near month #20. This new strategy brought a nearly 25% reduction in the project duration, which could be a good result *per se*. However, it still takes 50% longer than the first example that the project finished in ten months.

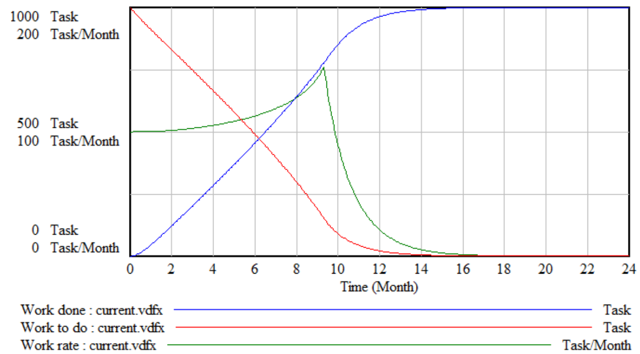

Figure 6. Schedule pressure & workforce model's outputs.

The reason for this is that neither the two existing delays present in the third example (i.e., "*QA duration*" and "*Time to adjust workforce*") nor the "*Work quality*" factor, which caused rework to occur," for simplicity, were taken into consideration when calculating the necessary adjustments to the workflow throughput. Thus, the "*Workforce adjustment rate*" was lower than required, and the project finished later than initially planned. This phenomenon also happens in real work, where project managers often mistakenly underestimate (or neglect) the effect of delays and quality issues.

Figure 7 below shows the causal loop diagram that accounts for the behavior described. Two main loops are driving the behaviors seen. A balancing feedback loop ("*B*"), on the right side, represents the project manager's decision to bring more resources to the project when it follows behind schedule. Besides, there is a reinforcing feedback loop ("*R*"), on the right, that accounts for the unintentional consequence of adding more people to the project, there is more rework to be done, as the "*Work rate*" increases, the "*Work to do*" will also increase, and thus, it will make the project to finish later than it was initially expected.

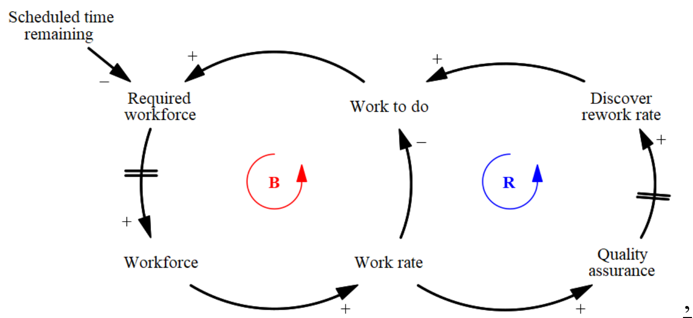

Figure 7. Causal loop diagram.

Another important consideration is that the schedule contraction obtained in this example, when compared to the second example, did not come for free. More human resources had to be added to the project to finish it earlier, and by doing so, increased the total effort employed, which, in this case, is directly associated with the incurred cost of the project. Here comes the last dimension of the iron triangle model, the **cost**.

4 Conclusion

As widely known, "all models are wrong, but some are useful" (Box and Draper 1987). This is also true for the straightforward examples described and discussed in the present text. These simulation models may not fit all contexts, and there may be several simplifications, missing elements, and concepts.

However, they helped introduce the system dynamics approach and demonstrate how it could be used to explore complex phenomena that emerged within project management scenarios. Besides, they revealed how basic structures could endogenously account for counterintuitive observations, how different evaluation dimensions (e.g., time, quality, and cost) are intrinsically intertwined and interrelated.

The iron triangle model corresponds to a multicriteria tradeoff decision that project managers constantly face during project execution. In any sight of a slip to one of these dimensions, the man in charge has to adjust the other levers to get the project back on the planned track. Yet, by playing with these interrelated properties, and as in any complex system, unintended and unexpected outcomes emerge (e.g., a project being late, cost overruns, low quality of the final work, and so on).

By analyzing project and project management through a holistic and systemic lens, as in the system dynamics approach, stakeholders can better understand these challenging issues and make better decisions by using simulation environments to evaluate potential future scenarios.

The examples discussed in this text introduced how the system dynamics approach can be used within the project management context to uncover underlying causal relations that account for expected behaviors that can be observed during project execution. However, these examples only scratched the underlying complexities and possibilities to the broad possible applications. Several other published works could be explored to investigate further and deepen the underlying understanding of the subject (Lyneis, Cooper, and Els 2001; Ford and Sterman 1998; Taylor and Ford 2006; Rahmandad and Hu 2010; Abdel-Hamid and Madnick 1991).

Figure 8 presents some of the reflections Lyneis et al. (2001) discussed regarding the existing dynamics found within projects' work & rework cycle. There are two key elements depicted in Figure 8, which are:

1) The rework cycle structure is shown in the center, composed of four stocks and four rates. This structure resembles the one developed in the examples previously discussed, where a fraction of the work to do flows back to the work backlog due to quality issues. Besides, in this new formulation, the rework also originated from change requests incurred after the task is considered completed.
2) There are several intertwined feedback structures that, due to the project's apparent progress assessment, come into play with different intensities and polarities (balancing or reinforcing). The positives and negatives feedback structures are distinguished by colors (i.e., green and red, respectively). The example developed in this text showed one possible formulation of the "***Hiring***" loop shown below, but other more sophisticated implementations can be better suited for different scenarios. Figure 8 also shows other counter-measures that managers can consider when a project falls behind schedule: force the Workforce to work ***Overtime*** and pressure them to increase their productivity (***Schedule Pressure***). However, these decisions have unintended consequences that may arise some time after their implementation and that are shown in red: workforce ***fatigue, stress***, and ***low morale***; lower ***quality level***; and ***work out of sequence, workplace congestion, and coordination problems***.

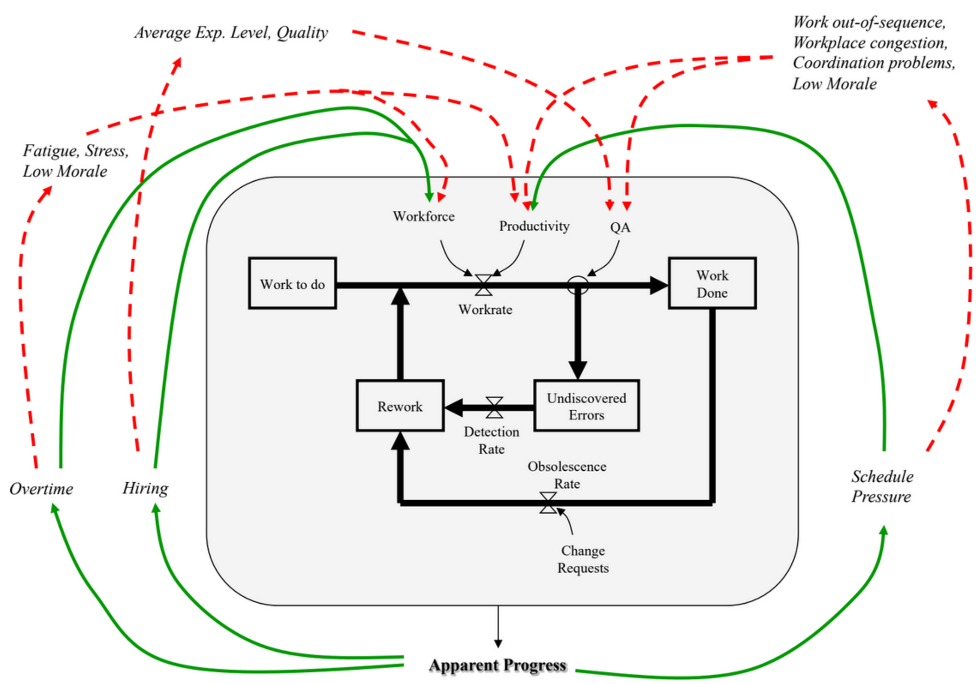

Figure 8. Work & rework cycle in projects - adapted from (Lyneis, Cooper, and Els 2001).

The extended reflections presented in Figure 8 are useful to demonstrate the numerous extensions and possibilities that can be brought to attention when formulating and developing a system dynamics simulation model. The decision on what to include or not will always rely on the problem under investigation and the selected model's boundaries that are sufficient for this matter.

References

Abdel-Hamid, T., and S. Madnick. 1991. *Software Project Dynamics: An Integrated Approach*. First edit. Prentice Hall.

Atkinson, Roger. 1999. "Project Management: Cost, Time and Quality, Two Best Guesses and a Phenomenon, Its Time to Accept Other Success Criteria." *International Journal of Project Management* 17 (6): 337–342. doi:10.1016/S0263-7863(98)00069-6.

Box, George E. P., and Norman R. Draper. 1987. *Empirical Model-Building and Response Surfaces*. 1st ed. John Wiley & Sons.

Ford, David N., and John D. Sterman. 1998. "Dynamic Modeling of Product Development Processes." *System Dynamics Review* 14 (1): 31–68. doi:10.1002/(SICI)1099-1727(199821)14:1<31::AID-SDR141>3.0.CO;2-5.

Forrester, Jay. 1961. *Industrial Dynamics*. 1st ed. Cambridge, MA: The MIT Press.

Lyneis, James M., Kenneth G. Cooper, and Sharon A. Els. 2001. "Strategic Management of Complex Projects: A Case Study Using System Dynamics." *System Dynamics Review* 17 (3): 237–260. doi:10.1002/sdr.213.

Mitchell, Melanie. 2011. *Complexity: A Guided Tour*. 1st ed. Oxford University Press.

PMI, Project Management Institute. 2013. *PMI's Pulse of the Profession In-Depth Report: Navigating Complexity*. Newtown Square, PA.

Rahmandad, Hazhir, and Kun Hu. 2010. "Modeling the Rework Cycle: Capturing Multiple Defects per Task." *System Dynamics Review* 26 (4): 291–315. doi:10.1002/sdr.435.

Sterman, John. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. US: McGraw-Hill/Irwin.

Taylor, T., and David Ford. 2006. "Tipping Point Failure and Robustness in Single Development Projects." *System Dynamics Review* 22 (1): 51–71. doi:10.1002/sdr.330.