

Analyzing the Trade-Off between Traditional and Agile Software Development – A Cost/Risk Perspective

Patrick Afflerbach¹, Simon Kratzer¹, Maximilian Röglinger², and Simon Stelzl¹

¹ University of Augsburg, FIM Research Center, Augsburg, Germany

² University of Bayreuth, FIM Research Center, Bayreuth, Germany

{patrick.afflerbach, simon.kratzer, maximilian.roeglinger,
simon.stelzl}@fim-rc.de

Abstract. Digitalization heralds a new era of enterprise IT. It challenges CIOs to find a balance between renovating legacy IT and seizing the opportunities of digital technologies to keep up with competitors and start-ups. This requires organizations to operate two software development modes simultaneously: the traditional and the agile mode. Despite substantial research on both modes, little is known about whether to implement distinct software development projects traditionally or agile. As a first step to addressing this gap, we propose a quantitative decision model that compares the cost and risk profiles of both modes associated with the implementation of a distinct project. The decision model integrates qualitative and quantitative characteristics of the project in focus and of the traditional and the agile mode. As for evaluation, we implemented the decision model as a software prototype and validated its behavior using sample projects as well as a sensitivity analysis.

Keywords: Software Development, IS/IT Management, Agile Methods, Decision Model.

1 Introduction

Digitalization is transforming many industries [1]. Digital technologies enable innovative business models and provide novel revenue opportunities. In their yearly CIO Survey, Gartner [2] relate digitalization to the third era of enterprise information technology (IT). While the first era refers to the initial implementation of corporate information systems, the second era focused on improving reliability and stability. Currently, new technological trends are shaping the third era in a way most CIOs are not prepared for [2]. There is a tension between “doing IT right [...], doing IT safely [...] and working the plan” and “doing IT fast, [...] doing IT innovatively [...] and adapting” [2].

Although the third era of enterprise IT requires different architectures, processes, and governance, it mostly impacts software development (SD) [1]. To tackle the challenges related to the tension mentioned above, researchers and industry experts suggest operating multiple SD modes within a company [1, 3]. The two most common modes are the traditional (TSD) and the agile SD mode (ASD) [2]. TSD is about upfront planning and maximizing control, ASD about risk-taking and dealing with uncertainty to

13th International Conference on Wirtschaftsinformatik,
February 12-15, 2017, St. Gallen, Switzerland

Afflerbach, P.; Kratzer, S.; Röglinger, M.; Stelzl, S. (2017): Analyzing the Trade-Off between Traditional and Agile Software Development - A Cost/Risk Perspective, in Leimeister, J.M.; Brenner, W. (Hrsg.): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017), St. Gallen, S. 439-453

create innovation at a high pace [4]. Both research and industry call for operating TSD and ASD in parallel. Thus, there is a shift from talking about why and how to do agile toward finding an appropriate trade-off between TSD and ASD.

Most approaches that support deciding about the appropriate SD mode for a distinct SD project (SDP) rely on qualitative analysis [3, 5]. Although there is research on quantifying the characteristics of both modes, there are, to the best of our knowledge, no approaches that quantitatively analyze the decision between TSD and ASD with respect to the implementation of a distinct SDP [6]. Thus, we investigate the following research question: *How can organizations decide whether to implement a distinct SDP in TSD or ASD mode regarding a cost/risk perspective?*

As a first step to answering this question, we adopted the design science research (DSR) paradigm and propose a quantitative decision model, taking a cost/risk perspective [7]. Our model incorporates characteristics inherent to the SDP in focus as well as characteristics specific to TSD and ASD mode. Mapping project characteristics to the cost/risk profiles of both SD modes, our decision model addresses the TSD/ASD trade-off for a distinct SDP. Thereby, our model does not strive for exact estimations of an SDP's business value. Rather, it aims to compare the economic effects associated with both SD modes. We thus take a tactical perspective on the TSD/ASD trade-off instead of a budget planning and cost estimation perspective.

Following the DSR reference process [8], we first provide theoretical background on both SD modes as well as on cost, benefit, and risk management for SDPs. We also derive design objectives that characterize proper solutions to our research problem (*objectives of a solution*). We then introduce the design specification of our decision model (*design and development*), before we report on our evaluation activities so far (*evaluation*). We conclude by summarizing key results, reviewing our limitations, and suggesting topics for further steps to answer this research question.

2 Theoretical Background and Design Objectives

2.1 Traditional and Agile Software Development

SD involves activities grouped into four phases, i.e., specification, design, implementation, and testing [9]. Although all SD methods follow these phases, they differ in how and how often these phases are traversed. TSD mode is sequential or iterative, while ASD is evolutionary or iterative and incremental [10]. The philosophy of TSD is plan- and process-driven as well as formal and tool-oriented [11, 12]. Aitken and Ilango [3] list methods for TSD mode. Methods related to TSD mode rely on the upfront planning of requirements, documentation, and long development cycles [11, 13]. This makes TSD mode beneficial for projects whose scope and features are known upfront. Defects or changes in scope or requirements, however, result in expensive iterations of previous phases [9]. To outline the characteristics of TSD, we use the waterfall model, which sequentially traverses the four SD phases without any iteration or repetition [14].

Due to long development cycles and rising demand for more flexible methods, ASD mode evolved in the 1990s. In 2001, experienced software practitioners issued the Agile Manifesto [15]. The idea behind ASD is the prioritization of individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [15]. Dybå and Dingsøy [13] as well as Lee and Xia [10] provide an overview of methods related to ASD mode. A well-known ASD method is SCRUM. The essential part of SCRUM is a sprint, an iteration that may last from one day to four weeks. After each sprint, there is a working software artifact ready to be released. ASD methods are primarily characterized as being able to react to changes in requirements and to encourage developers' creativity and autonomous work. As a disadvantage, ASD heavily relies on trustful relationships among customers, developers, and managers as well as a high amount of self-reliance of all participants. Furthermore, ASD often faces problems in a highly regulated and plan-driven environment [3]. Aitken and Ilango [3], Conboy [11], as well as Nerur et al. [12] compare TSD and ASD mode according to mode-specific characteristics.

The argumentation above revealed structural differences between TSD and ASD when implementing an SDP. These differences make the two modes differently suitable for specific projects. Thus, decisions about the appropriate SD mode must account for these differences. Against this background, we define the following design objective:

(DO.1) *Differences of TSD and ASD*: The decision on the appropriate SD mode must account for structural differences of TSD and ASD mode.

2.2 Costs and Risk in Software Development Projects

The dynamic environment in the third era of enterprise IT challenges managers to choose among different SDPs and to decide in which mode to implement these SDPs. In the last decade, researchers and practitioners proposed success criteria and frameworks for project valuation [16]. Techniques that help quantify an IS/IT project's value fit into five categories: net present value (NPV) methods, rate of return methods, ratio methods, payback methods, and accounting methods [17]. The cost-benefit analysis, a representative of the ratio methods, has evolved into the most popular approach to the valuation of IT projects [18]. With this popular approach, the qualitative and quantitative benefits of a distinct project are compared to implementation and operating costs. Finally, a project has a positive business value, if benefits exceed costs. Besides benefits and costs, risk evolved into a third recognized dimension of cost-benefit analysis [19]. As an SDP's largest benefits realize at run time of the related software product, they are out of our decision model's scope [20].

The estimation of an SDP's implementation costs is a well-covered topic in software engineering [6]. Thus, many software cost estimation models are available. These models can be grouped into five categories, i.e., model-based, expert-based, learning-oriented, regression-based, dynamics-based, and composite approaches. Model-based approaches propose algorithms that are specifically developed for the SDP context [9]. COCOMO is a model- and regression-based algorithmic approach for estimating the

costs of projects conducted in TSD mode based on input parameters such as project size, scaling factors, and effort multipliers [21]. As a widely recognized approach to software cost estimation, we use the improved version COCOMO II in our decision model for projects executed in TSD mode. There are also cost estimation models for projects conducted in ASD mode. Cao et al. [22] suggest dynamics-based methods that can be adjusted to changing requirements. Yang et al. [23] as well as Benediktsson et al. [24] modify COCOMO II for ASD mode, adjusting the number and length of iterations as well as by incorporating additional efforts. As COCOMO II and the extensions proposed by Benediktsson et al. [24] use the same input parameters, their recommendations are comparable. Thus, we rely on these approaches to consistently compare the costs for implementing an SDP in TSD and ASD mode.

In general, risk is an uncertain event or condition that, if it occurs, has a positive or negative effect on a project objective [20]. In our study, we focus on IT project risks, as we investigate SDPs from a stand-alone perspective and focus on the implementation phase. Studies about IT project risk are provided by Schmidt et al. [25], Boehm [26], Bannerman [20]. Referring to these studies, the most important risks are the following ones: lack of top management support, misunderstanding of requirements, changes in requirements, lack of user involvement, and lack of adequate project staff. We aggregate these risks into three risk types that represent the major impacts of the above-mentioned risk factors on an SDP, i.e., the risk of requirement changes, the risk of delay, and the risk of defect. In our model, we adjust an SDP's implementation costs by risky costs originating from the risk types just introduced [27].

The preceding reasoning underscores the relevance of an integrated cost/risk-perspective when deciding about the appropriate SD mode for a distinct SDP. An integrated perspective is not only necessary as for economic factors, but also when it comes to the comparison of both SD modes. We define the following design objective:

(DO.2) *Integrated economic valuation*: The decision on the appropriate SD mode requires considering economic dimensions (e.g., costs and risks) in an integrated manner. This decision requires ensuring the comparability of both modes.

3 Design Specification

Our decision model systemizes and combines factors relevant for determining the appropriate SD mode for a distinct SDP. Thus, our unit of analysis is the implementation of a distinct SDP. In our basic setting, the management has already strategically decided to implement the SDP, i.e., the benefits outweigh costs and risk independent from the chosen SD mode. Only the decision about the appropriate SD mode is open. We assume that the organization has the capabilities and expertise to conduct the SDP in focus in both modes (Assumption 1). In our decision model, this operative decision is based on the costs and the risks associated with TSD and ASD mode. Although researchers point to qualitative factors beyond a cost/risk perspective, we focus on their economic impacts and model them implicitly via the costs and risks effects of the SDP in focus [28]. As stated, we abstract from benefits as they materialize at run time. For the same reason, we focus on those costs and risks that accrue during the SDP's implementation.

The decision about the appropriate SD mode involves two major challenges. First and inherent to almost all economic decisions, the decision requires resolving trade-offs. In our TSD/ASD trade-off, the advantages and drawbacks of TSD mode face those related to ASD mode. Second, the decision is multi-dimensional. In order to address the TSD/ASD trade-off, the involved decision-makers must integrate quantitative and qualitative factors. We address these challenges by integrating these factors and evaluating their effects economically from a cost/risk perspective.

3.1 General Setting

We first introduce the general setting of our model and outline the TSD/ASD trade-off. We also define the model's objective function. From an economic perspective, we can break down the TSD/ASD trade-off into differences regarding the cost/risk profiles of an SDP in both SD modes. There are projects for which the cost/risk profiles dovetail, with the TSD/ASD trade-off clearly tending to one side. We refer to such SDPs as TSD- or ASD-type projects. TSD is the preferable SD mode if the cost and risk advantages of TSD exceed those of ASD – and vice versa. There are also projects whose profile cannot be unambiguously assigned to either type.

To facilitate a structured presentation of the decision model, we first consider costs and risks separately, before integrating them. The deterministic base costs C are defined as those costs associated with the SDP's implementation. We model the base costs by using the cost estimation models COCOMO II by Boehm et al. [21] for TSD mode and the extensions of Benediktsson et al. [24] for ASD mode. Furthermore, we define the risk R as additional costs taking the risks of defects, requirement changes, and delays into account. Basically, there are two reasons why we chose COCOMO II for modeling base costs: First, COCOMO is popular and well-known among researchers and practitioners, which is why it increases the comprehensibility of our model [9]. Second, COCOMO II and the extension by Benediktsson et al. [24] ensure comparability as both methods use identical input parameters. In fact, our aim is to offer decision support on the comparison of TSD and ASD for a distinct SDP, and not to estimate costs as accurately as possible.

To evaluate the SDP's costs and risk depending on the chosen SD mode, we use the certainty equivalent method. The certainty equivalent method is an acknowledged approach for corporate decision-making in risky situations [29]. It has also proven to be useful for decision-making in IS/IT projects as well as in IS/IT project portfolio management [16, 30]. The certainty equivalent represents the certain amount of money that creates the same subjective utility for the involved decision-makers as the corresponding risky situation [27]. Applying the certainty equivalent enables comparing the deterministic base costs C and the risky additional costs R . For decision-makers with a constant absolute risk aversion α , measured via the absolute Arrow-Pratt measure, and for a normally distributed risk with expected value μ_R and variance σ_R^2 , the certainty equivalent of a distinct SDP for a chosen SD mode is given by Eq. (1) [31]. As the certainty equivalent can be structurally decomposed into the expected costs and a risk premium, we refer to the certainty equivalent as risk-adjusted costs to outline both decisive components. On this foundation, we use the difference between the risk-adjusted costs that

accrue when implementing the SDP in TSD or in ASD mode as our objective function, which is shown in Eq. (2).

$$\Phi = \mathbb{E}[-(C + R)] - \frac{\alpha}{2} \cdot \text{VAR}[-(C + R)] = -C - \mu_R - \frac{\alpha}{2} \cdot \sigma_R^2, \quad \alpha \in \mathbb{R}^+ \quad (1)$$

$$\Delta\Phi = \Phi^{\text{TSD}} - \Phi^{\text{ASD}} \quad (2)$$

In case $\Delta\Phi > 0$, the SDP should be implemented in TSD mode, and vice versa. Below, we analyze the costs, their components, and the corresponding parameters. After that, we provide more details on the risk component.

3.2 Cost Perspective

First, we focus on the base costs of an SDP. The base costs for implementing an SDP depend on the chosen SD mode, because TSD and ASD mode feature different cost structures [3]. In TSD mode, the SDP is organized in a long phase which is processed sequentially in a single iteration. In contrast, ASD mode is organized and processed in several smaller iterations, so-called sprints. As mentioned above for modeling the base costs we use COCOMO II by Boehm et al. [21] for TSD mode and its extension from Benediktsson et al. [24] for ASD mode.

The base model of COCOMO II describes the SDP's effort in TSD mode in of person-months by taking the project size to the power of a scaling exponent E (Eq. 3). The number of person-months multiplied with the staff costs leads to the base costs. The variable size is measured in thousand source lines of code (KSLOC). Other measurement units for size (e.g., function points) are applicable as well. The constant A is an effort coefficient for productivity and the parameters EM_1, \dots, EM_{16} are effort multipliers that represent factors like product complexity, programmer capability, or platform volatility. The factor a thus adjusts the effort regarding productivity. The scaling exponent, which indicates economies ($E < 1$) or diseconomies of scale ($E > 1$), comprise a constant B and scale factors SF_1, \dots, SF_5 accounting, for example, for development flexibility, team cohesion, and process maturity. If all scale factors are rated "very high" and B is calibrated as COCOMO II suggests, E equals at least 0.91, implying economies of scale. Economies of scale can hardly be realized in SDPs. In almost every possible parameterization of COCOMO II, the cost function exhibits an exponent between 1.0 and its maximum 1.226, indicating diseconomies of scale. This is as communication and integration overheads increase over-proportionally with the project size. Eq. (3) represents the base model for TSD mode Boehm et al. [21].

$$PM_{\text{TSD}} = a \cdot \text{size}^E \quad \text{with } E = B + 0.01 \cdot \sum_{j=1}^5 SF_j \text{ and } a = A \cdot \prod_{i=1}^{16} EM_i \quad (3)$$

Benediktsson et al. [24] adjust the base model of COCOMO II for ASD mode. They divide the project size into sprints, a transformation that causes an initial effort for defining the breakdown structure. Moreover, sprints are interpreted as sub-projects. Consequently, Benediktsson et al. [24] apply the exponent E to the sprint sizes and sum up the resulting efforts, arguing that smaller sprints require less overhead and planning compared to the single large phase in TSD mode. In reality, the overall project length may vary between the different SD modes. However, there is no evidence that one mode is dominating the other. Therefore, and for reasons of comparability, we assume the

project length to be identical for TSD and ASD mode (Assumption 2). Our model setup would allow for length discounts in favor of ASD by subtracting a certain number of sprints from the length calculated for project implementation in TSD mode. Such a parameter, however, would be subject to extreme estimation inaccuracies and increase the complexity of our decision model. Furthermore, sprints can differ in size and content, which depends very much on the SDP at hand. We assume that all sprints take a constant amount of time (e.g. two weeks, four weeks) in case the SDP is implemented in ASD mode (Assumption 3). Thus, we divide the project length calculated for TSD mode into n equi-length sprints in ASD mode and apply the approach of Benediktsson et al. [24] [32]. Eq. (4) shows the base model for implementing the SDP in ASD mode as an adaption of the TSD mode estimation.

In Eq. (4), the parameter d_n models the overhead of the total effort required to define the breakdown structure. It depends on the total number of sprints [24]. Thereby, d_n is calibrated by a linear interpolation between the relative effort for defining one sprint e and for defining the largest number of sprints f (Eq. 5). The parameter c represents the overhead costs per sprint and accounts for the additional overhead within single sprints. The number of sprints n is gained from the calculation of the project length in CO-COMO II multiplied with the number of sprints per month. The exponent F consists of similar components as its counterpart for TSD mode. The constants G (called C in CO-COMO II) and D are obtained by calibrations based on former projects [21].

$$PM_{ASD} = d_n \cdot a \cdot size^E + n \cdot a \cdot \left(\frac{size}{n} \cdot (1 + c) \right)^E \quad (4)$$

with $n = G \cdot (PM_{TSD})^F \cdot \text{sprints per month}$ and $F = D + 0.02 \cdot 0.01 \cdot \sum_{j=1}^5 SF_j$ and

$$d_n = (f - e) \cdot \frac{n - 1}{n - N} + e \quad (5)$$

Figure 1 illustrates the cost/size relationships for projects implemented in TSD and ASD mode. Costs are measured in Monetary Units [MU], size in KSLOC. Since SDPs typically exhibit diseconomies of scale, we apply an exponent larger than 1 [21].

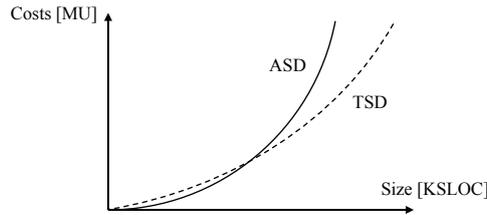


Figure 1. Costs of an SDP with diseconomies of scale depending on size

For $E \neq 1$, the two cost curves cross each other at a distinct intersection point. For small projects, ASD mode appears more favorable. This is as the initial effort required for creating a small number of sprints in ASD mode is small compared to the initial overhead for planning in TSD mode. Thus, TSD mode needs a minimum project size to justify the initial overhead. From a pure cost perspective, TSD mode is becoming more preferable with an increasing project size. This is as the initial overhead is distributed

over the project length. For large projects, the initial overhead for ASD mode is becoming relatively expensive as planning and managing sprints is getting complex [33].

3.3 Risk Perspective

In our analysis, we focus on three risk types, i.e., the risk of defect, risk of requirement changes, and risk of delay. Defects, changes in requirements, and delays can happen several times during implementation of an SDP. We model each risk type as a normally distributed variable with expected additional costs and a variance per sprint (Assumption 4) [27, 30]. Typically, the expected value μ per risk type is denoted in percent of the corresponding total costs C depending on TSD or ASD mode. Furthermore, the variance σ^2 of each risk type is measured in percent of the expected value.

We add this risk perspective to our decision model in order to incorporate the specific characteristics of TSD and ASD mode. The risk of defect covers the risk of having major defects in the software product due to developer mistakes [25, 26]. Boehm and Turner [34] model this risk based on an SDP's criticality. A large project size combined with a lack of upfront planning and inexperienced developers, for example, can cause such defects. A defect typically results in negative economic effects, which we model by expected additional costs resulting from a defect μ_{DEF} and their variance σ_{DEF}^2 . Since there is a detailed ex-ante planning in TSD mode, the expected additional costs and their variance are typically smaller as compared to ASD mode [12]. The risk of requirement changes, which Boehm and Turner [34] call the dynamism of a project, addresses change requests during project implementation. Due to regular releases, frequent stakeholder feedback as well as an environment geared toward refactoring and backlog refinement, the expected additional costs for requirement changes μ_{CHG} and their variance σ_{CHG}^2 are generally lower in ASD mode than in TSD mode [10]. The risk of delay addresses the risk of missing a deadline or not being fast enough regarding time-to-market. This risk is particularly high if the software product exhibits a short time-to-market. Delays result in expected additional costs μ_{DEL} and variance σ_{DEL}^2 from losing market share to competitors or from financial penalties due to missed deadlines. ASD mode can better mitigate the risk of delay as running software increments can be released more flexibly [5]. Thus, expected additional costs and the variance are generally higher in TSD mode.

In order to aggregate the three risk types, we must consider dependencies. There can be intra- and inter-temporal dependencies among the risk types. For example, in case of a major defect, the chance for a delay in the same period is higher than usual. We account for these dependencies between two risk types based on the Bravais-Pearson correlation coefficient ρ [35]. As an SDP lasts multiple periods or sprints, the risk types may also influence themselves over time. For example, if a major delay occurs, the chance for a greater delay in the following periods could be higher. Further, the risk of defect may tend to increase throughout the project in TSD mode. This drift that increases costs of defects over project time could be modeled via martingales. Alternatively, we could use individual distributions for every sprint, which would lead to more parameters and a more complex model. In our decision model, we therefore abstract from such auto-correlated structures, assuming that the distribution of each risk type is

identical over all periods (Assumption 5). This assumption leads to an underestimation of the costs of defects in TSD mode. In general, assuming inter-temporal independence underestimates the overall risk exposure. However, as the risk is underestimated in both SD modes and as we compare both SD modes regarding the implementation of a concrete SDP, this has not effect on the recommendations of our decision model. On this assumption, we can sum up the expected additional costs and variances of the three risk types per sprint t to derive the total expected additional costs μ_R and their variance σ_R^2 . The result is shown in Eq. (6) and (7).

$$\mu_R = \sum_{t=1}^n (\mu_{t,DEF} + \mu_{t,CHG} + \mu_{t,DEL}) = n \cdot (\mu_{DEF} + \mu_{CHG} + \mu_{DEL}) \quad (6)$$

$$\sigma_R^2 = \sum_{t=1}^n \sum_i \sum_j \sigma_{t,i} \sigma_{t,j} \rho_{i,j} = n \cdot \sum_i \sum_j \sigma_i \sigma_j \rho_{i,j} \quad (7)$$

with $i, j \in \{DEF, CHG, DEL\}$

On this foundation, the objective function from Eq. (2) can be rewritten as shown in Eq. (8). Thereby, C_{Dev} represents the costs for a software developer.

$$\Delta\Phi = -C_{Dev} \cdot a \left(size^E - d_n \cdot size^E + n \cdot \left(\frac{size}{n} \cdot (1+c) \right)^E \right) - \Delta\mu_R - \frac{\alpha}{2} \cdot \Delta\sigma_R^2 \quad (8)$$

4 Evaluation

To validate our decision model, we implemented its design specification as a software prototype [36]. The prototype enabled conducting sample calculations and a sensitivity analysis. As for our calculations, we consider two fictional sample SDPs specified in Table 1. This setting does of course not account for all possible projects. We chose the parameters such that one project favors each mode. The input parameters for the base costs are identical for both projects, but the risk parameters differ. Depending on the specific cost/risk profile, we have a TSD-type project with a high exposure to defects and an ASD-type project with a high exposure to requirement changes and delays. One example for such a TSD-type project is the implementation of a new core system within a bank. An example for the ASD-type project is the development of a new payment application for smartphones. For the TSD-type project, the expected additional costs of defect and the related variance are quite large, while the other risks are rather small. For the ASD-type project, it is the vice versa. Additionally, the risks of the TSD- and the ASD-type project vary according to the risk profiles of both SD modes. We keep the size flexible to enable a comparison between costs and risks for both SD modes with respect to different project sizes.

Table 2 shows the correlation coefficients of the risk types for both modes, which we use for our calculations. For both modes, the risk of defect is positively correlated with the risk of delay. As TSD is not as adaptive and flexible as ASD, a defect in TSD has a larger impact on the risk of delay compared to ASD. For TSD, a change of requirements increases the risk of delay, as heavy planning does not allow for changes. ASD mode can cope much better with changed requirements. As a consequence, the risk of requirement changes and risk of delay are uncorrelated in ASD mode.

Table 1. Specification of a TSD-type and an ASD-type sample project

Base costs input parameters				Risks TSD-type project			Risks ASD-type project		
a	2.94	SF_5	6.24		TSD mode	ASD mode		TSD mode	ASD mode
B	0.91	EM_i	1						
G	3.67	C_{Dev}	5,000 MU	μ_{DEF}	1.5 %	4 %	μ_{DEF}	1 %	2 %
D	0.28	c	0.15	μ_{CHG}	1.25 %	0.5 %	μ_{CHG}	2.5 %	1.5 %
SF_1	3.72	e	0.05	μ_{DEL}	1.25 %	0.5 %	μ_{DEL}	2.5 %	1.5 %
SF_2	3.04	f	0.4	σ_{DEF}	5 %	30 %	σ_{DEF}	5 %	15 %
SF_3	4.24	Sprints	2 per mo.	σ_{CHG}	15 %	10 %	σ_{CHG}	22.5 %	10 %
SF_4	4.38	Size	size	σ_{DEL}	15 %	10 %	σ_{DEL}	22.5 %	10 %

Table 2. Sample correlations between risk types for TSD and ASD mode

TSD mode				ASD mode			
$\rho_{i,j}$	DEF	CHG	DEL	$\rho_{i,j}$	DEF	CHG	DEL
DEF	1	0	0.5	DEF	1	0	0.25
CHG	0	1	0.5	CHG	0	1	0
DEL	0.5	0.5	1	DEL	0.25	0	1

We first analyze the ASD-type project from Table 1 and choose a constant absolute risk aversion $\alpha = 0.0002$. This is reasonable according to Bamberg and Spremann [37]. Figure 2 (left chart) illustrates the results, where we show different deltas between TSD and ASD mode in terms of costs, risk, and risk-adjusted costs while varying the project size between 0 and 300 KSLOC. The dotted line depicts the delta in terms of costs, the dashed line depicts the delta in terms of risk, and the continuous line depicts the delta in terms of risk-adjusted costs.

As outlined, the cost advantage for ASD is only given for small projects. The initial effort required for accomplishing a small number of sprints in ASD mode is relatively small compared to the overhead of upfront planning in TSD mode. With larger project sizes, the initial planning effort in TSD mode increases less than the overhead for planning larger amounts of sprints in ASD mode. Thus, TSD has a cost advantage for larger project sizes. This finding in our sample calculation complies with extant knowledge. Researchers have argued that TSD mode is more favorable for large projects in general [38]. From a risk perspective, ASD is the preferable SD mode for all investigated project sizes between 0 and 300 KSLOC. The risk advantage for ASD always exceeds the cost advantage for TSD mode. Therefore, ASD is the appropriate mode for the sample project at hand. For decision-makers with a lower level of risk aversion (α), the delta between TSD and ASD in terms of risk-adjusted costs is depicted by the continuous grey line in Figure 2 (left chart). In this case, ASD is not favorable for all project sizes. In case of more risk-seeking decision-makers, the cost advantage for TSD exceed the risk advantage for ASD for large project sizes. This example illustrates that deciding on the SD mode is similar to balancing the certain cost advantage of TSD against the

risk advantage of ASD. Economically speaking, decision-makers charge a risk premium for requirements changes on the cost advantage for TSD mode.

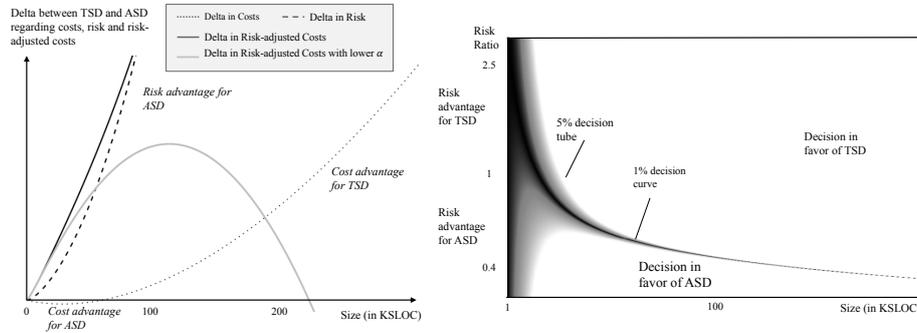


Figure 2. Calculation for the ASD-type sample project and grey-scale plot of relative delta

Next, we analyze the TSD-type project from Table 2. Table 1 shows the correlation coefficients of the risk types for both modes, which we use for our calculations. For both modes, the risk of defect is positively correlated with the risk of delay. As TSD is not as adaptive and flexible as ASD, a defect in TSD has a larger impact on the risk of delay compared to ASD. For TSD, a change of requirements increases the risk of delay, as heavy planning does not allow for changes. ASD mode can cope much better with changed requirements. As a consequence, the risk of requirements changes and risk of delay are uncorrelated in ASD mode. The cost advantages of TSD and ASD are the same as for ASD-type project, as the input parameters for the base costs are independent of the SD mode. Further, the risk advantage of TSD is positive for all project sizes, making TSD the preferable mode for nearly all project sizes. With the above examples, we demonstrated the suitability of our decision model by conducting sample calculations for a TSD- and an ASD-type sample project.

To demonstrate the robustness of our decision model, we now analyze the delta in risk-adjusted costs between both SD modes by varying the size and risk structure of a given project using a sensitivity analysis. When setting up the sensitivity analysis, two features must be considered: First, to stay comparable with an increasing project size, we rely on the relative delta in risk-adjusted costs, i.e., we divide the absolute delta in risk-adjusted costs by the absolute risk-adjusted costs of the respective SD mode. We do not consider the absolute risk-adjusted costs delta, which gets biased with an increasing project size. Second, as we only have a single dimension for describing the risk structure, we divide the risk of defect, which is typical for TSD-type projects, by the sum of the risk of requirement changes and delay, which are more typical for ASD-type projects. Thereby, we get a risk ratio that indicates a risk profile in favor of ASD for values from the interval $[0,1[$ and in favor of TSD for values from $]1, \infty[$. As a result, we get a three-dimensional analysis with project size and risk ratio on the horizontal and vertical axis, and the delta in relative risk-adjusted costs as a grey-scale plot. The delta in relative risk-adjusted costs is depicted on a grey-scale with bright indicating a high delta and dark indicating a low delta. The dark area shows settings where

decision-makers are indifferent, whereas the bright areas illustrate robust decisions in favor of either TSD or ASD. We start with the TSD-type project from Table 1. We then reduce the risk of defect successively, while equivalently increasing the risk of requirement changes and delay until we end up with the ASD-type project from Table 1. This way, we get comparable projects regarding costs and a balanced risk profile half way between both projects (i.e., risk ratio = 1). The black area and line in Figure 2 (right chart) shows the decision curve, i.e., the area of indifference between both SD modes. Along the decision curve, the relative delta in the risk-adjusted costs is less than one percent. The grey decision tube around the black line depicts the delta values that are less than five percent of the respective risk-adjusted costs. Within this tube, the decision is not profound, but slightly tends to one of both modes. Decision-makers have to be careful regarding decisions within the five percent decision tube, as estimation errors may strongly influence the decision. For small projects, the decision is mainly influenced by the risk advantage of TSD or ASD depending on the SD mode, since the cost advantage of ASD is negligibly small. The black area indicates that our decision model is indifferent for small projects as it strongly depends on the risk distribution of the project at hand and the SD modes. However, this indifference is plausible, as the decision on the appropriate SD mode for a small SDP does not carry as much weight in terms of total costs as for large SDPs. The managerial implication is that decision makers should focus on finding the appropriate mode for larger SDPs, since for small projects the cost and risk difference is negligibly small. With an increasing project size, the costs are strongly in favor of TSD mode, leading to a steep decrease of the decision curve. The decision curve then converges towards the horizontal axis, indicating a larger area for unambiguous decisions in favor of TSD for very large projects. This is due to the cost advantage of TSD. That is, it can be reasonable from a cost/risk perspective to implement an SDP in TSD despite a risk advantage for ASD mode. For strongly ASD-type SDPs, i.e., projects with a very strong risk of requirement changes and delays, however, ASD is appropriate. The managerial implication of this finding is that decision makers should carefully evaluate the project's risk in both modes, since it could reverse a strong cost advantage for large projects. Although the graph in Figure 2 indicates otherwise, there is no reason to believe that the majority of projects should be conducted in TSD mode. First, SDPs in practice are not equally distributed over the size and risk dimensions. Second, we analyzed only one particular sample SDP in Figure 2. In a practical setting, our model primarily serves as decision support. Decision-makers should nevertheless carefully evaluate the outcome by taking their experience and other models into account.

5 Conclusion

In this study, we investigated how organizations can decide whether to implement a distinct SDP in TSD or ASD mode. Building on the characteristics of the SDP in focus and the characteristics of both SD modes, we proposed a decision model that analyzes the costs and risks associated with the implementation of a distinct SDP. Our model builds on the cost estimation method COCOMO II for TSD mode and the extensions

proposed by Benediktsson et al. [24] for ASD mode to achieve comparability between both SD modes. To extend a purely cost-based view, the model accounts for three major risk types related to the implementation of an SDP, i.e., risk of defect, requirement changes, and delay. Besides these risk types, the decision model incorporates sprint length and overhead costs as characteristics of TSD and ASD mode. Our contribution is twofold. First, we bring the two cost estimation approaches together. Second, we extend the solely cost based view by a risk perspective. As for evaluation, we applied the decision model to sample projects with different input parameters. We also conducted a sensitivity analysis based on a software prototype to validate the decision model's suitability and robustness. The sensitivity analysis corroborated that the decision model yields plausible results. Overall, we contribute to the prescriptive knowledge on software development with a decision model as concrete artifact and its instantiation as a software prototype.

Our decision model is beset with limitations. First, we assume the risk types to be independent over time and normally distributed. Thus, the decision model underestimates risks associated with implementing an SDP. Risks may also increase or decrease over time, e.g., the expected additional costs for a change in requirements may increase over time if the project is executed in TSD mode. Research on inter-temporal dependencies of SD risks is required to implement these effects into our decision model. Future research should explore the risk structure of SDPs. Second, the decision model focuses on costs and risks that accrue during SDP implementation. Considering an SDP's business value also requires integrating benefits. In ASD mode, design and run time can no longer be separated as benefits accrue after each sprint. In TSD mode, however, benefits only realize after the SDP has been fully completed. Thus, we recommend that further research extends the decision model toward a cost-benefit analysis that includes benefits and a runtime perspective. Third, we focused on individual SDPs, ignoring the portfolio perspective. Future research should investigate how project dependencies influence the TSD/ASD trade-off. Fourth, as a first step, we demonstrated and analyzed the decision model via sample projects and a sensitivity analysis. Thus, the decision model would benefit from naturalistic evaluation, e.g., real-world case studies.

6 Acknowledgement

This research was (in part) carried out in the context of the Project Group Business and Information Systems Engineering of the Fraunhofer Institute of Applied Information Technology FIT.

References

1. Horlach, B., Drews, P., Schirmer, I.: Bimodal IT: Business-IT Alignment in the Age of Digital Transformation. In: Multikonferenz Wirtschaftsinformatik (MKWI) 2016. Universitaetsverlag Ilmenau, Ilmenau (2016)
2. Gartner: Gartner Executive Programs Survey of More Than 2,300 CIOs Reveals Many Are Unprepared for Digitalization: the Third Era of Enterprise IT. (2014)

3. Aitken, A., Ilango, V.: A Comparative Analysis of Traditional Software Engineering and Agile Software Development. In: Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS), pp. 4751-4760. IEEE, Wailea, Maui, Hawaii (2013)
4. Sondergaard, P.: Digital Business Demands Bimodal Capabilities. *ITNOW* 58, 8-9 (2016)
5. Ahimbisibwe, A., Cavana, R.Y., Daellenbach, U.: A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management* 28, 7-33 (2015)
6. Jorgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering* 33, 33-53 (2007)
7. Gregor, S., Hevner, A.R.: Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly* 37, 337-355 (2013)
8. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, 45-77 (2007)
9. Sommerville, I.: *Software Engineering*. Pearson, New York (2015)
10. Lee, G., Xia, W.: Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. *MIS Quarterly* 34, 87-114 (2010)
11. Conboy, K.: Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research* 20, 329-354 (2009)
12. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. *Communications of the ACM* 48, 72-78 (2005)
13. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, 833-859 (2008)
14. Royce, W.W.: Managing the development of large software systems. In: Proceedings of IEEE WESCON. IEEE, Los Angeles (1970)
15. Beck, K., Beedle, M., Bennekum, A.v., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., <http://agilemanifesto.org/> (Accessed: 11/01/2016)
16. Fridgen, G., Klier, J., Beer, M., Wolf, T.: Improving Business Value Assurance in Large-Scale IT Projects: A Quantitative Method Based on Founded Requirements Assessment. *ACM Transaction on Management Information Systems* 5, 1-17 (2014)
17. Remer, D.S., Nieto, A.P.: A compendium and comparison of 25 project evaluation techniques. Part 1: Net present value and rate of return methods. *International Journal of Production Economics* 42, 79-96 (1995)
18. Clermont, P.: Cost-Benefit Analysis: It's Back in Fashion, Now Let's Make It Work. *Information Strategy: The Executive's Journal* 18, 6-11 (2002)
19. Graham, D.A.: Cost-benefit analysis under uncertainty. *The American Economic Review* 71, 715-725 (1981)
20. Bannerman, P.L.: Risk and risk management in software projects: A reassessment. *Journal of Systems and Software* 81, 2118-2133 (2008)
21. Boehm, B.W., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R.: Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering* 1, 57-94 (1995)
22. Cao, L., Ramesh, B., Abdel-Hamid, T.: Modeling dynamics in agile software development. *ACM Transactions on Management Information Systems* 1, 1-26 (2010)
23. Yang, D., Wan, Y., Tang, Z., Wu, S., He, M., Li, M.: COCOMO-U: An Extension of COCOMO II for Cost Estimation with Uncertainty. In: Wang, Q., Pfahl, D., Raffo, D., Wernick, P. (eds.) *Software Process Change*, vol. 3966, pp. 132-141. Springer, Berlin, Heidelberg (2006)

24. Benediktsson, O., Dalcher, D., Reed, K., Woodman, M.: COCOMO-Based Effort Estimation for Iterative and Incremental Software Development. *Software Quality Journal* 11, 265-281 (2003)
25. Schmidt, R., Lyytinen, K., Mark Keil, P.C.: Identifying software project risks: An international Delphi study. *Journal of Management Information Systems* 17, 5-36 (2001)
26. Boehm, B.W.: Software risk management: principles and practices. *IEEE Software* 8, 32-41 (1991)
27. Fridgen, G., Mueller, H.-V.: Risk/cost valuation of fixed price IT outsourcing in a portfolio context. In: *Proceedings of the 30th International Conference on Information Systems (ICIS)*, pp. 1115-1131. Association for Information Systems (AIS), Phoenix (2009)
28. Conboy, K., Coyle, S., Xiaofeng, W., Pikkarainen, M.: People over Process: Key Challenges in Agile Development. *IEEE Software* 28, 48-57 (2011)
29. Buhl, H.U., Röglinger, M., Stöckl, S., Braunwarth, K.S.: Value orientation in process management - Research Gap and Contribution to Economically Well-Founded Decisions in Process Management. *Bus Inf Syst Eng* 3, 163-172 (2011)
30. Zimmermann, S., Katzmarzik, A., Kundisch, D.: IT Sourcing Portfolio Management for IT Services Providers-A Risk/Cost Perspective. In: *Proceedings of 29th International Conference on Information Systems (ICIS)*, pp. 1273-1290. Association for Information Systems (AIS), Paris (2008)
31. Pratt, J.W.: Risk aversion in the small and in the large. *Econometrica* 32, 122-136 (1964)
32. Shiohama, R., Washizaki, H., Kuboaki, S., Sakamoto, K., Fukazawa, Y.: Estimate of the appropriate iteration length in agile development by conducting simulation. In: *Proceedings of the 2012 Agile Conference (AGILE)*, pp. 41-50. IEEE, Dallas (2012)
33. Brown, A.W., Ambler, S., Royce, W.: Agility at scale: economic governance, measured improvement, and disciplined delivery. *Proceedings of the 2013 International Conference on Software Engineering*, pp. 873-881. IEEE Press, San Francisco, CA, USA (2013)
34. Boehm, B.W., Turner, R.: Using risk to balance agile and plan-driven methods. *Computer* 36, 57-66 (2003)
35. Pearson, K.: Mathematical Contributions to the Theory of Evolution.--On a Form of Spurious Correlation Which May Arise When Indices Are Used in the Measurement of Organs. *Proceedings of the Royal Society of London* 60, 489-498 (1896)
36. Sonnenberg, C., Brocke, J.: Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) *Proceedings of Design Science Research in Information Systems. Advances in Theory and Practice: 7th International Conference, DESRIST 2012, Las Vegas, NV, USA*, pp. 381-397. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
37. Bamberg, G., Spremann, K.: Implications of constant risk aversion. *Zeitschrift für Operations Research* 25, 205-224 (1981)
38. Barlow, J.B., Keith, M.J., Wilson, D.W., Schuetzler, R.M., Lowry, P.B., Vance, A., Giboney, J.S.: Overview and Guidance on Agile Development in Large Organizations. *Communications of the Association for Information Systems* 29, 25-44 (2011)