

Peaceful Coexistence: Agile Developer Perspectives on Software Architecture

Davide Falessi, Giovanni Cantone, and Salvatore Alessandro Sarcia',
University of Rome Tor Vergata

Giuseppe Calavaro, Paolo Subiaco, and Cristiana D'Amore, *IBM Software Group*

A survey of 72 IBM software developers suggests theoretical compatibilities between agile values and software architecture that bode well for future integration in practice.

The Agile Manifesto will be 10 years old in 2011. Agile methods are widespread in the software industry today, and a rapprochement between the agile and architecture-centric development communities has emerged. Nevertheless, some tension persists between these communities. To help separate facts from myths about the potential coexistence of agile development and software architecture, we conducted an exploratory study of experienced practitioners at a large company—specifically, the IBM Software Lab in Rome. By identifying and understanding the advantages and problems these professionals perceived, we aimed to facilitate the practical integration of these development approaches and illuminate future research directions. The participants had 18 years of developer experience on average, and most had already adopted agile approaches.

Although qualitative data is generally more suitable than quantitative data in exploratory studies, we developed a survey to capture quantitative data by first conducting focus groups in the lab and then synthesizing the comments into a survey that would capture quantitative data.¹ We piloted the initial survey and administered the final questions to 72 professional IBM developers.

Relevance of Software Architecture Uses

We wanted to know whether agile developers considered the use of software architecture relevant to their work, so our first question was, “In the context of agile development, how relevant is each of the following uses of software architecture?” Table 1 shows the list of uses, which we adopted

from the ISO/IEC WD4 42010 (IEEE P42010/D6) standard for systems and software engineering architecture descriptions.² The participating developers ranked the relevance level of each use from 0 (no relevance) to 3 (extremely relevant).

We calculated the results shown in Table 1 by averaging the scores for each use. The uses are listed in descending order according to their relevance rank. The results show that 13 of 17 uses rank higher than the scoring midpoint of 1.5. In other words, only three of 17 software architecture uses are more irrelevant than relevant to agile practice. We can conclude that the participants considered software architecture relevant in the context of agile development.

When to Focus on Software Architecture

As Grady Booch said, “You don’t need architecture to build a dog kennel, but you’d better have some for a skyscraper.”³ Along this line, our second survey question was, “In the context of agile development, when should you focus on software architecture?” Optional answers were “always,” “never,”

Table 1

Relevance levels of software architecture uses as perceived by agile developers

Rank	ISO/IEC 42010 uses of software architecture	Relevance level
1	To communicate among organizations involved in the development, production, fielding, operation, and maintenance of a system	2.16
2	As input to subsequent system design and development activities	2.04
3	To document assumptions made by the architect about the system and its intended use and environment	2.02
4	To analyze and evaluate alternative architectures	1.98
5	To communicate the characteristics, features, and design of a system to potential clients, acquirers, and integrators	1.98
6	To support review, analysis, and evaluation of the system	1.95
7	To aid planning for transition from a legacy software architecture to a new software architecture	1.80
8	As specification for a group of systems sharing a set of features	1.75
9	To support the scaling of agile practices to large projects	1.74
10	To document points of flexibility or limitations within the system for future requirements	1.69
11	As development and maintenance documentation	1.67
12	For operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, subsystems, and components	1.66
13	To establish criteria for certifying implementations for conformance to software architecture	1.62
14	To communicate among clients, acquirers, and developers as a part of contract negotiations	1.50
15	To support system planning and budgeting activities	1.35
16	To support preparation of acquisition documents	1.28
17	As input for the selection of system generation and analysis tools	1.12
Average		1.72

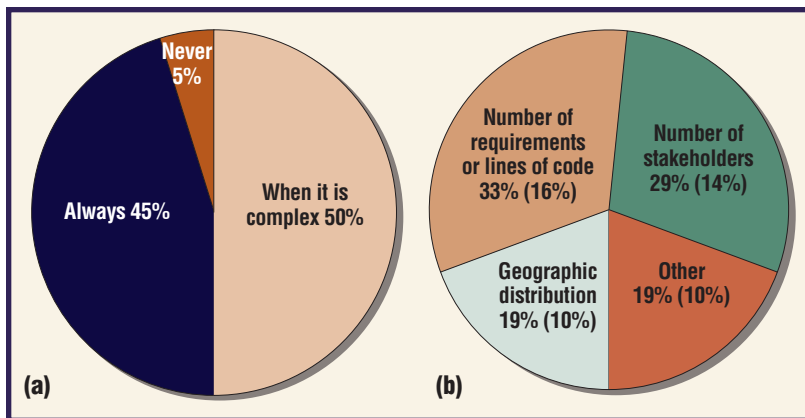


Figure 1. Results for the question of when to focus agile development on software architecture: (a) half the respondents selected project complexity; (b) of these, 33 percent (16 percent of all respondents) indicated the number of requirements or lines of code as the leading cause of complexity that requires software architecture.

and “when the project is complex.” Because complexity is a broad term, we asked respondents who selected it to choose geographic distribution, number of requirements or lines of code, number of stakeholders, and “other” as the leading cause of complexity.

As Figure 1a shows, half the respondents selected project complexity as a reason to focus agile development on software architecture. Figure 1b reports the percentage results characterizing this complexity. In particular, practitioners perceived the number of requirements or lines of code as the leading indicator of project complexity that requires a focus on software architecture, followed closely by the number of stakeholders.

Agile Values and Architecture-Centric Principles

We wanted to characterize the relationships between agile values and architecture-centric principles. We distilled three main principles of architecture-centric methods:⁴

- driven by nonfunctional requirements,
- requiring an upfront investment, and
- forcing software architecture compliance.

Participants characterized the relationships among all the combinations of these principles with the four values of the Agile Manifesto (<http://agilemanifesto.org>).

For the 12 combinations of agile values and architecture-centric principles, Figure 2 shows the distribution of relationships perceived as the most supportive, the most contrastive, and the overall average. According to Figure 2, the principles of architecture-centric method are, on the average, supportive (rather than contrastive or neutral) to agile values.

Other Results

Further results showed that a large majority of agile developers saw a need for new methods and special training to integrate architectural practices—such as software architecture analysis, design,

About the Authors



Davide Falessi is a postdoctoral researcher in the University of Rome Tor Vergata's Department of Informatics, Systems, and Production Engineering (DISP). His research interests focus on devising and empirically assessing scalable solutions for the development of complex software-intensive systems with a particular emphasis on quality, reuse, and evolution. Falessi has a PhD in software engineering (informatica ed ingegneria dell'automazione) from the University of Rome Tor Vergata. Contact him at falessi@ieee.org.

Giovanni Cantone is a professor of experimental software engineering and software-intensive systems' analysis and design in the University of Rome Tor Vergata's Department of Informatics, Systems, and Production Engineering (DISP). His research interests are all in software engineering. Cantone has a pre-form doctoral "Laurea" degree in electronics engineering from the University of Naples Frederick II. He is a founding member of the International Software Engineering Research Network. Contact him at cantone@uniroma2.it.



Salvatore Alessandro Sarcia is a researcher at the Italian Ministry of Defense—Army General Staff in Rome. His research interests are in advanced statistics, nonlinear modeling, organizational performance measurement, and computational intelligence applied to empirical software engineering. Sarcia received a PhD in software engineering (informatica ed ingegneria dell'automazione) from the University of Rome Tor Vergata. Contact him at sarcia@disp.uniroma2.it.



Giuseppe Calavaro is the information management services manager for the IBM Software Group in Italy. His research interests include software processes, architectures, and tools. Calavaro received a PhD in software engineering (informatica ed ingegneria dell'automazione) from the University of Rome Tor Vergata. Contact him at gcalavaro@it.ibm.com.



Paolo Subiacco is the Rational Service Europe Business Leader for IBM Software Group. His research interests include software processes, architectures, and tools as well as semantic and innovative smartphone applications. Subiacco received a master's degree from the University of Rome Tor Vergata. Contact him at paolo_subiacco@it.ibm.com.

Cristiana D'Amore is the central architectural team manager for the IBM Software Group in Italy. Her research interests are in software architecture—especially its role in agile software development. D'Amore received a master's degree in mathematics from the University of Rome La Sapienza. Contact her at cdamore@it.ibm.com.

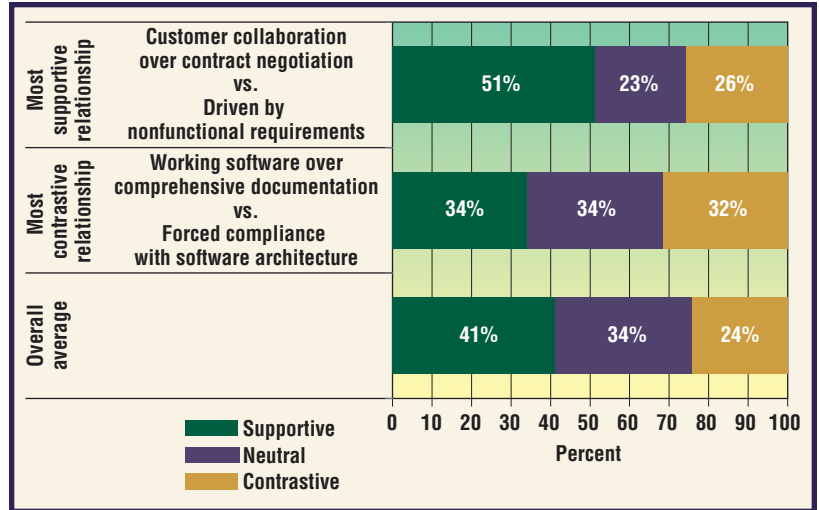



Figure 2. Relationship results for agile values versus architecture-centric principles. The top row shows the most supportive relationship among four agile values and three architecture-centric principles. The middle row shows the most contrastive relationship, and the last row shows the average.

review—into agile approaches. Because most of them indicated a supportive relationship between the approaches in terms of values and principles, we assert that the main problem in combining agile and architecture-centric methods resides not in theoretical issues but in practical matters of adoption.

Our results also showed that agile developers significantly agreed on the value of architectural design patterns for integrating architectural practice into agile methods.

Finally, nonagile developers appeared to be pessimistic compared to agile developers. In particular, the former overestimated the contrasts in agile and architectural approaches.

Our results show that agile developers perceived software architectures as important and supportive to agile values, as opposed to neutral or contrastive. This kind of positive perception bodes well for future efforts to integrate agile and architecture practices. 

References

1. D. Falessi et al., "Applying Empirical Software Engineering to Software Architecture: Challenges and Lessons Learned," *Empirical Software Eng.*, Dec. 2009, pp. 1–27; doi: 10.1007/s10664-009-9121-0.
2. *Draft International Standard ISO/IEC WD4 42010 (IEEE P42010/D6) Systems and Software Engineering—Architecture Description*, ISO/IEEE, 2009.
3. G. Booch, "The Future of Software," invited presentation, *Proc. 22nd Int'l Conf. Software Eng. (ICSE 00)*, IEEE CS Press, 2000, p. 3.
4. D. Falessi, G. Cantone, and P. Kruchten, "Do Architecture Design Methods Meet Architects' Needs?" *Proc. 6th Working IEEE/IFIP Conf. Software Architecture*, IEEE CS Press, 2007, p. 5.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.