



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Risk management in agile methods: a study of DSDM in practice
Author(s)	Coyle, Sharon
Publication Date	2009-05-25
Publication Information	Coyle, S. (2009) Risk management in agile methods: a study of DSDM in practice. In eXtreme Programming and Agile Processes in Software Engineering, 10th International Conference on.
Item record	http://hdl.handle.net/10379/1569

Downloaded 2024-04-25T04:02:06Z

Some rights reserved. For more information, please see the item record link above.



Risk Management in Agile Methods: A Study of DSDM in Practice

Sharon Coyle¹

¹ Business Information Systems Group, Centre for Innovation and Structural Change,
J.E. Cairnes Graduate School of Business and Economics,
National University of Ireland Galway
<http://www.nuigalway.ie/commerce/>

Abstract. Businesses are increasingly operating in extremely turbulent environments necessitating the need to respond and adapt to change more quickly and improve overall time to market. From an Information Systems Development perspective this has triggered a new wave of development, the most notable of these being agile methods. A principle objective of agile methods is to reduce well-known risks associated with common ISD project failures. While there is extensive academic literature on risk management and its growing importance in ISD, literature in relation to risk management in agile ISD projects is still in infancy. The purpose of this research was to ascertain the extent to which risk management practices are incorporated into agile development projects. The methodology deployed for this research involved a case study of a change management consultancy firm dedicated to the use of the Dynamic Systems Development Method (DSDM).

Keywords: Risk Management, Agile Methods, DSDM (Dynamic Systems Development Method)

1 Introduction

In ISD, the rapidly growing use of agile methods shows the urgency of organizations to adapt to change at a more speedy and efficient pace. Agile methods are known for their use of iterative development, active user involvement and their acknowledgement of the need to incorporate changing system requirements and “focus on generating early releases of working products using mostly collaborative techniques” [1]. This is a stark contrast to the traditional model for systems development which promotes “elicitation and freezing of requirements in advance” [2] with no overlap between project phases of analysis, design and implementation [3].

A principal objective of agile methods is to reduce well-known risks associated with common ISD project failures by for example, accepting that requirements will change. The Dynamic Systems Development Method is considered to be the first “truly agile method” where “it is preferred to fix time and resources and then adjust the amount of functionality accordingly” [4]. This highlights the element of flexibility

in DSDM with regard to adjusting system functionality where system requirements are open to change. However, no matter what the nature of change, there will always be associated risks involved.

1.1 Motivation for Research

While there is extensive literature on risk management, research of risk management in agile ISD projects is non-existent. This is surprising considering how quickly agile methods are being adopted in ISD where a survey conducted by Vijayasathy and Turk [5], having a total of 98 respondents indicated that sixty percent use agile approaches in seventy five percent or more of their projects. Many books on agile methods “have remarkably little to say about how a development team determines the risks it faces, prioritises them or takes action to negate their effects” [6]. Essentially agile approaches must tailor traditional risk management techniques meant for “years-long projects into a risk driven agile iteration lasting only seven to thirty days” [6]. How agile projects go about doing this remains unknown.

The primary objective of this research was thus to develop a better understanding of risk management practices in agile ISD projects and the level of formality with which these practices are executed. Specifically, this research focuses on three main elements of risk management, namely risk identification, estimation and evaluation.

2 Theoretical Foundations

In an ISD context, Barry Boehm highlighted the concept of managing risks and giving them priority as far back as 1988. Ten years later, Hall [7] described Boehm as being “the father of software risk management.” Boehm proposed a move away from the ‘staged’ SDLC to a more iterative or incremental process and this proposed concept in software development was an attempt to lower project risks [8]. Boehm’s aim was to eliminate any software difficulties or risks mainly by deriving “risk-driven documents” and “incorporating prototyping as a risk-reduction option” [9]. It resulted in what was called the Spiral Model that essentially created “a risk-driven approach to the software process rather than a primarily document-driven or code-driven process” [9].

Many associations can be made between Boehm’s proposals above and that which has developed in the approaches deployed by agile methods. While many authors highlight distinct approaches and frameworks for dealing with risk management (ranging from formal to informal), its basic fundamentals remain the same and are consistent across disciplines. The literature shows similar emphasis on the most important activities in risk management namely, those identified by Rowe [10] and Charette [11] – the early practitioners of risk management – who outline the three main elements of risk assessment as (i) Risk Identification, (ii) Risk Estimation and (iii) Risk Evaluation.

2.1 Risk Identification

Risk identification is the reduction of descriptive uncertainty [10] which involves “surveying the range of potential threats” [11]. This element of risk assessment involves detecting issues which could jeopardize or threaten the success of a project [12, 13]. Chapman [14] states that “the risk identification and assessment stages have the largest impact on the accuracy of any risk assessment.” It is therefore the most important stage of risk management. Of particular importance to ISD is the early identification of risks where “identifying and dealing with risks early in the development lessens long-term costs and helps prevent software disasters” [15]. Furthermore, it is important that risk identification carries on throughout the project’s lifecycle [13]. Therefore, risk identification is an ongoing, continuous process that requires regular screening and monitoring.

An important aspect of risk identification is categorizing the risks organizations encounter. According to Coppendale [16], “depending on the size and the complexity of the project there might be between five and fifteen” categories of risk. Categories attempt to group certain types of risk under a particular heading and in doing so “can help you find global risks that can be solved together” [17]. There are many sources of risk, some of which include senior management, the client or customer, the project team, organization of the project itself and even laws and standards which directly impact the project [18]. Such sources can be placed into their respective categories as being either internal or external risk as follows:

Table 1. Categories and Sources of Project Risk

		Risk Category	
		Internal	External
Sources of Project Risk	Senior Management		Acts of Nature
	Project Team & Management		The Client
	Organisation of the Project		Laws and Standards

The two most dominant sources of internal risk identified across the literature are Senior/Project Management and Project Team. The dominating external source of risk is the client. All of these were collectively identified by Mantel, Meredith et al. [18] and are represented in the above table. Every source of risk can have numerous risk factors. A risk factor is “a condition that forms a serious threat to the completion of an IT project” [19]. Some internal risk factors include project conflict and resource boundaries, which can be linked to sources of project team and senior management risk respectively. The following table shows some of the most dominant risk factors identified by Wieggers [20], who categorizes these factors by sector:

Table 2. Most common risk factors for various project types

Project Sector	Risk Factor	% of Projects at Risk
MIS	Creeping User Requirements	80
	Excessive Schedule Pressure	65
	Low Quality	60
	Cost Overruns	55
	Inadequate Configuration Control	50
	Commercial	Inadequate User Documentation
Low User Satisfaction		55
Excessive time to market		50
Harmful competitive actions		45
Litigation expense		30

On analysing the table, some direct link between sources and risk factors is evident. For example, the sources of ‘management’ risk or ‘organisation of the project itself’ could be linked to the risk factor of ‘inadequate configuration control’ due to a flaw in the project’s arrangement and organisation. However, the most interesting correlation is that of the ten risk factors listed in Table 2, at least six of these can be linked to the client as a source of project risk.

Finally, a dominating feature in recent literature deserves recognition where there is strong support among authors that an actual ‘source of risk’ can provide a ‘source of opportunity.’ Chapman and Ward [21] state “it is only once risk is seen as a good thing people begin to look for opportunities.” Very few people would acknowledge ‘opportunity’ as being a facet of risk as naturally there are negative connotations associated with risk. Hillson [22] however, states “the decision to encompass both opportunities and threats within a single definition of risk is a clear statement of intent, recognising that both are equally important influences over project success, and both need managing proactively.” In a general sense the above ideas represent something we all know and understand about risk and the nature of taking gambles – people and organisations usually undertake risks with the aim of benefiting from potential opportunities [23]. Taking on any form of risk can be a daunting task but as DeMarco & Lister [24, 25] state, “if a project has no risks, don’t do it.”

2.2 Risk Estimation

At this stage it is hoped that the project team have identified all potential risks and they can now move on to estimating those risks. Risk estimation is the reduction of measurement uncertainty [10] where “the values of the variables describing the system are determined, the various consequences of an event occurring are identified”

and finally, “the magnitude of the risk is determined” [11]. In ISD environments there are many generic risk factors (such as creeping user requirements) but very few instances of projects operating under similar circumstances. Therefore, allocating future estimates is undoubtedly different to those allocated for past events as there are so few exact comparable projects conducted in the past [26].

Risk estimation attempts to estimate “the chance (or probability) of potential loss” as well as “the exposure to potential loss i.e. the consequences or magnitude of the identified risks” [26]. The chance of potential loss is essentially the process of attaching a probability of occurrence to any identified risk. As Hall [7] states “estimation is the appraisal of risk probability and consequence.” Probability is categorized as being greater than zero and less than one hundred while consequence is decided relative to cost, schedule and technical goals [7]. If an event is certain to occur it has a probability of exactly one [27] or one hundred percent. According to McManus [27], probability data should be used to compute the risk. When no actual data on probabilities exist, estimates by individuals most familiar with the project, its risk factors and overall problems are a good substitute [18].

Headings should be capitalized (i.e., nouns, verbs, and all other words except articles, prepositions, and conjunctions should be set with an initial capital) and should, with the exception of the title, be aligned to the left. Words joined by a hyphen are subject to a special rule. If the first word can stand alone, the second word should be capitalized. The font sizes are given in Table 1.

Here are some examples of headings: "Criteria to Disprove Context-Freeness of Collage Languages", "On Correcting the Intrusion of Tracing Non-deterministic Programs by Software", "A User-Friendly and Extendable Data Distribution System", "Multi-flip Networks: Parallelizing GenSAT", "Self-determinations of Man".

Table 2. Font sizes of headings. Table captions should always be positioned *above* the tables.

Heading level	Example	Font size and style
Title (centered)	Lecture Notes ...	14 point, bold
1 st -level heading	1 Introduction	12 point, bold
2 nd -level heading	2.1 Printing Area	10 point, bold
3 rd -level heading	Headings. Text follows ...	10 point, bold
4 th -level heading	<i>Remark.</i> Text follows ...	10 point, italic

Lemmas, Propositions, and Theorems. The numbers accorded to lemmas, propositions, and theorems, etc. should appear in consecutive order, starting with Lemma 1, and not, for example, with Lemma 11.

2.1 Figures

Please check that the lines in line drawings are not interrupted and have a constant width. Grids and details within the figures must be clearly legible and may not be written one on top of the other. Line drawings should have a resolution of at least 800 dpi (preferably 1200 dpi). The lettering in figures should have a height of 2 mm (10-point type). Figures should be numbered and should have a caption which should always be positioned *under* the figures, in contrast to the caption belonging to a table, which should always appear *above* the table. Please center the captions between the margins and set them in 9-point type (Fig. 1 shows an example). The distance between text and figure should be about 8 mm, the distance between figure and caption about 6 mm.

To ensure that the reproduction of your illustrations is of a reasonable quality, we advise against the use of shading. The contrast should be as pronounced as possible.

If screenshots are necessary, please make sure that you are happy with the print quality before you send the files.

Remark 1. In the printed volumes, illustrations are generally black and white (halftones), and only in exceptional cases, and if the author is prepared to cover the extra costs involved, are colored pictures accepted. Colored pictures are welcome in the electronic version free of charge. If you send colored figures that are to be printed in black and white, please make sure that they really are legible in black and white. Some colors show up very poorly when printed in black and white.

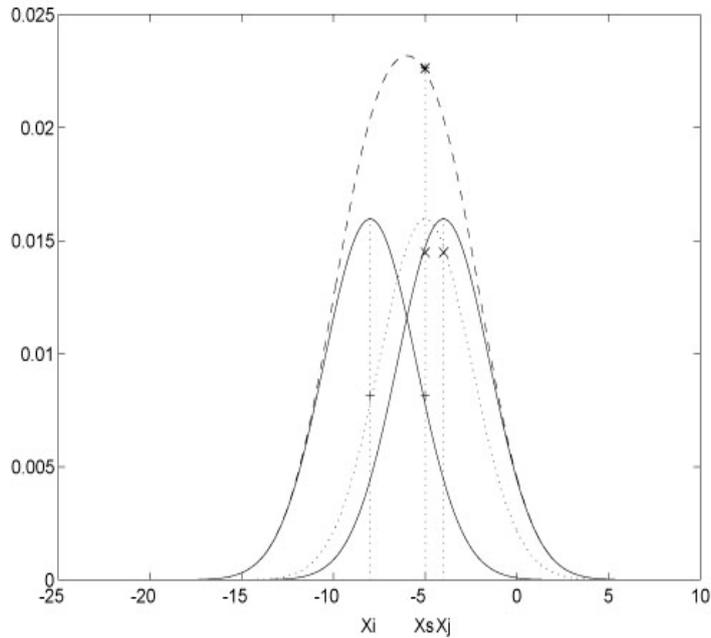


Fig. 1. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in italics, in parentheses, as shown in this sample caption.

2.2 Formulas

Displayed equations or formulas are centered and set on a separate line (with an extra line or halfline space above and below). Displayed expressions should be numbered for reference. The numbers should be consecutive within each section or within the contribution, with numbers enclosed in parentheses and set on the right margin.

$$x + y = z . \tag{1}$$

Equations should be punctuated in the same way as ordinary text but with a small space before the end punctuation mark.

2.3 Footnotes

The superscript numeral used to refer to a footnote appears in the text either directly after the word to be discussed or – in relation to a phrase or a sentence – following the

punctuation mark (comma, semicolon, or period). Footnotes should appear at the bottom of the normal text area, with a line of about 5cm set immediately above them.

2.4 Program Code

Program listings or program commands in the text are normally set in typewriter font, e.g., CMTT10 or Courier.

Example of a Computer Program from Jensen K., Wirth N. (1991) Pascal user manual and report. Springer, New York

```
program Inflation (Output)
  {Assuming annual inflation rates of 7%, 8%, and
  10%,...  years};
const MaxYears = 10;
var   Year: 0..MaxYears;
      Factor1, Factor2, Factor3: Real;
begin
  Year := 0;
  Factor1 := 1.0; Factor2 := 1.0; Factor3 := 1.0;
  WriteLn('Year 7% 8% 10%'); WriteLn;
  repeat
    Year := Year + 1;
    Factor1 := Factor1 * 1.07;
    Factor2 := Factor2 * 1.08;
    Factor3 := Factor3 * 1.10;
    WriteLn(Year:5,Factor1:7:3,Factor2:7:3,
            Factor3:7:3)
  until Year = MaxYears
end.
```

2.5 Citations

For citations in the text please use square brackets and consecutive numbers: [1], [2], [3], etc.

2.6 Page Numbering and Running Heads

There is no need to include page numbers. If your paper title is too long to serve as a running head, it will be shortened. Your suggestion as to how to shorten it would be most welcome.

¹ The footnote numeral is set flush left and the text follows with the usual word spacing.

3 LNCS Online

The online version of the volume will be available in LNCS Online. Members of institutes subscribing to the Lecture Notes in Computer Science series have access to all the pdfs of all the online publications. Non-subscribers can only read as far as the abstracts. If they try to go beyond this point, they are automatically asked, whether they would like to order the pdf, and are given instructions as to how to do so.

4 BibTeX Entries

The correct BibTeX entries for the Lecture Notes in Computer Science volumes can be found at the following Website shortly after the publication of the book: <http://www.informatik.uni-trier.de/~ley/db/journals/lncs.html>

Acknowledgments. The heading should be treated as a 3rd level heading and should not be assigned a number.

5 The References Section

In order to permit cross referencing within LNCS-Online, and eventually between different publishers and their online databases, LNCS will, from now on, be standardizing the format of the references. This new feature will increase the visibility of publications and facilitate academic research considerably. Please base your references on the examples below. References that don't adhere to this style will be reformatted by Springer. You should therefore check your references thoroughly when you receive the final pdf of your paper. The reference section must be complete. You may not omit references. Instructions as to where to find a fuller version of the references are not permissible.

The following section shows a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4] and [5], as well as a URL [6]. Please note that proceedings published in LNCS are not cited with their full titles, but with their acronyms!

References

1. Reifer, D.: How good are agile methods? *IEEE Software*, 19(4), 16--18 (2002)
2. Fitzgerald, B.: Systems Development Methodologies: The problem of tenses. *Information Technology and People*, 13(3), 174--185 (2000)
3. Walters, S.A., Broady, J.E. et al.: A Review of Information Systems Development Methodologies. *Library Management*, 15(6), 5--19 (1994)

4. Abrahamsson, P., Warsta, J. et al.: New Directions on Agile Methods: A Comparative Analysis. 25th International Conference on Software Engineering. IEEE Computer Society, pp. 244--254 (2003)
5. Vijayasarathy, L., Turk, D.: Agile Software Development: A survey of early adopters. *Journal of Information Technology Management* 19(2), (2008)
6. Smith, P., Pichler, R.: Agile Risks/Agile Rewards. *Software Development*, 13(4), 50--53 (2005)
7. Hall, E. M.: *Managing Risk: Methods for Software Systems Development*, Addison Wesley (1998)
8. Powell, P. L., Klein, J.H.: Risk Management for Information Systems Development, *Journal of Information Technology*, 11(4), 309--319 (1996)
9. Boehm, B. W.: A Spiral Model of Software Development and Enhancement, *IEEE Computer*, 21(5), 61--75 (1988)
10. Rowe, W. D.: *An Anatomy of Risk*, John Wiley & Sons Ltd. (1977)
11. Charette, R. N.: *Applications Strategies for Risk Analysis*, Multiscience Press Inc. (1990)
12. Coppendale, J.: Manage Risk in Product and Process Development and Avoid Unpleasant Surprises, *Engineering Management Journal*, 5(1), 35--38 (1995)
13. Grey, S.: *Practical Risk Assessment for Project Management*, John Wiley & Sons Ltd. (1995)
14. Chapman, R. J.: The Effectiveness of Working Group Risk Identification and Assessment Techniques, *International Journal of Project Management*, 16(6), 333--343 (1998)
15. Boehm, B. W.: Software Risk Management: Principles and Practices, *IEEE Software* 8(1), 32--41 (1991)
16. Coppendale, J.: Manage Risk in Product and Process Development and Avoid Unpleasant Surprises, *Engineering Management Journal*, 5(1), 35--38 (1995)
17. Williams, R. C., Walker, J. A. et al.: Putting Risk Management into Practice, *IEEE Software*, 14(3), 75--81 (1997)
18. Mantel, S. J., Meredith, J. R. et al.: *Project Management in Practice*, John Wiley & Sons, Inc. (2001)
19. Keil, M., Cule, P. E.: A Framework for Identifying Software Project Risks, *Communications of the ACM*, 41(11), 76--83 (1998)
20. Wiegers, K. E.: Know Your Enemy: Software Risk Management, *Software Development*, 6(10), 38--42 (1998)
21. Chapman, C., Ward, S.: *Project Risk Management: Processes, Techniques and Insights*, John Wiley & Sons Ltd. (1997)
22. Hillson, D.: Extending the Risk Process to Manage Opportunities, Fourth European Project Management Conference, PMI Europe 2001, London, UK. (2001)
23. Schwalbe, K.: *Information Technology Project Management*, Course Technology, Thomson Learning (2000)
24. DeMarco, T., Lister, T.: Risk Management during Requirements, *IEEE Software*, 20(5), 99--101 (2003)
25. DeMarco, T., Lister, T.: *Waltzing with Bears: Managing Risk on Software Projects*, Dorset House Publishing Co. (2003)
26. Charette, R. N.: *Software Engineering Risk Analysis and Management*, Multiscience Press (1989)
27. McManus, J.: *Risk Management in Software Development Projects*, Elsevier Butterworth-Heinemann (2004)

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195--197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) *Euro-Par 2006. LNCS*, vol. 4128, pp. 1148--1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181--184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: *The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration*. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, <http://www.ncbi.nlm.nih.gov>

Appendix: Springer-Author Discount

LNCS authors are entitled to a 33.3% discount off all Springer publications. Before placing an order, they should send an email to SDC.bookorder@springer.com, giving full details of their Springer publication, to obtain a so-called token. This token is a number, which must be entered when placing an order via the Internet, in order to obtain the discount.

Checklist of Items to be Sent to Volume Editors

1. A final Word or RTF file
2. A final PDF file
3. A copyright form, signed by one author on behalf of all of the authors of the paper
4. A readme giving the name and email address of the corresponding author

