

2012

Contribution of Agile Software Development Methods to Business-IT Alignment in Non-profit Organizations

Arjan Aarnink
HU University of Applied Sciences Utrecht

Gijsbert Kruithof
HU University of Applied Sciences Utrecht

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/ciima>

Recommended Citation

Aarnink, Arjan and Kruithof, Gijsbert (2012) "Contribution of Agile Software Development Methods to Business-IT Alignment in Non-profit Organizations," *Communications of the IIMA*: Vol. 12 : Iss. 2 , Article 1.

Available at: <https://scholarworks.lib.csusb.edu/ciima/vol12/iss2/1>

This Article is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in Communications of the IIMA by an authorized editor of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

Contribution of Agile Software Development Methods to Business-IT Alignment in Non-profit Organizations

Arjan Aarnink

HU University of Applied Sciences Utrecht, The Netherlands
arjan.aarnink@student.hu.nl

Gijsbert Kruithof

HU University of Applied Sciences Utrecht, The Netherlands
gijsbert.kruithof@student.hu.nl

ABSTRACT

Business-IT alignment (BIA) is still a major IT management concern. Agile Software Development (ASD) is becoming a commonly used method for IT development. An important characteristic of this method is high involvement of business members in software development. The potential benefits of the use of ASD on the business-IT alignment within organizations have not yet been investigated. This paper addresses the question whether ASD can contribute to business-IT alignment within organizations where software development is being done by the organization itself. We identify the characteristics of agile software development by using an Agile Maturity Model. Business-IT alignment is operationalized by using Luftman and Kempaiah's Business-IT Alignment Maturity Model. Our hypothesis is that the use of agile software development will increase the BIA-indicators Governance, Communication, 'Competence & Value Measurements' and Partnership. The hypothesis is tested at a non-profit organization in the Netherlands consisting of 600 employees. Our findings support the hypothesis: Three of four BIA-indicators show considerable improvement where one BIA-indicator is showing minor improvement. Further research is recommended at more organizations, involving all BIA-indicators.

Keywords: Business-IT alignment, agile software development, business-IT alignment maturity

INTRODUCTION

Regularly articles about the importance of alignment between business and IT are published. For example, according to the results of the "16th strategic Information Management / Trends in IT" by Derksen and Luftman (2011), the topic is still a big concern for IT managers. Where 'IT and business alignment was the 3th most important concern in 2010, in 2011 it is the number one concern for IT management in the Netherlands (see Table 1).

Articles and papers often focus on the importance of business-IT alignment and describe the use of frameworks to measure the business-IT alignment maturity (e.g. the BIA Maturity model, Luftman and Kempaiah (2007)). However, the actions to accomplish a better alignment are often harder and more complicated to define.

IT management concerns	2011-NL	2010	2009	2008	2007	2006	2005	2004	2003
Business Productivity & cost reduct.	2	1	1	7	4				
Business agility and speed to market	7	2	3	13	17	7		5	7
IT and business alignment	1	3	2	1	2	1	1	1	1
IT reliability and efficiency	3	4	6						
Business Process Man./reengineering	6	5	4	18	15	11	5	10	10
IT strategic planning	8	6	7	3	8	4	4	4	
Revenue generating IT innovations	16	6	8						
IT cost reduction	5	8	5	7	4				
Security and privacy	4	9	9	8	6	3	2	3	3
Globalization	20	10	15						
Knowledge management	9								
Change management	10								

Table 1: Top 10 IT Management Concerns NL Versus Global (Derksen & Luftman, 2011).

Agile software development is a group of software development methods based on iterative and incremental development in small multifunctional teams. "Agile software development" was coined by the *Agile Manifesto*. (Beck et al., 2001)

The main characteristics are:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile software development claims to be more responsive to changing business needs than traditional software development methods (Beck et al., 2001). Through means of early, systematic and continuous involvement of business members in the software development team a high degree of involvement of the business is guaranteed. A few well known agile software development methods are:

- Extreme programming
- Dynamic software development method (DSDM)
- Crystal methodologies
- Scrum

Dybå and Dingsøy (2008) reviewed all empirical studies on agile software development methods up to 2005. They concluded that there was not yet any substantial empirical evidence on the effectiveness, strength and limitations of agile software developments. They recommend much more empirical investigations should be conducted, especially on the popular agile method Scrum. In line with these recommendations some empirical data will be provided in this paper on the effects of the agile software development Scrum.

Our aim is to provide insight into improvement of business-IT alignment maturity by using the agile software development method Scrum within non-profit organizations. Scrum is chosen because it is one of the most used and researched upon (Dybå & Dingsøy, 2008) agile software development methods. The associated research question is: "How can the agile software development method Scrum be used such that it contributes to business-IT alignment?" The

scope of this investigation are organizations where software development is being done within the organization itself.

If the use of agile will contribute to a better business-IT alignment, not only software development projects would benefit from the method. It is also a justification to use the agile method as a tool to improve alignment between business and IT in general. In that case the benefits will last longer than the project. The benefits will be permanent.

In this paper the correlation will be investigated between the use of the agile software development method and the business-IT alignment maturity by measuring the maturity of those employees that are using agile as a software development method and those who are not. A case study at a not-for-profit organisation in the Netherlands will be used to verify the artefact.

In the next paragraph the main theoretical constructs: Business-IT alignment and agile software development will be derived from literature. Business-IT alignment will be operationalized using Luftman and Kempaiah's Business-IT Alignment Maturity Model (2007). Business-IT alignment will be measured at a not-for-profit organisation in the Netherlands among employees using agile software development and those who are not, using Luftman and Kempaiah's questionnaire. The agile software development method Scrum will be operationalized using interviews and a questionnaire to assess the actual application of this software development method.

In the chapter 'Framework validation' the results of the case study will be presented. And finally conclusions will be drawn and recommendations be made for further enquiry.

THEORETICAL FRAMEWORK

Business-IT alignment has been introduced in the literature by Henderson and Venkatraman (1993). They describe the lack of integration of the business- and IT-domain as a possible cause of non-realization of IT-value. The Strategic Alignment Model that they developed supports organisations in assessing and developing the use of information technology. Henderson and Venkatraman identified four cross domains relationships: The business-strategy domain and IT strategy-domain both accounting the internal and external domain. Maes (1999) developed an extension of this model. Two dimensions were added: the internal domain was divided in a structural and operational part. The second extension was the relationship between business and IT.

Luftman and Kempaiah (2007) elaborate on the Henderson and Venkatraman (1993) model and puts focus on the necessity of business transformation. In Luftman and Kempaiah's model for business-IT alignment, five levels of alignment maturity are defined: Initial/ad-hoc; Committed process; Established focused process, Improved/managed process and Optimized process.

Each of the maturity levels is focussing on a set of six indicators to determine the alignment between business and IT: 1. Communication: In what way do employees of business and IT-departments understand each other. Do they communicate frequently and effectively with each other, with vendors, consultants and partners? 2. Competency/Value Measurements: How well does the IT-department demonstrate its value to the business and how well is the organization measuring its project performance. 3. Governance: Is there a decision-making authority

available. And in what way are decisions based on the company's strategy 4. Partnership: How good is the relationship between the IT- and business departments? Does the IT-department fulfill an equal role in developing business opportunities? And in what way are rewards and risks shared between IT and business departments. 5. Scope and architecture: What is the maturity of the information technology and in what way is IT more than just a department to support business departments. 6. Skills: Do the employees have the right skills to perform their jobs. Is the company ready to change in its environment? In this paper, Luftman and Kempaiah's Business-IT Alignment Maturity Model is used because it is operationalized in a questionnaire which has been often used and researched.

Software development methods have evolved steadily over the decades (Zhang, Hu, Dai, & Li, 2010). Starting in the 1950s with a two-step process: analysis followed by coding, in 1970s life cycle models were developed. The best known method being the Waterfall method: a set of sequential phases starting with requirements analysis, through program design, coding, testing and operation. In the 1990s a new set of software developing methods came into being which came to be known as "agile". Their main characteristic as opposed to the waterfall method is an incremental and iterative approach.

In an agile process software is developed and tested in short repeated cycles. Each cycle ends with a deployment to the users of the software. In 2001 the *Agile Manifesto* was published. The practitioners claimed agile software development to be more responsive to business needs than other traditional methods (Beck et al., 2001). This claim has not been substantiated with much empirical evidence. A thorough review of all literature up to 2005 by Dybå and Dingsøy (2008) concluded that much more empirical research is needed. Hossain, Babar and Paik (2009) did a literature review on one specific agile method called Scrum. They also concluded that the strength of evidence on an agile strategy is very low. Recently Lee and Xia (2010) have developed a construct of software agility in terms of dimensions, determinants and effects on software development performance. Based upon extensive quantitative and qualitative data analysis they discern two dimensions: 1. Software Team Characteristics and 2. Software Development Agility. The team characteristics are Software Team Autonomy and Software Team Diversity. Team autonomy refers to the extent to which the software team is empowered with authority and control in making decisions. Team diversity refers to the extent to which team members differ in their functional backgrounds. Software development agility has two aspects: 1. Software team response extensiveness and 2. Software team response efficiency. Team response extensiveness is defined as the proportion of requirements that is realized. Team response efficiency is defined as the minimal time, cost, personnel and resources that the team needs to respond to a particular requirement change.

In table 2 the agile software development dimensions are plotted against the business-IT alignment (BIA) indicators that are expected to be influenced by agile development. The autonomy of the software team is expected to have a positive effect on Governance and Partnership because of the given empowerment of the team to make its own decisions. The diversity of the software team is expected to have a positive effect on Partnership and Communication because both business and IT people are present in the team. Response extensiveness is expected to have a positive effect on Value because the requirements of the Business can all be governed and monitored by the team itself. The response efficiency is

expected to have a positive effect on Value and Governance because changing or new requirements can directly be addressed by the team itself.

Agile Software Development	BIA
Software team autonomy	Governance, Partnership
Software team diversity	Partnership, Communication
Software team response extensiveness	Value
Software team response efficiency	Value, Governance

Table 2: ASD Dimensions Plotted Against Business-IT Alignment Indicators.

Agile software development within the organisation is expected to influence business-IT Alignment along to the above mentioned aspects. To measure the usage of the agile software development methodology, the Agile Maturity Model (AMM) for agile software development environments developed by Patel and Ramachandran (2009) will be adopted.

The hypothesis tested in this paper is that the business-IT alignment maturity in an organisation on the indicators Communication, Value, Governance and Partnership after an agile software development project is higher than before. The theoretical framework derived from this hypothesis can be found in figure 1. In the next paragraph this theoretical framework will be operationalized.

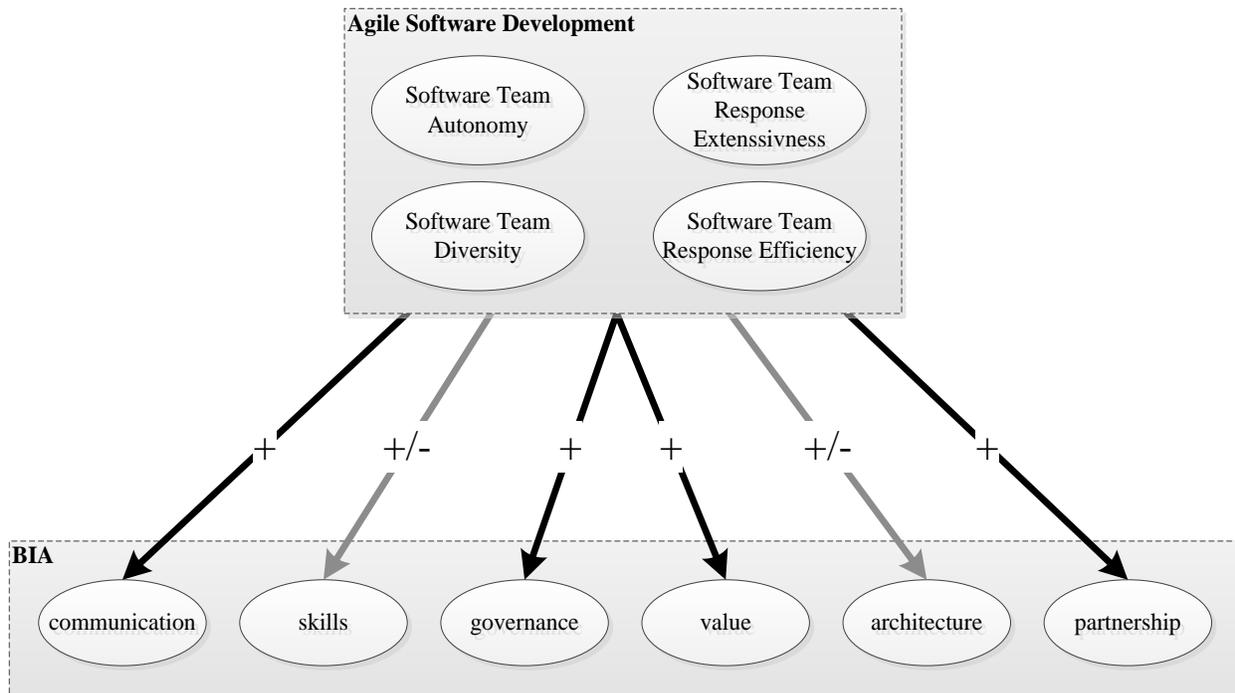


Figure 1: Research Framework: the Four BIA-Indicators Affected by Agile Software Development.

FRAMEWORK OPERATIONALIZATION

Business-IT Alignment

In this research Luftman and Kempaiah's (2007) Strategic Alignment Maturity (SAM) Model will be adapted. This model is widely accepted and has been operationalized in a questionnaire which has been commonly used in research (Silvius, de Haes, & van Grembergen, 2010). Luftman and Kempaiah's model identifies six indicators assessing BIA-maturity. These six indicators are discussed in the previous chapter and can be found in the model below (figure 2).

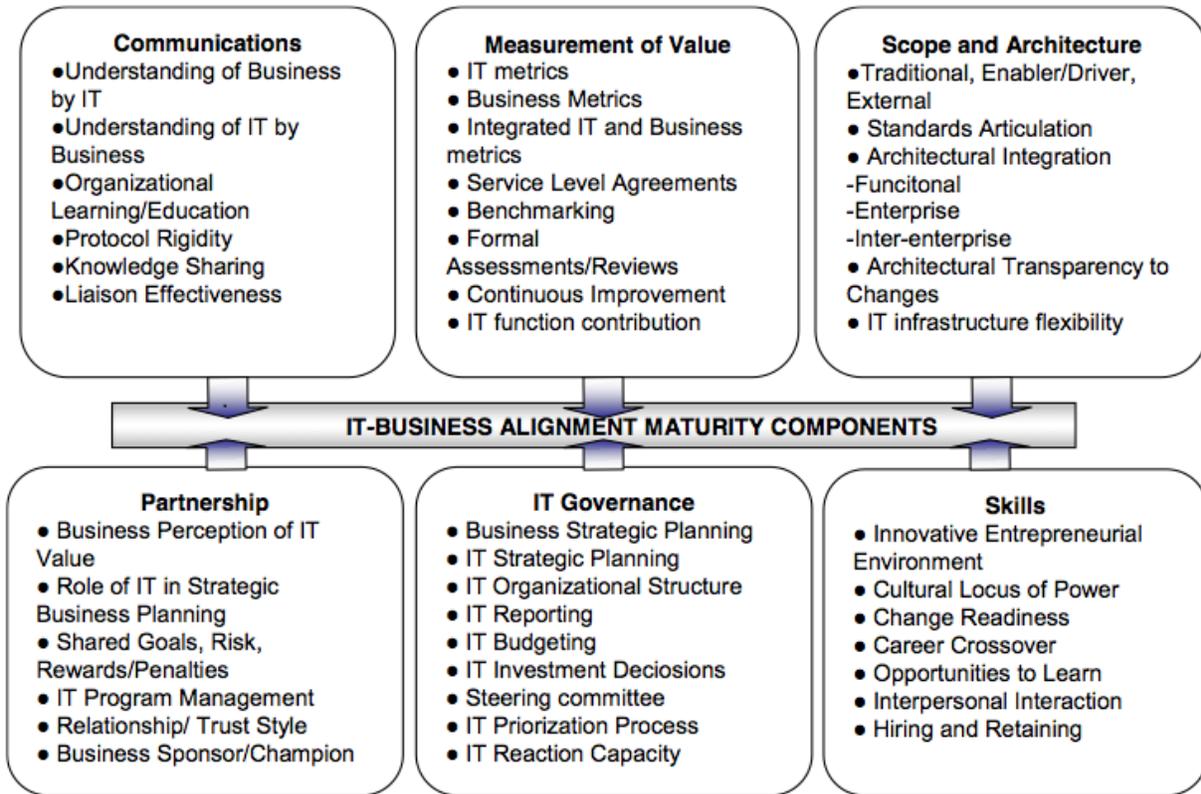


Figure 2: Six Indicators That Show BIA-Maturity.

Each of the six indicators can be assessed to determine the maturity. Based on the individual scores an overall maturity can be determined. In this research the four indicators (see Figure 1) where alignment improvement is suspected will be assessed. The assessment consists of an questionnaire which upon request can be obtained from the authors..

The SAM model consists of 5 levels of maturity: Initial/Ad Hoc processes; Committed processes; Established, focussed processes; Improved, managed processes; Optimized processes.

Agile Software Development has no established tradition of how to be operationalized. In the previous chapter the dimensions of software agility as defined by Lee and Xia (2010) were discussed. The extent to which the agile software development methodologies are implemented within projects or organizations however calls for another approach. To measure usage of the

agile software development methodology, the Agile Maturity Model (AMM) for agile software development environments developed by Patel and Ramachandran (2009) will be adopted. This model has been operationalized in a questionnaire and validated.

The model (see Figure 3) consists of 5 levels of maturity. From level 1(initial) where no agile software development process is defined, to level 5 (Sustained).

Each level of maturity can be assessed with a questionnaire. The assessment questionnaire for level 2 (explored) can upon request can be obtained from the authors.

In level 2 four Key Performance Areas (KPA) are defined: 1.Project Planning, 2. Story cards driven Development, 3. On-site Customer availability, 4. Introduction of Test Driven Design (TDD).

The AMM aims to identify problems, improve and enhance the usage of the agile application development methodology. In our research we will not look at these aspects: the model will be used to identify the level of agile maturity.

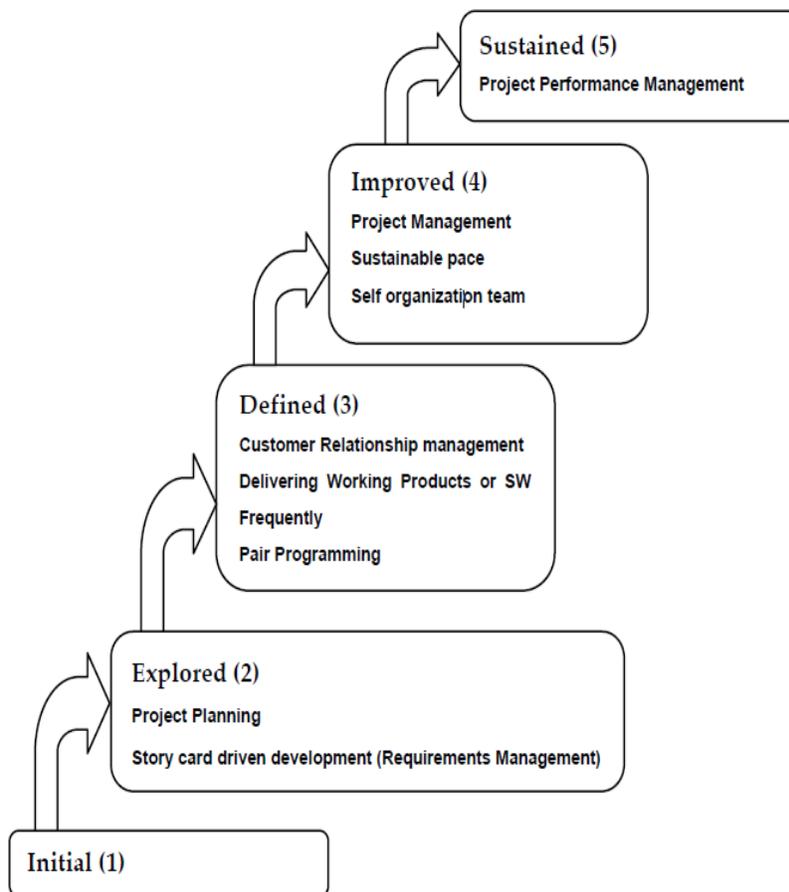


Figure 3: Five Levels of Strategic Alignment Maturity.

BIA-Indicators Affected by Agile Software Development

The use of agile software will be measured by the agile maturity. A maturity level of at least 2 or 3 will be regarded as substantial evidence for the use of agile software development methods. Use of agile software developments within a business unit is expected to result in a higher business-IT alignment maturity score as opposed to the BIA maturity of a business unit that does not use agile software.

FRAMEWORK VALIDATION

The framework was applied at a not-for profit organization in the Netherlands. The company consists of about 600 employees and has its own IT-department for software development. During the application of this framework the company started to use of the agile software development methodology on a small scale. In this period both agile and non-agile methods were used. Our main validation criterion was applicability for the organization, because of the potential value of the use of agile software development in bridging the gap in business-IT alignment.

To determine the agile maturity, an interview including a questionnaire was used. The interview was conducted with the scrum-master associated with the agile-projects. The questionnaire was an operationalization of the Agile Maturity Model (AMM) by Patel and Ramachandran (2009). The questionnaire will result in a rating per Key Process Area (KPA). We decided to calculate maturity in the same way as Patel and Ramachandran (2009) to be able to compare results.

$$\frac{\sum (Y_n) + \frac{1}{2} \sum (P_n) * 100}{\sum (T_n) - \sum (NA_n)}$$

Where Y_n = Number of 'YES' answers
 P_n = Number of partially answers
 T_n = Total number of questions
 NA_n = Number of n/a answers

Patel and Ramachandran (2009) assess agile maturity on the basis of the resulting percentage of this formula, the so-called KPA-rating. They consider different levels of achievement: fully Achieved: 86% to 100% there is evidence of a complete and systematic approach to and full achievement of the defined key practices in the assessed KPA. No significant weaknesses exist across the defined organization unit. Largely Achieved: 51% to 85% there is evidence of sound systematic approach to and significant achievement of the defined key practices in the assessed KPA. Performance of the key practices may vary in some areas. Partially Achieved: 16% to 50% there is evidence of sound systematic approach to and achievement of the defined key practices in the assessed KPA. Some aspect of achievement may be predictable. Not Achieved: 1% to 15% there is little or no evidence of achievement of the defined key practices in the assessed KPA.

Four indicators are expected to be effected by the use of agile software development: communications, Competency & Value Measurements, Governance and Partnership. In order to

assess the BIA-maturity of these four indicators, two categories of employees were identified: agile-users and non-agile-users. Within these two categories a questionnaire was conducted by both business and IT-employees. The questionnaire we used is developed by Luftman and Kempaiah (2007) operationalizing six BIA-indicators. We extracted the 25 questions measuring the four BIA-indicators relevant to our investigation. The measurement will result in a maturity level per indicator. The maturity level can reach from level1 (initial-ad-hoc) to level5 (optimized Process).

RESULTS

The agile maturity was assessed for AMM level 3. Level 3 consists of 7 Key Process Area's (KPA'S). The average capability level of these KPA's resulted in a KPA-rating of 53%. Based on this rating we can conclude that in average a level 3 maturity is largely achieved, if it is allowed to use the KPA-rating in such a small sample. There is evidence of sound systematic approach to and significant achievement of the defined key practices in the assessed KPA's. Performance of the key practices may vary in some areas.

The results of the questionnaire to determine the BIA-maturity are presented in table 3. This table shows the BIA-score on the 4 indicators divided by 4 groups: Business and IT both divided in agile and non-agile users. These results are only indicative. Especially the agile and non-agile means cannot be compared with a full scale Luftman/Kempaiah BIA-assessment because we omitted two BIA indicators.

	Business		IT		Business and IT	
	Non-Agile	Agile	Non-Agile	Agile	Non-Agile	Agile
Communications	2,78	2,92	2,33	2,67	2,67	2,79
Competency & Value Measurements	2,69	3,00	1,86	2,43	2,36	2,73
Governance	2,86	3,21	3,29	2,86	3,00	3,06
Partnership	3,00	3,30	2,40	3,10	2,89	3,20

Table 3: BIA-Maturity.

Based on these results the effect of the use of the agile software development method on the business-IT alignment maturity can be derived. In Figure 4 the maturity per BIA-indicator is presented for both agile and non-agile users. The figure shows a higher maturity on all four BIA-indicators.

All BIA-indicators show a slight difference between IT and business. IT-people tend to evaluate business-IT alignment somewhat less favorable than business-employees, with the exception of governance. The difference in evaluations pertains in agile and non-agile circumstances.

Taking a closer look to the BIA-indicator 'governance' (Figure 4) we will find minimal improvement: 0.06. Table 3 shows improvement on indicator for business-employees using agile. However the score on governance shows a lower score for IT-employees using agile.

Although there is a lower maturity on the BIA-indicator ‘governance’ for IT-employees, the average on this indicator is higher for agile-users. Since the difference on this indicator is minimal, there is no substantial improvement. The other indicators however show higher maturity for agile-users.

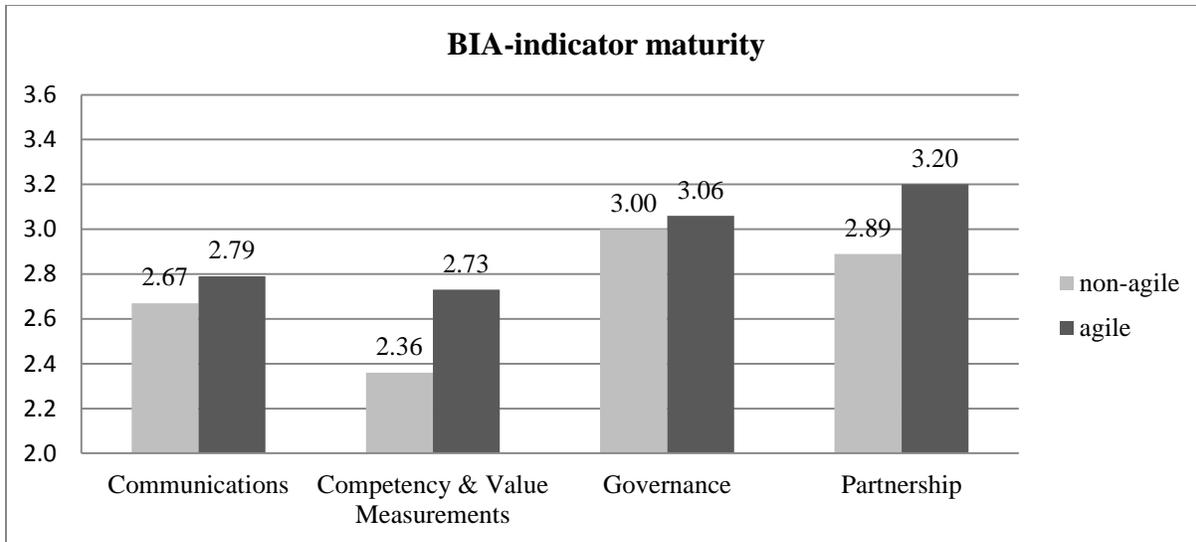


Figure 4: BIA-Improvement.

CONCLUSION AND FURTHER RESEARCH

This research shows a difference between the use of the agile software development method Scrum and the business-IT alignment maturity. The hypothesis that the use of agile software development methods results in a higher business-IT alignment maturity is consequently not to be rejected. The BIA maturity factors communication, partnership and competency and value measurements indicate a higher value in the business unit using agile than in the Business unit not using agile. The BIA maturity factor governance shows no substantial difference. We conclude that the use of agile software development can contribute to business-IT alignment of organizations; consequently organizations can apply an agile software development method in order to strive for business-IT alignment. Evidence however is limited. The research was conducted on a small population within one not-for profit organization. More research is necessary within more organizations to support the research question. The effect of agile software development in this research is based on a comparison of two groups: one group using agile software, the other group does not. Further research is recommended measuring BIA maturity in the same group before and after the use of agile software development methods. Research in this paper has been restricted to the use of the agile software development method scrum. It would be of interest to invest other agile software methods such as Extreme Programming or DSDM.

Further research is recommended to differentiate the factors within agile software development which influence the different factors of business-IT alignment, also taking in consideration the business-IT alignment factors that have not been measured in this research (Skills and

Architecture). Of interest also is the question why the Governance factor of BIA maturity was not influenced to a degree as was expected.

The research results however are encouraging for further investigation of the hypothesis that the use of the agile software development method can contribute to business-IT alignment.

REFERENCES

- Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., . . . Marick, B. (2001). *Manifesto for agile software development*. Retrieved December 9, 2011, from <http://www.agilemanifesto.org/>
- Derksen, B., & Luftman, J. (2011). Persbericht: Trends in IT. Retrieved December 11, 2011, from <http://www.bitti.nl/nieuws/actueel-nieuws/trends-in-it-&-informatiemanagement>
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*(9-10), 833-859. doi: 10.1016/j.infsof.2008.01.006
- Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal, 32*(1), 4-16.
- Hossain, E., Babar, M. A., & Paik, H. -Y. (2009). Using Scrum in global software development: A systematic literature review. In *Fourth IEEE International Conference on Global Software Engineering*, 175-184. doi: 10.1109/ICGSE.2009.25
- Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly, 34*(1), 87-114.
- Luftman, J., & Kempaiah, R. (2007). An update on business-IT alignment: "A line" has been drawn. *MIS Quarterly Executive, 6*(3), 162-177.
- Maes, R. (1999). A generic framework for information management (PrimaVera Working Paper 99-03). Retrieved from <http://imwww.fee.uva.nl/~pv/PDFdocs/99-03.pdf>
- Patel, C., & Ramachandran, M. (2009). Agile maturity model (AMM): A software process improvement framework for agile software development practices. *International Journal of Software Engineering, 2*(1), 3-28.
- Silvius, A. J. G., de Haes, S., & van Grembergen, D. (2010). Explorative study on the influence of national cultures on business/IT alignment maturity. *International Journal on IT/Business Alignment and Governance, 1*(2), 26-45. doi: 10.4018/jitbag.2010040103
- Zhang, X., Hu, T., Dai, H., & Li, X. (2010). Software development methodologies, trends and implications: A testing centric view. *Information Technology Journal, 9*, 1747-1753.

This Page Was Left Blank Intentionally.