

Bridge Methods: Using a Balanced Project Practice Portfolio to Integrate Agile and Formal Process Methodologies

Stephen J. Cohen
Microsoft Corporation

William H. Money
The George Washington University

Abstract

This paper examines the non-linear nature of progress on software projects. It asserts that this uneven accomplishment stems from the continuous need to drive value quickly in response to a project manager or owner perceiving that a project is failing, near failure, or at least that the owning organization is “loosing value” from either poor process or poor organizational choices of tools and development methods.

It describes how using a balanced project practice portfolio to integrate agile and formal process methodologies can theoretically reduce the observed variance in project productivity caused by poorly applied or poorly selected “best practices”.

Finally, it proposes a methodology that uses expected practice performance and combinations of practices to;

identify missing practices,

off-set poorly applied practices, and

assess the quantity and types of practices required to smooth and optimize project performance outcomes.

1. Introduction: Using a portfolio of practices to integrate agile and formal practices

Principles and practices of Agile are rapidly, but unintentionally being combined into a generic “portfolio of practices” (called by different names) similar to processes and practices already accepted in the formal/traditional development methodologies. This is confusing to the developers in the field, project and program managers, and all but the truly informed or involved. The general practitioners and managers typically view Agile in an optimistic light as a highly streamlined and adaptive delivery methodology or in the more pessimistic view as a messy, confusing, ill defined, and rigor adverse excuse for failing to deliver in a predictable and measurable way.

Practices, viewed as traditional or standard methods of acting, are key to this method of improving project

performance. Practices are postulated to have attributes that make them more or less effective under varying project conditions (e.g., stage of development, scale, expertise, staff experience, etc.). The resultant effects of implementing practices are therefore not predictable, always uniform, or always positive for a project. Case descriptions of the associated consequences of implementing specific practices as measured through cost benefits, performance, and risk mitigation results are widely varying. One can conclude that great practices – best practices as described in surveys - do not always result in project success. Most practitioners would concur that the most carefully documented, commonly measured projects in terms of software developed, defects fixed, time to task completion, and time to delivery do not reflect the impact of one or several specific practices, but depicts the effects of combinations or practices for the project at the moment of the measurement.

However, it is likely that some percentage of the progress on a project is attributable to a practice or combination of practices. Of concern is how much, and under what conditions was the project’s success measure attributable to the specific practice performed. Further are some practices highly useful (resulting in great results), and other perhaps negatively associated under a given situation, with the overall accomplishment? This paper proposes a method that identifies the practices, assembles of groups of practices, and measures outcomes (via practitioner survey) to enable managers to implement selective practices in response to changing project conditions and increase the likelihood of overall project success.

This conceptualization of the work performed in a project environment as small units of activity or “practices” follows the thinking and frameworks that have been found in the reengineering, process innovation, and process redesign field for approximately 15 years. [9][13] That literature focuses on measured groups of activities that are established to produce and output for a customer or market as a process. A process will order the activities by time and location; set start and end dates, and identify inputs and outputs. The process (by definition) represents a pre-

determined way that work is structured. We contend that the software development “field” and practitioners do not yet have broad and strong agreement on the practices / components of successful projects (required activities in the process literatures), nor do they agree fully on the unique sequences, durations, inputs or outputs for the practices (and process designers might agree for a process composed of a number of activities). Without careful development of the practices and understanding about how the practices are combined (structured) and how they may permissibly vary when placed in use, it will be difficult to advance the field. Ongoing disagreements about the definition of Agile and specific practices might even prove to be detrimental, and obscure developing a more complete understanding of precisely why Agile (under a specific situation or implemented in a unique way) contributes to success while it may not have the same results in a another circumstance.

This analytical and object approach to breaking projects down into practices and examining individual effects leads to a number of associated research assumptions. First, it is assumed that project practices themselves will have measurable and individual outcome results that will vary under different conditions for situations. Secondly, the outcome results will be measureable and quantifiable in terms of their productivity. Finally, the results of several practices will be able to be combined to determine the objective results of a practice or groups of practices.

With measurement data, one may develop and test hypotheses that enable the software development field to demonstrate more definitively which project practices performed in projects (groups of specific practices following a required structure) can contribute highly to a projects success. Further, one may empirically measure and offer strong research evidence that supports anecdotal or testimonial data regarding the effectiveness of specific Agile or Formal practices. High quality data will enable quantifiable tests of what is believed about specific practices’ contribution to results in the form of standard hypothesis.

For example, we believe that various practices can be combined to produce better project outcomes. We hypothesize that the standard deviations (of the outcome measures of practice’s effectiveness) in a specific situation will be less than the average of the standard deviations of the practices separately. Secondly, we hypothesize that practices may or may not co-vary with respect to their effectiveness, and that practices with a smaller covariance between two practices will prove to be ineffective, “incompatible” or not synchronized with respect to inputs, timing, or structure. This will be shown in the smaller the variances from a predicted baseline. Data such as this

would enable managers to make informed decision and choices using a model that shows combined standard deviations and co- variances. By choosing carefully, a manager might well address a large risk or even a really bad entrenched practice (that is mandated or required for some reporting or organizational purpose) with a carefully selected off-setting practice or set of practices.

We can speculate that research may ultimately enable more advanced questions to be addressed regarding the possible existence of underlying constructs that support the use of Agile (or Formal) practices. For example, related literatures have enabled academics and practitioners to understand the constructs that are determinants of the acceptance of technology and conditions that can limit its introduction in organizations. Similar research of projects and practices may aid in determining if there are underlying constructs (technical, social, political, or economic effects; beliefs; perceptions, etc.) that support or limit the ability of an organization to employ specific practices successfully and how to address the limits or enhance positive situations.

The broad benefit to managers from closely examining practices is clear. Many different concurrent negative situations (loss of time, costs too high, no progress, business opportunity loss, and growing technical risk) can be addressed, but not all at the same time or in the same way. The concept explored herein is analogous to the medical treatment processes which readily recognize that patients (in this instance projects) have co-morbidity conditions, and that it greatly matters exactly what is treated. (Co-morbidity describes the effect of all other diseases an individual patient might have other than a primary disease of interest.)

2. Bridge: the development environment

The authors have previously described a Bridge [25] development environment that 1) deals well with changing requirements because of its focus on low level and iterative development and testing; 2) meets the traditional schedule and control demands for management reporting well because the iterative development efforts by mapping to traditional milestones; and 3) generates performance metrics that report progress against defined objectives.

This paper extends the original work by providing a more detailed examination of the Formal and Agile practices and processes that are co-implemented in software development programs. The objective is to ensure that critical processes deliver results (working systems), and that creating and maintaining the proper

mix of Agile and formal practices will deliver more consistent results and correct for variance from a projects baseline goals more effectively than strict adherence to prescriptive Agile or formal processes.

Formal and Agile methodologies both have their own principles and prescriptive versions of best practices each intended to drive continuous progress and do so as efficiently as possible. The Bridge method draws practices from both to create a malleable portfolio of practices unique to the specific project at hand. Additionally, the Bridge method separates the initial practice portfolio, constructed to achieve and maintain a predicted baseline, from the on-going portfolio of practices which is continuously reviewed and modified to address issues arising from unpredicted events, issues resulting as a side effect from another practice, and the inevitable changes discovered during solution development.

Principles and practices taken from Formal development processes such as CMMI [6], PRINCE2 [25], Rational Unified Process [14], Waterfall, [20], Spiral [4], Dynamic Systems Development Methodology [8], and Agile processes including Agile Modeling [12], Agile Data [1], Crystal Methods [7], eXtreme Programming [24], and Scrum [21] contain approximately 300 discrete practices and 75 stated principles. Some clearly overlap;

XP	small releases
CMMI	Pilot process and technology improvements to select which ones to implement
PRINCE2	Work Packages
Scrum	Sprint Goals
Agile Manifesto	continuous delivery of valuable software
Spiral	refining prototypes
Dynamic Systems Development Method	Prototype refinement
RUP	Demonstrate value iteratively

while other are uniquely stated as an integral part of their development process.

XP	Pair Programming
CMMI	Staged Audits
PRINCE2	Giving Ad Hoc Direction
Scrum	Burn Down Charting
Agile Manifesto	harness change for the customer's competitive advantage

Spiral	the entire project can be aborted if the risk is deemed too great
Dynamic Systems Development Method	MoSCoW - Must Should Could Would
RUP	Elevate the level of abstraction

The Bridge methodology seeks to empirically combine practices to match the project context, and further to specify more concisely when a practice might be more or less useful – and what might be the expected outcome. Frequently data and project direction of this nature is only available after project have been completed, and managers conduct a “hot wash” or lessons learned session to determine how to proceed with the next project.

3. Practice risk-reward indices

As was previously postulated, the Agile practices are generally high risk / high reward while formal practices are generally low risk and low reward where reward is; significant contribution to delivery, reduced cost to develop, and increased effectiveness in tasking. Bridge combines Agile and formal practices to effectively “offset” negative results and reduce variations from a predicted baseline. This is important because of the costs in time and dollars required to correct a variance from the projects baseline is directly proportionate to the magnitude and duration of the variance.

The figure 1 below illustrates the dampening effect of executing covariant practices across a projects lifecycle as compared to the more traditional sequential practice approach.

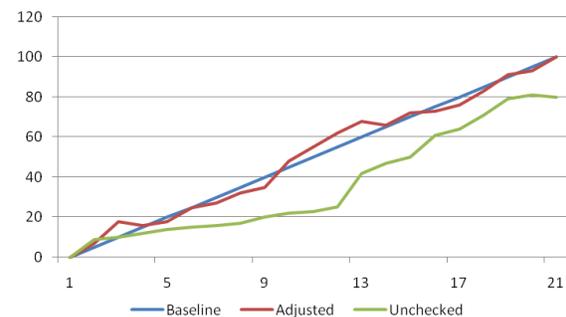


Figure 1: Covariant versus Sequential practices over time

This is accomplished by creating classes of practices which group practices by their weighted risk and

+1.5 use of simple tools you would be +.38 from your baseline.

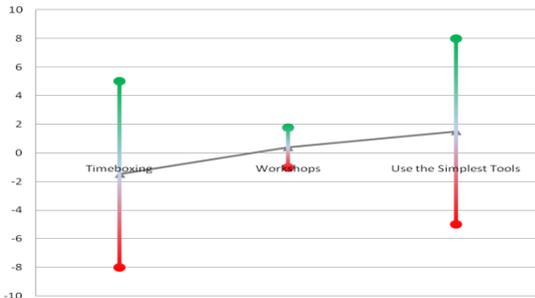


Figure 4: Example offsetting practices

Practices group in to classes

- Leadership
- Team cohesion
- Communication
- Skills
- Tools
- Complexity
- Quality
- Process (rigor)
- Repeatable (reusable)
- Definition

Initial project practice portfolios need to identify a diverse set of practices balanced for each phase of the software development lifecycle. During each phase the portfolio needs to be reviewed whenever possible to ensure issues, whether external, discovered, or internal, are being addressed by the covariant class. These challenge the traditional response to seek change within the class if not within the very practice associated with the issue.

Consider a project struggling with the Skills class practice “Plan for knowledge and skills needed to perform the project.” Once a deficiency in the execution of this practice is identified a common approach to address it would be to either attempt to better execute the practice, in this case that would mean sending resources to training, or at least seek to implement another skills class practice such as the “bringing in ringers” Both skills practices have significant average risks that would combine with the existing risk increasing the variance from the intended baseline of the project. It is unreasonable to believe a project having Skills class issues will not end up on the high risk side of another Skills practice. By using one or more practices from other classes such as Process or Repeatability they will reduce the need to attempt a high risk Skills practice. Instead a moderate risk practice such as “strict promotion from dev to test” and

“reuse before building” will, by their covariant nature (Skills increase is offset by Process decreases and Skills decreases are offset by many reusing the work of few skilled), offset the Skills issue.

The authors conducted a literature survey to develop a list of practices potentially in use that can be included in a use survey. The literature review of practices available for use was drawn from the 37 established SDLCs.[17] It includes well known schools of thought such as Agile, Waterfall, and CMMI, as well as some lesser known software development philosophies such as Cowboy coding, Hollywood principle, and Worse-is-better. The individual risks and rewards examples in this paper, as well as their weighting are currently estimates (researcher best guesses). They are provided for illustration purposes.

4. Research methodology

The researchers are engaged in a two part strategy to advance the understanding of Agile and Formal practices, and their combination through the Bridge approach. First, the researchers are submitting the list and the 400 definitions of the practices to a panel of 12 industry leaders to obtain standardized definitions, remove overlap in the practices (create new smaller unit practices where overlaps are identified, and remove duplicate practices and/or identify the practices holding multiple recognized/accepted names.

Secondly, the researchers will distribute a survey to 1200 IT Architects. The survey asks for project data associating the employment of a practice that has delivered or been very closely associated with specific and quantified project outcomes. A survey research methodology will be used to ascertain practice existence/usage and the practice’s perceived contribution to the organization’s objective of developing a system in a highly compressed time frame while still meeting the functional requirements; of the system users.

Research data will be collected through a voluntary and anonymous survey posted on the internet. The survey will be made accessible to the software project community and directed toward personnel in organizations developing or having developed systems with Agile or Formal project practices. Potential survey respondents will access the survey from their desktop computers. The survey will be posted for a period of about three weeks. An e-mail encouraging/inviting survey response will be promulgated to all candidate respondents. A hyperlink “shortcut” to the web-survey will be embedded in the email inviting/encouraging survey participation.

The survey includes a very included a limited number of demographic items, some contextual items, and provisions for discretionary respondent comments. Respondent survey inputs will be recorded electronically in a data base available only to the researchers, and no mechanism will be available to trace survey inputs to individual respondents.

Standard statistical procedures will be employed to examine the association between measures of success and specific practices. In addition, the researchers will examine the degree to which hypothesized propositions/relationships are supported by the collected survey data. Statistical procedures/methods will include simple and multiple linear regression, analysis of variance, and correlation analysis.

The basic structure and methodology of the research embodies several essentially implicit assumptions including: respondents will be engaged in or will actually have worked on projects involving the practices; and that voluntary survey inputs reflect their candid/objective individual perceptions. All inputs will be weighted equally in data analysis, assuming that the responders possess equivalent understanding of practices and organizational results.

5. Analogy to financial sector analysis

The concept of a managed portfolio, as well as a portfolio manager is commonplace within the financial sector. Portfolios contain a mix of investments each with its own risks and historical records of return. A common approach is to diversify a financial portfolio in such a way as to minimize risk and maximize return. One of the more modern portfolio management schools of thought, the efficient frontier [3], expands on the basic diversification concept by back testing various classes of stocks and identifying combinations which effectively cancel out fluctuations in market returns. In essence the efficient frontier approach seeks to find two asset classes having a very small if not negative covariance. The result is a reduced standard deviation within the portfolio which combines them thereby reducing the risk and increasing their return.

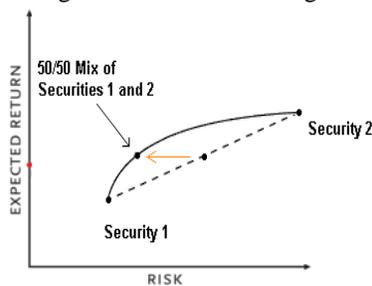


Figure 5: Efficient Frontier Risk - Return [19]

Much like financial assets within the financial market, project practices are a holistic set of activities interacting with each other, and driven by both natural and governed forces. No project practice is truly independent of the others. Determining estimates of effort and cost (CMMI [6] is impacted by the effectiveness of Stakeholder participation (Agile [3]). Both are affected by Time-boxing (Dynamic Systems Development Method [8]) which in turn is affected by choosing to prove it with code (Agile Modeling [12]). Changing the mix of just these 4 practices will likely result in changes to schedule and cost which ultimately equates to increasing or reducing project risk. In the financial sector, the natural forces; Supply and Demand and the governed forces; regulation and controls have extensive records of influence, albeit nondeterministic, on the market. Similarly projects have the natural forces of Time, Skill, Quality, and Scope and governed forces of Organization, Communication, Team Cohesion, and Leadership. Left alone a natural order will determine the quality of a projects solution as well as the likelihood of it delivering on-time and within budget.

The efficient frontiers core value; seeking to offset one class of assets with another, is necessary to address the systemic and unavoidable risks inherent in a given class of assets. The same is true for applying offsetting practices to a project portfolio. A simple example exists in the common practice of using new technologies to leverage advances exposed by the new technology. Inherent in this practice is the risk of increasing required training to aid inexperienced teams to use the new technology as well as the risk associated with being an early adopter (will the new technology actually deliver, how will it interact with other technologies never before paired with it, etc...). Instead of absorbing the risks they can be offset by applying the practices of Bringing in Ringers and avoid educating during production (Project Recovery). This illustrates using practices from the Skills Class to offset a practice from the Tools class.

A notable failing in this analogy lays in the noticeable absence of project practice historical performance data. The financial sector has been collecting and analyzing very fine grained performance data since its inception. The formal community has many metrics but few approach the completeness or consistency necessary [15] and the Agile community has produce even less than the formal practitioners [19]. Another, potentially more significant variant from the financial model is the limit of external forces. The financial market has become an all encompassing worldwide entity while projects are vastly smaller, and bound by corporate or organization limits.

This analogy is helpful in restating the initially proposed hypothesis using the concepts of expected risks or potential for reward. The individual practices may be viewed as having estimable risk/reward values that vary under conditions/situations within a given project. Thus, a portfolio of practices (using this risk-return concept) can be constructed whose covariance will offset the disruptive forces applied to a project when a necessary practice is implemented.

The authors argue that managers and developers select processes carefully, and via a balanced project practice portfolio learn how to address a large risk or even a really bad entrenched practice with a carefully selected offsetting practice or set of practices.

We seek to identify such activities, codify them, and further explain why they are successful.

6. Standard Risk-Reward Benefits

Bridging Agile and Formal practices has, for most, been more a philosophical exercise than an empirical or practical one to date. This Project practice portfolio approach is intended to create a concrete implementation of the Bridge. The Bridge proposes a middle ground where Agile practices are freely combined with formal practices. The Project Practice portfolio further details how all practices, without regard for original school of thought or methodology, can be independently valued and classified. The resulting values represent an initial set for distribution, discussion, and revision.

The perceived valuation of the application of a practice in the eyes of the project participants will also be assessed by asking respondents for their view toward the practice on a scale that uses two bi-polar words (semantic differential). The initial analysis will use risk and reward in these questions. The goals of these questions will be to ascertain the attitude or disposition of the responders to the specific practices. This analysis can be helpful in possibly determining if certain practices are being avoided for unknown reasons, if agreement is generally being given (positive bias), and if responses tend to be influenced by perceptions that positive self reporting is attributable to a perceived socially desirable connotation that the responder wishes to receive from indicating success or usage of a specific practice.

Causes project failure					Assures project success					
-10	-8	-6	-4	-2	0	2	4	6	8	10

Figure 6: Survey Scale

Risk is assumed to always be negative and can result from a failure to implement, a failure to implement well, a failure to implement completely, or even a

failure to stop implementing a practice. Reward is assumed to always be positive and is the result of partial implementation or strict adherence resulting in something that moves the project closer to delivering a solution to users.

Creating a complete portfolio of practices that are being applied to a project makes it possible to identify areas that have not been addressed well. This is most clearly evidenced when a phase or activity has no practices. More likely than a complete absence of practices is a phase or activity that seems to rely on the successful execution of only high risk, high return practices. Besides the obvious indication of a project in jeopardy taking wild risks in the hope of beating all the odds, project constraints might be driving poor management decisions which will manifest in practice selection and execution. Systematic (such as perceptions of risk reward) and routine review of current practice quality provides another feedback loop and opportunity to re-balance the portfolio.

For example a common scenario is a large multi-technology solution has decided to use iterative development (Agile [16]), daily scrums (scrum), and continuous builds (XP [22]). After multiple iterations progress seems by most measures to be slowing. The builds are far less than daily, and the users involved are increasingly dissatisfied with the product. Being iterative means the project practice portfolio should be a mix of practices from all phases. In this case, you find unit testing (DSDM [8]) as the only test phase practice and no practices from the deployment phase (MSF [18]). While conducting retrospectives (scrum [13]) has been declared, it has been deferred after several iterations that ran long, and many of the prototypes (spiral [4]) parts have been left as-is without being accepted by any process to ensure completeness (spiral [4]).

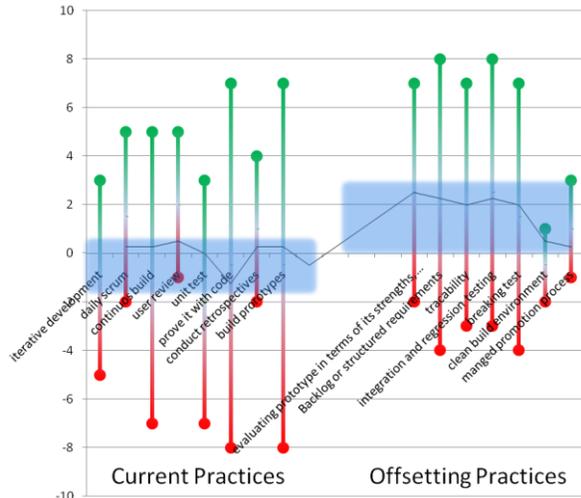


Figure 7: Evaluating a practice portfolio

Figure 7 above depicts the current practices with a low or negative average index and a set of offsetting practices with a predominately 1 to 2 index, more than enough to return the project to its baseline.

7. FBI Virtual Case File (VCF) Example

For research and example purposes, we have applied the Project Practice Portfolios and the Bridge method analysis to a failed past project (FBI Virtual Case File) that is still being pursued due to its critical value to the owning agency. [5] [26] [9] The Virtual Case File (VCF) project is a useful example of where best practices inappropriately applied or applied too late in a project have resulted in major deficiencies.

The VCF was to be the final portion of Trilogy, a 3 part enterprise IT upgrade including new computers, network infrastructure and collaborative case management system designed for tracking criminal cases. The case information system was to have been designed to enable case data sharing and searching. It was intended to replace what was widely and publicly described as an outdated paper-based records system that relied on keyword searches, mainframe technology, and did not support what is commonplace in today's computing environments; graphics, photos, or scanned items. All of these 'special features' were to have been included according to a House investigations and inspector general report. The report placed responsibility for the overall failure and estimated loss of \$104 million with the FBI, not its contractor SAIC. [5]. The problems was summarized as: management turn-over; 36 design changes to the contract; 400 testing problems identified by the government and not delivered to the contractor; and the widely criticized

execution of a \$17 million pilot test after the decision had already been accepted that the delivered system would be discarded. [5]

It is rather easy to observe that some practices were being applied without fully realizing their value. Testing, usually a high value, low risk practice, failed to achieve project transparency and seems to have allowed quality to be schedule if not politically driven. The VCF's apparent intent to "launch all of the software at once"[9], typically evidence of a waterfall process, lacked the necessary validation of artifacts at the end of the phase, and then using the results of the validation as input for the next phase. Instead there was a (presumably failed) attempt to deploy the entire system at once and delay end-to-end debugging until the system was in production.[26] This insight is specifically called out by the formal assessment conducted by a review team from the National Research council (NCR) composed of more than a dozen independent scientists, researchers and industry representatives. They concluded: that the rollout was poorly planned; agents would not be able to take cases to the field for reference; common features (bookmarking and histories) were missing; sorting was not functioning; and the launch was planned as an all at one time event with limited testing (and no backup plan) that could possibly have led to very large disruptions of work. [9]

The FBI is now engaged in a second try at developing the required technology with a new systems and software program called Sentinel that is designed for collaboration, sharing information between agents, and indexing to enable searches for people, places, and events. [26] They have adopted several specific practices in an attempt to limit repeating the issues from the earlier VCF project. The Waterfall has been replaced with an incremental approach consisting of four phases, each with a number of smaller segments. Exemplifying "application of lessons learned", "Buy before Build", "incrementally constructing useful segments", and "Doing the hard stuff first", the Sentinel project has reduced its risk and increased its project practice portfolios' overall index presumably to offset some of the legacy practices that remain a negative risk value.

Reported progress is not all positive, and requirements were scaled back on the first phase with 57 requirements being shifted to phase 2. However, measurement data are being derived in a more timely fashion. The September 2007 report noted that the project was about \$4.4 million over budget, and that the work for the first phase took 14 months instead of the planned 12 months. No explanation was given for the report in the EVM data that the contract was within budget on the project. [25] Hindsight would imply that

a balanced and more formal analysis of practice adoption, implementation, and ongoing usage might have suggested practice outcome metrics to "watch" closely, and that the more recently implemented practices in place today be adopted much earlier in the life cycle of this project.

We are only able to speculate from the limited second and third hand data available on this public project. Many practices were performed that might have been considered appropriate during the first four years of the project. The continued (restarted) project is clearly employing a number of different and changed practices, and reporting on the results of these practices in a much more complete and systematic fashion.

8. Future work

Project Practice Portfolios and the Bridge method can be used to measure past projects and capture future projects in a consistent way that will further the industry.

It is our intent to expand and improve the estimated dataset by engaging the individual practice communities and the industry at large. We have published the list of software development philosophies, their methods and practices, and the individual practice index values to a wiki at ProjectPracticePortfolio.org. We are soliciting each of the represented communities to review and update the lists and their values.

The resulting improved data will be consolidated and combined with research findings. It will be made available our next report on individual practices and practice combinations. It is our expectations that having a persistent public source of individual and combination practice values with lower the bar for projects to incorporate project practice portfolios into their software development lifecycle.

9. References

- [1] *Agile Data Home Page*. AmbySoft. 9 30, 2007. <http://www.agiledata.org/> (accessed 6 12, 2008).
- [2] Ambler, Scott. "Survey Says...Agile Has Crossed the Chasm Examining the effectiveness of agile practices." *Dr. Dobb's Portal: The world of Software Development*. Think Services. July 2, 2008. <http://www.ddj.com/architect/200001986> (accessed 6 12, 2008).
- [3] Bernstein, William J. *Efficient Frontier: An Online Journal of Practical Asset Allocation*. 2008.
- [4] <http://www.efficientfrontier.com/ef/index.shtml> (accessed 6 12, 2008).
- [5] Boehm, Barry W. "A Spiral Model of Software Development and Enhancement." (IEEE Computer) 21, no. 5 (1988).
- [6] Carlstrom, Gregg. "FBI reports progress in second try at case management software system ." *FederalTimes.com*. 1 30, 2008. <http://www.federaltimes.com/index.php?S=3342889> (accessed 5 28, 2008).
- [7] Chrissis, Konrad, and Shrum. *CMMI Second Edition, Guidelines for Process Integration and Product Improvement*. Boston Ma.: Pearson Education, Inc., 2007.
- [8] Cockburn, Alistair. *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley Professional, 2004.
- [9] Cohen, Stephen and Money, William. *Bridge Methods: Complementary Steps Integrating Agile Development Tools & Methods with Formal Process Methodologies*. Waikoloa, Hawaii: Hawaii International Conference on System Sciences, 2008.
- [10] Davenport, T. H. *Process Innovation*. Boston, Massachusetts: Havard Business School Press, 1993.
- [11] *Effective Practices for Modeling and Documentation*. AmbySoft. 5 24, 2008. <http://www.agilemodeling.com/> (accessed 6 12, 2008).
- [12] Eggen, D., Washington Post Staff Writer. "FBI pushed ahead with troubled software." *WashingtonPost.com*. Washington Post. June 6, 2005. <http://www.washingtonpost.com/wp-dyn/content/article/2005/06/05/AR2005060501213.html> (accessed 6 12, 2008).
- [13] Gloger, Boris. *Scrum Delivers*. Scrum Alliance, Inc. . 5 15, 2006. <http://www.scrumalliance.org/articles/22-scrum-delivers> (accessed 6 12, 2008).
- [14] Jacobson, I., Booch, G., Rumbaugh, J. *The unified software development process*. Reading Ma: Addison-Wesley, 1999.
- [15] Jones, Capers. *Applied Software Measurement: Global Analysis of Producton and Quality*. McGraw-Hill, 2008.
- [16] Larman, Craig. *Agile and Iterative Development*. Prentice Hall, 2003.
- [17] *List of software development philosophies*. June 6, 2008. http://en.wikipedia.org/wiki/List_of_software_development_philosophies (accessed June 12, 2008).

- [18] Microsoft. "MSF Resource Kit." *Microsoft TechNet*. Microsoft. 2003. http://download.microsoft.com/download/3/a/6/3a6b7624-da2a-4883-a413-99281adadff3/MSF_v3_Overview%20Whitepaper.pdf (accessed 6 12, 2008).
- [19] MoneyChimp. "The Efficient Frontier and Portfolio Diversification." *Money Chimp*. MoneyChimp. 1997. http://www.moneychimp.com/articles/risk/efficient_frontier.htm (accessed 6 12, 2008).
- [20] Royce, Winston. "Managing the development of large software." Proceedings of IEEE WESCON , 1970.
- [21] *SCRUM Alliance*. SCRUM Alliance. 2008. <http://www.scrumalliance.org/> (accessed 6 12, 2008).
- [22] Stapleton, Jennifer. *DSDM—Dynamic Systems Development Method*. Essex, England: Addison-Wesley, 1996.
- [23] Tenner, A.R., and I.J. DeToro. *Process Redesign*. Reading, Massachusetts: Addison Wesley Longman, Inc., 1997.
- [24] *The Rules and Practices of Extreme Programming*. Don Wells. 1999. <http://www.extremeprogramming.org/rules.html> (accessed 6 12, 2008).
- [25] Venning, Great Britain. Office of Government Comm and Chris. *Managing Portfolios of Change (with MSP for Programmes and PRINCE2 for Projects)*. Stationery Office, 2007.
- [26] Wells, Don. *Integrate Often*. ExtremeProgramming.org. 1997. <http://www.extremeprogramming.org/rules/integrateoften.html> (accessed 6 12, 2008).
- [27] Witte, Dan Eggen and Griff. "The FBI's Upgrade That Wasn't." *WashingtonPost.com*. Washington Post. 8 18, 2006. <http://www.washingtonpost.com/wp-dyn/content/article/2006/08/17/AR2006081701485.html> (accessed 5 20, 2008).