Risk management in software development projects: an evaluation

INF11103 Management of Software Projects: Coursework 2

Matriculation number 40070877

6 May 2013

Word Count: 3000, not including abstract, footnotes and works cited

Abstract

Risk management (RM) is presented as very important. It is claimed that 5% investment in RM saves over 50% of overrun, and hence that RM is worthwhile. However, it is possible to question this claim, and to perceive issues in RM's theory and application. These issues are centred around human factors in RM. Evidence for and against RM is presented, and it is concluded that RM can be beneficial but it will not completely prevent runaway disasters.

1. Introduction

1.1. What are risk and risk management?

There are many definitions of risk (Hillson, 2002) but the two major project management (PM) methodologies, PRINCE2 (The Stationery Office, 2009) and PMBoK (Project Management Institute, 2013) define risk as 'an uncertain event or set of events that, should it occur, will have an affect on the achievement of objectives'. This definition is used in this paper. Hence risk management (RM) is managing risks to obtain the best possible combination of costs (time and money) and outcomes.

1.2. RM emphasis and importance

Many studies quote 189% software project average cost overruns (The Standish Group, 1995) but these are actually between 20 and 40% (Jørgensen, 2013). To contextualise this, the UK government annually spends around £14bn on IT projects (Dunleavy, Margetts, Bastow, & Tinkler, 2008), while a current larger project is expected to cost around £3bn (Centre for economic and social inclusion, 2013). 30% over-run would be £900,000,000 – equivalent to around 9000 senior nurses' annual salaries (Royal College of Nursing, 2013). If RM cost 5% of budget but saved 50% overrun (McConnell, 1997), it would save £1·35bn net.

An implication that can be drawn from this is that RM is worthwhile. In fact Williams (2005) went so far as to say 'PM … procedures are self-evidently correct: … project failure is indicative of inadequate PM' (presumably including inadequate RM). Such statements are open to question: RM has been described as 'walzing with bears' (DeMarco & Lister, 2003) and 'broken' (Hubbard, 2009).

1.3. Scope and questions

1.3.1. Is RM worthwhile?

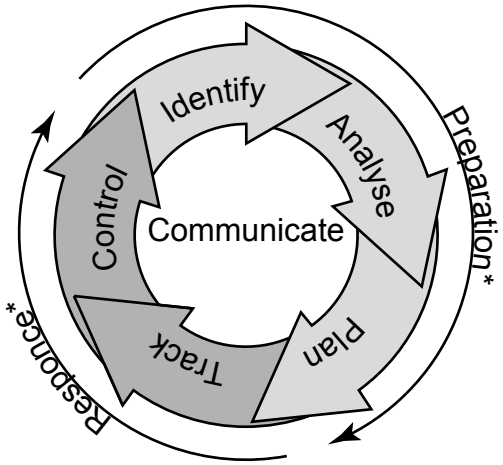
To answer this question, one can rephrase it as *Does RM work?*, where *work* is defined as *reduce the **quantity** of materialised 'negative' risks* ('threats'), *and/or increase the **quantity** of materialised 'positive' risks* ('opportunities').

To allow for a small quantity of costly threats outweighing a large quantity of small benefits, *work* can be framed in cost-benefit terms: *Does RM reduce materialised-threat **costs** and increase materialised-opportunity **benefits**?*

It is possible to envisage RM's costs outweighing its benefits, in which case RM would most likely be counter-productive. Hence the question becomes *Does RM reduce materialised-threat costs and increase materialised-opportunity benefits **enough to outweigh its costs**?*

1.3.2. Factoring in RM process

RM activities are often modelled as



*These super-classes are defined in this paper.

(Software Engineering Institute, 2012)

Clearly, if a threat has not been identified or its impact has been mis-assessed, plans may be absent or inappropriate.

Control activities are classified as avoidance, transference, mitigation and acceptance (Peltier, 2004). In this paper, acceptance is considered to be 'passive' RM because it involves no activity except using pre-allocated contingency budget and time, or accepting that the deliverables will not meet all requirements. So the central question in this paper is refined to *Q1: Does **active** RM reduce materialised-threat costs and increase materialised-opportunity benefits enough to outweigh its costs?*

1.3.3. Further questions

If RM does work, one can then ask *Q2: What contributes to such successes?* and *Q3: What are RM's limits?* If RM does not work, one can then ask *Q4: Can RM **never** work or can it be made to work?* An overall yes/no answer may not be possible: some risk classes, project types and scales may be generally amenable to RM while others are not.

This paper is an attempt to answer these questions in the field of software projects.

2. Methodology, structure of this paper

It would be difficult to perform a experiment in which some live projects deliberately omitted **all** RM – all PM activities can be seen as RM activities. So the relevant experiment would be to omit **formal** RM activity from some live projects, then compare them with suitable controls. However, RM's deliberate omission would be unethical so 'circumstantial' evidence has been sought in academic literature.

This paper now sets out RM theory and process, then explores potential issues around these, before examining whether issues materialise, and RM's contributions and limitations.

3. RM theory

Much RM theory and practice originates from Barry Boehm. For example, Boehm (1991) suggested analysing RM costs and benefits: a threat's cost ('risk exposure') was defined as the product of its materialisation probability and the loss that materialisation would cause. In his made-up example, independent testing reduced (risk exposure + cost of testing) and so was worthwhile, i.e. the answer to *Q1* was 'yes' in this case. This probabilistic approach was expressed as expected utility theory by Kutsch & Hall (2005) and is part of PRINCE2 (pp. 83-84).

4. Potential issues in RM process

It is possible to envisage several potential problems with Boehm's theory and its application:

4.1. Potential issues in identification and analysis

- Unidentified threats may materialise – these cannot be planned for.
- Identified threats' probabilities and impacts may be unknown.
- Costs may be unpredictable.
- The theory treats threats as independent.
- The theory does not consider temporal aspects, such as risk probabilities and costs changing over time.
- The theory is project-centric: it does not explicitly cover 'fringe benefits'[1] or post-project events.

4.2. Potential issues in control

- A materialised threat's cost is its full cost, not a percentage thereof. In Boehm's example, if a critical fault occurs the whole project fails.

Such 'holes' are peered into below.

5. Issues around risk preparation

The weak links in preparation are identification and analysis: specific plans cannot be made for unidentified risks (de Bakker, Boonstra, & Wortmann, 2012, p. 451).

---

1 Such as the 'space race' being responsible for the invention of velcro

5.1. Risk classification

Because there are many risk types (Houston, Mackulak, & Collofello, 2001), preparation may be simplified by risk classification. One system divides risks into technical, project and business risks (Hoodat & Rashidi, 2009), after (McGregor & Sykes, 2001). Because project managers are fulcra between project teams and project sponsors, they need to be well aware of business/external factors.

5.2. Risk identification

Methods include questionnaires and constructing taxonomies (Carr, Konda, Monarch, Ulrich, & Walker, 1993), (Higuera & Haimes, 1996), pondering, interviewing, brainstorming and checklists (Chapman, 1997). Identification methods directly affect RM contributions (Chapman, 2001). Project characteristics (e.g. distributed or co-located) also influence risks and can be used to predict risks unforeseen at project start (Lamersdorf, Munch, del Viso Torre, Sánchez, Heinz, & Rombach, 2010).

One form of checklist is what occurred in similar previous projects. The most common and important threats are

- Creeping user requirements
- Unavailability of key staff
- Reliance on a few key personnel
- Project manager unavailable
- Instability and lack of continuity in project staffing

(Houston, Mackulak, & Collofello, 2001)

In this list, important potential threats are human rather than technical factors.

5.3. Risk analysis

5.3.1. Software metrics in risk analysis

Metrics such as lines of code (LoCs), e.g. (Zhang, 2009), cyclomatic complexity, e.g. (Sadiq, Rahman, Ahmad, Asim, & Ahmad, 2010), and Bayesian networks have been used to predict defects (Fenton & Neil, 1999). Such methods are probabilistic and sensitive (Song, Jia, Shepperd, Ying, & Liu, 2011). **Process** metrics may be better defect-predictors than static code metrics (Moser, Pedrycz, & Succi, 2008), implying that process, and hence human factors, are important. A recent literature review concluded that ‘metrics researchers may need to refine their empirical methodology before they can answer useful empirical questions’ (Kitchenham, 2010).

5.3.2. Probability prediction in risk analysis

Boehm's approach requires 'tremendous' data collection (Hans & Diekmann, 2001), so it is unsurprising that much analysis is subjective (Du, Keil, Mathiassen, Shen, & Tiwana, 2007). Boehm's approach assumes that probabilities are determinable and that individual risks can be treated independently. Pender (2001, p. 80) critiqued such assumptions, and provided an entry into using fuzzy logic. Dependent risks, such as key staff perceiving threats and so quitting the project[2], can be envisaged. There are theoretical techniques for addressing risk dependencies (Wu, Song, Li, Cai, & Li, 2010), (Kwan & Leung, 2011) but these have not yet been fully tested. Also, there is research into statistical modelling, using techniques such as Bayesian networks, e.g. (Wagner, 2010), (Hu, Zhang, Ngai, Cai, & Liu, 2012), to cover this issue.

5.3.3. Other issues in risk analysis

Hall (2001) suggested that assessments are subjective and hence major error-sources – the recommended cure was repeated risk assessments. Later, Kwak & Stoddard, (2004) suggested 'RM must be a natural part of the software development process to be most effective' and suggested several lessons, including

- A documented process does not guarantee the process will be followed.
- As the size and complexity of the project increases, the effort for RM increases exponentially.
- The people that are actually doing the development work … must be empowered … to change practices.

A message that may be drawn from this section is that human factors have a large influence on risk assessment.

5.4. Risk planning

Arguably, the biggest issue around planning is whether it is undertaken. In a survey of software companies, less than half planned for risks (Greer & Conradi, 2009). Regular re-planning has been shown to be important (Hoermann, Schermann, & Krcmar, 2011) – and is fundamental to PRINCE2. Cloud computing is a currently important trend, wherein consumers may have little power over providers, so risk planning may require even more thought (Baldwin, Pym, & Shiu, 2013).

5.5. Risk preparation costs

There appears to be very little literature data about RM costs. Computing various RM strategy combinations' costs is possible (Shan, Jiang, & Huang, 2009): differences would equate to the costs of more or less effective combinations.

---

2 This may have happened in the above-mentioned government IT project (Wintour, 2013).

6. Issues around risk response

Two important issues are discussed in some detail here.

6.1. Scope creep

The top factor in Houston, Mackulak, & Collofello's list (2001) is often known as 'scope creep'. Implementation and test costs will grow as creep occurs. 'Avoidable' creep can be managed by applying costs to change requests, but some creep is inevitable 'from just not knowing what the system is supposed to do' (Pühl & Fahney, 2011). Creep also arises from stopping requirements analysis too early (Berry, Czarnecki, Antkiewicz, & AbdElRazik, 2010). Judging when to cease requirements analysis is hence part of RM.

It may be possible to intuitively select which requirements to implement but a formal method has been proposed (Jung, 1998). This depends on all requirements' implementation costs and values being known. While agile projects accept that scopes change and hence discover requirements 'on the fly' (Williams & Cockburn, 2003), arguably this method could be applied at each iteration to explore which new requirements would provide best overall value.

Modern software tends to involve complex supply-chains, thus increasing RM's complexity and cost – a proposed solution is 'systemic' risk analysis (Alberts, Dorofee, Creel, Ellison, & Woody, 2011).

A message which may be drawn from this section is that scope creep needs strong RM.

6.2. Software testing

This factor is considered because unless software does what is required, the effort to create it is likely to have been wasted. Hence risk control includes testing.

A simplistic way to predict testing costs would be summing predictable spend, factoring in 'opportunity' costs (cost of capital and taking testers from other projects). This assumes that all desirable testing can be achieved in predicted periods but it is impossible to test everything[3], so studies have concentrated on deciding how and what to test, and modelling test costs.

Agile development models include client-tests as development proceeds, not least so the product delivers truly required functionality. It should be possible to predict client-tests costs, based on the time allotted to such tests.

---

3 *When to stop testing?* is by no means a new question (Musa & Ackerman, 1989)

Costs for tests can be predicted by quasi-renewal methods (Pham & Wang, 2001), fuzzy-logic estimations (Engel & Last, 2007) or entropy/chaos methods (Song, Wu, Li, Cai, & Li, 2010). The initial quasi-renewal research assumed that all faults are independent and that all detected faults are removed immediately, without introducing new faults ('perfect' debugging). Later work has tried to deal with 'imperfect' debugging (Kapur, Pham, Anand, & Yadav, 2011).

With regard to deciding where to concentrate testing and debugging, there is currently no clear answer: Shin & Williams (2013) state that fault and vulnerability prediction models require 'significant improvement', while Catal (2011) concluded that practical prediction techniques do not yet work.

Testing becomes even more difficult in web applications. These typically consist of JavaScript, HTML and CSS, and run in different browsers, leading to some techniques being nearly impossible to apply (Dallmeier, Burger, Orth, & Zeller, 2013). Similarly, system 'heisenbugs' are very difficult to handle (Spinellis, 2013).

The overall message is that it is difficult to predict how much testing will be sufficient and hence how much testing will cost.

6.3. Issues around implementation: the human factor

Ineffective RM implementation might be seen as a problem with people, rather than with RM, but if RM's benefits are to be gained, RM needs to be perceived as achievable. Further, human factors pervade the potential issues noted so far. This area can be probed by considering whether active RM is done, and the drivers for such decisions.

Project managers tend to delay dealing with risk, thus harming their perceptions of RM – and their projects (Kutsch & Hall, 2005). This is due to imperfect knowledge (of probabilities, impacts and consequences), avoiding alarm and uncertainty (such as whether a threat is 'real') and unawareness (e.g. of non-technical risk). Perfectly rational RM decisions are impossible, although pleasing clients is rational.

While error (imperfect knowledge) is inevitable, deliberate ignorance ('irrelevance') is a human factor that can be mitigated (Kutsch & Hall, 2010). For example, managers may choose to concentrate on 'tame' problems and ignore risks that are not immediately pertinent.

While IT project failure-causes are well known, there is little evidence that this knowledge is used or that risks really are manageable (de Bakker, Boonstra, & Wortmann, 2010). This research concluded that RM only works in specific situations but did not state these conditions.

In a further sample, only 14% of projects executed plans in response to materialised threats (13 had made plans). Managers did not perceive preparation as useful or cost-effective, suspending action until threats materialised (Kutsch, Denyer, Hall, & Lee-Kelley, 2012). In some cases, delay was associated with lack of authority or with overdesigned, rule-based environments.

6.4. Does preparation determine response?

It can be asked whether preparation prompts response selection or whether managers just react 'on the fly' as risks materialise. If the latter is true, and projects still generally succeed, then preparation may be pointless.

There is evidence that risk preparation is **not** helpful, and is not done (Kutsch, Denyer, Hall, & Lee-Kelley, 2012) but this is not yet conclusive. This work showed experienced managers not making, or ignoring, plans but did not say what happened to the projects. In other fields, preparedness is beneficial (Järveläinen, 2013).

7. Evidence: the case for and against RM

This section considers whether RM's benefits are outweighed by its costs. There is much literature guidance on how to do RM, and a large array of models for different project- and industry-types but no hard data for RM costs have been found, but there is 'circumstantial' evidence both for and against RM.

McConnell (1997) suggested that to obtain a 50–70% chance of avoiding time overrun, RM requires only 5% of the total project budget. Raz & Michael (2001) found that when used, RM practices seemed to work, mostly improving time and budget achievements. They found that simple tools such as impact assessment, classification, periodic reviews, subcontractor management and customer surveys were likely to be associated with better PM performance. Unfortunately, they did not investigate the tools' implementation-costs, or the drivers for their adoption.

RM is associated with successful projects (Verner, Cox, & Bleistein, 2006). Conversely, around half of troubled projects are associated with ineffective RM (Nelson, 2007).

Other research concluded that RM occasionally contributes to success (de Bakker, Boonstra, & Wortmann, 2012), and that success is attributed to social effects – making teams perceive and accept risk, and stimulating action by stakeholders (de Bakker, Boonstra, & Wortmann, 2011). Similarly, there is evidence of 'self-efficacy bias': managers with strong self-belief underestimate risks, thus continuing with projects that should be halted. Also, whether the risk comes from endogenous or exogenous affects reactions (Jani, 2011). RM has been shown to be a source of competitive advantage (Elahi, 2013).

There are, of course, projects that go horribly wrong ('black swans'): their likelihood is estimated to be 17%. These failures have been attributed to optimism bias (Budzier & Flyvbjerg, 2011) and to 'impossible' (very low probability) problems occurring. Perceptions of 'impossibility' may be due to their causal events' rarity (Buhl, 2012). Also, political (exogenous) factors play a significant role (Budzier & Flyvbjerg, 2011).

It has also been suggested that enforcing RM is counter-productive, leading to a 'vicious cycle of unsatisfactory results' and rules being promulgated but ignored (Wiesche, Schermann, & Krcmar, 2013).

8. Conclusions and further work

There is no hard evidence for RM costs in the literature, but there is a claim that a small investment is likely to prevent over-runs. RM is associated with successful projects and competitive advantage, and its absence is associated with unsuccessful projects. So the answer to *Q1* is a qualified 'yes'.

Concerning *Q2* and *Q3*, where RM works, it seems to act by improving human factors. Where RM fails, this is due to human factors – either RM is misapplied (e.g. by not implementing plans, thus wasting the time spent preparing those plans), or threats are deliberately ignored or low-probability, high-impact risks are not considered. Its limits hence lie in how well it is applied and how alive managers are to their own limitations and external threats, and how much power they have to deal with these. Because RM can work **when used**, *Q4* is redundant but there is definitely room for improvement in RM.

Further work would include directly examining omission and retention of RM. To do this effectively and ethically, all involved would need to be unaware of the experiment and the projects would need to be unimportant but not trivial.

In summary, RM does not appear to be 'broken'. Rather, it is an inexact art that does not guarantee success. In keeping with risk assessment's statistical basis, RM can reduce the likelihood of failure but it will not prevent black swans from occasionally devouring resources and careers.

9. Works Cited

Alberts, C. J., Dorofee, A. J., Creel, R., Ellison, R. J., & Woody, C. (2011). A Systemic Approach for Assessing Software Supply-Chain Risk. *Proceedings of 44th Hawaii International Conference on System Sciences* (pp. 1-8). Kauai, Hawaii: IEEE. http://dx.doi.org/10.1109/HICSS.2011.36

Baldwin, A., Pym, D., & Shiu, S. (2013). Enterprise information risk management: Dealing with cloud computing. In S. Pearson, & G. Yee, *Privacy and Security for Cloud Computing* (pp. 257-291). London, UK: Springer. http://dx.doi.org/10.1007/978-1-4471-4189-1_8

Berry, D. M., Czarnecki, K., Antkiewicz, M., & AbdElRazik, M. (2010). Requirements Determination is Unstoppable: An Experience Report. *Proceedings of 18th IEEE International Requirements Engineering Conference* (pp. 311-316). Sydney, NSW, Australia: IEEE. http://dx.doi.org/10.1109/RE.2010.44

Boehm, B. W. (1991). Software Risk Management: Principles and Practices. Software, 8 (1), 32-41. http://dx.doi.org/10.1109/52.62930

Budzier, A., & Flyvbjerg, B. (2011). Double whammy–How ICT projects are fooled by randomness and screwed by political intent. *Saïd Business School working papers*. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2069975

Buhl, H. U. (2012). The Contribution of Business and Information Systems Engineering to the Early Recognition and Avoidance of “Black Swans” in IT Projects. *Business & Information Systems Engineering*, 4 (2), 55-59. http://dx.doi.org/10.1007/s12599-012-0206-8

Carr, M. J., Konda, S. L., Monarch, I., Ulrich, F. C., & Walker, C. F. (1993, June). *Taxonomy-based risk identification*. No. CMU/SEI-93-TR-06. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST. Retrieved May 2, 2013, from http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA266992

Catal, C. (2011). Software fault prediction: A literature review and current trends. *Expert Systems with Applications*, 38 (4), 4626-4636. http://dx.doi.org/10.1016/j.eswa.2010.10.024

Centre for economic and social inclusion. (2013). *Universal credit*. Retrieved April 27, 2013, from Centre for economic and social inclusion: policy guides: http://www.cesi.org.uk/keypolicy/universal-credit

Chapman, C. (1997). Project risk analysis and management-- PRAM the generic process. *International Journal of Project Management*, 15 (5), 273-281. http://dx.doi.org/10.1016/S0263-7863(96)00079-8

Chapman, R. J. (2001). The controlling influences on effective risk identification and assessment for construction design management. *International Journal of Project Management*, 19 (3), 147-160. http://dx.doi.org/10.1016/S0263-7863(99)00070-8

Dallmeier, V., Burger, M., Orth, T., & Zeller, A. (2013). WebMate: Generating Test Cases for Web 2.0. *Lecture Notes in Business Information Processing*, 133, 55-69. http://dx.doi.org/10.1007/978-3-642-35702-2_5

de Bakker, K., Boonstra, A., & Wortmann, H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management*, 28 (5), 493-503. http://dx.doi.org/10.1016/j.ijproman.2009.07.002

de Bakker, K., Boonstra, A., & Wortmann, H. (2011). Risk management affecting IS/IT project success through communicative action. *Project Management Journal*, 42 (3), 75-90. http://dx.doi.org/10.1002/pmj.20242

de Bakker, K., Boonstra, A., & Wortmann, H. (2012). Risk managements' communicative effects influencing IT project success. *International Journal of Project Management,* 30, 444-457. http://dx.doi.org/10.1016/j.ijproman.2011.09.003

DeMarco, T., & Lister, T. (2003). *Waltzing with Bears: Managing risk on software projects*. New York, New York, USA: Dorset House. http://www.systemsguild.com/pdfs/bearsample.pdf

Du, S., Keil, M., Mathiassen, L., Shen, Y., & Tiwana, A. (2007). Attention-shaping tools, expertise, and perceived control in IT project risk assessment. *Decision Support Systems*, 43 (1), 269-283. http://dx.doi.org/10.1016/j.dss.2006.10.002

Dunleavy, P., Margetts, H., Bastow, S., & Tinkler, J. (2008). *Digital Era Governance: IT Corporations, the State, and e-Government*. Oxford, UK: Oxford University Press. http://ideas.repec.org/b/oxp/obooks/9780199547005.html

Elahi, E. (2013). Risk management: the next source of competitive advantage. *Foresight*, 15 (2), 117-131. http://dx.doi.org/10.1108/14636681311321121

Engel, A., & Last, M. (2007). Modeling software testing costs and risks using fuzzy logic paradigm. *The Journal of Systems and Software*, 80 (6), 817-835. http://dx.doi.org/10.1016/j.jss.2006.09.013

Fenton, N. E., & Neil, M. (1999). A critique of software defect prediction models. *IEEE Transactions on Software Engineering*, 25 (5), 675-689. http://dx.doi.org/10.1109/32.815326

Greer, D., & Conradi, R. (2009). Software project initiation and planning – an empirical study. *IET Software*, 3 (5), 356-368. http://dx.doi.org/10.1049/iet-sen.2008.0093

Hall, D. C. (2001). Development program risk management: a case study. *INCOSE Proceedings of a symposium on risk management*, (p. 3)

Hans, S., & Diekmann, J. (2001). Approaches for Making Risk-Based Go/No-Go Decision for International Projects. *Journal of Construction Engineering and Management*, 127 (4), 300-308. http://dx.doi.org/10.1061/(ASCE)0733-9364(2001)127:4(300)

Higuera, R. P., & Haimes, Y. Y. (1996, June). *Software Risk Management. No. CMU/SEI-96-TR-012. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST*. Retrieved May 2, 2013, from http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA310913

Hillson, D. (2002, April 2002). *What is risk. Results from a survey exploring definitions*. Retrieved May 2, 2013, from http://www.takkabutr.com/EAU/MIS-2010-2553-EAU/MIS-2010/RISK/RiskDefinition%2011Apr02.pdf

Hoermann, S., Schermann, M., & Krcmar, H. (2011). When to manage risks in IS projects: An exploratory analysis of longitudinal risk reports. *Proceedings of 10th International Conference on Wirtschaftsinformatik*. Zurich, Switzerland. http://vmkrcmar23.informatik.tu-muenchen.de/1857/1/WI_2011_When_to_manage_risks_in_IS_projects.pdf

Hoodat, H., & Rashidi, H. (2009). Classification and Analysis of Risks in Software Engineering. *World Academy of Science, Engineering and Technology*, 56. http://ebitik.com/wp-content/uploads/kitabevi/9645.pdf

Houston, D. X., Mackulak, G. T., & Collofello, J. S. (2001). Stochastic simulation of risk factor potential effects for software development risk management. *Journal of Systems and Software*, 59 (3), 247-257. http://dx.doi.org/10.1016/S0164-1212(01)00066-8

Hu, Y., Zhang, X., Ngai, E., Cai, R., & Liu, M. (2012). Software project risk analysis using Bayesian networks with causality constraints. *Decision Support Systems*, in press. http://dx.doi.org/10.1016/j.dss.2012.11.001

Hubbard, D. W. (2009). *The failure of risk management: Why it's broken and how to fix it*. Hoboken, New Jersey, USA: John WIley & Sons, Inc. http://www.weibnc.com/wp-content/uploads/brkpdfs/The-Failure-of-Risk-Management-Why-Its-Broken-and-How-to-Fix-It-by-Douglas-W-Hubbard-Brilliant.pdf

Jani, A. (2011). Escalation of commitment in troubled IT projects: Influence of project risk factors and self-efficacy on the perception of risk and the commitment to a failing project. *International Journal of Project Management*, 29 (7), 934-945. http://dx.doi.org/10.1016/j.ijproman.2010.08.004

Järveläinen, J. (2013). IT incidents and business impacts: Validating a framework for continuity management in information systems. *International Journal of Information Management*, 33 (3), 583-590. http://dx.doi.org/10.1016/j.ijinfomgt.2013.03.001

Jørgensen, M. (2013). Myths and Over-Simplifications in Software Engineering. *Lecture Notes on Software Engineering*, 1 (1). http://dx.doi.org/10.7763/LNSE.2013.V1.2

Jung, H.-W. (1998). Optimizing Value and Cost in Requirements Analysis. *Software*, 15 (4), 74-78. http://dx.doi.org/10.1109/52.687950

Kapur, P. K., Pham, H., Anand, S., & Yadav, K. (2011). A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation. *IEEE Transactions on Reliability*, 60 (1), 331-340. http://dx.doi.org/10.1109/TR.2010.2103590

Kitchenham, B. (2010). What's up with software metrics? – A preliminary mapping study. *Journal of Systems and Software*, 83 (1), 37-51. http://dx.doi.org/10.1016/j.jss.2009.06.041

Kutsch, E., Denyer, D., Hall, M., & Lee-Kelley, E. (2012, August 07). Does risk matter? Disengagement from risk management practices in information systems projects. *European Journal of Information Systems*. http://dx.doi.org/10.1057/ejis.2012.6

Kutsch, E., & Hall, M. (2005). Intervening conditions on the management of project risk: Dealing with uncertainty in information technology projects. *International Journal of Project Management*, 23 (8), 591-599. http://dx.doi.org/10.1016/j.ijproman.2005.06.009

Kutsch, E., & Hall, M. (2010). Deliberate ignorance in project risk management. International *Journal of Project Management*, 28 (3), 245-255. http://dx.doi.org/10.1016/j.ijproman.2009.05.003

Kwak, Y. H., & Stoddard, J. (2004). Project risk management: lessons learned from software development environment. *Technovation*, 24, 915–920. http://dx.doi.org/10.1016/S0166-4972(03)00033-6

Kwan, T. W., & Leung, H. K. (2011). A Risk Management Methodology for Project Risk Dependencies. *IEEE transactions on software engineering*, 37 (5), 635-648. http://dx.doi.org/10.1109/TSE.2010.108

Lamersdorf, A., Munch, J., del Viso Torre, A. F., Sánchez, C. R., Heinz, M., & Rombach, D. (2010). A Rule-Based Model for Customized Risk Identification in Distributed Software Development Projects. *Proceedings of 5th IEEE International Conference on Global Software Engineering* (pp. 209-218). IEEE. http://dx.doi.org/10.1109/ICGSE.2010.32

McConnell, S. (1997). *Software Project Survival Guide: How to Be Sure Your First Important Project Isn't Your Last*. Redmond, Washington, USA: Microsoft Press. http://www.stevemcconnell.com/sg.htm

McGregor, J. D., & Sykes, D. A. (2001). *A Practical Guide to Testing Object-Oriented Software*. Upper Saddle River, New Jersey, USA: Addison-Wesley Professional

Moser, R., Pedrycz, W., & Succi, G. (2008). A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. P*roceedings of ACM/IEEE 30th International Conference on Software Engineering*. Leipzig, Germany: ACM/IEEE. http://dx.doi.org/10.1145/1368088.1368114

Musa, J.D. & Ackerman, A.F. (1989) Quantifying software validation: when to stop testing? *IEEE Software*, 6 (3), 19-27. http://dx.doi.org/ 10.1109/52.28120

Nelson, R. R. (2007). IT project management: infamous failures, classic mistakes, and best practices. *MIS Quarterly Executive*, 6 (2), 67-78. http://misqe.org/ojs2/index.php/misqe/article/view/128

Pühl, S., & Fahney, R. (2011). How to assign cost to "Avoidable Requirements Creep": A step towards the waterfall's agilization. *Proceedings of IEEE 19th International Requirements Engineering Conference* (pp. 307-312). Trento, Italy: IEEE. http://dx.doi.org/10.1109/RE.2011.6051623

Peltier, T. R. (2004). Risk Analysis and Risk Management. *EDPACS: The EDP Audit, Control, and Security Newsletter*, 32 (3), 1-17. http://dx.doi.org/10.1201/1079/44581.32.3.20040901/83426.1

Pender, S. (2001). Managing incomplete knowledge: Why risk management is not sufficient. *International Journal of Project Management*, 19 (2), 79-87. http://dx.doi.org/10.1016/S0263-7863(99)00052-6

Pham, H., & Wang, H. (2001). A quasi-renewal process for software reliability and testing costs. *IEEE Transactions on Systems, Man and Cybernetics*, Part A: Systems and Humans, 31 (6), 623-631. http://dx.doi.org/10.1109/3468.983418

Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledg*e. Newtown Square, PA, USA: Project Management Institute

Raz, T., & Michael, E. (2001). Use and benefits of tools for project risk management. *International Journal of Project Management*, 19 (1), 9-17. http://dx.doi.org/10.1016/S0263-7863(99)00036-8

Royal College of Nursing. (2013). *Pay rates 2013/14*. Retrieved 04 30, 2013, from http://www.rcn.org.uk/support/pay_and_conditions/pay_rates_201314

Sadiq, M., Rahman, A., Ahmad, S., Asim, M., & Ahmad, J. (2010). esrcTool: A Tool to Estimate the Software Risk and Cost. *Proceedings of Second International Conference on Computer Research and Development*, (pp. 886-890). Kuala Lumpur, Malaysia. http://dx.doi.org/10.1109/ICCRD.2010.29

Shan, X., Jiang, G., & Huang, T. (2009). The Optimal Model of Software Project Cost Risk Resolution Strategy Combination Based on COCOMOII. *Proceedings of International Conference on Management and Service Science*, (pp. 1-4). http://dx.doi.org/10.1109/ICMSS.2009.5303576

Shin, Y., & Williams, L. (2013). Can traditional fault prediction models be used for vulnerability prediction? *Empirical Software Engineering*, 18 (1), 25-29. http://dx.doi.org/10.1007/s10664-011-9190-8

Software Engineering Institute. (2012). *A Framework for Software Product Line Practice, Version 5.0*. Retrieved May 2, 2013, from Software Engineering Institute: http://www.sei.cmu.edu/productlines/frame_report/technicalRM.htm

Song, H., Wu, D., Li, M., Cai, C., & Li, J. (2010). An Entropy Based Approach for Software Risk Assessment: A Perspective of Trustworthiness Enhancement. *Proceedings of 2nd International Conference on Software Engineering and Data Mining* (pp. 575-578). Chengdu, China: IEEE

Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011). A General Software Defect-Proneness Prediction Framework. *IEEE Transactions on Software Engineering*, 37 (3), 356-370. http://dx.doi.org/10.1109/TSE.2010.90

Spinellis, D. (2013). Systems Software. *IEEE Software*, 30 (3), 18-19. http://dx.doi.org/10.1109/MS.2013.61

The Standish Group. (1995). *The Standish Group Report*. Retrieved April 28, 2013, from http://www.projectsmart.co.uk/docs/chaos-report.pdf

The Stationery Office. (2009). *Managing successful projects with PRINCE2* (5th ed.). Norwich, UK: The Stationery Office

Verner, J., Cox, K., & Bleistein, S. J. (2006). *Proceedings of the 2006 ACM/IEEE international symposium on empirical software engineering*, 154-163. http://dx.doi.org/10.1145/1159733.1159758

Wagner, S. (2010). A Bayesian network approach to assess and predict software quality using activity-based quality models. *Information and Software Technology*, 52 (11), 1230-1241. http://dx.doi.org/10.1016/j.infsof.2010.03.016

Wiesche, M., Schermann, M., & Krcmar, H. (2013). When IT Risk Management Produces More Harm Than Good: The Phenomenon of 'Mock Bureaucracy'. *Proceedings of 46th Hawaii International Conference on System*. http://vmkrcmar23.informatik.tu-muenchen.de/2114/3/When_IT_RM_produces_more_harm_than_good_v7.pdf

Williams, L., & Cockburn, A. (2003). Agile software development: it's about feedback and change. *IEEE Computer*, 36 (6), 39-43

Williams, T. (2005). Assessing and Moving on From the Dominant Project Management Discourse in the Light of Project Overruns. *IEEE Transactions on Engineering Management*, 52 (4). http://dx.doi.org/10.1109/TEM.2005.856572

Wintour, P. (2013, March 5). *Universal credit benefits system 'in meltdown', claims Labour*. Retrieved April 28, 2013, from *The Guardian*: http://www.guardian.co.uk/politics/2013/mar/05/universal-credits-meltdown-claims-labour

Wu, D., Song, H., Li, M., Cai, C., & Li, J. (2010). Modeling risk factors dependence using Copula method for assessing software schedule risk. *Proceedings of 2nd International Conference on Software Engineering and Data Mining* (pp. 571-574). Chengdu, China: IEEE

Zhang, H. (2009). An Investigation of the Relationships between Lines of Code and Defects. *Proceedings of IEEE International Conference on Software Maintenance*. Edmonton, Alberta, Canada: IEEE. http://dx.doi.org/10.1109/ICSM.2009.5306304