January 2007

A Framework for Aligning Testing and Development

Colin Onita
University of Memphis

Jasbir Dhaliwal
University of Memphis, jdhaliwl@memphis.edu

Recommended Citation
Colin Onita and Jasbir Dhaliwal. "A Framework for Aligning Testing and Development" *Proceedings of the 5th Workshop on Advances and Innovations in Systems Testing* (2007).

A Framework for Aligning Testing and Development

Jasbir Dhaliwal and Colin Onita
Systems Testing Excellence Program, University of Memphis
jdhaliwl@memphis.edu, cgonita@memphis.edu

Abstract

Strategic alignment between an organization's business strategy/capabilities and those of its information technology (IT) unit has been researched extensively and has found significant application in industry in the last decade. Given the critical interdependencies between development and testing groups within a corporate IT unit, this paper presents a similar alignment model for ensuring that these two functions work together effectively in meeting corporate IT goals pertaining to building new systems. This development-testing alignment (DTA) model is described and an overarching research framework for investigating its value and application is presented.

1. Introduction

As the business environment becomes more and more complex and companies become more globalized and integrated, the speed of doing business increases. This accentuates the need for accurate, valid, real time IT systems that support the business function and provide unique competitive advantage. In order to build and sustain such competitive advantage, companies have to rely more and more on their IT systems [11, 14] which have become integrated in virtually every aspect of their business operations. In spite of this, and despite their critical operational, tactic and strategic role, many new and old IT systems have either not offered what they were created for, or have failed outright. According to Gartner [8], on average, only 7% of software functionality that was paid for is actually used, with 85% of IT projects failing to meet objectives (32% being cancelled outright). Many of these failures and inadequacies result from a poorly executed development process. The development processes used employ either inadequate development models or flawed implementation due, in part, to the lack of proper testing and effective collaborative mechanisms between the development and testing functions. Testing has been defined in many ways, from software testing which is the process used to help identify the correctness, completeness, security, and quality of computer software to system testing which is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements [2]. This paper takes testing to mean all activities and actions (automated or otherwise) taken to ensure that systems are valid in relation to the real world that they model and in which they will operate; verified in relation to requirements and the series of specifications that follow each step of the systems development lifecycle or the V-model; and free of all types of errors.

A review of the testing literature reveals that relations between the development and testing functions are somewhat dysfunctional [5, 13] and there are no empirically sound findings pertaining to how the two functions can be aligned in both strategic and execution terms for success. Most research in the area of systems testing deals with technical issues related to code testing [7, 19], and other technical aspects. An integrated framework that focuses on aligning development and testing at all stages of system building and that ties development strategies and capabilities to testing strategies and capabilities within the corporate IT environment has not yet been developed.

This paper proposes a DTA model which posits that such alignment leads to beneficial effects such as lower costs and shorter time of development, greater system quality, fewer errors and a better relationship between the corporate IT unit and customers in business functions who have commissioned new systems. Measuring alignment is an important requirement for any proposed model given the "measure and manage" basis of modern technology management. Alignment models and measurements have been studied in other related contexts [6, 15,16] but never ***within*** corporate IT units and specifically between the development and testing functions. Figure 1 shows two distinct levels at which alignment takes place in relation to building business systems in corporate settings. Levels one focuses on alignment between the overall business strategy and the corporate IT unit that supports it. Level two, which is the specific focus of this paper, looks at alignment between the development and testing functions within the corporate IT unit.

Six alternative conceptualizations of this fit can be proposed as adapted from Venkatraman [20]:

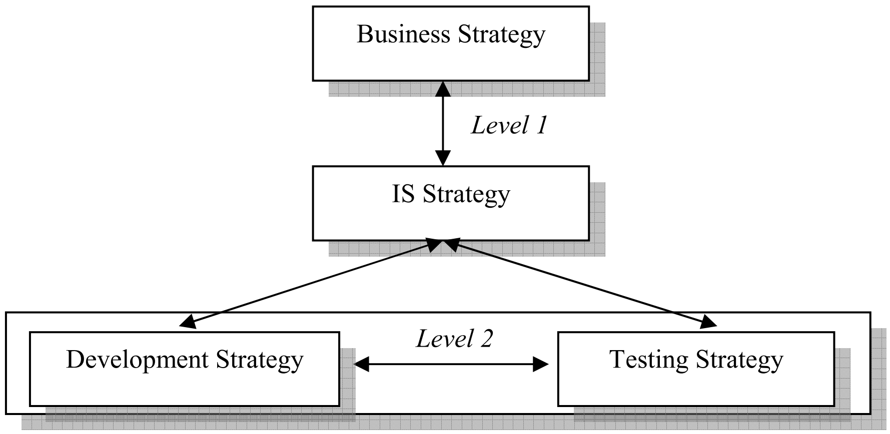

Figure 1: Levels of Alignment

2. Aligning Development & Testing

Business strategy is at the core of all organizational activity. Information technology has to directly support business strategy in order to create the synergistic effect of technology contributing effectively to business success. Similarly, since systems development is an integral part of corporate technology acquisition strategies, it too has to be aligned with testing to ensure business success. In many organizations, there is a gap, at both strategic and functional levels, between development and testing groups as well as between individual testers and developers. To bridge these gaps, this paper proposes a series of methods, grouped under the DTA model that draws upon the strategic alignment model initially proposed by Henderson and Venkatraman [6]. This DTA model focuses on the **fit** between the development and testing functions and how they operate collaboratively to support each other to achieve the goals of the corporate IT unit.

Alignment has emerged as a key managerial concept in relation to the strategic management of business systems. While it has largely been studied and applied in terms of the mapping between business strategy/capabilities and IT strategy/capabilities, it can also be usefully applied at another level of granularity. This is the alignment between development and testing strategy/capabilities and can be defined as the strategic and operational fit between the development and testing functions.

1. DT Alignment as Moderation – this fit between development and testing is considered to be a third-party composite variable that influences the direction and strength of the relationship between each function's strategy and their outcomes. For example, just as DTA moderates the relationship between development strategy and the success of the development effort, it also moderates the relationship between testing strategy and the success of the testing effort.

2. DT Alignment as Mediation: this fit is a necessary intervening or intermediate variable between the strategy and success of effort of either function. In Figure 1, this would mean that the effectiveness/success of both development and testing efforts is explained or fully mediated by DTA. While it could be possible for partial mediation to exist, this is not pursued here for purposes of conceptual clarity given the goals of this paper.

3. DT Alignment as Matching: this fit is the correspondence or equivalence between development and testing strategies and capabilities. For example, a close mirroring of capabilities, tools and resources in both the development and testing functions would suggest strong DTA leading to successful implementation of business systems.

4. DT Alignment as Gestalts: this fit is defined in terms of the degree of internal coherence between development and testing strategies/ capabilities. Here, alignment would be measured through comparison to other groups of companies with related development and testing strategies/capabilities.

5. DT Alignment as Profile Deviation – this fit is viewed in terms of what would constitute as an ideal profile of well aligned development and testing. The tops 10% of best performing companies can be investigated in relation to their DTA strategies/capabilities and the implementation process of a sound alignment arrangement. Next, differences between this ideal profile and the current state of affairs in a particular organization can be brought to focus.

6. DT Alignment as Covariation – this fit between development and testing is seen as a covariation of attributes that characterize each separate function. Here we look at the attributes of development and testing separately and investigate how they covary or diverge.

A key goal of this research is to present a methodology for applying these concepts within the corporate IT unit tasked with building and implementing business system applications.

Teo and King [18] hypothesize that a high level of integration of business and IT plans may facilitate communication and collaboration. In the areas of development and testing, a high level of integration at both the functional and strategic levels may also facilitate communication and collaboration between them. There also has to be close correspondence between testing and development in capabilities, skills, methods and governance. For example, a centrally governed development function misaligned with a decentralized testing function may not be optimal for success.

Related to these ideas, are the dual concepts of integration and correspondence. Integration represents the level of linkage between development and testing, while correspondence represents how closely their capabilities mirror and complement each other. As shown in Figure 2, there is a recursive relationship between integration and correspondence that either promotes or reduces alignment. Similarly, varying levels of alignment can either induce or minimize integration and correspondence. This is a common characteristic of all alignment models in the literature as verified by Reich and Benbasat [15].

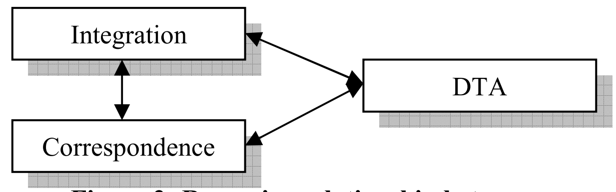

Figure 2: Recursive relationship between Integration, Correspondence and DTA

To test the validity of these theoretical ideas, this research will devise an instrument for testing and quantifying the value of DT alignment. It is imperative that for a new model to be useful, it has to improve on the current state of affairs. Figure 3 details the key structural and flow components of our DT alignment model for development and testing within the corporate IT unit. This model decomposes the alignment of the development and testing functions along three key flow dimensions: 1) strategic alignment, 2) capabilities alignment, and 3) strategy-execution alignment.

Both the strategy and capabilities levels are investigated along the two functional dimensions of development and testing. To achieve alignment all four dimensions have to be matched in capabilities, resources, structure, etc. This does not mean that they have to be similarly matched, but that testing complements development and acts as an enabler of development success by providing verification, validation and bug-finding services.

The process of aligning testing and development needs to be clearly linked to benefits for the organization. If the testing process seamlessly supports the development process and makes it more robust, efficient and effective, then testing is adding value to IT systems development and to the business. This directly translates into shorter development times, higher quality of output, and indirectly into lower development costs, and higher revenues through the enhanced capability for taking advantage of opportunities and mitigating threats.

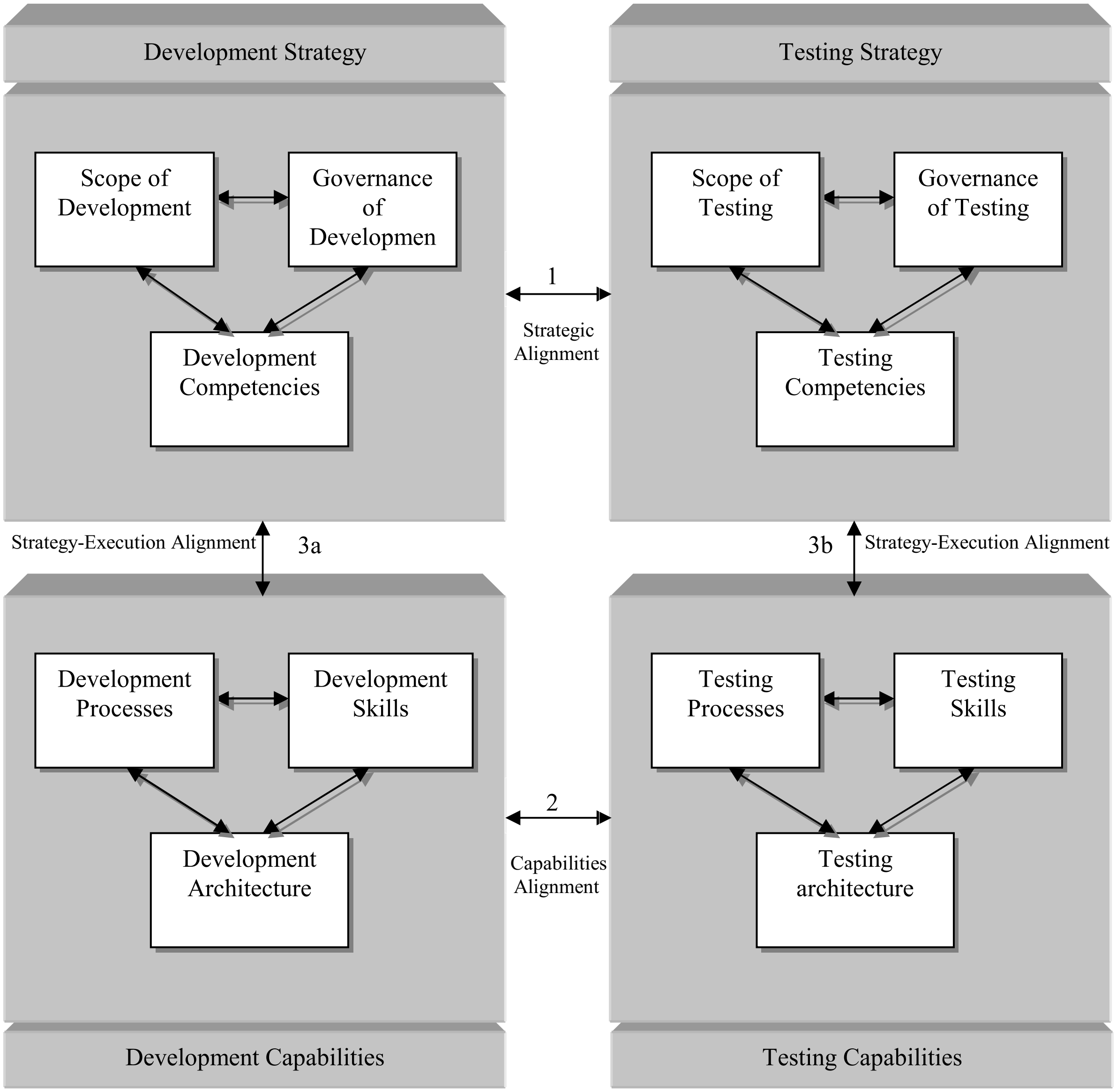

Fig 3: Alignment model for testing and development (adapted from Henderson and Venkatraman [6])

When looking at aligning development and testing, we have to take into consideration external and internal factors affecting alignment. The external domain is comprised of environment attributes that are common to all companies engaged in the industry. These attributes include the level of technological sophistication in development and testing, distinctive IT attributes and applications that individual firms use and that differentiate firms within an industry.

Each structural component (box) of the Alignment Model deals with alignment from a double perspective: strategy/capabilities and development/testing.

The first structural component, development strategy looks at the specific IT artifacts that the firm uses in order to function. Here the scope of IT development is defined in terms of specific information artifacts and specific IT goals that support the business strategy. The formal makeup of the IT development departments and teams, buy or build decisions, as well as the overall competencies and state of art in the IT field are also taken into consideration.

The second structural component, development capabilities, deals with the internal makeup of the development function and the processes that are exogenous to it. This directly impacts applications being developed, tools used in development processes, as well as models or frameworks employed in the development process. Decisions about development models, such as SDLC, RAD, prototyping, etc. and the skills of the development personnel and particular tools that are used in development processes are also considered at this level.

On the testing side, which represents the third structural component, testing strategy focuses on the testing goals and competencies, the scope of testing within the development framework, the available tools and methods for testing software. In-house or outsourced testing decisions are considered as well as the structural makeup of the testing department.

The fourth structural component describes the testing capabilities involved in software testing. The specific methods of testing (traditional, V-mode, iterative), as well as specific choices about testing tools, architecture, communication structure, etc. are considered and brought to focus from an alignment perspective. The individual skills of testing personnel are also assessed.

DT Alignment has three flow dimensions (as represented by the numbered vertical and horizontal arrows): strategic alignment, capabilities alignment, and strategy-execution alignment.

2.1 Strategic Alignment

Strategic Alignment (arrow 1) deals with the fit between the development strategy and the testing strategy. There needs to be a synergy between the way things are done at the development and testing strategy levels. The structure of both functions has to be similar, the purpose and scope of development and testing have to be in harmony, and the level of authority and autonomy (governance) of the two functions has to be complementary.

2.2 Capabilities Alignment

Capabilities Alignment (arrow 2) focuses on the comparative capabilities of development and testing at the operational/execution level. Here, the processes, skills and architectures of the development and testing functions have to be synergistic and complementary. Development methodologies and philosophies have to match testing tools and methods. The skills of testers and the procedures of testing have to supplement and support the skills of developers and procedures used in development. Dysfunctional relations between development and testing are often the result of misalignment of capabilities.

2.3 Strategy-Execution Alignment

2.3.1 Development Strategy-Execution Alignment

Strategic-Execution Alignment within development (arrow 3a) refers to the ability of the development function to execute its stated strategy. Here strategic decisions have to be supported by the operational infrastructure that is in place. Strategic decisions must be supported by the competencies of the operational staff and by their tools and methodologies. The functionality provided by the operational level has to empower and support the strategic goals and decisions made by the strategic decision makers such as CIOs, CTOs and so on. For example, decisions made about development strategy have to be feasible in relation to the skills of the developers and the tools that developers have available.

2.3.2 Testing Strategy-Execution Alignment

The testing Strategy-Execution Alignment (arrow 3b) is similar to the above in that testing capabilities (competencies, tools and methodologies) have to support the execution of stated testing strategies. Examples of this type of alignment are testing operations support, in the form of skills, tools, and testing procedures, that enable execution of testing strategies devised by upper management in testing. It is imperative that testing executives focus on ensuring that the testing strategies they devise are executable from the perspective of the testing capabilities they have built in their organizations over time.

3. Research Framework for Investigating DTA

3.1 Three stage model for empirical assessment

It is imperative that the theoretical DTA model presented in Section 2 be empirically tested in relation to its underlying assumptions and practical implications.

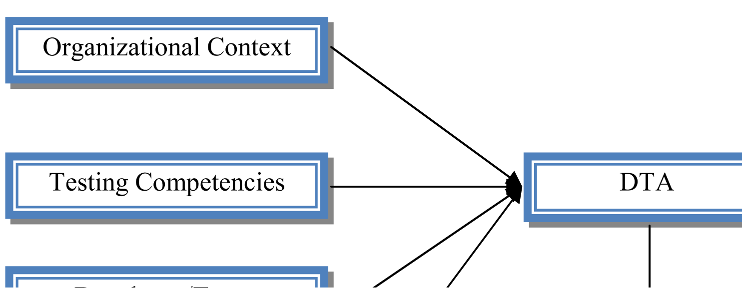

Figure 4 presents a three stage model for this purpose that highlights not only the key subcomponents of DTA but also the key antecedents that influence the level of DTA together with the key outcome variables resulting from positive DT alignment.

Brown and Magill [4] investigated a host of antecedents to strategic alignment in the IT strategy area. By adapting from their study, our framework identifies four key antecedent factors that influence the level of DTA within a corporate IT unit. These are the: 1) the organizational context, 2) testing competencies, 3) the relations between developers and testers, and 4) methodologies used for development and testing. An understanding of these critical antecedents is important to the implementation of a proper DT alignment strategy. A key dimension of organizational context is the support testing receives from various stakeholders (managers, employees, process owners, etc). A key dimension of testing competencies is the experience of the testing personnel together with the robustness and power of the tools and procedures available. The quality and dynamics of developer-tester relations within the corporate IT unit represents the third factor, while the type of methodologies used for development and testing efforts represents the fourth key antecedent factor.

The model also identifies three key impact variables for testing the value that DTA brings to the organization. These are reductions in development cost and time, as well as an increase in the quality of systems build and implemented.

Section 3.2 presents more detail on the antecedent constructs, while Section 3.3 elaborates on the impacts of DTA. The detailed dimensions of DTA have already been presented in section 2.

3.2 Antecedents of DT Alignment

The organizational context, tester competencies, developer/tester dynamics and the specific development methods used all influence the level of alignment between testing and development.

The first construct is the organizational context in which development and testing occurs. This influences alignment between the development and testing functions. Organizational context, as an antecedent construct, has four sub-constructs which are presented in the Table 1.

The second construct, testing competencies, describes the experience and testing know-how available to the testers. Here we investigate specific characteristics of both the individual testers and of the overall testing function as exemplified by documentation, best practices, tools, etc.

Developer/tester dynamics include all aspects of the relationship between developers and testers – communication, trust, credibility, power distance, personal relationships, shared domain knowledge, understanding – that promote a harmonious and cooperative interaction between the development and testing functions.

The methodologies used are focus on the ways in which development and testing are being done. Certain methodologies are more suitable for testing purposes, while others are more standardized and relegate testing to the end of the development process.

The focus of the Model (DTA dimension) represents developer/tester alignment (DTA). Here we focus on the four dimensions of alignment – Development and Testing Strategy and Development and Testing Infrastructure/Processes (for further clarification, please refer to section 2).

Table 1: Organizational context breakdown (adapted from Brown and Magill [4])

Organizational context	Organizational structure	Reporting structure Formal Structure Informal Structure
	Organizational environment	Communication, trust, support, learning
	Testing function status	Authority of Testing Function Autonomy of Testing Function Attitude towards Testing Function
	Leadership style	Authoritarian or autocratic Participative or democratic Delegative or Free Reign

3.2.1 Development Methods

Choosing development methods is an important decision when building new IT systems. In the following we will briefly present the most used development methods.

The method of choice for Large Scale Complex Systems is the SDLC [10] and it is also the method that can benefit the most from the idea of aligned testing, since, large systems are most likely to affect the competitive advantage. There have been multiple modifications of the traditional SDLC model that integrate testing at various stages of the SDLC. The most well known modification is the V-Model which will be presented below.

The traditional SDLC framework is typically composed of consecutive stages, where the output of each step becomes the specifications (input) for the next stage. In general, once a stage has produced its output, it is considered complete, and the development process will not revisit it again. There are many versions of the SDLC, with the number of phases ranging from 3 to 10, but, basically, the individual stages of the SDLC are Analysis, Design, Development, Integration and Testing and Installation and Acceptance Phase. Testing is usually relegated to a verification and validation of the built components that addresses only the surface issues of whether the coded product complies with the gathered requirements. The requirements themselves are seldom tested and the problem specification is seldom verified.

Rapid Application Development (RAD) is a software development methodology that focuses on building applications in a very short amount of time; traditionally with compromises in usability, features and/or execution speed. The term generically describes applications that can be designed and developed in a short time frame, but it was originally intended to describe a process of development that involves application prototyping and iterative development. Speed and quality are the primary advantages of Rapid Application Development, while reduced scalability and feature sets are the disadvantages.

The most known model of integrated testing is the V-model, which has as its basis Sommerville's [17] V&V (validation and verification) process. The V&V process is a whole life cycle process, which is applied at each stage of the software development process. Its main objectives are to discover defects in the system and to assess whether the system is usable in production. Boehm [3] specifies the basic V-Model that has been heavily modified since its first specification in 1979.

3.4 Impact of DT Alignment

This paper posits that DT alignment will lead to benefits for the firm. The impacts of alignment are threefold.

The time it takes to develop a new system or software should be reduced if alignment between the testing function and development function is achieved. This is because close support, integration and cooperation between the functions will promote better and more efficient development. Testing would act as a gatekeeper between development steps and would provide verification and validation services for all development activities. As errors are found early in the development process, and remedied efficiently this

would lead to less time spent in subsequent phases of development and would also lead to a higher quality of the end result – the developed system. This is the second outcome of DT alignment – quality of the developed system. There are many ways to assess the quality of the system – the level of satisfaction of users with the system, the level to which requirements have been met, the efficiency and effectiveness of the system, the number of errors per lines of code, etc. Alignment between testing and development influence with all of these quality characteristics.
Finally, the cost of development would be impacted by the degree of alignment between development and testing. If testing and development are in misalignment resources would be wasted either by duplication, misuse or no use. An increase in development time also translates into increased development costs.

To investigate the research model, we propose a set of hypotheses that map the antecedents and outcomes of DTA.

In relation to the antecedents of DT alignment we propose the following:

H1: Organizational structure has an impact on the level of alignment between development and testing.

H2: The organizational environment has an impact on the level of alignment between development and testing.

H3: The status of the testing function has an impact on the level of alignment between development and testing.

H4: Leadership style has an impact on the level of alignment between development and testing.

H5: Increased experience positively influences the level of alignment between development and testing.

H6: The relationship between developers and testers positively influences the level of alignment between development and testing.

H7: Specific methodologies influence the level of alignment between development and testing.

Figure 5 demonstrates the directional influence of these hypotheses.

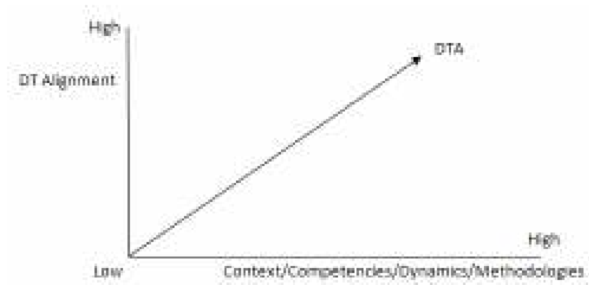

Figure 5: Antecedents of DTA

In relation to the impacts of DTA we propose the following:

H8: An increase in the level of alignment between development and testing will lead to an increase in quality of developed product.

H9: An increase in the level of alignment between development and testing will lead to a decrease in development time.

H10: An increase in the level of alignment between development and testing will lead to a decrease in development cost.

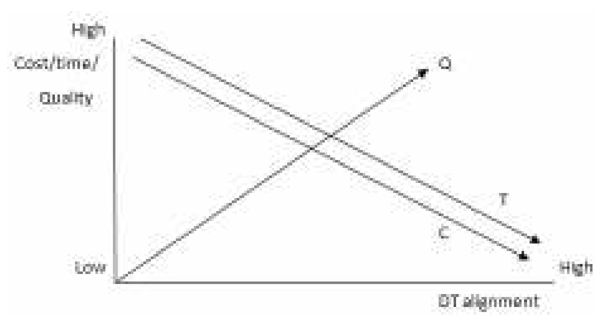

Figure 6: Impact of DTA

Figure 6 highlights the directional impact of the hypotheses H8 to H10.

4. Research design

The best way to research DTA would be by employing a field study, coupled with a sample survey, and ending with a field experiment. The initial field study would investigate the antecedent variables that were discussed in Section 3. A problem is that these variables may not be stable over time and the development process span many months, introducing additional complexity into the study. Ideally, we would like to see two or more projects run at the same time, with the same purpose and in the same environmental setting that would differ only on whether or not the development and testing functions are aligned. This is, however, unrealistic. What we could realistically obtain is archival data about how systems were developed prior to the introduction of DT alignment, and the outcomes of those development processes. These archival results would then be compared in a field study to the results of a current project that was recently completed by employing the DT alignment model. This design suffers from the fact that the environment at the time the archival data had been collected may have been different in a significant way from the current environment.

A more realistic approach to the research would be to employ a case study to measure the structural and flow dimensions that make up the Alignment Model within the IT unit of a large IT organization. Once the measurement is complete, a survey instrument along with actuarial and archival data can be employed to measure the levels of the outcomes of DTA.
Action research should also be used at a later stage to implement an alignment strategy, and, once alignment has been achieved, the survey tool, archival and actuarial data analysis strategies could be employed to measure the levels of the outcomes.

Table 2: Research Plan

Research Plan	
Step 1:	Operationalize DTA
Step 2:	Operationalize Antecedents
Step 3:	Operationalize Outputs
Step 4:	Survey project without DTA
Step 5:	Implement DTA
Step 6:	Survey project with DTA
Step 7:	Analyze results

5. Conclusion

The paper presents a model for aligning the development and testing functions. This alignment approach is posited to be beneficial by decreasing the cost and the time needed to build new IT systems, as well as by increasing the quality of the developed system.

6. References:

[1] Bernard H. Boar; "Application prototyping: a requirements definition strategy for the 80s", JWS, 1984

[2] Black, Rex; Managing the Testing Process (2nd ed.) (2002). Wiley Publishing.

[3] Boehm B. W. "Guidelines for verifying and validating software requirements and design specifications." EURO IFIP'79, pages 711-719, 1979

[4] Brown Carol, Magill Sharon "Alignment of the IS Functions with the Enterprise: Toward a model of Antecedents.", MISQ, December 1994

[5] Cohen, C. F., Birkin, S. J., Garfield, M. J. and Webb, H. W. "Management conflict in software testing." *Communications of the ACM,* 47(1), 76-81, 2004

[6] Henderson, J. and Venkatraman, N. "Strategic alignment: leveraging information technology for transforming organizations.", IBM Systems Journal, No1, 1993

[7] Alan R. Hevner, Harlan D. Mills "Box-Structured Methods for Systems Development with Objects." IBM Systems Journal 32(2): 232-251, 1993

[8] Hunter and Blosch "Managing the New IT Risks". Gartner, 2003

[9] Li, Eldon; "Software testing in a system development process: a life cycle perspective", Journal of Systems management 41, 8, 1990

[10] Mahmood, M. "System Development Methods – A comparative Investigation", MIS Quarterly, September 1987

[11] McFarlan, F, W "Information Technology Changes the way you compete", Harvard Business Review (62,2), 1984

[12] McKeen, James D. "Successful Development Strategies for Business Application Systems", MIS Quarterly, 1983

[13] Pettichord, "B. Testers and developers think differently: Understanding and utilizing the diverse traits of key players on your team." *Software Testing & Quality Engineering,* 2(1), 42-45, (2000)

[14] Porter, M, and Millar V. "How information technology gives you competitive advantage"", Harvard Business Review (63,1) 1985

[15] Reich and Benbasat "Factors that influence the social dimension of alignment between business and information technology objectives." MISQ 24(1), 81-113, 2000

[16] Sabherwal, Hirschheim and Goles; "Information systems – business strategy alignment: The dynamics of alignment: Insights form a punctuated equilibrium model. Strategic information management: Challenges and strategies in managing information systems", (Galliers and Leidner, Eds) pp 311-346, Butterworh-Heinemann, Oxford, 2003

[17] Sommerville I. "Software Engineering" 7th Edition. Addison-Wesley, Menlo Park, CA, 1999

[18] Teo and King; "Assessing the impact of integrating business planning and IS planning." Information and Management 30, 309 – 321, 1996

[19] Tomas Vagoun, Alan R. Hevner "Feasible Input Domain Partitioning in Software Testing: RCS case Study." Ann. Software Eng. 4: 159-170, 1997

[20] Venkatraman, N. "The Concept of Fit in Strategy Research", Academy of Management Review, Volume 14, Number 3, 1989a, pp. 423-444

[21] Venkatraman, N., Prescot, J.E. "Environment-strategy-Coalignmnet: An Empirical Test of its Performance Implications." Strategic Management Journal, Number 3, 1984, pp 513-525