Agile Tailoring Tool (ATT): A Project Specific Agile Method

Article · March 2009

DOI: 10.1109/IADCC.2009.4809266

3 authors:

Shady Mohammed
Ahram Canadian University

Maha Hana
Helwan University

Ahmed Sharaf Eldin
Sinai University

2009 IEEE International Advance Computing Conference (IACC 2009)
Patiala, India, 6–7 March 2009

Agile Tailoring Tool (ATT):A Project Specific Agile Method

Shady Mohammed El-Said(1), Maha Hana(2), Ahmed Sharaf Eldin(2)

(1)Faculty of Computer Science - October University for Modern Sciences and Arts –Egypt,
(2)Faculty of Computers and Information - Helwan University – Egypt

***Abstract*- after decades from introducing and using agile methodologies, project mangers realized that no methodology is sufficient by itself. Thus, merging their principles is the solution yet no formal solution has been proposed. Relying on previous work, ATT provides a mathematical model to act as a tailoring tool to formulate a new agile method based on experienced agile methods and the project specifications. It requires project managers to understand well the project requirements in terms of SDLC phases, and accordingly the new agile methodology is tailored.**

I. Introduction

Traditional software development methodologies had proved failure to address several problems among them the requirement change problem, while agile development methodologies appeared as a solution to this problem. Practical experience proved that there is no agile method that works as a standalone one. Thus, there must be a sort of customization for the selected method or merging principles from several methods. Selecting a method from the agile family, requires the project manager to be aware of all the agile family members or at least the famous ones, which adds burdens of understanding and experiencing the different agile methods without making use of others' previous experience.

Software development corporations provide tools that support some of the technical aspects of the agility [1],[2] with lack of support to the nontechnical aspects as metaphor and pair programming, which leads to difficulties when adopting agility in a software project. This paper shows a formal way that helps project managers to tailor an agile methodology that meets the project development requirements based on their understanding to the project itself rather than the selected agile family members themselves.

The incoming section in this paper illustrates previous trails in the same interest; while section three is a brief illustration of some selected agile methods. In section four a detailed illustration to the tailoring tool. Final results are stated in section four and concluded in section five.

II. Related Work

Among the scientific researches target agile methods, there are few distinct researches [3],[4] done in aggregating agile methods. Both researches conducted a detailed comparison and illustration to the most famous agile family members, yet [4] shows up a new evaluation way using metalanguage and fuzziness. Using these comparisons is not enough to perfectly help project managers to select the suitable parts from each method and merge them to suite their requirements. Two trials in agile customization were introduced [6][7]. Research in [6] was intended to discover how to adopt extreme programming – XP – in large organizations. It used a pilot project with a number of cases to measure XP capabilities. The work done in [7] can be considered as the first considerable trail that gives the project mangers ability to formulate their own agile method depending on the project requirements. The main drawback of [7] is that it relies on the understanding of project manager for both the considered agile methods and their project requirements, which adds the burden of understanding the entire agile methods on the project managers rather than concentrating on the desired tasks from it.

III. Adopted Agile Methods

Agile methods adopted here are extreme programming –XP, Scrum, adaptive software development – ASD, dynamic systems development method –DSDM, and feature driven development – FDD. These methods are described in details in [3]-[5] showing their process, role, practices, adoption, and scope of use. Crisp and fuzzy quantification found in [4], [5] give the inspiration for this work. Tables 1-6 show respectively the values estimated for each method against the systems development life cycle – SDLC – in 3D space of process, role, and practice as a crisp values, fuzzy theoretical, and fuzzy practical from experimental pilot software.

Methodology		SDLC Phases Group I			
		Feasibility Study	Requirement Analysis	Planning	Design
XP	Process	0	1	1	1
	Role	0	1	1	1
	Practice	0	1	1	1
Scrum	Process	0	1	1	1
	Role	0	1	1	1
	Practice	0	1	1	0
ASD	Process	0	1	1	0
	Role	0	1	1	0
	Practice	0	0	1	0
DSDM	Process	0	1	1	1
	Role	0	1	1	1
	Practice	0	1	1	1
FDD	Process	1	1	1	1

	Role	1	1	1	1
	Practice	1	1	1	1

Table 1 Selected Agile Methods Binary Membership values against SDLC Phases Group I [4]

Methodology		SDLC Phases Group II			
		Coding	Testing	Deployment	Maintenance
XP	Process	1	1	0	1
	Role	1	1	0	1
	Practice	1	1	0	1
Scrum	Process	1	1	0	1
	Role	1	1	0	1
	Practice	1	0	0	1
ASD	Process	1	0	0	1
	Role	1	0	0	1
	Practice	1	0	0	1
DSDM	Process	1	1	1	1
	Role	1	1	0	1
	Practice	1	1	0	0
FDD	Process	1	1	0	0
	Role	1	1	1	0
	Practice	1	1	0	0

Table 2 Selected Agile Methods Binary Membership values against SDLC Phases Group II [4]

Methodology		SDLC Phases Group I			
		Feasibility Study	Requirement Analysis	Planning	Design
XP	Process	0.117	0.555	0.529	0.633
	Role	0.283	0.498	0.324	0.4
	Practice	0.167	0.28	0.4	0.417
Scrum	Process	0.067	0.453	0.571	0.375
	Role	0.2	0.56	0.479	0.442
	Practice	0.092	0.462	0.243	0.192
ASD	Process	0.433	0.41	0.243	0.25
	Role	0.283	0.282	0.243	0.242
	Practice	0.167	0.443	0.371	0.192
DSDM	Process	0.667	0.379	0.332	0.533
	Role	0.411	0.291	0.379	0.317
	Practice	0.567	0.395	0.279	0.367
FDD	Process	0.2	0.348	0.443	0.567
	Role	0.533	0.304	0.324	0.517
	Practice	0.267	0.197	0.129	0.383

Table 3 Selected Agile Methods Theoretical Fuzzy Membership values against SDLC Phases Group I [5]

Methodology		SDLC Phases Group II			
		Coding	Testing	Deployment	Maintenance
XP	Process	1	0.51	0.071	0.483
	Role	0.408	0.31	0.027	0.55
	Practice	0.488	0.6	0.017	0.367
Scrum	Process	0.163	0.2	0	0.467
	Role	0.35	0.14	0	0.55
	Practice	0.075	0.08	0.017	0.311
ASD	Process	0.15	0.135	0.017	0.15
	Role	0.35	0.02	0.017	0.3
	Practice	0.2	0.16	0	0.367
DSDM	Process	0.325	0.39	0.417	0.411
	Role	0.525	0.24	0.063	0.367
	Practice	0.5	0.46	0	0.4
FDD	Process	0.8	0.4	0	0
	Role	0.417	0.21	0.192	0
	Practice	0.338	0.06	0	0

Table 4 Selected Agile Methods Theoretical Fuzzy Membership values against SDLC Phases Group II [5]

Methodology		SDLC Phases Group I			
		Feasibility Study	Requirement Analysis	Planning	Design
XP	Process	0.3	0.6	0.386	0.717
	Role	0.467	0.647	0.238	0.5
	Practice	0	0.293	0.386	0.65
Scrum	Process	0	0.42	0.386	0.333
	Role	0	0.45	0.364	0.375
	Practice	0.083	0.39	0.276	0.225
ASD	Process	0.167	0.27	0.179	0.25
	Role	0.083	0.2	0.229	0.208
	Practice	0.033	0.26	0.371	0.225
DSDM	Process	0.6	0.49	0.273	0.458
	Role	0.378	0.338	0.45	0.383
	Practice	0.35	0.31	0.25	0.4
FDD	Process	0.067	0.34	0.443	0.55
	Role	0.15	0.277	0.267	0.442
	Practice	0.233	0.29	0.143	0.383

Table 5 Selected Agile Methods Practical Fuzzy Membership values against SDLC Phases Group I[4]

Methodology		SDLC Phases Group II			
		Coding	Testing	Deployment	Maintenance
XP	Process	1	0.42	0.208	0.483
	Role	0.583	0.48	0.333	0.583
	Practice	0.417	0.54	0.1	0.389
Scrum	Process	0.113	0.2	0.067	0.6
	Role	0.275	0.13	0.067	0.55
	Practice	0.1	0.1	0.083	0.311
ASD	Process	0.6	0.32	0	0.283
	Role	0.5	0.34	0	0.433
	Practice	0.5	0.34	0	0.367
DSDM	Process	0.375	0.43	0.583	0.367
	Role	0.525	0.273	0.129	0.367
	Practice	0.5	0.2	0	0.5
FDD	Process	0.8	0.3	0	0
	Role	0.371	0.13	0.133	0
	Practice	0.275	0.06	0	0

Table 6 Selected Agile Methods Practical Fuzzy Membership values against SDLC Phases Group II[4]

IV. Agile Tailoring Tool (ATT) Algorithm

Step 1

Input:

1. Crisp contribution of the selected agile methods against 3D SDLC phases.
2. Fuzzy Theoretical contribution of the selected agile methods against 3D SDLC phases.
3. Fuzzy Practical contribution of the selected agile methods against 3D SDLC phases.

Output:

- Aggregated contribution of the selected agile methods against 3D SDLC.

Process:

1. Select three weights corresponding to the three inputs
2. Calculate the weighted average for each dimension D from the binary comparison (B), fuzzy theoretical (FT) and fuzzy practical (FP) as

$$\frac{W_1 D_B + W_2 D_{FT} + W_3 D_{FP}}{\sum_{i=1}^{3} W_i} \quad (1)$$

Step2

Input:

1. Aggregated contribution of the selected agile methods against 3D SDLC phases
2. Project specifications estimates against 3D SDLC phases

Output:

The new agile method against 3D SDLC phases

Process:

1. For each method, calculate the absolute difference between the two inputs
2. for each dimension in each phase, find the minimum absolute difference which indicates the chosen method for that dimension

V. Agile Tailoring Tool (ATT)

ATT aggregates data from the crisp, fuzzy theoretical and fuzzy practical tables. The resultant values are calculated as follows:

For a methodology M, let (P_B) be a crisp process value, (P_{FT}) be a fuzzy theoretical value and (P_{FP}) be a fuzzy practical value. Let (P_R) equals the resultant process value, which is computed from the following weighted average [8] formula:

$$P_R = \frac{W_1 P_B + W_2 P_{FT} + W_3 P_{FP}}{\sum_{i=1}^{3} W_i} \quad (2)$$

The same formula is used for the roles and practices for each method at each phase. The weight parameters are chosen to equal 1, 2 and 3 respectively and the ATT values are in tables 7-8.

Methodology		SDLC Phases Group I			
		Feasibility Study	Requirement Analysis	Planning	Design
XP	Process	0.189	0.652	0.536	0.736
	Role	0.328	0.656	0.394	0.55
	Practice	0.056	0.407	0.493	0.631
Scrum	Process	0.022	0.528	0.55	0.458
	Role	0.067	0.578	0.508	0.502
	Practice	0.072	0.516	0.386	0.177
ASD	Process	0.228	0.438	0.337	0.208
	Role	0.136	0.361	0.362	0.185
	Practice	0.072	0.278	0.476	0.177
DSDM	Process	0.522	0.538	0.414	0.573
	Role	0.326	0.433	0.518	0.464
	Practice	0.364	0.453	0.385	0.489
FDD	Process	0.267	0.453	0.536	0.631
	Role	0.419	0.407	0.408	0.56
	Practice	0.372	0.377	0.281	0.486

Table 7 ATT Weighted Sum Values against SDLC Phases Group I

Methodology		SDLC Phases Group II			
		Coding	Testing	Deployment	Maintenance
XP	Process	1	0.547	0.128	0.569
	Role	0.594	0.51	0.176	0.642
	Practice	0.538	0.637	0.056	0.484
Scrum	Process	0.278	0.333	0.034	0.622
	Role	0.421	0.278	0.034	0.625
	Practice	0.242	0.077	0.047	0.426
ASD	Process	0.517	0.205	0.006	0.358
	Role	0.533	0.177	0.006	0.483
	Practice	0.483	0.223	0	0.473
DSDM	Process	0.463	0.512	0.597	0.487
	Role	0.604	0.383	0.086	0.473
	Practice	0.583	0.42	0	0.383
FDD	Process	0.833	0.45	0	0
	Role	0.491	0.302	0.297	0
	Practice	0.417	0.217	0	0

Table 8 ATT Weighted Sum Values against SDLC Phases Group II

Next, the project managers required to estimate in percentages the relevance of each development phase to the ideal SDLC in terms of process, role and practice.

For the estimated values, compute the absolute difference between them and the resultant table corresponding value. The minimum value indicates the methodology that should be adopted in the corresponding SDLC phase.

Tables 9-10 show an example for a project (Z), with the following set of percentages for process, role, and practice for each SDLC phase. Then tables 11-12 show the absolute difference between the proposed project and the correspondence at each considered methodology. Tables 13-14 show the ATT method by taking the corresponding method to the minimum value at tables 11-12.

Project (Z)	SDLC Phases Group I			
	Feasibility Study	Requirement Analysis	Planning	Design
Process	0.506	0.436	0.053	0.315
Role	0.691	0.575	0.929	0.273
Practice	0.014	0.59	0.853	0.955

Table 9 Estimated Percentages for Project (Z) SDLC Phases Group I

Project (Z)	SDLC Phases			
	Coding	Testing	Deployment	Maintenance
Process	0.669	0.893	0.228	0.355
Role	0.696	0.842	0.74	0.127
Practice	0.037	0.024	0.181	0.726

Table 10 Estimated Percentages for Project (Z) SDLC Phases Group II

Methodology		SDLC Phases Group I			
		Feasibility Study	Requirement Analysis	Planning	Design

XP	Process	0.317	0.216	0.483	0.421
	Role	0.363	0.082	0.535	0.277
	Practice	0.041	0.184	0.36	0.325
Scrum	Process	0.484	0.092	0.497	0.143
	Role	0.624	0.004	0.421	0.229
	Practice	0.058	0.074	0.468	0.779
ASD	Process	0.279	0.003	0.285	0.107
	Role	0.555	0.214	0.567	0.088
	Practice	0.058	0.312	0.377	0.779
DSDM	Process	0.016	0.102	0.361	0.258
	Role	0.365	0.142	0.411	0.191
	Practice	0.35	0.137	0.469	0.466
FDD	Process	0.24	0.017	0.483	0.316
	Role	0.272	0.168	0.521	0.287
	Practice	0.358	0.213	0.572	0.469

Table 11 Absolute Difference Between Project (Z) Percentages and Resultant Fuzzy Values against SDLC Phases Group I

Methodology		SDLC Phases Group II			
		Coding	Testing	Deployment	Maintenance
XP	Process	0.331	0.347	0.1	0.214
	Role	0.102	0.332	0.565	0.515
	Practice	0.501	0.613	0.125	0.243
Scrum	Process	0.391	0.56	0.194	0.267
	Role	0.275	0.564	0.707	0.498
	Practice	0.205	0.053	0.134	0.3
ASD	Process	0.152	0.688	0.222	0.003
	Role	0.162	0.665	0.735	0.356
	Practice	0.446	0.199	0.181	0.254
DSDM	Process	0.206	0.382	0.37	0.132
	Role	0.092	0.459	0.655	0.346
	Practice	0.546	0.396	0.181	0.343
FDD	Process	0.165	0.443	0.228	0.355
	Role	0.205	0.54	0.443	0.127
	Practice	0.38	0.193	0.181	0.726

Table 12 Absolute Difference Between Project (Z) Percentages and Resultant Fuzzy Values against SDLC Phases Group II

New Methodology	SDLC Phases Group I			
	Feasibility Study	Requirement Analysis	Planning	Design
Process	DSDM	ASD	ASD	ASD
Role	FDD	Scrum	DSDM	ASD
Practice	XP	Scrum	XP	XP

Table 13 ATT Method for the Proposed Project (Z) against SDLC Phases Group I

New Methodology	SDLC Phases Group II			
	Coding	Testing	Deployment	Maintenance
Process	ASD	XP	XP	ASD
Role	DSDM	XP	FDD	FDD
Practice	Scrum	Scrum	XP	XP

Table 14 ATT Method for the Proposed Project (Z) against SDLC Phases Group II

Results in tables 13-14 infer that for each SDLC phase there is an agile method that can perform process, role and practice better than the other considered agile methods. Detailed knowledge about the considered agile methods is found in [3]-[5] or any other reference can help in adopting the ATT method.

VI. Conclusion and Future Work

No agile method can be considered as a perfect or a complete one by itself and it must be integrated with other method(s) to obtain the most possible profit. In this work, ATT is an intuitive, simple and yet a rational tool to create a new agile method according to the project specifications. For each project, there is a unique specific tailored agile method that suits the project manager point of view. Project manager's awareness of the proposed project requirements and understanding to the SDLC phases become the criterion for the success of ATT method.

ATT has several advantages. First, it needs one requirement that is to understand the project phases and estimate the relevance of each SDLC phase. This requirement is already inspected in order to increase the chances of the project success. Second, ATT algorithm is simple and straightforward. Third, ATT presents each project its specific agility methodology which will be appealing to the end user. Fourth, the weights given for each of the crisp, theoretical fuzzy and fuzzy practical 3D SDLC phases are variable.

Future work is to experiment with different weights in order to measure ATT sensitivity and effect. Another work may be is to put it into practice to measure its success. The last is to use the project managers' feedback to refine ATT.

VII. References

[1] Morris, Richard. Refactoring in Visual Basic 2005. MSDN. [Online] Microsoft, April 2005. [Cited: November 18, 2008.] http://msdn.microsoft.com/en-us/vbasic/ms789084.aspx.

[2] Narang, Sanjay. Software-Development Methodologies and Visual Studio Team System. MSDN. [Online] Microsoft, October 2006. [Cited: November 17, 2008.] http://msdn.microsoft.com/en-us/library/aa905317.aspx.

[3] P. Abrahamsson,O. Salo, J. Ronkainen, J. Warsta "Agile Software Development Methods: Review and Analysis", VVT Publications, Vol. 478, pp. 7–94, 2002

[4] S. M. El-Said, "Agile Software Development Methodologies: A Comparative Study", M.S. Thesis, Dept. Info. Sys., Comp. and Info. Fac., Helwan Univ., Egypt. 2008

[5] A. Sharaf, M. A. Hana , S. M. El-Said, " Agile Software Development Methodologies: Metalanguage Crisp and Fuzzy Comparisons", International Journal of Intelligent Computing & Information Sciences – Ain Shams University, Vol. 8, No. 1, PP 25-41, 2008

[6] Lindvall, M.; Muthig, D.; Dagnino, A.; Wallin, C.; Stupperich, M.; Kiefer, D.; May, J.; Kahkonen, T.;" Agile software development in large organizations" IEEE, Vol 37, Issue 12, pp26-34, 2004.

[7] Mnkandla, Ernest. 2008. Tool for Tailoring Agile Methodology Practices. The Joburg Centre for Software Engineering . [Online] April 2008. [Cited: November 2008, 19.] http://www.jcse.org.za/upload/events/15/jcse_xp_forumprofiling_agile_practices_10_april_2008.pdf.

[8] Douglas A. Lind, William G.Marchal and Samuel A. Wathen, "Basic Statistics for Business and Economics", 5th ed, McGraw Hill, 2006