HANDLING DEVELOPMENT TIME UNCERTAINTY IN AGILE RELEASE PLANNING

Kevin Logue and Kevin McDaid
Software Technology Research Centre, Dundalk Institute of Technology, Dundalk, Ireland

Keywords: Release Planning, Requirement Prioritization, Uncertainty, Agile Methodology.

Abstract: When determining the functionality to complete in upcoming software releases, decisions are typically based upon uncertain information. Both the business value and cost to develop chosen functionality are highly susceptible to uncertainty. This paper proposes a relatively simple statistical methodology that allows for uncertainty in both business value and cost. In so doing it provides key stakeholders the ability to determine the probability of completing a release on time and to budget. The technique is lightweight in nature and consistent with existing agile planning practices. A case study is provided to demonstrate how the method may be used.

1 INTRODUCTION

The importance of release planning within a modern software development project cannot be understated. Even as recent as 2004 studies have shown that a considerable number of software development projects were found to finish over budget, late or missing key functionality (Johnson, 2006). Reasons for these overruns include unrealistic goals, inaccurate estimates, an ill-defined system, poor monitoring of project status and poor project management (Charette, 2008). Through the implementation of an honest and reliable release plan the chance of completing a project within the allocated time and budget can increase considerably (Cohn, 2006)

The task of selecting what functionality to complete in upcoming software releases is a complex process (Regnell & Brinkkemper, 2005). Typically a development has a finite amount of resources available, forcing developers to prioritize certain functionality or user stories to determent of others. The question of which stories to prioritize requires considerable understanding of both the technical complexity of the project and also a keen understanding of the product's market.

Little (Little, 2005) identifies four major areas of uncertainty that affect a software development project; market uncertainty, technical uncertainty, project duration and project dependencies. Market uncertainty relates to the understanding of customers' needs. Attempting to satisfy a smaller client base offers considerable less uncertainty, as developers can more easily prioritize desired requirements. Technical uncertainty occurs where the technical needs are not fully understood. Typically, technical uncertainty arises when using a new technology or when a problem is not well defined. The duration of a project is a major contributing factor to uncertainty. The more time required to complete a project the more likely that either market or technical uncertainties may arise. Project dependencies refer to the flexibility of a project, and in the event that one part of the project must be completed before another part starts the project's ability to absorb the occurrence of uncertain events diminishes.

This paper proposes a relatively simple statistical methodology designed to help developers to determine the likely completion time for a plan and its expected business value. The method recognises uncertainty in business value, required effort and resources available to develop a software project. The method is demonstrated and investigated through a case study. A comparison with the results derived using an alternative methodology due to Cohn (2006) is also presented.

The remainder of the paper is set out as follows. Section 2 discusses related work in the area of release planning, requirement prioritisation and project uncertainty. Section 3 describes the methodology while Section 4 explains how it can fit

Logue K. and McDaid K. (2008).
HANDLING DEVELOPMENT TIME UNCERTAINTY IN AGILE RELEASE PLANNING.
In *Proceedings of the Third International Conference on Software and Data Technologies - SE/GSDCA/MUSE*, pages 192-199
DOI: 10.5220/0001885401920199

within a development process. Section 5 presents a case study designed to explore the feasibility of the methodology. Section 6 compares the results and recommendations with those derived from an alternative statistical methodology due to Cohn (2006) Section 7 contains some concluding remarks on the methodology.

2 BACKGROUND

Release planning in Extreme Programming (XP) is carried out through the “Planning Game”. This is used to select the requirements to include in the next and upcoming releases. The process begins by first eliciting all requirements and expressing them in terms of user stories where a user story is a representation of a requirement written in a high level form free of technical jargon. The team is then asked to provide estimates of the size of each story in terms of ideal development days or story points, where ideal development days represents the length of time it would take a developer to complete a story assuming that they are in a position to engage solely in development activity. A story point on the other hand is an amalgamation of the size of task, its complexity and also its risk (Cohn, 2006). Next, the team, usually under the direction of the customer, prioritises the stories. This time consuming activity requires a series of decisions based on complex cost-benefit analyses to determine which stories offer maximum overall benefit.

Prioritisation techniques can be absolute or relative (Karlsson et al., 2007). An absolute technique assigns a priority to each requirement based upon its importance to the project. An example is the MoSCoW prioritisation technique which separates requirements into “must have”, “should have”, “could have” and “would like to” have sub groups. An alternative technique is the “Cost-Value” approach (Karlsson & Ryan, 1997). This approach compares all possible pairs of requirements, in order to determine which requirement is of higher priority. However while conceptually simple the method does not offer a simple means for developers to compare stories and as such does not guarantee maximum value for minimum cost (Jung, 1998).

A more involved approach, termed EVOLVE (Greer & Ruhe, 2004), takes into account stakeholder priorities, requirement constraints and also effort limits. The method is not ideal for use within agile processes for two reasons. Firstly, it requires the coordination of stakeholders in prioritizing requirements. Secondly, the effort estimation is based on requirements and not explicitly on the constituent tasks that deliver those requirements. While this is a common approach in estimating, estimation at the task level has been shown to offer a more realistic representation of the effort required (McConnell, 2006).

Most research to date has taken a deterministic approach to the problem of release planning, ignoring the inherent uncertainty in the development time and business value of software. An exception is the Fuzzy Effort Constraint method (Ngo-The et al., 2004), in which the development team provides minimum, maximum and most likely values to allow assuming estimates take the form of fuzzy triangular numbers. Once these estimates are in hand the method compares the required resources with the resources that are available to the project using a probabilistic approach. This comparison yields a measure of the degree to which the resource constraint has been met. In so doing it provides the decision makers with a series of possible plans with varying degrees of satisfaction and the level to which the resource constraint has been met. While we also adopt triangular models for development time we do so in the context of a probabilistic model which we believe is more intuitive and is structured to reflect the estimation process in agile methods.

We are not the first to propose a statistical approach. While not detailing any evaluation activity Cohn (2006) discusses the use of both a feature buffer and a schedule buffer. When using a feature buffer the team first commit to a set of user stories to be completed within the next release. The remaining stories are placed in a buffer, if time allows these will be completed. A schedule buffer is different in that it acknowledges the uncertainty within the estimates provided by the team. Using a schedule buffer developers are asked to provide both a 50% estimate and a 90% estimate for the size of each feature. Using these two values, and based on a normal distribution, the additional time to allow for the project to ensure completion with 90% certainty is determined. Section 6 compares the results of this approach with the one outlined herein for the data presented through the Case Study in Section 5.

3 METHODOLOGY

3.1 Overview

Due to the nature of software development projects uncertainty is an inherent problem (Ziv et al., 1996).

The method proposed herein provides key stakeholders with the opportunity to manage in a more effective manner the uncertainty in the release planning process. It has been designed with two scenarios in mind. The first scenario exists when the development team have a preconceived plan. In this situation the methodology can be used to determine likely value and the time required to complete the project to a satisfactory probability.

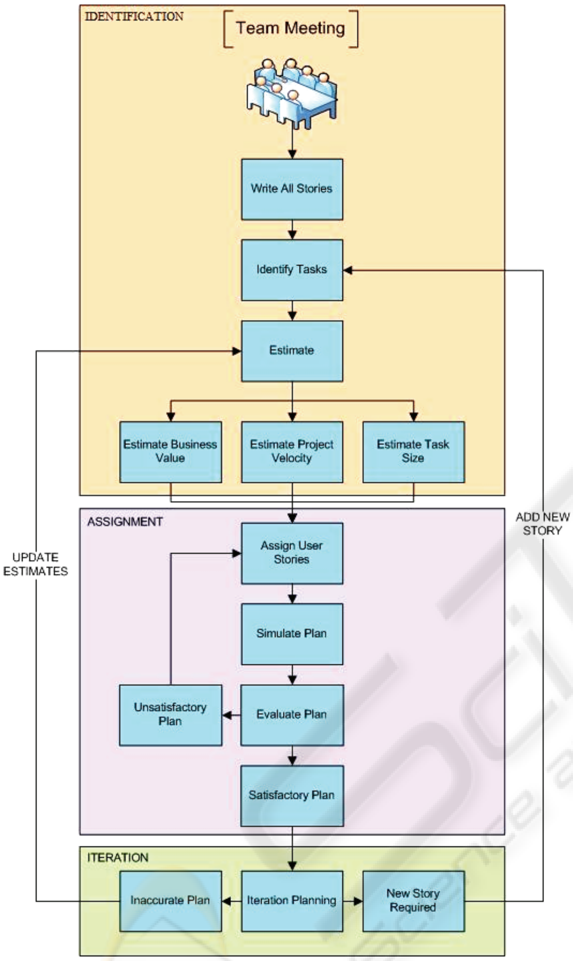

Figure 1: Release Planning Methodology.

The second scenario attempts to support the decision maker by proposing a set of high valued story assignment. This technique is not designed to generate a single optimal combination of stories; rather it is concerned with developing a set of story combinations which are optimal or near optimal. The decision makers can then choose from these combinations. In this way the method guides the decision makers, an approach advocated in many works (Ruhe & Saliu, 2005).

3.2 Expert Knowledge

Our method is compatible with the existing planning game in XP and involves the gathering of all user stories and estimating both the size of each story, expected value and also project velocity. The method recognises that these values are subject to uncertainty.

As with XP's Planning Game this process should involve the entire development team and a customer representative called the product owner. In an ideal situation the product owner will be an end user of the product or actual project customer. However, more often than not the product owner is a member of the sales or management team who have expert knowledge of the market and customer needs.

The process, illustrated in Figure 1, starts with the identification of user stories. Once all the candidates have been identified the size of each story is estimated by the development team. Unlike traditional agile methods this research suggests a finer level of granularity be used and that at this point the constituent tasks of each story be identified. The advantage of this approach is twofold. Firstly overlap between stories can be easily recognized and isolated, simplifying the dependencies that may exist between stories. Secondly this approach encourages developers to examine the underlying architecture of the project (Nord et al., 2000) and places them in a better position to provide more accurate estimates (McConnell, 2006).

Once all tasks have been identified the development team is asked to estimate the size of each task in ideal development days. Traditionally developers are asked to estimate single values, however, to allow for uncertainty in the size of stories, this methodology asks for the provision of three estimates, a most likely, a pessimistic and an optimistic value, in ideal days for the size of each task.

Next the development team estimates the project velocity. The project velocity represents how much work can be carried out during an iteration and can be found by examining previous iterations while taking into account the experience of the development team. Similar planning approaches are used in other agile methods with extensive guidance given in (Cohn, 2006) and (Beck & Andres, 2001). Due to the uncertain nature of these estimates and the variation across iterations, the team is again asked to provide most likely, pessimistic and optimistic values.

With both story size estimates and the expected velocity in hand the team estimates the business value of each story. Due to the difficulty in the provision of monetary estimates for software projects, a simplified 1 to 10 scale is used (Cohn, 2006). This technique allows the team to express the value of a story in relation to others (Boehm, 1981). For example in the event that a story was assigned a value of 2 it is half the value of a story assigned 4. Once again to allow for the inherent uncertainty within these values a most likely, pessimistic and optimistic values are elicited for each user story is required.

A factor which can affect the value of a story is the release in which it is completed. Delaying the time to market of a particular feature can reduce its financial return. To allow for this, the methodology asks the team to provide a weighting in the range 0 to 1 to each story in the event it is completed in a release other than the first. For example taking a story with a value of 4, a development team may weight it as 0.8 in the second release. As such if this story is completed in the first release, its value will remain 4, however if it is completed in the second release its value will have decreased to 3.2.

3.3 Assignment

Once all estimations have been gathered the team is ready to begin the assignment phase. The goal of this phase is to explore possible plans by one of two techniques, either an automated process or through manually exploring plans to decide on an optimal one using a simple tool developed in Microsoft Excel.

Even for a particularly small development the number of possible plans can be extremely large. As such it may not be possible or practical for a team to determine an optimal assignment of stories, without devoting considerable time and effort. To this end, other work by the first author is examining the use of Genetic Algorithms to automate the exploration of the search space. Genetic Algorithms have been used to good effect in other optimization problems (Greer & Ruhe 2004) and (Ngo-The & Ruhe, 2007).

In this paper the focus is on the manual exploration of plans and as such best represents the situation where the team wish to explore the characteristics of a small number of plans. In this case the methodology can be used to determine the likely duration and value of a release and ensure the project will finish on time and generate a required business value.

To that end, once distributions are known for the size of stories and for the project velocity, Monte Carlo simulation is performed to obtain the distribution for the real time to complete a selection of stories in a release. Similarly, the distribution for the combined business value of the stories can be obtained.

Currently the model can use a Triangular (Miranda, 2002) or a PERT (Douglas, 1978) distribution to represent the possible uncertainty within all estimates. Likely completion time and project value can be statistically simulated from either of these distributions which can be specified based on minimum, most likely and maximum estimates. The Triangular probability distribution is well recognized as a suitable distribution when the true distribution of data is unknown. The PERT distribution is similar to the triangular and preferred in cases where the extreme values are asymmetrically spread about the mode. In this paper the size of tasks, the project velocity and the business value of stories are assumed to follow a PERT distribution.

The methodology is implemented through a spreadsheet tool which takes the inputted estimates and, using Monte Carlo simulation methods, generates statistical distributions for the overall business value and time to complete a selected set of stories. The tool displays these outputs graphically and also shows the probability of completing the given plan within the target release size and also the probability of achieving the target business value where the target times and values are set by the user. Average time and business values are also presented. In the event that the assignment is judged to be of insufficient quality the team can adjust the story assignment and simulate the plan again. Once a plan of sufficient quality is found and the development team is confident in its quality the plan can then be put into practice.

4 ITERATIVE PLANNING

Currently the methodology does not dictate the order in which tasks are to be completed, instead it is more concerned with determining which stories to complete in each release to maximise business value and minimize risk. Iteration planning, as opposed to Release planning, requires considerable more understanding of the technical requirements of a story, and also the skills of each individual within the team. While the methodology could be adopted to account for the extra data required, it is questionable whether the significant additional planning time associated with such an approach would be consistent with agile techniques.

The methodology can be adapted as development proceeds and further information becomes available. As tasks are completed the simulation methods can be used to provide most up to date information on the likelihood of completion on schedule. Information on the time needed to complete one task can also be used to alter the distributions for other related tasks. Again, through the spreadsheet tool, a manual approach can be used to establish the new best plan.

5 CASE STUDY

Following on from the work of (McDaid et al., 2006) in which real data from a small Irish company was used to illustrate an early version of the method, this paper now presents a simple case study with the same firm. The purpose of the study was to investigate the feasibility of the method. Company Z is a small software firm that develops a single very high value product.

The company operates on the basis of quarterly releases of their main product. Typically, a release would include new functionality driven by the needs of key customers, new functionality and modifications to improve existing functionality. They wish to be able to plan two releases in the future, selecting from a wide range of possible features for their innovative product.

Table 1: Story Details.

	Tasks	Business Value		
		min	mode	max
1	1	6	8	9
2	2, 3, 4, 5, 6	6	7	8
3	7, 8, 9, 10, 11, 12, 13	5	7	8
4	14, 15, 16	4	6	7
5	17, 18, 19	4	8	9
6	20, 21, 22, 23	4	6	6
7	24	3	5	8
8	25, 26, 27, 28, 29, 30, 31, 32, 33	8.5	9	9.5
9	34, 35, 36, 37, 38, 39, 40	5	6	8
10	41, 42, 43, 44, 45	5	6	8

The firm uses an agile development process., most closely related to Extreme Programming. While this provides them with a good understanding of the functionality that could be added to the project, they have in the past struggled to provide accurate estimates of the size of stories. This has led to time and cost overruns. These overruns have been exacerbated by the need to perform ongoing maintenance and repair work driven by the needs of their key customers, a practice that impacts on the project velocity through the amount of time that can be spent during iterations on new development.

Table 2: Task Details.

Task	Min	Mode	Max	Task	Min	Mode	Max	Task	Min	Mode	Max
1	2	6	10	**16**	3	4	8	**31**	2	3	4
2	3	3	4	**17**	8	10	25	**32**	0.5	0.5	1
3	4	4	10	**18**	0.1	4	10	**33**	0.5	0.5	1
4	4	4	4	**19**	3	5	6	**34**	4	8	10
5	3	3	8	**20**	2	3	15	**35**	1	1	1
6	1	1	1	**21**	2	2	2	**36**	2	3	4
7	4	4	5	**22**	1	2	3	**37**	1	1	1
8	2	2	2	**23**	2	5	15	**38**	3	4	5
9	2	2	2	**24**	20	25	30	**39**	1	1	1
10	5	6	7	**25**	1	1	3	**40**	1	1	2
11	2	2	2	**26**	3	3	4	**41**	0.5	0.5	0.5
12	1	1	1	**27**	3	3	5	**42**	6	10	15
13	1	4	6	**28**	1	1	2	**43**	3	5	10
14	3	3	4	**29**	1	1	2	**44**	1	1	3
15	8	8	11	**30**	1	1	1	**45**	1	1	1

The case study involved the Chief Technology Officer for the firm who, besides being an expert on the development of the candidate stories, was also, through his regular contact with customers and management, very aware of the relative business importance of the functionality. As such he provided a good representation of the product owner.

Initially, the product owner identified a number of stories for inclusion in upcoming releases. A set of constituent tasks were elicited while ensuring the level of granularity was chosen to isolate activities that overlap between stories. Details of the stories and tasks are given in Table 1. Information on prerequisite or co-requisite stories was also provided. This is not presented as it is not directly relevant to the results of the case study.

To allow for uncertainty in the size of stories, the product owner was then asked to provide three estimates, a most likely, a pessimistic and an optimistic value, in ideal days for the size of task. This data is shown in Table 2. Next the product owner was asked to consider the likely project velocity and estimated the value at 4 ideal days per developer per 5 day working week. Arising from discussion it was found that this value could range from a minimum of 3.5 ideal days to a maximum of 4.5.

To complete the elicitation of the expert information, the difficult issue of the business value of the candidate stories was addressed. Again, minimum, most likely and maximum values were elicited, this time on a scale of 1 to 10. The values, shown in Table 1 should reflect the likely long term financial return of developing the features. Provision of these values, which must combine short term initial return with longer term resulting business, proved to be a difficult task for the participant.

Having obtained the required data, the study then addressed whether the methodology, which provides the stakeholder with information on the uncertainty of planning outcomes, can support the decision maker under two different release planning scenarios. In the first scenario, the study looked at the length of release that should be planned for a specified combination of stories. In the second part, which shall be described later, it asked what functionality should be selected for inclusion in a release of a fixed duration. The case of more than one release was also examined but this work is not detailed here.

The product owner decided that the first release should include Stories 1, 2, 4, 5, 8 and 9 involving the completion of tasks 1-6, 14-19 and 25-40. To determine the distribution for the development time of this set of stories, each individual task time is simulated and the resulting values added. This gives one possible size in ideal days for the release. Repeating this process over a large number of runs, 10,000 say, results in a statistical distribution for the size of the release.

To establish the real time it might take to develop these stories the method combines, again through simulation, the uncertainty in the project velocity, represented by a PERT distribution, with the uncertainty in the size of the stories. In the current model these project velocity values are assumed to be independent in that a high or low velocity for an iteration does not mean the next iteration will witness the same fluctuations.

In this way the method can produce a distribution for the likely duration for developing the proposed plan for the next release. The distribution for the real development time for these stories is shown in Figure 2. Based on 3 developers the average time is found to be 39.4 days or 7.8 weeks. The graph shows that the real time could vary from 34 to 46 days.

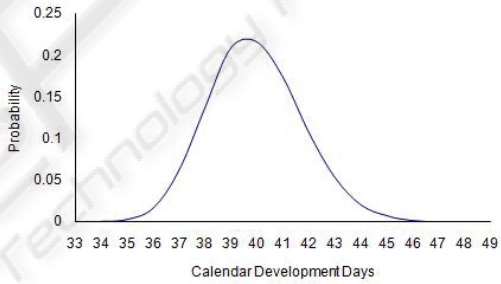

Figure 2: Calendar days to complete plan.

Through the spreadsheet tool the participant was provided with the probability of completing the stories within different release times. In this case the data in Table 3 was provided informing him that these would take on average 39.4 calendar days and that the probability of completion within 8 weeks was 64%. However, to be 90% certain of completing the desired functionality, management would have to plan a release date 9 weeks into the future. The additional 1.2 week period represents the slack time that should be included in the plan to ensure that the release is completed on time. In this scenario the participant felt that the method provided clear reasoning for the inclusion of an extra week and concluded that he would, on this basis, discuss with management the allocation of an additional week to reduce the risk of running overtime.

Next, the study addressed the selection of stories within a constrained release time. In practice, this is a very complicated decision problem that involves balancing the return on investment, as represented by the business value, with the cost of development. Assuming a release of 12 week duration the product owner was, through the tool, provided with information on the distribution of the business value that might result from a selected group of stories.

Table 3: Likelihood of completing plan.

Iterations	6	7	8	9	10
Probability	0%	0.2%	64%	99%	100%

Following a number of iterations the participant settled on a combination that included stories 1, 2, 3, 4, 5, 8, 9 and 10, choosing to reject stories 6 and 7. The distribution for the business value of these stories is shown in Figure 3. The corresponding likelihood of achieving at least certain business values are given in Table 4 below. Note that the expected business value is 58.2. For this combination the probability of completion within the 12 week release was found to be 97%.

Table 4: Business Value.

Value	55	56	57	58	59	60
Probability	98%	92%	78%	55%	30%	11%

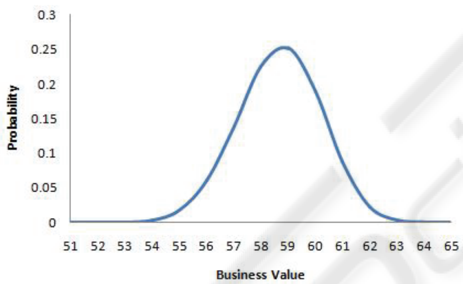

Figure 3: Business Value of plan.

Following discussions it was clear that, while the product owner examined the distribution of likely business value, the participant based his decision of which plan to choose on the average business value of the combination. This was likely due to the fact that for the data provided there were not a number of plans with similar averages value.

6 COMPARISON

Cohn (2006) describes a statistical methodology, based on the normal distribution, for release planning that can be used to specify a project and a feature buffer. It also advocates the elicitation of an upper bound on the size of stories to establish the distribution, in his case using a 90% value. Using approximate methods, based on the standard deviation of the normal distribution, it establishes the average time to complete a set of stories and proposes a time two standard deviations above the average to represent the ideal days that should be scheduled for development. Unlike our method it does not allow for the uncertainty in the project velocity and instead specifies a schedule buffer in terms of ideal development days.

Notwithstanding, it is possible, based on some approximation, to compare this method with ours based on a release plan that selects Stories 1,2, 4, 5, 8 and 9 as outline in the Section 5. Figure 4 shows a plot of the size in ideal days of the selected stories according to both ours and Cohn's methods.

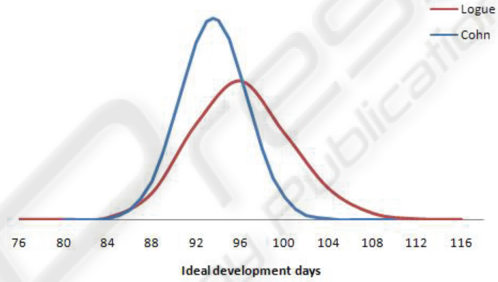

Figure 4: Probability distribution function for size of release in ideal development days using Cohn and Logue methods.

The graph shows that the methods indicate slightly different modal values for the size of the release plans. It also shows that Cohn's method yields a distribution with less variation than ours, due in some part to the asymmetric character of the PERT distribution. These differences are the subject of ongoing research.

7 CONCLUSIONS

The creation of a release plan poses a major difficulty for even the most experienced of development teams. Uncertainty in available resources and business value of candidate requirements makes prioritization a complex and daunting task. Traditional methods for handling uncertainty fail to recognise the often skewed nature of estimates. The method proposed within this paper seeks to support decision makers, it uses probabilistic methods to provide statistical distributions for the time to complete releases and the likely business value. While the method increases the data required in the planning process, it remains relatively lightweight. The method has been designed so as to fit an agile environment however it should be possible to incorporate it within most planning methodologies. The ability to update data and to easily re-evaluate a plan make it well suited for iterative and incremental development methodologies.

The development of the methodology described herein is in its early stages. While method has been shown to have some potential, there is important empirical research to be done to fine tune its application within an agile environment. Important work on the derivation of optimal plans when the number of requirements is reasonably large is also required. These advances will be the focus of future research by the authors.

ACKNOWLEDGEMENTS

This work is supported by the Technological Sector Research measure funded under the NDP and ERDF.

REFERENCES

Beck, K., Andres, C., 2001. Extreme Programming Explained, Addison Wesley, Reading, MA.

Beliakov, G., 2005. Universal Nonuniform Random Vector Generator Based on Acceptance-Rejection, ACM Transactions on Modeling and Computer Simulation, Volume 15, Issue 3, pp. 205 – 232

Boehm, B., 1981. Software Engineering Economics", Prentice Hall, Eaglewood Cliffs, NJ.

Brighton Webs, 2008. Triangular Distribution [website] http://www.brighton-webs.co.uk/distributions/, February 2008

Charette, R., 2008. Why Software Fails, [website] http://spectrum.ieee.org/sep05/1685

Cohn, M., 2006. Agile Estimating and Planning, Prentice Hall.

Douglas, D. E. 1978. PERT and simulation. In *Proceedings of the 10th Conference on Winter Simulation - Volume 1*. H. J. Highland, Ed. Winter Simulation Conference. IEEE Press, Piscataway, NJ, 89-98.

Greer, D., Ruhe, G., 2004. Software Release Planning: An Evolutionary and Iterative Approach, J. Information and Software Technology, vol. 46, issue 4.

Johnson, J., 2006. My Life is Failure: 100 Things You Should Know to be a Successful Project Leader, Standish Group International.

Jung, Ho-Won, 1998. Optimizing Value and Cost in Requirements Analysis, IEE Software, July / August,

Karlsson, L., Thelin, T., Regnell, B., Berander, P., Wohlin, C., 2007. Pair-wise Comparisons Versus Planning Game Partitioning – Experiments on Requirements Prioritisation Techniques, Empirical Software Engineering, Volume 12, Issue 1, pp 3-33

Karlsson, J., Ryan, K., 1997. A Cost-Value Approach for Prioritizing Requirements, IEEE Software, Volume 14,, pp. 67-74

Little, T., 2005. Context Adaptive Agility: Managing Complexity and Uncertainty, IEEE Software, April, Vol. 22, No. 2

McConnell, S., 2006, Software Estimation: Demystifying the Black Art, Microsoft Press

McDaid, K., Greer, D. , Keenan, F., Prior, P., Taylor, P., Coleman, G., 2006. Managing Uncertainty in Agile Release Planning, Proc. 18th Int. Conference on Software Engineering and Knowledge Engineering (SEKE'06), pp 138-143.

Miller, G., 1997. OO Process and Metrics for Effort Estimation, Conference on Object Orientated Programming Systems Languages and Applications, Addendum to the 1997 ACM SIGPLAN, pp 150-151.

Miranda, E., 2002. Planning and Executing Time-Bound Projects. *Computer* 35, 3 (Mar. 2002), 73-79

Ngo-The, A., Ruhe, G., 2007. A Systematic Approach for Solving the Wicked Problem of Software Release Planning, Soft Computing, Volume 12, Issue 1, pp. 95-108.

Ngo-The, A., Ruhe, G., Shen, W., 2004. Release Planning under Fuzzy Effort Constraints, icci, pp. 168-175, Third IEEE International Conference on Cognitive Informatics (ICCI'04).

Nord, R., Paulish, D., Soni, D., 2000. Planning Realistic Schedules Using Software Architecture, Software Engineering, 2000. Proceedings of the 2000 International Conference, pp. 824 – 824.

Regnell, B., Brinkkemper, S., 2005. Market-Driven Requirements Engineering for Software Products, Engineering and Managing Requirements, Springer

Ruhe, G., Saliu, O., 2005. The Art and Science of Software Relapse Planning, IEEE Software, issue 6, vol. 22, no. 6, pp 47-53.

Thomas, P., Hawking, D., 2007. Evaluating Sampling Methods for Uncooperative Collections Proceedings of the 30th annual international ACM SIGIR.

Ziv, H., Richardson, D., Klosch, R., 1996. The Uncertainty Principle in Software Engineering, Technical Report UCI-TR-96-33, University of California, Irvine