IS Alignment improved with co-evolutionary principles: 
An Open Source approach

Belbaly Nassim 
GSCM-Montpellier Business School 
n.belbaly@supco-montpellier.fr

Frank Robert 
GSCM-Montpellier Business School 
f.robert@supco-montpellier.fr

Abstract
Despite extensive research, results on the sources, components and mechanisms to create or sustain information systems (IS) alignment are still lacking. The quest for an IS alignment remains a critical unsolved problem. In an attempt to address this situation, we present and test a comprehensive theoretical model in an OSS context. We explain how IS alignment at the individual level between the Open source system (OSS) project performance and open source developer's activity level is improved. We integrate the co-evolutionary theory (adaptive tension, change rate and modular design) to understand how IS alignment is improved. This study is based on data gathered from over 750 open source projects developing enterprise applications on sourceforge.net over one year. Results support all the hypotheses and conclude that using co-evolutionary theory help better IS alignment in an OSS context.

1. Introduction
For many years, researchers and business executives have drawn attention to the importance of IS alignment ([28], [29]). Studies trying to establish alignment at the strategic level (alignment of IS strategy with business strategy), operational level (alignment of organizational structure and IS structure) and individual level (alignment of developers behaviors and users needs) ([11]; [13]; [2] have been suggested to impact organizational outcomes, performance, and competitive advantage ([23]; [48]; [42]; [17]; [4]; [20]; [6]). Approaches to align IS and corporate objectives, however, fall short as they rely mainly on traditional planning and focus on strategic and operational level. These approaches are based on simple cause-effect deterministic logic that favors short term and static balance over sustained alignment ([52]; [2]). This static balance often considers alignment as an end-state process, with short-term performance implications ([43], [19]). And presumes that IS alignment should be identical in different organizational contexts or that the same design approach should be useful in all situations ([4], [43], [19], [5]).

Consequently, current approaches provide little guidance towards studying alignment between ongoing IS and organizational systems in different or changing contexts. This logic ignores the co-evolutionary nature of IS alignment and the complex mutual influences between IS and various organizational domains that maybe operating simultaneously over time [2]. We argue that the co-evolutionary and emergent nature of alignment has rarely been taken into consideration in IS research and that this is the reason behind why IS alignment is so difficult. Taking a co-evolutionary stance allows us to frame the process of mutual adaptation not just as a matter of static alignment, but as a dynamic interplay of coevolving interactions, relationships and effects in complex turbulent environments. Using the definition of Benbya & Mckelvey [2], we view IS alignment in an OSS context as a continuous co-evolutionary dynamic process that reconciles top-down ‘rational designs’ and bottom-up ‘emergent processes’. This definition needs to integrate the individual level component to complete the understanding of IS alignment dynamics. For that reason, we have concentrated our efforts on developers and considered OSS projects as an unit of analysis.

Our study distinguishes itself from previous ones on IS alignment by: (1) studying IS alignment at the individual level in an OSS context; (2) proposing a dynamic interaction between OSS project performance and developer’s activity level to better understand alignment; (3) using co-evolutionary theory. We examine the following broad research question: How OSS project performance and developer’s activity level co-evolves to better shape IS alignment? To carry this out, we test a model where we: (1) Look for ways to speed up the co-evolution of OSS project performance and developer’s activity level (Total Bugs, closed Bugs, Bugs life, Number of participants, Number of
Developers); (2) Use the co-evolution based on three “1st Principles” of evolutionary efficacy (adaptive tension, change rate and modular design). Our empirical testing uses data from 750 OSS projects over one year, and leads to a model for analyzing the nature and role of IS alignment in an OSS context.

This paper is organized as follows. We first define IS alignment concept and the potential contribution of co-evolutionary perspective. We then introduce our research model where we examine the influence of co-evolutionary principles on IS alignment that occurs between the OSS project performance and developer’s activity level. Finally, the paper’s findings, limitations and implications are discussed.

2. IS Alignment

Prior conceptualizations of IS alignment have each focused selectively on one dimension of alignment at the expense of others, with the result that the current state of knowledge about IS alignment in organizations is ambiguous and conflicting. Early approaches devised top-down strategic planning models based on the idea that an IS strategy can be planned in close association with business strategy. The driver behind the process of IS strategy formulation in this view is to evaluate the impact of IS on business strategic options, while also aligning IS and business strategies [11]. Yet, researchers recognized early that alignment of business and IS strategies also requires structural alignment between IS and organizational infrastructures [13]. Structural alignment stresses the importance of structural fit between IS components and other organizational domains, specifically in the areas of IS decision-making rights, reporting relationships, provision of IS services and infrastructure, and the deployment of IS personnel. Acknowledging that alignment is not an event but a dynamic process, Henderson and Venkatraman ([17]; [18]) propose a model representing the dynamic alignment between corporate and IS strategic context, stressing the necessity for both strategic and structural alignment.

Their Strategic Alignment Model brings strategy and IS together by balancing internal and external factors in the business and IS domains. The focus of research within the most recent school of thought is on the integration and fusion of IS and business activity domains; attention has turned out to the mechanisms and enablers of alignment in a context that is increasingly viewed as dynamic [37]. Despite this evolution in perspective, IS alignment research has been criticized in terms of three recurring themes characterizing studies of IS alignment [2]:

- The predominance of a prescribed strategy process as a pre-condition of IS alignment;
- An over emphasis on reaching static balance among the different components of alignment;
- A confusion in the literature over the terms used to refer to alignment.

The concept of alignment goes forward with little agreement on a consistent definition. Indeed, definitions of alignment range from high-level, broadly encompassing definitions such as “the fit between an organization and its strategy, structure, processes, technology and environment” [21] to more focused definitions such as “convergent intentions, shared understanding, and coordinated procedures” [46]. Two well-received views are that IS alignment is “the degree to which the IS mission, objectives, and plans comply with organizational objectives and plans” [42] and that IS alignment involves “fit” and “integration” among business strategy, IS strategy, business infrastructure, and IS infrastructure [18]. In other words, misalignment arises when an organization’s IS capability fails to provide adequate support for the organizational units attempting to combine their efforts in support of its business strategy. This inconsistency and confusion is also reflected by the usage of different terms and alternative concepts to refer to the phenomenon of alignment as demonstrated by the following terminology: coordination [26], balance [18], linkage [41], fit [53], harmony [57], integration [55], and strategic fit [24], among others.

This review of literature reveals the need to develop a trend of research on:

- IS alignment at the individual level, as the majority of studies have dealt with strategic and operational level. In this study, we explicitly focus on developer’s activity and consider OSS projects as a unit of analysis.
- IS alignment from a co-evolution perspective, in this study, we consider the process of mutual adaptation as a dynamic interplay of coevolving interactions, relationships and effects in complex turbulent environments.

The potential contribution of coevolutionary perspective

The term, co-evolution was coined by Ehrlich and Raven [12] in their study of butterfly/plant symbiosis. They describe how species of butterflies and their host plants evolve simultaneously, each adapting to the other. Kauffman [22] observes that all “evolution” is really co-evolution. He argues that co-evolution is at the root of self-organizing behavior, constant change in systems, the production of novel
macro structures, and nonlinearities. Co-evolution theory applies to organizations ([31];[27]). McKelvey [32] considers co-evolution and competitive behavior of firms, defining co-evolution as “mutual causal changes between a firm and competitors, or other elements of its niche, that may have adaptive significance”. He stresses that co-evolution is a multi-level phenomenon and that it is necessary to “take a more emergent natural systems perspective and pick parts naturally emerging as evolutionarily significant (those most likely to change which offer selective advantage for the firm as a whole)”.

Mitleton-Kelly [34] uses co-evolutionary theory to study the relationship between business and IS domains to gain insight into the problems of legacy systems. She adopts a multi-level analysis, looking at the interaction: among individuals, between individuals and IS, between Business and IS domains, and between the organization and its environment. Peppard and Breu [37] apply co-evolutionary theory to Business/IS strategic alignment.

Our review of suggested alternatives to rational, top-down, static IS designs shows IS scholars calling for a more dynamic, interactive, mutual adaptation approach (e.g., [7], [36], [52]), in short, co-evolution. As our focus is at the individual level, our co-evolutionary perspective allows us to frame the process of mutual adaptation and change between user’s need and developer’s behavior not just as a matter of alignment but as a dynamic interplay of co-evolving interactions, interrelationships and effects. Thus, we adopt a holistic rather than a bivariate conceptualization of alignment because of its ability to retain complex and interrelated nature of the relationship between the different components.

Our framework relies on simple rules [14] — which typically stem from stylized facts and foundational principles. We draw on three “Principles” (suggested by Benbya and Mckelvey, [2]), in our search for ways to increase the flexibility and adaptability of both user’s need and developer’s behavior and their co-evolution. They are:

1. Adaptive Tension: Environmentally imposed tensions (energy differentials) stimulate adaptive order creation— Prigogine’s dissipative structures theory [38].
2. Change Rate: Higher internal rates of change offer adaptive advantage in a changing environment— Fisher’s genetic variance theorem [15].
3. Modular Design: Nearly autonomous subunits increase complexity and rate of adaptive response— Simon’s near decomposability principle [47].

Though they originate from different disciplines, these Principles are now all seen as contributing to more efficacious adaptation under conditions of environmental complexity and turbulence. They act as jointly probable elements, any one of which gives an organism, species, organization, or industry adaptive advantage. Having none is a disaster; having all greatly fosters adaptive success. These principles are said to be “interdependent” in the sense that they should not be applied in isolation if one wants to reach valid conclusions regarding co-evolutionary adaptations. In fact, their use in isolation from one another could arguably lead to inertia. Hereafter, we discuss how the Principles at the individual level connect to the premises of our co-evolutionary framework of IS alignment in an OSS context.

3. Research Model

Our suggested theoretical model of the IS alignment occurring in the OSS context relates to mainly:

- The individual level, where developer variable is linked to the OSS projects performance. We have retained the developer’s activity level as the developer’s behavior variable because it contributes to OSS projects performance by performing activities which objective are mainly to respond to user’s needs (coding, bugs, communication..). As the unit of analysis is the OSS project, the developer’s activity level will influence the OSS projects download because the more the developers will be active and will respond better to the user’s need and the more the OSS projects will be downloaded.

- The state of alignment is reached when the influence between the developer’s activity level and OSS project performance is positive.

![Figure 1: IS alignment improved with co-evolutionary principles.](image-url)
3.1 Assessing Developer’s Activity level

In OSS development, the developer community tends to “release early and release often,” which implies that an active release cycle is a sign of a healthy development process and project [8]. The level of developer’s activity requires an involvement in the interaction, which is measured by counting the number of times they are actively engaged with the other participants or content. For example, SourceForge computes and reports a measure of project activity based on the bug reports by developers. Thus we can evaluate the developer’s activity by focusing on bug reports and on the developer’s number. In fact, detailed examination of bug-fixing might yield useful process data indicative of developer’s level activity. Bug reports and feature requests are typically managed through a task-management system that records the developer and community discussion, permits labeling of priority items, and sometimes includes informal voting mechanisms to allow the community to express its level of interest in a bug or new feature. In our research, we have focused on bugs total numbers, closed bugs (response rate), bugs life as variables influencing open source project performance. To collect data on bugs, we have extracted from 750 SourceForge OSS projects the length of time taken to fix bugs (bug life) as well as the response rate to open bugs (closed bugs) and finally total number of bugs for the project.

On the other hand, as OSS projects are dependent on volunteer developers, being able to attract developers to a project on an on-going basis to maintain the level of activity is crucial. Thus, the number of developers involved in a project could be an indicator of developer’s activity level as well as the developers who are formally associated with each project.

OSS project performance

As open source projects are not always driven by direct profit motives [25]. There is usually no pre-determined deadline, a priori budget, or a set of specifications [44] for assessing the OSS project performance. Raymond [40] defined successful OSS projects as those characterized by a continuing process of volunteer developers fixing bugs, adding features and releasing software “often and early”. In our study, we consider success from the demand side (end users) by counting the number of downloads. On the demand side, project popularity which measure the download effect is a key measure of the OSS project’s success: high popularity shows that there are many users using or are interested in using the open source software. OSS project performance is related to the capacity of developers to understand respond and adapt to the user’s needs. This understanding enables developers to develop the right OSS application that users will download and use.

On the other hand, an OSS project will cease to exist or progress if there is no demand or if no developer activity is performed to adapt this product to the user’s needs.

Principle of Adaptive Tension—Cause of Adaptive Non-response

Adaptive tension takes effect when differences in energy (usually heat) imposed on a system rise to the point where they cause phase transitions resulting in increased connectivity. These in turn cause new kinds of order—whirlpools in turbulent fluid flows, or the rolling boil in a teapot ([38], [39]; [33]). IS alignment is not an event but rather a dynamic process where the several dimensions intersect. Indeed, IS alignment is characterized by multiple and conflicting realities that surface as time goes by. These tensions exist, between the user’s needs and developer’s activity level. Thus, alignment results in not one but several development spirals, each instigated by the initial conceptions and interests of the various actors. The adaptive tension is with the number of visited pages and total of visits. Our co-evolutionary perspective allows us to frame this process of adaptive tension not just as a matter of alignment facilitating short-term success—which could lead to an inertia that in turn could lead to failure when the environment suddenly shifts [16] — but as a dynamic interplay of coevolving interactions between the OSS project performance and developer’s activity level.

Hypothesis 1: Adaptive tension will positively influence OSS project performance

Principle of Change Rate—Cause of Adaptive Failure

Fisher's [15] work made a key link between variation and adaptation that is now all but axiomatic in the biological and social sciences. His basic theorem stated: “The rate of evolution of a character at any time is proportional to its additive genetic variance at that time” (quoted in Depew and Weber [9]). In other words, adaptation can proceed no faster than the rate that usable variation—e.g., in the form of new knowledge, learning, innovation, networking, agent skills, etc.—becomes available. Evolution in the changing competitive environment to which an organization’s strategy attends, requires IS systems to be continually re-aligned so as to keep them in tune with its changing strategy; they also need to keep up
with changes in the broader information technology environment external to the organization. Therefore, it is necessary to keep user’s needs and developer’s activity level changeable enough so that they can keep pace with a firm’s evolving strategies. This is particularly true in the case of uncertain, disruptive and rapidly changing environments. In this sense, the term dynamic refers to “the capacity to renew competences so as to achieve congruence with the changing business environment” [49]. We assess the change rate of the IS alignment with the flow of information between the OSS performance and developer’s activity level.

Hypothesis 2: Change rate will positively influence OSS project performance

Principle of Modular Design—Cause of Inefficient Adaptation

Simon’s [47] classic essay on the “architecture of complexity” articulates the design principle for modular systems. He argued that complex systems that are hierarchical—but which consist of “nearly decomposable” subunits (meaning that they are mostly independent from top-down control or interdependencies with other subunits)—tend to evolve faster and toward stable, self-generating configurations, has been influential in the way modularity has been conceived. Schilling [45] suggests that modularity is a continuum describing the degree to which a system's components can be separated and recombined. Modularity is, thus, the ability to easily reconfigure (add, modify, or remove) technology components by minimizing interdependencies among modules and is considered one of the most important dimensions in IS flexibility [10]. A higher degree of modularity means consequently a greater speed in developing new, or modifying existing, applications. As mentioned by Torvalds [51] about the concept of modularity, "for without it, you cannot have people working in parallel". Modularity means that the kernel itself and plans for its future development is organized around small, manageable pieces. Advocates of OSS argue that having a large team means that OSS is by necessity ‘modular’ (made up of discrete units, each with a specific function). Modularity simplifies software design and can increase the reliability as well as flexibility of software. High modular nature of OSS allows developers to carry out development of specific parts of the system with autonomy and without any need to coordinate their efforts with other sub-projects.

Modularity allows parallel development of new components, modules and substantial improvement of the overall design of the system via module innovation and competition between similar projects (both completely new modules and variation and improvements in existing ones) [35].

This principle applies in particular to IS alignment between the OSS project performance and developer’s activity level when we focus on the development stage. The stage of development defines the level of modularity. As increased modularity helps in the designer’s understanding of the problem, it increases the chances of the designer’s contribution to the project problem. As a result, OSS projects development stage assesses the modular design that influences the interaction between the user’s needs and developer’s activity level.

Hypothesis 3: Modular design will positively influence OSS project performance

4. Research Method

Open source software (OSS) is developed by voluntary developers working in teams. In contrast to traditional software development, which follows an engineering archetype, OSS follows an evolutionary archetype for software development ([3]; [54]). In an evolutionary archetype, roles are not assigned, and the problem definition evolves over time. Software is developed under a modular structure, which allows developers to work in parallel [1]; [30]; [54]). Developers follow an adaptive process to solve technical problems.

There are an infinite number of paths to solve a problem and choosing among various alternatives involves a high degree of experimentation, requiring leeway and discretion among developers [54]. This process requires an effectual normative environment that promotes trust and cooperation among developers.

We have collected the data using 750 OSS projects. We have decided to choose one specific category of OSS projects. Our choice was enterprise applications which were exclusively hosted by the SourceForge Web site. The sub-category of enterprise application concerned: CRM, ERP, Business Intelligence, Data warehousing and Workflow projects. We have collected data from the projects that have been downloaded at least once and have related forums where we could collect data on bugs. The quantitative data has been used to test the co-evolutionary hypotheses of adaptive tension, modular design and change rate influence on IS alignment between OSS project performance and developer’s activity level. We used the PLS regression as it’s help to overcome certain constraints of the classical linear regression [56].
The PLS regression permit to perform an independent analysis of the variables that explain the IS alignment in open source projects. This method present many advantages and gives good results when data are missing, but also when there is colinearity between variables (Tenehaus 1998). The variables are presented on table 1.

<table>
<thead>
<tr>
<th>OSS Projects Characteristics</th>
<th>Variables</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSS project Performance</td>
<td>Total visits</td>
<td>Number of downloads</td>
</tr>
<tr>
<td>Adaptive Tension</td>
<td>Number of visited pages</td>
<td>Total visits</td>
</tr>
<tr>
<td>Change rate</td>
<td>Information flows</td>
<td>Modular design</td>
</tr>
<tr>
<td>Modular design</td>
<td>Development stages</td>
<td>Developer Activity level</td>
</tr>
<tr>
<td></td>
<td>Total Bugs</td>
<td>Closed Bugs</td>
</tr>
<tr>
<td></td>
<td>Bug life</td>
<td>Number of participants</td>
</tr>
<tr>
<td></td>
<td>Number of Developers</td>
<td>Development stages</td>
</tr>
</tbody>
</table>

**Table 1: Variables of IS alignment**

5. Results

5.1. Correlations Matrix

The correlation matrix shows the potential relationship between the variables (cf. Table 2). For example, there is a positive correlation \(0.43\) between the total visits, number of visited pages and information flows. On the other hand, the total visits (measuring the number of connections to the project website) is positively related to the number of participants \(0.38\) and number of developers respectively \(0.39; 0.40\). The number of visited pages has a positive effects on information flows which is highly correlated to the total number of bugs and bugs closed \(0.70; 0.732\). We note also that there is a very strong link between the total number of bugs and the bugs closed \(0.97\). The number of participant and the number of developers present also a very high correlation coefficient. Thus we can predict the presence of colinearity between these variables.

5.2. Model

The objective of our analysis is to test the existence of a causality link between the co-evolution principles, the developers’ activity level and OSS project performance in order to assess its alignment. As we have seen earlier when we have studied the correlation matrix, there are some variables that are collinear. The colinearity statistics (Tolerance and VIF) issued from the multivariate regression confirms it (Table 3).

**Table 2: Correlation Matrix**

**Tableau 3: Colinearity statistics**
Their tolerance is near zero (under 0.5) and their VIF indicates important values. In order to include all the observed variables and avoid excluding the variable that are collinear we have put together an efficient econometric model using a PLS regression: Hereafter, we will verify the validity of the PLS regression over all the independent variables \( X_i \) and the dependent variable \( Y \).

The quality of the model represent the tradeoff between the PLS regression with the collected data. The quality of the model is confirmed when \( R^2Y(\text{cum}) \) and the \( Q^2(\text{cum}) \) are \( \geq 0.5 \) [50].

In fact, 64.3% of the IS alignment variance is measured by the number of OSS projects download and is explained by our model through \( R^2Y(\text{cum}) = 0.643 \). We obtain a model with two axes that can predict 63% of the IS alignment variance \( (Q^2(\text{cum}) = 0.63) \) says that the phenomenon explaining \( Y \) is well framed by \( h \) parts \( t_1, t_2, \ldots, t_h \) if \( Q^2(\text{cum}) \geq 0.5 \).

Significance of the Axes is done by the estimation of variables weight in the building of PLS components. The results are presented in the table 4:

<table>
<thead>
<tr>
<th>Variables</th>
<th>( W^*c(1) )</th>
<th>( W^*c(2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total of visits ( (X_{1,a}) )</td>
<td>0.419</td>
<td>0.31</td>
</tr>
<tr>
<td>Number of Page visited ( (X_{1,b}) )</td>
<td>0.317</td>
<td>0.173</td>
</tr>
<tr>
<td>Information flows ( (X_2) )</td>
<td>0.364</td>
<td>-0.0784</td>
</tr>
<tr>
<td>Total Bugs ( (X_{3,a}) )</td>
<td>0.287</td>
<td>-0.386</td>
</tr>
<tr>
<td>Closed Bugs ( (X_{3,b}) )</td>
<td>0.267</td>
<td>-0.438</td>
</tr>
<tr>
<td>Bugs life ( (X_{3,c}) )</td>
<td>0.256</td>
<td>-0.173</td>
</tr>
<tr>
<td>Number of participants ( (X_4) )</td>
<td>0.334</td>
<td>-0.192</td>
</tr>
<tr>
<td>Number of Developers ( (X_{5,a}) )</td>
<td>0.304</td>
<td>-0.282</td>
</tr>
<tr>
<td>Development stages ( (X_{5,b}) )</td>
<td>0.411</td>
<td>0.618</td>
</tr>
</tbody>
</table>

Table 4: Variables Weight

The \( W^*c \) weigh show the importance of each independent variable \( X_i \) during the construction of the PLS regression, \( t_i \) result resume better the \( X_i \) and still have a good capability to predict \( Y \) (IS alignment). We note that the variables \( X_{1,a} \) et \( X_{1,b} \) which measure the adaptive tension and information flows \( (X_2) \) explain the change rate, and contribute highly to the first axe \( (W^*c(1)) \). The Number of developers and participants\( (X_4, X_{5,a}) \) influenced also to the first axe. Concerning the second axe, it’s mainly influence by the modular design, (measured by development stages \( (X_5) \)). The results are presented and show that some variables are close to \( W^*c(1) \) and \( W^*c(2) \) which allow to identify how close the dependent and independent variable are linked.

Figure 2. \( W^*c(1)/W^*c(2) \)

The quality of the data collected is assessed with the Hotelling ellipse that show that majority of OSS projects are clearly represented by \( t_1 \) et \( t_2 \) components of our PLS regression.

Figure 3: OSS projects presented with the PLS regression.

Some projects are situated out of the Hotelling ellipse but they remain a small size compared to the total number, thus their effect is not so important, and we can state that PLS regression model has a good quality. Table 5 the VIP shows that the entire variables are contributing to the two axes. The result shows that the three co-evolutionary principles influence the IS alignment. Their first hypothesis shows that the modular design through the development stages, the change rate through information flow, the adaptive tension through total visits, number of page visited explain relatively 35%, 13%, 27% and 18% of OSS performance. As we have stated earlier, the more the OSS project is performing, the more it’s aligned with the developer’s activity.
The IS alignment with the rest of an organization remains a critical and chronic unsolved problem in todays complex and turbulent world. Top-down approaches are accused of being static whereas bottom-up approaches allowing more autonomy remain vague in concept and application. To overcome this problem, we have reconciled top down rational designs and bottom up emergent processes. We have focused on the co-evolution nature at the individual level of IS alignment in an OSS context. We have proposed an IS alignment model based on co-evolutionary dynamic process between OSS projects performance and developer activity level. This model tries to respond to the following research question: how can we help the OSS projects performance and developer activity level to co-evolve to better shape alignment? This response folds on two main parts, first, we are looking for ways to analyze IS alignment process between OSS project performance and developers activity level. Second, by investigating if the co-evolution principles will influence positively the IS alignment through adaptive tension, change rate and modular design as the three 1st Principles of the co-evolution.

We have assessed the model using a PLS regression on 750 projects (enterprise applications). The analysis of IS alignment process between developers activity level (total bugs, closed bugs, bugs life, number of participants, number of developers) and OSS project performance are significant (respectively 0.27, 0.19, 0.13, 0.35) and the relationships are positive (cf table 5). On the other hand, we show that adaptive tension (Total visits, Number of visited pages), change rate (Information flows) and modular design (Development stages) variables are significant (respectively 0.27, 0.19, 0.13, 0.35) and influence more positively the OSS project performance than the developer activity level alone. Thus, we can conclude that we have found that all our hypotheses are significant and support our research question by showing that OSS projects performance and developer activity level shape IS alignment and that the co-evolutionary principles have a stronger influence. Though perhaps surprising, it is nonetheless totally logical to find that all of our 1st Principles evolve and influence positively IS alignment. It is hard to imagine the Principles not being embedded within the model findings. The results are in concordance with Greenwood and Hinings [16] who consider adaptive tension as a dynamic interplay of co-evolving interactions, Teece et al. [49] who stated that change rate is the capacity to renew competences and finally Narduzo and Rossi (2005) who understand modular design as a variation and improvement in existing modules. Thus we see that: Adaptive tension act to improve change rate of finding better IS alignment (speed of improvement in the model); Modularity acts to improve both change rate and Adaptive tension (diversity in the model). The combined action of the foregoing three Principles fosters improved co-evolution because status quo, and inertia would not result in improved IS alignment under these circumstances. In fact, if we do not take in consideration the co-evolution principles, we see that the OSS project performance and the developer level activity level are significant but the influence on IS alignment is weaker.

To become more significant and achieve a better IS alignment and improve OSS project performance project, there is a need to push toward a continuous dynamic interplay. First by draining users toward visiting their projects, proposing a stable applications (at a good development stage), and create a fruitful interaction between developers and users. In that specific context, we can assume that an OSS project is performing because it responds to the users need, enable the communication between the developers and users and propose the right stable application to users. Thus, to be efficient an OSS project needs to develop a favorable context where the information flows are enabled between the users and developers of OSS projects.

Tableau 5: Weight of the independent variables of the PLS components

<table>
<thead>
<tr>
<th>Variables Characteristics</th>
<th>Measures</th>
<th>VIP*</th>
<th>Coeff CS**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Tension</td>
<td>Total visits</td>
<td>1.2164</td>
<td>0.26714</td>
</tr>
<tr>
<td></td>
<td>Number of visited page</td>
<td>0.90401</td>
<td>0.18492</td>
</tr>
<tr>
<td>Change rate</td>
<td>Information flows</td>
<td>1.0138</td>
<td>0.13383</td>
</tr>
<tr>
<td>Modular design</td>
<td>Development stages</td>
<td>1.3417</td>
<td>0.35176</td>
</tr>
<tr>
<td>Developer Activity level</td>
<td>Total Bugs</td>
<td>0.90753</td>
<td>0.012261</td>
</tr>
<tr>
<td></td>
<td>Closed Bugs</td>
<td>0.89106</td>
<td>0.011178</td>
</tr>
<tr>
<td></td>
<td>Bugs life</td>
<td>0.73359</td>
<td>0.082417</td>
</tr>
<tr>
<td></td>
<td>Number of participants</td>
<td>0.95335</td>
<td>0.088904</td>
</tr>
<tr>
<td></td>
<td>Number of Developers</td>
<td>0.90044</td>
<td>0.050107</td>
</tr>
</tbody>
</table>

*VIP = Importance of the independent variables to explain IS alignment
**CoeffCS = Weight of the regression coefficient for the dependent variable

6. Discussion and conclusion

The IS alignment with the rest of an organization remains a critical and chronic unsolved problem in todays complex and turbulent world. Top-down approaches are accused of being static whereas bottom-up approaches allowing more autonomy remain vague in concept and application. To overcome this problem, we have reconciled top down rational designs and bottom up emergent processes. We have focused on the co-evolution nature at the individual level of IS alignment in an OSS context. We have proposed an IS alignment model based on co-evolutionary dynamic process between OSS projects performance and developer activity level. This model tries to respond to the following research question: how can we help the OSS projects performance and developer activity level to co-evolve to better shape alignment? This response folds on two main parts, first, we are looking for ways to analyze IS alignment process between OSS project performance and developers activity level. Second, by investigating if the co-evolution principles will influence positively the IS alignment through adaptive tension, change rate and modular design as the three 1st Principles of the co-evolution.

We have assessed the model using a PLS regression on 750 projects (enterprise applications). The analysis of IS alignment process between developers activity level (total bugs, closed bugs, bugs life, number of participants, number of developers) and OSS project performance are significant (respectively 0.27, 0.19, 0.13, 0.35) and the relationships are positive (cf table 5). On the other hand, we show that adaptive tension (Total visits, Number of visited pages), change rate (Information flows) and modular design (Development stages) variables are significant (respectively 0.27, 0.19, 0.13, 0.35) and influence more positively the OSS project performance than the developer activity level alone. Thus, we can conclude that we have found that all our hypotheses are significant and support our research question by showing that OSS projects performance and developer activity level shape IS alignment and that the co-evolutionary principles have a stronger influence. Though perhaps surprising, it is nonetheless totally logical to find that all of our 1st Principles evolve and influence positively IS alignment. It is hard to imagine the Principles not being embedded within the model findings. The results are in concordance with Greenwood and Hinings [16] who consider adaptive tension as a dynamic interplay of co-evolving interactions, Teece et al. [49] who stated that change rate is the capacity to renew competences and finally Narduzo and Rossi (2005) who understand modular design as a variation and improvement in existing modules. Thus we see that: Adaptive tension act to improve change rate of finding better IS alignment (speed of improvement in the model); Modularity acts to improve both change rate and Adaptive tension (diversity in the model). The combined action of the foregoing three Principles fosters improved co-evolution because status quo, and inertia would not result in improved IS alignment under these circumstances. In fact, if we do not take in consideration the co-evolution principles, we see that the OSS project performance and the developer level activity level are significant but the influence on IS alignment is weaker.

To become more significant and achieve a better IS alignment and improve OSS project performance project, there is a need to push toward a continuous dynamic interplay. First by draining users toward visiting their projects, proposing a stable applications (at a good development stage), and create a fruitful interaction between developers and users. In that specific context, we can assume that an OSS project is performing because it responds to the users need, enable the communication between the developers and users and propose the right stable application to users. Thus, to be efficient an OSS project needs to develop a favorable context where the information flows are enabled between the users and developers of OSS projects.

One of the limitations of this paper is the group of analysis, as we have tested the model for one specific OSS group (enterprise application), thus we need to extend it to other groups in order to facilitate the extrapolation. We can add also other 1st principles of evolution that exist like: Requisite Complexity, Causal Intricacy, Positive Feedback, and Irregular
Oscillation. The second limitation concern the scope of the study where we can analyze more in details the elements collected to find other influencing variables like the impact of the size of the project. Third limitation concerns, the problem of collecting the right data to assess the variable (for example modular design: the only way to measure it impact was to incorporate the development stages, because there is no way to extract measure of coupling and cohesion). We have also excluded the measure of code submitted, or the activity mailing list was unavailable.

Futures research, will concern the collection of data from other OSS group, but we will also try to link the IS alignment to Design science of OSS projects and putting in place a OSS design theory where we will mix the natural and design science principles.

7. References


