An A Priori Method to Assess the Potential Effectiveness of a Software Development Curriculum

By
Robert J. Boncella,
Gerrald Reed,
and
Nan Sun*

WASHBURN UNIVERSITY
SCHOOL OF BUSINESS
WORKING PAPER SERIES
Number 47

July 2005

Washburn University
School of Business
1700 SW College Ave.
Topeka, KS 66621
785-231-1010, extension 1308
www.washburn.edu/sobu

*Robert J. Boncella is professor of computer information systems at the School of Business at Washburn University, Topeka, Kansas. Gerrald Reed is lecturer of computer information systems at the College of Arts and Sciences at Washburn University, Topeka, Kansas. Nan Sun is assistant professor of computer information systems at the College of Arts and Science at Washburn University, Topeka, Kansas. Comments should be directed to Robert J. Boncella, School of Business, Washburn University, 1700 SW College Ave. Topeka, Kansas 66621, 785-231-1010, extension 1839, bob.boncella@washburn.edu.

An A Priori Method To Assess The Potential Effectiveness Of A Software Development Curriculum

Robert J. Boncella
Washburn University
School of Business
bob.boncella@washbburn.edu

Gerrald Reed
Washburn University
School of Business
gerrald.reed@washburn.edu

Nan Sun
Washburn University
CIS Department
nan.sun@washburn.edu

Executive Summary

The following is a demonstration of an a priori method to assess the potential effectiveness of a software development curriculum. This method can be used to determine the extent to which a curriculum will address the causes of software project failure (SPF). If the curriculum does not address these causes the method should suggest the ways to modify it so it will.

We assume that a single course will not be sufficient to address the causes of software project failure. As a result we developed a sequence of courses for our department's software development curriculum. This course sequence consists of four courses. They are: Software Engineering (SE), Systems Analysis and Design (SAD), Software Project Management (SPM), and Senior Capstone (SC). SE, SAD and SC are required, while SPM is strongly recommended. The course titles are descriptive of their content. The ideal sequence for these courses would be SE fall semester of junior year, SAD- spring semester of junior year, SPM – fall semester of senior year, and SC – spring semester of senior year.

This curriculum is intended to prevent or at least mitigate the causes of software project failure (SPF). There is no single SPF cause. Practitioners have identified and ranked the most critical SPF courses. This list and ranking is specified below from most frequent to least likely. The SPC causes and ranking are: vague requirements, poor user input, poor estimation of required resources, communications breakdown, stakeholder conflicts, changing requirements during development, failure to plan, poor cost estimation, poor schedule estimation, skills do not match project requirements, poor architecture, inadequate testing, poor quality work, and the "to get along - go along" attitude.

Besides acquiring information from practitioners, we are interested in knowing how researchers address the causes of SPF. Our literature review leads us to the article "Software Project Risks and Their Effect on Outcomes" (Wallace and Keil 2004). In this paper, a software-project-risk framework is developed to classify individual risk factors according to their perceived importance and whether project managers view them as controllable.

In order to evaluate the potential effectiveness of the proposed curricula we present four application matrices, one for each course in the proposed curriculum. In each of the four matrices the rows are labeled with the causes of SPF in decreasing order of importance. The column of each matrix is labeled with the topics presented in that course. The cells of the four matrices contain either an "X" or nothing. If the cell contains nothing then that SPF cause is not addressed in detail by the course topic in the associated column. An "X" indicates a detailed treatment of how the topic will counter a particular SPF cause. These form the basis for a fifth application matrix that combines the total number of topics per SPF per course. The total number of topics per SPF cause is then ranked

A Spearman's one-tail test on SPF Rank and Total Topics Rank yields R=. 265 and p=. 025 indicating a statistically significant (p=. 025) positive correlation between the ranked SPFs and

the number of topics devoted to it. That is, the more likely something is to be an SPF, the more topics are devoted to it.

This paper is a demonstration of an a priori method to assess the potential effectiveness of a software development curriculum. This method was used to determine the extent to which a curriculum may address the causes of software project failure (SPF). This type of evaluation is not intended to replace traditional methods of posteriori evaluation (e.g. surveys of alumni and employers) but we hope that it will be used to enhance curriculum development.

Introduction

The following is a demonstration of an a priori method to assess the potential effectiveness of a software development curriculum. This method can be used to determine the extent to which a curriculum will address the causes of software project failure (SPF). If the curriculum does not address these causes the method should suggest the ways to modify it so it will. The designers of this sequence of courses hold: if attention is paid to the causes of software project failure in an academic setting the effects of these causes can be reduced in practice.

A Proposed Course sequence

We assume that a single course will not be sufficient to address the causes of software project failure. As a result we developed a sequence of courses for our department's software development curriculum. This course sequence consists of four courses. They are: Software Engineering (SE), Systems Analysis and Design (SAD), Software Project Management (SPM), and Senior Capstone (SC). SE, SAD and SC are required, while SPM is strongly recommended. The course titles are descriptive of their content. The specifications for each of these courses, in terms of course description, objectives and text to be used follow. The ideal sequence for these courses would be SE fall semester of junior year, SAD- spring semester of junior year, SPM – fall semester of senior year, and SC – spring semester of senior year. Listed below are the course descriptions and objectives of each course in this sequence. All of these courses have been presented either as special topics or as required courses. However they have never been offered as sequence intended to address the problem of software project failure.

Software Engineering (SE)

Course Description:

SE is a study of disciplined approaches to the production of quality software products and an examination of some social and professional issues related to software production and use. Topics covered are: software requirements and specifications, lifecycle models, analysis, design, validation and evolution of software, project managements, as well as social and ethical considerations. The text for this course is: *Software Engineering* by Sommerville (2004).

Course Objectives:

The course objectives are: to acquire an understanding of the basic software engineering methods/techniques/tools and be familiar with the industry standards in the software engineering field.

Systems Analysis and Design (SAD)

Course Description:

This course will examine what is required to analyze and design software information systems. The software development life cycle (SDLC) will be studied. The topics of Information gathering methods, communication techniques and the nature of the decision making process will be

studied. Additional topics will be the defining logical and physical requirements through the use object-oriented methods. The text for this course is: *Systems Analysis and Design with UML version 2.0: An Object Oriented Approach* by Dennis, Wixom, Tegarden (2005)

Course Objectives:

The course objectives are: to acquire an understanding of the basic concepts of Object-Oriented Systems Analysis and Design and UML and to apply these concepts through a case study in order to see how they benefit the systems development process.

Software Project Management (SPM)

Course Description:

This course is to prepare graduates for future work on a project team. The software project environment will be studied, with emphasis on the role of a Project manger, as well as project team members. The role of the project manager is examined as it relates to project conception, project planning, team organization, monitoring and management, risk management, project reporting, and installation. The format includes best practice discussions with community Project Managers, guidelines from the Project Management Institute, the use of project management tools, as well as case studies and text material. The text for this course is: Gary, F.C., E. W. .Larson, *Project Management: The Managerial Process*, Boston, MA., McGraw-Hill (2000).

Course Objectives:

The course objectives are for students to understand how businesses in the community conduct the management of a project, the life cycle of their projects, and the tools used to aid in the successful completion of software projects. The concepts of project planning are reinforced through the use of software tools, such as MS Project, by student application to selected case studies.

Senior Capstone (SC)

Course Description:

SC is designed to provide closure for the major. Group projects are assigned which allow the student to analyze, design, and implement systems.

Course Objectives:

The course objectives are: to assimilate and synthesize skills acquired during the course of study for the major.

This curriculum is intended to prevent or at least mitigate the causes of software project failure (SPF). There is no single SPF cause. Although there are many causes some occur more frequently or are more damaging than others. Regardless, practitioners have identified and ranking the most critical SPF courses. This list and ranking is specified below.

SPF Causes And Ranking

In "A Course Sequence to Address Causes of Software Project Failure" (Boncella, Reed, and Sun 2003), software project failure (SPF) is defined as any software project with severe cost overruns, quality problems, or a project that is cancelled before completion. Fourteen SPF causes were defined and their definitions are provided below. These causes are presented in an order based on how frequently they were the leading cause of SPF.

The SPF causes ranking of importance were synthesized from industry sources (IT-Cortex), (Numerical Science), and (Pfleeger). The SPF causes start from Vague Requirement, which is ranked as number one importance, to The "Get Along – Go Along" Attitude with the relative importance of fourteen.

Vague Requirements

Vague requirements have several causes. Some of these are: requirements become clear as the project evolves; the scope of the project has not been fixed and users are unwilling to set a baseline of requirements (they may not know them).

Poor User Input

There are several causes of poor user input. Among them are: lack of domain experts; user attitude; "real" users were not interviewed; users are asked how the application should look and feel, not what the application should do; the users assumed the analysts knew more about the users' job than they really knew.

Poor Estimation of Required Resources

Poor estimation of required non-human resources (e.g. hardware & software) causes SPF when the estimation is either excessive or below what is required. Of these two an underestimation is the more damaging. In addition there may be the lack of reliable historical data and lack of sound techniques to do estimation.

Communications Breakdown

In a large project all those involved, managers and developers, can cause SPF by losing sight of the "big picture". If the members of project do not know how their work fits into the whole they can easily lose their way.

Stakeholder Conflicts

Stakeholder conflicts cause SPF in several ways: no early resolution of recognized conflicts; stakeholders have a personality conflict – they don't like each other, as a result they refuse to communicate with each other; analysts cannot identify the relevant stakeholders and stakeholders cannot agree on priorities.

Changing Requirements during Development

Changing requirements during development is normal for any project. This can lead to SPF if there is not a change management process in place to identify change, control change, ensure that change is properly implemented, and report changes to others that may have an interest.

Failure to Plan

Failure to plan as a cause of SPF can be explained as the belief that the coding and testing phase of the project is the "real work" of the project. Since the coding and testing can easily and to some degree accurately measured it is natural to think of the project consisting of only coding and testing.

Poor Cost Estimation

Poor cost estimation are caused by an unwillingness to accept the natural project minimum cost. In addition there may be the lack of historical data and lack of sound techniques to do estimation.

Poor Schedule Estimation

As with poor cost estimation, poor schedule estimation is caused by an unwillingness to accept project's natural minimum schedule, as well as the lack of historical data and sound techniques to do estimation. For both cost and schedule estimation failure to apply available estimating methodologies will lead to SPF.

Skills That Do Not Match Project Requirements

A dissonance between the skill sets available and the project requirements can cause SPF. Some of these are: project manger skills that are too technical; programmer skill sets that are inadequate; both manger and staff skills are of poor quality (you get what you pay for); lack of a feasibility study that determines the required skill sets needed.

Poor Architecture

Poor architecture can cause SPF by not being flexible enough to adapt to changing requirements during the lifetime of the project.

Inadequate Testing

Commercial off-the-self (COTS) software systems or components may save development time but if they are not adequately tested with respect to the project requirements they can be a cause of SPF. There could be the lack of testing plans; testing techniques. And finally, not sufficient amount of time and resources are allocated to carry out adequate testing. Components developed in-house also need to be tested with respect to the project requirements. The system as a whole also needs to be tested with respect to the project requirements.

Poor Quality Work

Unrealistic schedules can lead to poor quality work that causes SPF. A product delivered on time but is in need of constant updates and patches can be considered a failure.

The "To Get Along - Go Along Error" Attitude

The inability or environment in which developers and/or mangers cannot say no will lead to SPF if the schedule and budget are set by edict or if the users have the final say.

Software Project Risks

Besides acquiring information from practitioners, we are interested in knowing how researchers address the causes of SPF. Our literature review leads us to the article "Software Project Risks and Their Effect on Outcomes" (Wallace and Keil 2004). In this paper, a software-project-risk framework is developed to classify individual risk factors according to their perceived importance and whether project managers view them as controllable. Fifty-three software project risk factors known to affect development efforts were compiled and were mapped into four quadrants. These four quadrants were formed by the intersection to two axes. The vertical axis measures from bottom to top on a scale ranging from moderate to high the perceived relative important of a software project risk. The horizontal measures from left to right on a scale ranging from low to high the perceived level of control the project manager has over the particular software project risk. The quadrants are labeled starting form upper left and in a clockwise motion Q1, Q2, Q3, and Q4.

Q1 or Customer Mandate risk factors) focuses on risk factors relating to customers and users, including lack of top management commitment and inadequate user involvement. Q2 or Scope and Requirements risk factors focuses on risk factors associated with a project manager's inability to judge a system's scope. Q3 or Execution risk factors focuses on risk factors such as inadequate project staffing, failure to defines roles and responsibilities, and poor project planning and control. And Q4 or Environment risk factors focuses on risk factors in both internal and external environments.

It is important to note that the risk factors in Q1 are high in risk but low in managerial control; risk factors in Q2 are high in risk but high managerial control; risk factors in Q3 are moderate in risk but high managerial control; and risk factors in Q4 are moderate in risk but low managerial control;

In this classification scheme risk are considered factors that can adversely affect a software project.

EFFECTIVENESS OF THE CURRICULUM

We would like to know, before we implement the curriculum specified, how effective it would be. In particular we would like to know how the causes of SPF and project risk factors are addressed in the course sequence.

In order to do this we present four application matrices. In each of the four matrices the rows are labeled with the causes of SPF in decreasing order of importance. The column of each matrix is

labeled with the topics presented in that course. The cells of the four matrices contain either an "X" or nothing. If the cell contains nothing then that SPF cause is not addressed in detail by the course topic in the associated column. An "X" indicates a detailed treatment of how the topic will counter a particular SPF cause. The instructor of each course determined whether a particular topic addressed a particular SPF. The four application matrices are listed in the following figures.

RANK	Course Topics SE / SPF Causes	Ethical and professional issues	System dependability	Software Process	Management - Project, Process, Quality, Configuration	Requirements Engineering	Design	Software Evolution	Verification and validation	Total Topics
1	Vague Requirements			X		X			X	3
2	Poor User Input			X		X			X	3
3	Poor Estimation of Required Resources				X					1
4	Communicatons Breakdown				X	X				2
5	Stakeholder Conflicts				X	X				2
6	Changing Requirements During Development			X	X				X	3
7	Failure to Plan				X					1
8	Poor Cost Estimation				X					1
9	Poor Schedule Estimation				X					1
10	Skills Do Not Match Project Requirements				X					1
11	Poor Architecture		X				X			2
12	Inadaquate Testing		X						X	2
13	Poor Quality Work		X		X			X	X	4
14	The "To Get Along - Go Along" Attitude	X			X					2

Figure 1 SPF Causes X SE Course Topics Application Matrix

RANK	Course Topics SAD / SPF Causes	SDLC	RAD	Object-Oriented Analysis & Design	OOA, OOD & UML	Project Team Roles and Skills	Business Process Automation	Business Process Improvement	Business Process Reengineering	Developing an Analysis Plan	Interviews	JAD Sessions	Questionnaires	Document Analysis	Use-Case Descriptions	Use-Case Diagrams	Class Diagrams	Sequence Diagrams	Collaboration Diagrams	Statechart Diagrams	Total Topics
1	Vague Requirements	X	X								X	X	X	X	X	X	X	X	X	X	12
2	Poor User Input						X	X	X		X	X	X								6
3	Poor Estimation of Required Resources																				0
4	Communicatons Breakdown											X				X	X	X	X	X	6
5	Stakeholder Conflicts																				0
6	Changing Requirements During Development		X	X	X																3
7	Failure to Plan	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	19
8	Poor Cost Estimation																				0
9	Poor Schedule Estimation																				0
10	Skills Do Not Match Project Requirements																				0
11	Poor Architecture		X	X	X																3
12	Inadaquate Testing																				0
13	Poor Quality Work																				0
14	The "Too Get Along - Go Along" Attitude																				0

Figure 2 SPF Causes X SAD Course Topics Application Matrix

RANK	Course Topics SPM / SPF Causes	Org Strategy with projects	Project Definition	Develop Cost Budgets	Develop Network Flows	Managing Risk	Project Time Reduction	Scheduling Resouces	Organization	Leadership	Managing Project Teams	Progress and Performance	Total Topics
1	Vague Requirements												0
2	Poor User Input												0
3	Poor Estimation of Required Resources				X			X					2
4	Communicatons Breakdown								X	X	X	X	4
5	Stakeholder Conflicts	X	X			X			X				4
6	Changing Requirements During Development					X						X	2
7	Failure to Plan	X	X	X	X								4
8	Poor Cost Estimation			X		X	X						3
9	Poor Schedule Estimation		X		X		X						3
10	Skills Do Not Match Project Requirements					X		X	X				3
11	Poor Architecture												0
12	Inadaquate Testing												0
13	Poor Quality Work												0
14	The "Too Get Along - Go Along" Attitude								X	X	X		3

Figure 3 SPF Causes X SPM Course Topics Application Matrix

RANK	Course Topics SC / SPF Causes	Project Planning	Requirements Gathering & Specification	Design	Coding	Testing	Total Topics
1	Vague Requirements		X				1
2	Poor User Input		X				1
3	Poor Estimation of Required Resources	X					1
4	Communicatons Breakdown		X				1
5	Stakeholder Conflicts		X				1
6	Changing Requirements During Development		X				1
7	Failure to Plan	X					1
8	Poor Cost Estimation	X					1
9	Poor Schedule Estimation	X					1
10	Skills Do Not Match Project Requirements	X					1
11	Poor Architecture			X			1
12	Inadaquate Testing					X	1
13	Poor Quality Work				X	X	2
14	The "To Get Along - Go Along" Attitude						0

Figure 4 SPF Causes X SC Course Topics Application Matrix

We also created a summary matrix with association of SPF causes to project risk quadrants. The 14 SPF causes were mapped to the four quadrants and then to the subject matrices in the course sequence. The authors performed the mapping independently. While there was a general agreement on the placement of most SPF causes, quadrants, and subject matrices, discussions were held on the occasional disagreements until a consensus was reached.

Figure 5 illustrates the result of the mapping. Column one and two are the SPF causes and their rank of importance. Column three, four and five are the quadrants and their perceived importance and level of project manager's control, followed by the number of topics in each of the four courses that address the SPF causes. Column 10 indicates the total number of topics in the course sequence that addresses the SPF causes. The last column is a ranking of the total topics that address each SPF cause. So for example the SPF cause "Vague Requirements" is the second most frequently addressed cause.

Rank	SPF Causes	Quadrant	Risk	Controls	SE	SAD	SPM	SC	Total Topics	Total Topics Rank
1	Vague Requirements	Q2	H	H	3	12	0	1	16	2
2	Poor User Input	Q1	H	L	3	6	0	1	10	4
3	Poor Estimation of Required Resources	Q3	L	H	1	0	2	1	4	13
4	Communications Breakdown	Q3	L	H	2	6	4	1	13	3
5	Stakeholder Conflicts	Q1	H	L	2	0	4	1	7	6
6	Changing Requirements During Development	Q2	H	H	3	3	2	1	9	5
7	Failure to Plan	Q3	L	H	1	19	4	1	25	1
8	Poor Cost Estimation	Q3	L	H	1	0	3	1	5	10.5
9	Poor Schedule Estimation	Q3	L	H	1	0	3	1	5	10.5
10	Skills Do Not Match Project Requirements	Q3	L	H	1	0	3	1	5	10.5
11	Poor Architecture	Q2	H	H	2	3	0	1	6	7.5
12	Inadequate Testing	Q3	L	H	2	0	0	1	3	14
13	Poor Quality Work	Q3	L	H	4	0	0	2	6	7.5
14	The "To Get Along - Go Along" Attitude	Q4	L	L	2	0	3	0	5	10.5

Figure 5 Summary Matrix

At first glance on the basis of number of topics it appears that we adequately address of the causes of SPF in order of importance. A Spearman's one-tail test on SPF Rank and Total Topics Rank yields R=. 265 and p=. 025 indicating a statistically significant (p=. 025) positive correlation between the ranked SPFs and the number of topics devoted to it. That is, the more likely something is to be an SPF, the more topics are devoted to it.

Further analysis shows four SPF causes that are ranked highly important (in the top six) by the industry sources are also perceived as high risk factors by the project managers. These SPF causes are Stakeholder Conflicts, Poor User Input, Vague Requirements, and Changing Requirements during Development. All of them have to do with stakeholders and requirements – in quadrants Q1 or Q2. These high-risk causes are considered non-technical (soft skills) areas of software development. The concern is to how we as academics can best address these causes.

After taking the suggested sequence, the students will have a theoretical understanding of:

- Why eliciting and understanding requirements are a difficult task. They also come to realize that imprecision in the requirements specification is the cause of many software engineering problems and that whenever possible, one should write requirements quantitatively.
- What a stakeholder is, who are the possible stakeholders given a project and why conflicts exist among different stakeholders.
- Why requirements change during development, what configuration management is and why configuration management is important.

In the SC course, students get the chance to work on a real world project. The drawback of the SC course is not all students have the chance to work on real world projects, and even if they do, their primary goal is to pass the course. Based on current grading criteria, passing the course is quite different from delivering the exact product a customer desires.

One possibility is to introduce adequate number of case studies/projects that simulate vague requirements and stakeholder conflicts. Another possibility is for us to make the Senior Capstone course as realistic as possible. Rather than view this course as an academic exercise we design the course so it will the student will encounter these non-technical causes of SPF.

Conclusion

The preceding was a demonstration of an a priori method to assess the potential effectiveness of a software development curriculum. This method was used to determine the extent to which a curriculum may address the causes of software project failure (SPF). This type of evaluation is not intended to replace traditional methods of posteriori evaluation (e.g. surveys of alumni and employers) but we hope that it will be used to enhance curriculum development.

References

Boncella, R., Reed, G., and Sun, N., A Course Sequence to Address Causes of Software Project Failure, Proceedings of the 6th Annual Conference of the Southern Association for Information systems, (2003).

Boncella, R., Reed, G., and Sun, N., Software Project Management: Scenarios of Failure and Potential Solutions, Proceedings of the 7th Annual Conference of the Southern Association for Information systems, (2004).

Humphrey, W.S., "Why Projects Fail", Computerworld, May 20, 2002, http://www.computerworld.com/printthis/2002/0,4814,71209,00.html (Current Nov. 8, 2002)

IT-Cortex (2002) "Failure Rate: Statistics Over IT Projects Failure Rate," http://www.it-cortex.com/Stat_Failure_Rate.htm (Current Nov. 8, 2002)

Keil, M., Cule, P., Lyytinen, K., and Schmidt, R., A Framework for Identifying Software Project Risks, Communications of the ACM, November 1998/Vol. 41, No. 11, pp.76-83.

May, L.J., (July 1998), "Major Causes of Software Project Failure", STSC, http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1998/07/causes.asp (Current Nov. 8, 2002).

Numerical Science (2002), "Statistics", http://www.numerical-science.com/stats.htm (Current Nov. 8, 2002).

Pfleeger, S.L., *Software Engineering: Theory and Practice*, Upper Saddle, N.J., Prentice Hall Inc. (1998).

Tiwana, A., and Keil, M., The One-Minute Risk Assessment Tool, Communications of the ACM, November 2004/Vol. 47, No. 11, pp.73-77.

Wallace, L., and Keil, M., Software Project Risks and Their Effect on Outcomes, Communications of the ACM, April 2004/Vol. 47, No. 4, pp.68-73.