Why large and complex ICT and digital transformations fail?

Framework for managing success and failure factors

midaGon

Success is a Choice

BACKGROUND

Large scale systems projects have a high likelihood of failure. According to (Shaul & Tauber 2013)

a) 90% of ERP implementations are delivered late or are over budget,
b) enterprise initiatives show a 67% fail rate in achieving corporate goals and are considered negative or unsuccessful,
c) more than 40% of all large-scale projects fail.

This is indicative of a real issue that ultimately influences the vitality of organizations. To improve this situation, Midagon wanted to get an objective understanding of the reasons underlying success and failure and to find ways to increase project success rates. We, therefore, requested professor Matti Rossi from Aalto University School of Business and professor Kari Smolander from Lappeenranta University of Technology to study the success and failure factors of complex digital transformations. As part of the study, we also wanted to understand if new methodologies (namely agile) have changed the success rates and if they ensure the success of digital transformations.

Based on the findings from academic research, we introduce a Midagon framework to systematically address the success and failure factors in digital transformations. This framework can be applied in digital transformations, independent of the methodology used.

These issues are covered in three parts:

1. This first part looks at large scale system projects through academic studies. The goal is to identify critical success factors, pitfalls and why so many systems projects fail. As a result, success and failure factors are identified.
2. The second part looks at the impact of project methodologies. In particular, we study how new implementation approaches, especially agile methods and continuous delivery, can limit the risks of these endeavors. This section is based on both academic research and Midagon experience.
3. The third part introduces a Midagon framework through which
   - the existence of success and failure factors can be assessed and
   - success and failure factors can be systematically addressed throughout the project lifecycle

CONTENTS

Critical success and failure factors of systems development

What does the academic research state about factors increasing the likelihood of success or failure? In the following table, a slightly condensed list based on the studies of Somers and Nelson, 2001 and Wong et al., 2005 are presented. The lists have been very stable over the years. Therefore, they can be seen as definitive.

	Critical success factors (CSF)	Critical failure factors (CFF)
1.	Top management support and CEO sponsorship	System misfit
2.	Project team competence	High turnover rate of project team members
3.	Inter-departmental cooperation and open communication	Over-reliance on heavy customization
4.	Clear goals and objectives	Poor consultant effectiveness
5.	Project management	Poor IT infrastructure
6.	Management of expectations	Poor knowledge transfer
7.	Vendors' support	Poor project management effectiveness
8.	Data quality and conversion	Poor quality of Business Process Re-engineering (BPR)
9.	User training on software and education on new business processes	Poor quality of testing
10.	Dedicated resources	Poor top management support
11.	Enough time and resources for business process redesign	Too tight project schedule
12.	Minimal customization	Unclear concept of the nature and use of the system from the users' perspective
13.	Change management	Unrealistic expectations from top management concerning the System
14.	Use of third-party consultants	Users' resistance to change
15.	Working partnership with vendors and consultants	

Table 1: Success and failure factors

These lists are compiled based on large number of empirical studies of the success factors and are in the order of importance.

The well-known issue related to these Critical Success Factors -lists (later CSFs) is that not following the CSFs almost certainly leads to failure, whereas following the lists will not guarantee success. If all of the CSF's are in place, it means that there are good reasons to believe in successful delivery. However, even with all of the CSFs in place, the likelihood of delivering on time and budget is far less than 50 %.

There are several reasons for failure. The comprehensive review by Shaul and Tauber (2013) provides the following key reasons for the high failure rate.

- First, systems are often **promoted as a unified platform for all business processes**. Since the system involves a large portion of the organization, companies can experience difficulties in convincing employees to commit to the implementation process, who then fail to implement the system in an effective manner (Davenport 2000).
- Second, many organizations start complex projects, despite the fact that near-term success and long-term survival are difficult to predict.
- Third, most enterprise systems follow best practices, such as a maximum integration of information flows and standardization, that are **less suitable for firms that have decentralized, non-hierarchical structures and non-uniform cultures** (Mattila et al. 2017).
- Fourth, organizations increasingly find they are **obligated to accept the project outcomes that emerge from compromises** between an installed consultancy base or software vendor solutions and the local context (Wagner & Newell 2004).
- Fifth, **legacy systems and shadow systems are still being used**. (Tanriverdi et al. 2007; Mattila et al. 2017).

PART 2: IMPACT OF METHODOLOGY

Is the methodology the key to project success?

Project methodologies are an integral part of the project setup, because all companies using project methodologies expect greatly improved project performance (Joslin et al., 2014). In the following section, we go through new project management methodologies and evaluate, if the selection of the right project methodology alone, is enough to eliminate critical failure factors and to put in place critical success factors.

Traditional waterfall approaches, that have been widely used for decades, have become a target for criticism. Long planning and development cycles attributed to these approaches are not able to predict and consider the changes coming from the fast-changing and chaotic business environment. High customization is often a consequence of the unexpected changes in the environment.

If an ICT project takes two to three years from the planning table to the first go live, the business environment will have changed and the result will most likely no longer match the needs. The development process is also a learning process, where all stakeholders learn and understand the potential and the possibilities of a new solution, as opposed to having enough knowledge upfront and being able to control all of the variables in advance.

During the past 20 years, new approaches have emerged in the project management area. These methods are especially rooted in software development. However, they have been adopted more widely in ICT system implementation projects and environments. The methods referred to in this white paper primarily include Agile / SCRUM, Scaled Agile Framework (SAFe) and continuous delivery or DevOps.

The promises of agile and scaled agile

The emergence of agile methodologies is, without a doubt, the most important change in project management practices in recent decades. They are influenced by lean manufacturing in Japan but have since been developed on their own. However, elements such as Kanban boards, with their origin in lean manufacturing, are widely used by agile teams.

Agile practices emphasize direct interaction and co-located development teams, in order to solve the immediate problem of software developers working together. In the modern, fast-changing business environment, responding to change should be valued more than following a pre-defined plan until the bitter end. The customer is seen as an integral part of the development team. The benefits are obvious and, therefore, agile methods make a significant contribution to answering how development teams should work and reflect together. However, in large, complex and geographically distributed projects, the essential problems may be totally different from the idea of agile.

To address the problem of project and organization size, complexity and geographical distribution, during the past 10 years much of the progress in Agile development has occurred in distributed and large-scale contexts. Scaled Agile (SAFe) is perhaps the most prominent of the Agile methodologies that is intended to address these issues. According to Scaled Agile Inc. (2018) (*https://www.scaledagileframework.com/why-safe/*), the business benefits of SAFe are improved

quality, employee engagement, productivity, faster time to market and program execution. Indeed, SAFe and other similar methods aim at bridging the gaps between how one co-located development team operates and how to operate agile on an organizational level.

Whether Agile works or not, is difficult to study. How much of the success can be attributed to Agile and how much to other items in the context? Large quantities of detailed and correct project data across industries is difficult to get. The development context is often extremely complex, which makes projects incomparable. For example, Clarke and O'Connor (2012) identify from the literature 48 essential factors that can have an effect on project success.

Despite this difficulty, some attempts have been made to evaluate the effects of Agile on project success. Serrador and Pinto (2015) tested the effect of Agile use in more than 1,000 projects on two dimensions of project success: efficiency and overall stakeholder satisfaction against organizational goals. The findings suggest that Agile has a positive impact on efficiency, stakeholder satisfaction, and perception of overall project performance. On the other hand, in the study by Budzier and Flyvbjerg (2013) agile methods seemed to reduce schedule risk, but not the other risks, such as benefit or cost risk.

Some studies have found Agile approaches to be positive for health and well-being at work (e.g. Syed-Abdullah et al. 2006) through enthusiasm and autonomy. Other studies have found that Agile increases stress in the workplace (Laanti 2013). Laanti explains that it is management's responsibility to create a working environment that is suitable for Agile. If that does not happen, increased stress will be the result.

The move from waterfall to agile is itself a major transformation. The challenges of such a transformation should not be overlooked. Since organizations, their practices and culture are quite different, there is no single concrete recipe for doing agile transformation successfully. Agile and large-scale agile methodologies are frameworks. Each organization must design how to apply it in their own organizational context and how to transition from the old way-of-working to the new. A step-wise approach to transition from waterfall to Agile is possible and often recommended.

Dikert et al. (2016) point out that in a large organization, contexts beyond development must be involved in the transformation. Functions ranging from marketing and sales to human resources also need to align with agile. If this does not happen, it might cause trouble for the transformation. To enable successful large-scale agile deliveries (for example by using SAFe), a major top-down agile transformation needs to take place on the enterprise level – from top management sponsorship to individual development teams. A common failure factor for agile deliveries is the mismatch between organizational culture and delivery methods: if the underlying structures are based on waterfall types of processes, then the odds for efficiently leveraging the benefits of agile methods are limited.

In many organizations that use waterfall, development teams are used to doing several projects simultaneously with partial allocations to each project. When moving to agile ways of working, these resourcing principles need to be fundamentally changed. Agile methods should provide more flexibility for parallel development. However, the principles for work allocation and resourcing can be very different from the waterfall type of deliveries. Agile teams and release trains are typically specialized around certain competence areas and, therefore, a major transformation may require a comprehensive reallocation of ongoing tasks and projects.

Continuous delivery

While Agile development is iterative and gradual, modern software applications and information systems are also expected to be continuously developed, enhanced and delivered. Continuous Delivery (or DevOps) is an extension to agile, with short iterations and emphasized interaction. However, the more profound difference is that continuous delivery is a change from time-limited project delivery to a continuous activity that is happening in parallel with business development. In this regard, continuous delivery goes beyond the definition of a project, which, by its nature, is a temporary and time-limited process with progressive elaboration towards a goal. However, many of the common components of continuous delivery (for example, short build cycles and test automation) can be implemented under any development methodology (and provide similar benefits as with agile methods).

Continuous delivery is important, as companies build digitalized capabilities and services, which continuously interact with the company's

customers. In this future environment, Fitzgerald and Stol (2017) point out that the systems development operations and business contexts cannot be as separate as they used to be. Their continuous nature goes together through continuous planning and budgeting (biz) to continuous integration and development (dev), and to continuous improvement and experimentation (ops).

The other major reason of using Continuous Delivery goes back to the "non-functional" needs that may emerge suddenly and require immediate changes to systems. These non-functional needs may include information security updates and technical upgrades to fix patches. Atlassian writes: "In the DevOps community, those with Agile experience acknowledge that SCRUM is useful for tracking planned work. Some work in operations can be planned: releasing a major system change, moving between data centers, or performing system upgrades. However, much of the work of operations is unplanned: including performance spikes, system outages, and compromised security. These events demand an immediate response. There's no time to wait for the items to be prioritized in a backlog or for the next sprint planning session. For this reason, many teams that have come to embrace DevOps thinking, look beyond Scrum to Kanban. This helps them track both kinds of work and to understand the interplay between them. They may also adopt a hybrid approach, often called Scrumban or Kanplan (kanban with a backlog)." For more information about this side, please see: *https://www.atlassian.com/agile/devops*

Holmström-Olsson et al. (2012) provide a process for adopting continuous delivery. They propose a ladder model, with actions that have to be taken in a certain order: 1) from traditional to agile R&D; 2) from agile R&D to continuous integration; 3) from continuous integration to continuous deployment; and 4) from continuous integration to innovation system including continuous experimentation.

When continuous delivery permeates business, development and operations, the lifecycle principle of projects and software and systems development is really challenged. The project is no longer the (only) organizing principle of development. At the same time, many organizations will find that the nature of digitalized services, which directly interact with company's customers require nothing less.

Takeaways and lessons learned from agile and continuous development

Academic research does not provide a definitive answer on the impact of emerging methodologies on digital transformation success. Agile methodologies have important positive effects on transformation outcomes, but they do not give a universal answer to handling success and failure factors. There are important takeaways to be acknowledged.

Agile methods' most important principle is that responding to change is more important than following the plan. More profound planning does not solve the problems, because the business environment is continuously changing. This requires continuous learning. Many of the current development approaches are based on this urgent need to be able to respond and be flexible. Agile and continuous practices may help in making objectives and conflict resolution more open and transparent. Short iterations and continuous deliveries keep developers and other stakeholders more aware of risks, related to conflicting objectives.

However, complex and large ICT projects are not only about systems development. Business cases need to be calculated and business benefits realized, sometimes long after the deployment phase itself has ended. They often involve selecting and buying new technology and selecting implementation partners. They change underlying business processes, alter peoples' roles and responsibilities and require people to obtain new skills. In the case of new product or service introduction, marketing and sales must time their go to market efforts.

Simply put, they involve a substantial amount of change which must be managed in a structured way to succeed. This kind of broad change cannot be managed simply based on prioritized product backlog and visibility of the next three sprints. The same requirement to plan and be structured applies to deploying changes. This is at least equally important and requires, since deployments require careful planning upfront, e.g. what resources are allocated to this activity, what kind of organization model supports change deployments etc.

Different business problems or projects require different solutions. There is no one right way of doing everything. Each methodology has its advantages and disadvantages. If the expected

project outcome is well-known in advance or can be predicted with high accuracy, waterfall is well-suited. On the other hand, if there is considerable uncertainty about the expected outcome, e.g. what will customers value, Agile is the obvious choice. When digital services development becomes an ongoing process, with no clear start or end, Continuous Delivery methods like DevOps will possibly provide the best tools for continuous and synchronized business development and technical development execution. Finally, there is a possibility to combine Waterfall and Agile into what we call a hybrid methodology.

Combining Waterfall with Agile is typically called Disciplined Agile. This is often used in large ICT system implementations, since they require certain logical and progressive paths to be followed (e.g. SAP and IBM Maximo). For more information, please see *http://www.disciplinedagiledelivery.com/*. There are also two rather well-known hybrid models that combine the "good stuff from SCRUM and the Continuous Delivery from Kanban / DevOps: Scrumban and Kanplan. *https://www.atlassian.com/agile/kanban/kanplan*. In this white paper, our focus is on the combination of Waterfall and Agile.

Especially in complex and large ICT projects, many organizations would benefit greatly by combining the best parts of waterfall and agile methods. Hybrid methodology takes the structured planning elements from waterfall methodology to establish a milestone or gate-based approach. Various practices from agile methods, like shorter development cycles, are then added to the framework: within the milestones each phase of the project or program can be structured into sprints, e.g. design sprints and system development sprints. Other agile practices to be adapted, are typically related to DevOps, team collaboration and measurement. Finally, the overall solution is tested end to end and deployed or launched in desired phases or sequence.

There are main benefits of using Agile as a key component of hybrid methodology.

- First, Agile provides more built-in tools and processes on how the team should work together on a day-to-day basis. Waterfall methods have less focus on these things, as they mainly assume that a phase has a starting and end point, but how work gets accomplished is another problem. In many complex ICT projects, shared understanding and shared purpose is becoming increasingly important.
- Second, Agile provides more tools for ensuring that there is transparency and visibility to progress across all the levels of organization. It provides more flexibility to adjust the plan as the team and key stakeholders learn what adds value, as the business environment changes. The challenge will be to "slice and dice the elephant" into a minimum viable product that can be further built on. Especially when teams are geographically distributed in several locations, building co-located agile teams is still a task to be resolved and may require restructuring or re-purposing existing teams.

Success and failure factors are largely universal and independent of the methodology chosen. The selection of the most suitable methodology offers a partial solution. However, it does not systematically address all of the success and failure factors identified by academic research. An over-focus on methodology easily creates biases that methodology solves everything or is the answer to existing problems. In many organizations, the discussion about methodology is dogmatic. The old methodology is bad, while the new methodology is good. This is not the case and over time organizations learn that each methodology has its pitfalls and strengths.

PART 3: FRAMEWORK FOR SUCCESS

In the previous section, we observed that new methodologies bring elements that positively affect the success of digital transformations and that the optimal methodology is dependent on the context where it is being applied in. We also learned that the selection of optimal methodology alone does not ensure success.

Understanding the success and failure factors

Due to the limitations of available methodologies, a systematic approach is needed to address the success and failure factors identified by the academic research. A knowledge of these factors alone is pointless, unless it leads to actions. What should then be done to:

- Put the missing success factors in place and to strengthen existing ones, and to
- Eliminate or mitigate the failure factors

First, one needs to understand what success and failure factors mean in practice. The number one success factor "Top management support and CEO sponsorship" can be used as an example.

In the various projects we have run, the level of top management support has varied greatly. Still, the top management has typically believed that the support they are giving is on a good level, even when this is not actually the case. A better way to objectively assess whether the success and failure factors are properly addressed, is needed.

The following table provides a framework for assessing the maturity of top management support and CEO sponsorship. It describes typical indicators from Stage 1 with limited or no support to Stage 5 with excellent support.

Maturity assessment: Typical indicators of stage				
1.	2.	3.	4.	5.
Top management and CEO are not aware of the project.	Top management and CEO are aware of the project but do not actively support it.	Top management and CEO are aware of the project. Some executives support it, but not all.	Top management and CEO have approved the initiation of the project.	Top management and CEO initiated the project.
There is a conflict between the organization's strategic intent and project objectives.	There is no strategic fit between organizations' strategic intent and the project objectives.	There is some strategic alignment between the project and the organization as a whole, but it is not obvious.	Project is aligned with top management strategic intent.	There is a strong strategic fit between project and organization as a whole.
Top management is not involved in project planning or steering the project.	Top management is aware of project planning and progress but does not contribute to it.	Top management comments on project planning and progress.	Top management and CEO are involved in project steering group.	Top management and CEO are involved in project steering group and daily operations.
There is no connection between expected project outcomes and top management incentives.	There is limited connection between expected project outcomes and top management incentives.	There is some connection between expected project outcomes and top management incentives.	There is a strong connection between expected project outcomes and top management incentives.	There is very strong connection between expected project outcomes and top management incentives.
			Top management publicly support the project.	Top management very visibly and publicly support the project.
			Top management contributes to project planning and progress.	Top management directs project planning and progress.

Table 2: Example of maturity assessment – Top management support and CEO sponsorship

Similarly, the maturity of each of the other success and failure factors can be assessed. Midagon has created a maturity assessment framework systematically covering the success and failure factors.

Maturity assessment serves two purposes:

1) A low maturity score for an individual factor highlights a potential problem. Areas to improve can be selected and corrective actions taken, if the score is lower than expected.

2) Total score – whether low or high - gives an indication of the overall likelihood of success and failure. One should consider postponing project and taking corrective actions, if the total score is lower than expected.

EMBEDDING SUCCESS INTO METHODOLOGY

Once success and failure factors are understood, a systematic approach is needed to ensure that they are proactively acted on. Our point of view is that success and failure factors should be addressed in and be an integral part of organizations' project methodologies. We also see that these factors are universal i.e. independent of the methodology applied.

Let us use Midagon's project management methodology as a reference point. It includes the following phases:

- Initiation
- Project planning
- Execution
- Stabilization
- Project closing
- Business as usual

Using these project phases as a basis, the table below indicates in which phase the maturity of each of the success and failure factors should be evaluated. It is important to acknowledge that maturity assessment is not a one-off exercise. Maturity may change over time for a better or worse. The assessment is needed in each of the phases, where the factor in question is relevant.

The responsibility for assessment and the forum in which the results are gone through, varies depending on the methodology. The evaluation can be done – for example – at project check points or in sprint reviews.

CSFs/CFFs		Initiation	Project planning	Execution*	Stabili-zation	Project closing	Business as usual
Critical Success Factors	1. Top management support	x	x	x	x	x	x
	2. Project team competence		x	x	x		
	3. Cooperation and open communication across organization	x	x	x	x	x	x
	4. Clear goals and objectives	x	x	x	x	x	x
	5. Effective project management		x	x	x		
Critical Failure Factors	1. System misfit	x	x	x	x	x	x
	2. High turnover of team members		x	x	x		
	3. Over-reliance on heavy customization		x	x	x		
	4. Poor consultant effectiveness		x	x	x		
	5. Poor IT infrastructure		x	x	x	x	x

Table 3: A subset of the success and failure factors identified in the study done by Rossi and Smolander.

As a result, we have a capability to continuously and systematically monitor the maturity of key success and failure factors. Deviations can be quickly identified, and corrective actions taken. Follow-up ensures that the impact of corrective actions is measured and understood.

An additional benefit for organizations with large project portfolios, is the accumulation of data on maturity scores and success of projects. This allows organizations to learn more about the relevance of different factors.

CONCLUSIONS

It is important to keep your eye on the ball. Based on our experience, critical success and failure factors identified by the academic research are not systematically addressed in the project methodologies applied by the organizations. The likelihood of success can be easily improved by solving this issue.

We propose methodologies and acceptance criteria to be revised. A systematic and regular approach for evaluating the critical success and failure factors is needed. Regardless of the chosen methodology, pay close attention to securing success factors and removing failure factors. You should also make sure that you apply the project methodology prudently to the project at hand.

Reported failures of large-scale, complex systems development and contemporary approaches to development, agile and continuous development emphasize that most problems underlying success and failure factors are managerial. They relate to the way of working, communicating and interacting across stakeholders. We recommend that you specifically focus on these aspects.

A successful program requires strong and visible sponsorship from the top management. The leaders' role is vital in creating success. The company's executives, who are sponsors, need to create premises for success. In many cases, management's objectives destroy any chance of success, by setting overly ambitious targets.

An experienced and competent program manager is essential. Leading a project is different from administration. Project plans need to be created and progress tracked, which is administration. In a small and trivial project, administration may be enough. The bigger the project and more complex it is, the more it requires leadership. People need to be led and subject matter expertise is crucial.

Having the right team is also critical, while the cost of an incompetent team can be detrimental. If you have the wrong team on the project, it is possible that nothing will save the project. On the other hand, when everything fails in a project, it is the key individuals who will rise to the challenge and create success. This has been proven in many projects.

Great results are seldom realized without inspiring leadership, a high performing team and a vision of the future or outcome of the project. Take as an example, sports teams and teams that win championships. You seldom hear them saying that the key success factor was tactics, a playbook or well-managed weekly practice routines. It is true that sometimes a team invents new tactics that had not been thought of before. However, other teams quickly learn and catch up. Exceptional teams are made up of individuals who want to contribute to each other's and their team's success. This is called team spirit. They typically have coaches who find ways to put the right players into the right roles, unlock the true potential of the team and go beyond the potential. Truly exceptional teams grow together. Creating and leading high performing project teams isn't that much different.

To conclude:

1) Select an optimal methodology for the context in question,
2) actively and systematically manage success and failure factors in every stage of the transformation and
3) focus on people. This is what success depends on.

REFERENCES

BUDZIER, A. & FLYVBJERG, B., 2013. Making Sense of the Impact and Importance of Outliers in Project Management Through the Use of Power Laws. Proceedings of IRNOP, 11.

CLARKE, P. & O'CONNOR, R.V., 2012. The situational factors that affect the software development process: Towards a comprehensive reference framework. Information and Software Technology, 54(5), 433–447.

DAVENPORT, T. 2000. Mission Critical: Realizing the Promise of Enterprise Systems, Boston, Massachusetts, Harvard Business School Press.

DIKERT, K., PAASIVAARA, M. & LASSENIUS, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software, 119, 87–108.

FITZGERALD, B. & STOL, K.-J., 2017. Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123, 176–189.

HOLMSTRÖM-OLSSON, H., ALAHYARI, H. & BOSCH, J., 2012. Climbing the "Stairway to Heaven" -- A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In 2012 38th Euromicro Conference on Software Engineering and Advanced Applications. 2012 38th Euromicro Conference on Software Engineering and Advanced Applications. 392–399.

JOSLIN, R. & MÜLLER, R. (2014). The impact of project methodologies on project success in different contexts. Paper presented at Project Management Institute Research and Education Conference, Phoenix, AZ. Newtown Square, PA: Project Management Institute.

LAANTI, M., 2013. Agile and Wellbeing -- Stress, Empowerment, and Performance in Scrum and Kanban Teams. In 2013 46th Hawaii International Conference on System Sciences. 2013 46th Hawaii International Conference on System Sciences. 4761–4770.

MATTILA, M., NANDHAKUMAR, J. & ROSSI, M. 2017. Balancing of fluid and cemented routines in Enterprise Systems.

SHAUL, L. & TAUBER, D. 2013. Critical Success Factors in Enterprise Resource Planning Systems: Review of the Last Decade. ACM Computing Surveys, 45, 55- 55:39.

SERRADOR, P. & PINTO, J.K., 2015. Does Agile work? — A quantitative analysis of agile project success. International Journal of Project Management, 33(5), 1040–1051.

SOMERS, T. & NELSON, K. The Impact of Critical Success Factors across the Stages of Enterprise Resource Planning Implementations. Hawaii International Conference on Systems Sciences, 2001. 1-10.

SYED-ABDULLAH, S., HOLCOMBE, M. & GHEORGE, M., 2006. The Impact of an Agile Methodology on the Well Being of Development Teams. Empirical Software Engineering, 11(1), 143–167.

TANRIVERDI, H., KONANA, P. & GE, L. 2007. The choice of sourcing mechanisms for business processes. Information Systems Research, 18, 280-299.

WAGNER, E. L. & NEWELL, S. 2004. 'Best' for whom? the tension between 'best practice' ERP packages and diverse epistemic cultures in a university context. Journal of Strategic Information Systems, 13, 305-328.

WONG, A., SCARBROUGH, H., CHAU, P. & DAVISON, R. 2005. Critical failure factors in ERP implementation. Pacis 2005 Proceedings, 40.

Lauri Eskola

CEO
+358 50 484 5662
lauri.eskola@midagon.com

Ilkka Töyrylä

VP Business Transformation Services
+358 45 12 696 12
ilkka.toyryla@midagon.com

midaGon

Keilaranta 1
02150 Espoo

+358 45 126 9600
Business ID: 2058234-3