

The Agile Developers' Perspective on Software Architecture

Daive Falessi, Giovanni Cantone, Salvatore Alessandro Sarcia' *University of Rome Tor Vergata*,
Giuseppe Calavaro, Paolo Subiaco, Cristiana D'Amore, *IBM Software Group*

Abstract. This study aims to explore the relationship between agility and architecture by surveying 72 software developers at IBM. Results suggest that software architecture and agility are not like oil and water. In particular, Agile developers perceived architectures as important and supportive (rather than in contrast or neutral) to agile values. This kind of positive perception of software architectural principles and practice bodes well for future efforts to integrate agile and architecture practices.

Keywords: Software architecture, Agile software development.

Context, aim and motivation

With an industrial strategic vision to investigate the relationships between agility and architecture, we carried out a survey in a large software development company. The main aim of the survey was to explore practitioners' perceptions about the coexistence of agile and software architecture approaches. Our assertion is that the integration of agile and architectural practices can be facilitated by identifying and understanding how agile values can be aligned with architectural principles.

This survey had 72 participants from the IBM Software Lab in Rome, Italy. The participants had eighteen years of experience on the average and most of them had already adopted agile approaches.

Question 1: How can agile developers use software architecture?

We wanted to know if agile developers considered the use of architecture relevant to their work. We asked the participants, "*In the context of agile development, how relevant is each of the following uses of software architecture?*" As uses of software architecture, we adopted those proposed in the standard ISO/IEC WD4 42010 (IEEE P42010/D6). The level of relevance of each use ranged from 0 to 3.

Table 1 shows the level of relevance of each software architecture use on average among agile developers; these uses are ranked based upon the resulting relevance. Results show that thirteen out of seventeen uses have an average level of relevance higher than the average (i.e. last column Table 1 higher than 1.5). In other words, only three out of seventeen software architecture uses are more irrelevant than relevant. We can conclude that ***software architecture is considered relevant in the context of agile development.***

Question 2: When should agile developers focus on software architecture?

“You don’t need architecture to build a dog kennel, but you’d better have some for a skyscraper” (Grady Booch). On the same vein, results from our survey show that a half of the Agile developers believed system complexity to be the reason of focusing on software architecture.

We wanted to investigate the characteristics that make a system so complex to require architecting activities by agile practitioners. We defined three main orthogonal characteristics of system complexity:

1. Geographical distribution,
2. Number of requirements / lines of code, and
3. Number of stakeholders.

Figure 2 shows that a significant number of respondents indicated the number of stakeholders as a reason for architecting, while the highest number considered the number of requirements/lines of code as the main reason.

Question 3: How are agile values and Architecture-centric methods related? Do they support one another or are they in contrast?

Because the integration of agile and architecture can be facilitated by identifying and understanding how agile values can be aligned with architectural principles, we wanted to know the type of relationships between the principles behind agile values and architecture-centric methods. We first distilled three main “principles of architecture-centric methods”:

1. Being driven by non-functional requirements (NFRs),
2. Requiring an up-front investment,
3. Forcing software architecture compliance.

Then participants defined the type of relation (*Supportive* or *Neutral* or *In contrast*) among all of the combinations of such “principles of architecture-centric methods” with agile values, as defined in the agile manifesto.

Among the twelve combinations of agile values and architecture-centric principles, Figure 2 shows the distribution of relationships concerning the most supportive, the most in contrast, and the overall average. Based upon Figure 2, **principles of architecture-centric method are**, on the average, **supportive** (rather than *in contrast* or *neutral*) **to agile values**. Even the most *in contrast* combination among agile values and architecture-centric principles is more *supportive* than *in contrast* (see second row of Figure 2). Moreover, the most *supportive* combination has been considered *supportive* twice as much as *neutral* or *in contrast*.

Additional questions

Further results from the survey show that:

- A large majority of Agile developers felt the need of both **new methods and special training** for integrating architectural practices (software architecture analysis, design, review) into agile approaches. Based on these findings, we assert that the main problem in combining agile and

architecture-centric methodology does not dwell on theoretical issues but it is a matter of practical adoption.

- Agile developers significantly agreed on the **valuable support provided by the architectural design patterns for integrating architectural practice into agile methods.**
- **Non-agile developers appeared to be pessimistic** by overestimating the contrasts regarding the coexistence of agile and architectural approaches.

ID	Uses of Software Architecture	Level of relevance
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 scale of agile practices to large projects.	1.74
10	To document points of flexibility or limitations with the system with respect to 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 system generation and analysis tools.	1.12
Average		1.72

Table 1: Level of relevance of software architecture uses as perceived by agile developers, calculated by averaging the obtained scores (0, 1, 2, 3).

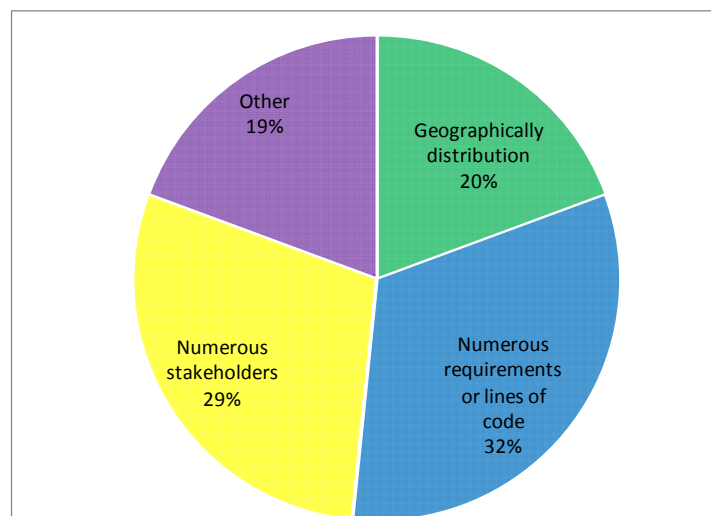


Figure 1: Clustering of system complexity characteristics that require architecting activities.

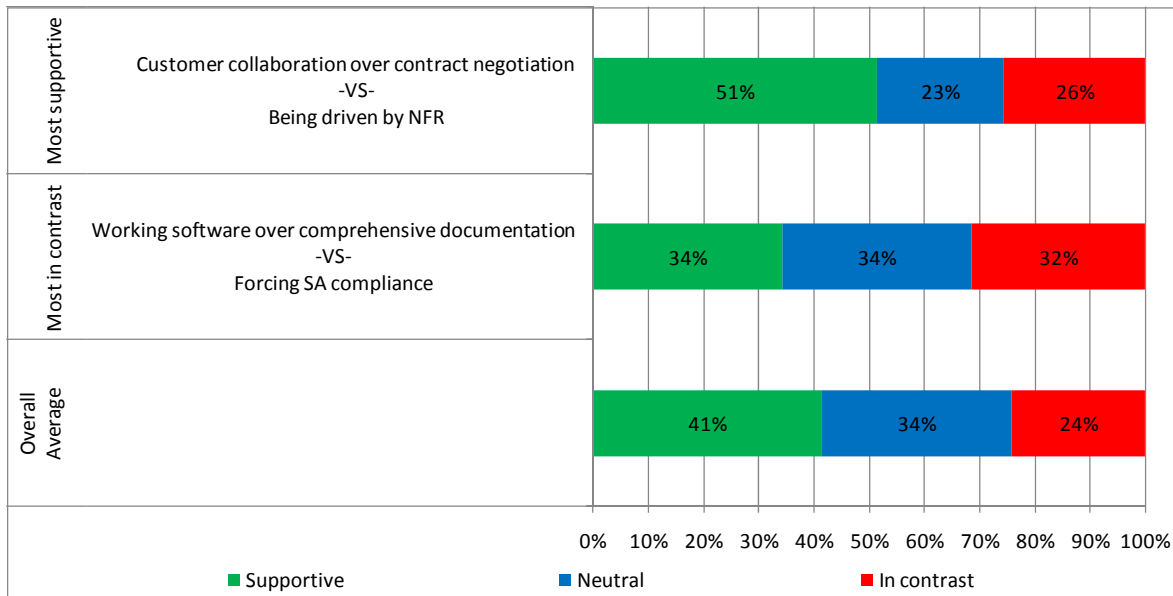


Figure 2: Type of relationship among agile values and Architecture-centric principles.

Our results show that Agile developers perceived architectures as important and supportive (rather than in contrast or neutral) to agile values. This kind of positive perception of software architectural principles and practice bodes well for future efforts to integrate agile and architecture practices.

Biography

Davide Falessi is a postdoctoral researcher at the Department of Informatics, Systems, and Production engineering (DISP) University of Rome “TorVergata”. His main research interests is in devising and empirically assessing scalable solutions for the development of complex software-intensive systems with a particular emphasis on quality, reuse, and evolution. He achieved a PhD and Master degree from the University of Rome “TorVergata”, Italy. Contact him at Viale del Politecnico 1, DISP, 00133 Rome, Italy; falessi@ing.uniroma2.it.

Giovanni Cantone is professor of Experimental Software Engineering, and of Analysis and Design of Software Intensive Systems in the Department of Informatics, Systems, and Production engineering (DISP), University of Rome “Tor Vergata”. From the 70s, his main interests have been and still are in Software Engineering. He is a founding member of the International Software Engineering Research Network (ISERN). He achieved the pre-reform doctoral “Laurea” degree in Electronics Engineering from the University of Naples “Frederick II” (of Hohenstaufen), Italy. Contact him at Viale del Politecnico 1, DISP, 00133 Rome, Italy; cantone@uniroma2.it.

Salvatore Alessandro Sarcia is a researcher at the Italian Ministry of Defense - Army General Staff in Rome. His research interests fall on advanced statistics, non-linear modeling, organizational performance measurement, and computational intelligence applied to (empirical) software engineering. He achieved a PhD from the University of Rome “Tor Vergata”, Italy. Contact him at Palazzo Esercito, Via XX settembre 123/A, 00187 Rome, Italy; sarcia@disp.uniroma2.it.

Giuseppe Calavaro is the Information Management Services Manager in Italy for IBM Software group. His main research interests include software processes, architectures, and tools. He received a PH.D. and a master degree from University of Rome "Tor Vergata", Italy. Contact him at Via Sciangai 53, 00144 Rome, Italy; gcalavaro@it.ibm.com.

Cristiana D'Amore is the Manager of Central Architectural Team at IBM Software group. Her main research interest is software architecture and especially its role in agile software development. She received a master degree in Mathematics from the university of Rome "La Sapienza". Contact her at Via Duchessa di Galliera, 42, 00144 Roma, Italy; cdamore@it.ibm.com.

Paolo Subiaco is the Rational Service Europe Business Leader for IBM Software Group. His main research interests include software processes, architectures, and tools. Paolo has a passion for semantic and innovative smartphone applications. He received a master degree from Tor the University of Rome "Tor Vergata", Italy and he continues to contribute at academic activity as IBM University Ambassador. Contact him at Via Sciangai 53, 00144 Rome, Italy; paolo_subiaco@it.ibm.com.