A FEW INSIGHTS INTO ROMANIAN INFORMATION SYSTEMS ANALYSTS AND DESIGNERS TOOLBOX

Marin FOTACHE¹, Gabriela MESNITA², Florin DUMITRIU³, Georgiana OLARU⁴

DOI: 10.1515/tjeb-2017-0001

Information Systems (IS) analysts and designers have been key members in software development teams. From waterfall to Rational Unified Process, from UML to agile development, IS modelers have faced many trends and buzzwords. Even if the topic of models and modeling tools in software development is important, there are no many detailed studies to identify for what the developers, customers and managers decide to use the modeling and specific tools. Despite the popularity of the subject, studies showing what tools the IS modelers prefer are scarce, and quasi-non-existent, when talking about Romanian market. As Romania is an important IT outsourcing market, this paper investigated what methods and tools Romanian IS analysts and designers apply. In this context, the starting question of our research focuses on the preference of the developers to choose between agile or non-agile methods in IT projects. As a result, the research questions targeted the main drivers in choosing specific methods and tools for IT projects deployed in Romanian companies. Also, one of the main objectives of this paper was to approach the relationship between the methodologies (agile or non-agile), diagrams and other tools (we refer in our study to the CASE features) with other variables/metrics of the system/software development project. The observational study was conducted based on a survey filled by IS modelers in Romanian IT companies. The data collected were processed and analyzed using Exploratory Data Analysis. The platform for data visualization and analysis was R.

Keywords: IS modeling, diagrams, agile methods, CASE features

JEL Classification: M15, L86, C61, C89

¹,²,³ Professor PhD, “Al. I. Cuza” University of Iasi, Romania
⁴ PhD, “Al. I. Cuza” University of Iasi, Romania
1. Introduction

Modeling is a pivotal activity in system/software development that has challenged both developers and researchers due to the dynamics of methodology, methods, models and approaches. In the first years of modeling (1960's), the purpose of the models was to provide a conceptual blueprint for rationally managing the development of the software systems. Such a scheme could therefore serve as a basis for planning, organizing, staffing, coordinating, budgeting, and directing software development activities (Scacchi, 2001). Furthermore, different development methods generate problems due to different notation, semantics, and synonyms of the modelling notations used by these methods, so developers can waste time to understand and agree on the signification of notations (Evans et al., 1998).

The evolution of methodologies for system/software lifecycle development was accompanied by a plethora of diagrams adapted to the methodological approach. Each software/system lifecycle model maps a specific development context, with different characteristics and aims. For example, (Ludewig, 2003, Magdaleno, A.M. et al.) pointed out that for the waterfall the main goals are predictability, stability and reliability. Agile software development focuses on minimal documentation and a greater importance of testing. Another difference concerns the tools supporting the development tasks. Because of their pace and "frugality", one might question how much modeling do the agile methodologies entail?

Despite half a century of IS/software modeling, the major papers showing the extents developers use various methods and tools are scarce. Some studies have been based on the information provided by academia within an experimental (laboratory) context. Some others were based on interviews with developers. A couple of studies used large samples of professionals investigating the relationships among various variables and metrics of modelers' activities.

However, after reviewing the literature in the field, we failed to find any paper approaching the relationship between the methodologies (agile or non-agile), diagrams and other tools (we refer in our study to the CASE features) with other variables/metrics of the system/software development project. This was one of the main objectives of this paper. As Romania has become an important IT outsourcing market, we were interested in finding similarities and differences in terms of IS modeling methods and tools, relative to the modelers' experience in source countries.

Therefore, this paper aimed to provide fresh insights into current modeling features in IT projects deployed in Romanian IT companies. The starting question was "Do the developers currently prefer agile or non-agile methods in IT projects?" Subsequently we dive in the following questions:
1. What are the most popular types of diagrams and tools (CASE features) for Romanian modelers? Does the project industry (the industry of the project customer) influence the use of diagrams?

2. Do project size (budget and duration of project) and the modelers' experience relate somehow with the type of diagrams in use?

3. Is there any correlation between the platform chosen for the application logic and the types of diagram?

4. Is there diagram use (no matter what type/ category) associated with the development method category of the project? What are the diagrams and tools preferred within the agile methods? Are there differences between agile and non-agile development methods in terms of types of diagrams in use?

The remaining of the paper will review the literature dedicated to IS/software modeling, the type of diagrams and tools used in software development for both non-agile and agile methods. Next we will present the research methodology and the results of the observational study. Finally, the paper will include sections of discussion and conclusions.

2. Background and related works

Generally, authors define modeling as a representation of a system and its operating functionalities as an abstraction (da Silva, 2015, Broy et al., 2012, Gorschek et al., 2014) or as a partial and simplified view of a system. The early models for software development date back in the 1960's - see Hosier and Royce (Scacchi, 2001) and Stevens and Jackson (Gorschek et al., 2014). According to da Silva (2015), Broy et al. (2012), Martinez et al. (2013), Topper et al. (2013), Ludewig (2003), the modeling main aims are sharing a view/understanding among technical and non-technical stakeholders, enabling and supporting the communication among them, the documenting and the code generation. Also, models are used as source for planning, coordinating and budgeting the activities of project software development (Scacchi, 2001).

The benefits of modeling are related to the type of models and methodologies employed in software development projects, and also to the tools deployed. Some common perceived benefits of the modeling have been identified as follows: to facilitate communication, promote reuse of components, enable sharing of the information and control of project, increase documentation traceability, enhance the development processes, improve productivity, quality of software, automate code generation, and enhance problem solving capabilities. These all benefits are presented in Gorschek et al. (2014), da Silva (2015), Broy et al. (2012), Scacchi (2001), Martinez et al. (2013), Topper et al. (2013), Oprea (1999).
Frequently presented in the literature (Martinez et al., 2013, Topper et al., 2013, Ludewig, 2003) are the domain models, use cases, functional models, structural models, architecture models, models of user interaction, design models. According to Davies et al. (2005) models are mostly graphic and represent both the state and the dynamics of software component. For each type of the model CASE tools are available that provide code generation, documentation set, versioning, verification and validation (Martinez et al., 2013, Gorschek et al., 2014, da Silva, 2015).

Related to the influence of project size on the modeling, Davies et al. (2005) investigated the circumstances modelers employ conceptual modeling, i.e. "the factors that uniquely influence the decision of analyst to continue to use modeling". They found a relationship between organization size, modelers' experience and the usability of techniques and modeling tools. The bigger the size of the organization the higher is the usage of these techniques. The second identified correlation was: as developers gain experience, become experts, they less need for techniques and modeling tools. Gorschek et al. (2014) find the developers with bachelor degree or higher use models more frequently than others.

Another extensive (as number of respondents and the period of data gathering) study was conducted by Dobing (2008). Here interest was in the frequency for use of diagrams, their goals and their role. This study was addressed mainly to analysts, and less to the other categories of developers. As variables, the study aimed to identify the correlation between the experience of team members (Dobing, 2008; Gorschek et al., 2014), the size of the project in terms of budget (Grosman et al., 2005, Davies et al., 2006, Gorschek et al., 2014), number of lines of code, and type of the project (development of a new system, replacement of the system, improvement).

Gorschek et al. (2014) showed that most studies aim to answer the question how to use patterns, especially UML, disregarding the question of whether they are used or not. That is because, as also specified by Dobing (2008), the literature assumes that UML is de facto standard (Grossman et al., 2005). Therefore, the intention of Gorschek et al. (2014) study was to examine "if, and to what extents, software developers use design models". From the statistics presented, it appears to be the largest study conducted, with 3785 respondents (in 2010). Relations were identified among variables such as: experience in programming, open or closed development (refer to open source and proprietary), level of education, language software used and the type of development - leased consultancy (in house + team from outside), bespoke development (soft command), off-the-shelf products / components development (own products for market or soft key), embedded software (incorporated in other products, like cars, robots, etc.). The main finding was that the use of models, and UML, in particular, seems not to be a de facto standard in software development. Also, the study confirms the result of Davies et al. (2006) concerning the reverse relationship between modeling and experience.
Grossman et al. (2005) and Dobing (2008) found also some correlation between use of diagrams and industry of software application. Other studies have also analyzed the relation between the use of diagrams and the type of projects like open source or closed, in house or outsource, own product or demand one.

Concerning diagrams and tools used, Davies et al. (2006) concluded that the most popular was ER diagram and data flow diagram, rather than specific UML diagrams. That finding was somehow justified considering the pioneering stage of UML. In contrast, the study of Dobing (2008) revealed that UML is the most popular modeling tool, but had not necessarily become a standard in the world of developers. Developers did not directly consider themselves as UML practitioners. Similar findings for the Romanian market were in Avram (Olaru) and Fotache (2014), and Fotache et al. (2015). As tools Visio, Rational Rose, Oracle9i Developer Suite were nominated more frequently. In Grossman et al. (2005) the most popular tools was Enterprise Architect (producer of this tool is Australian, the country where the study was conducted), followed by Rational Rose and Visio. Cherubini et al. (2007) argued that modeling is not always in a CASE tool, but often on the board or on paper. Hence, it is not always possible to apply reverse engineering and any integration models. In fact, this difficulty in integrating the various models and synchronizing them was revealed also in Peter (2013) and Madanayake et al. (2016).

Most articles address modeling issues from the perspective of communication or of a designer toolbox. There are scant references on the relationship between diagrams and programming languages or platforms for application logic. Such a reference is Gorschek et al. (2014). They show the developers working in C# or C++ use modeling tools more often than those who develop using Java.

3. Research methodology, data collection, variables

In this paper, the use of diagrams and tools in agile methodologies for IT projects deployed in Romania was investigated using an observational (non-experimental) study based on a survey targeting software developers, mainly information systems analysts and designers. The survey’s main topics were: types of IT projects developed, methodologies and tools used in information systems analysis and design, analyst/designer profile (educational and professional background, demographic structure), main issues encountered when recruiting for analysts/designers, and the opinion of the professionals regarding the university curriculum and information systems analysis and design teaching. Collected data were processed and analyzed using Exploratory Data Analysis (Behrens, 1997). R was the software platform for performing data exploration and analysis. Graphics were created using ggplot2 package (Wickham, 2016).
The survey was distributed through e-mail and a social (professional) network. In the first half of 2013, 139 people filled the questionnaire. Most of the respondents were alumni of Business Information Systems program in Alexandru Ioan Cuza University of Iasi. Each respondent could fill details for up to three projects she/he had been involved in.

Of questionnaire variables, those necessary in this study only one variable was related to respondent profile: experience (in years) as information system analyst/designer (ordinal). The remaining ones were related to IT projects:

- Industry of the project customer (Automotive, Education, Finance, etc.);
- Development method used in the project (Incremental, Kanban, RUP, Scrum, Spiral, V-model, Waterfall, XP, Other Agile, None);
- Development method category (Agile, Non-Agile, None) which was determined by grouping the values for variable Development method;
- Binary variables indicating whether diagrams of type Use Case, Activity, BPMN, Sequence, Class, Communication, Entity-Relationship were used or not in the project;
- Binary variables indicating which CASE features (tools) were exploited within the project: Code Generation, Diagrams creation, Documentation Generation, and Teamwork;
- An ordinal variable indicating the size of the project budget;
- A numeric variable indicating the project duration (in months);
- Nominal variables (five closed answers, one open answer) for specifying the platform for application logic used in the project (Java/Java-based Framework, .NET, C/C++, etc.);
- According to the nature of each question and the available data, the following data analysis tools were used and explained in detail:
  - Data visualization: bar plots, mosaic plots, scatter plots, and boxplots;
  - Tests of Independence of nominal variables (the chi-square independence test and Fisher's Exact Test for Count Data);
  - Normality tests;
  - Kruskal–Wallis test (one-way ANOVA on ranks).

4. Results

The first question under scrutiny was "What the most type of diagrams and CASE features are most used by Romanian modelers IT projects? The barplot in figure 1 (left) shows that Use Case was the most popular, followed by Entity-Relationship and Class.

As for the CASE features used within the project (zero, one on more for each project), figure 1 (right) shows that the most popular CASE feature was Diagram creation, followed by Teamwork and Code Generation.
Figure 1. Diagrams (left) and CASE features (right) used in the projects

Next, the relation between the types of diagrams and the CASE features was investigated. Figure 2 displays a strong association between the number of CASE features and the number of diagram types used within the projects. Reported Spearman correlation coefficient (variables are not continuous and have a limited number of values so Spearman correlation coefficient was preferred to classical Pearson's) was 82.3% which is high.

Figure 2. Association between the project number of CASE features and the number of diagrams
The next research question under examination was: *Do the specific CASE features exploited in the projects relate to specific diagrams?* The mosaic plot in figure 3 shows little variances of type of diagrams among nominated CASE features.

![Mosaic plot showing CASE Features vs. Types of Diagrams](image)

**Figure 3. Association between CASE features and diagrams**

The chi-square independence test, whose null hypothesis was H0: Types of diagrams and CASE features are independent, returned $X^2 = 6.3215$, df = 12, $p$-value = 0.899. That confirmed that the types of diagrams in use do not manifest a statistically significant relationship with the nominated CASE features used in the projects.

Next research question was: *Does the industry of projects influence the use of diagrams?* Figure 4 shows that the diagram usage differs among industries of the project. Projects targeting customers in Data Management services, Education, Finance and Telecom seem to use in a lesser extend the diagrams than projects in E-commerce, healthcare, and manufacturing. Projects targeting IT security seems to not be use diagrams al all whereas all projects in Logistics exploited at least one diagram but in these two cases the number the projects of these type is very low.
The variation among industries does not seem to be significant, since the chi-square independence test between variables diagram usage and industry (H0: Use of diagrams and Project Industry are independent) returned $X^2 = 21.24$, $df = 15$, p-value = 0.1293, and Fisher's Exact Test for Count Data computed a p-value of p-value = 0.1242 (fail to reject H0).

Further questions were: Is the influence of project size (budget and duration of project) and the modelers experience related to the type of diagrams used?

Mosaic plot Figure 5 displays the association between the types of diagrams and the project budget magnitude.

Apparently (figure 5), only two types of diagrams (both, non-UML) - BPMN and Communication manifests a slightly different proportion in budget size structure. The difference is not statistically significant, as the chi-square independence test ($X^2 = 13.703$, $df = 18$, p-
value = 0.748) and Fisher's Exact Test for Count Data (p-value = 0.8672) confirmed the lack of relationship between two variables.

Figure 5. Association between diagram usage and the project budget

Next part question was: *Are the types of (modeling) diagrams related to the project duration (e.g. are specific type of diagram more appropriate for longer projects; or, do longer projects tend to use specific types of diagrams, different from the diagrams used in shorter projects)?*

Boxplot in figure 6 displays some variation of project duration along different types of diagrams. But project duration median is strikingly similar for all types of diagrams. Since variable `proj_duration_months` is not normally distributed (W = 0.69179, p-value < 2.2e-16), the non-parametric Kruskal-Wallis test was used for testing the significance of project duration median among different types of diagrams. For this test the null hypothesis was: H0: there is no difference in terms of project durations among different types of diagrams methods used by the project. Kruskal-Wallis test returned chi-squared = 1.4653, df = 6, p-value = 0.9618, so we fail to reject H0! So there are no statistically significant relationship between different types of diagrams and the project duration. Also pgirmess package, which provides the nonparametric multiple comparisons, found no significant differences between any two types of diagrams.
Figure 6. Distribution of project duration, by type of diagrams

Next questions: Are the types of diagrams and the modeler experience related? Do more experience designers tend to use different types of diagrams than less experiences ones?

Figure 7. Association between types of diagrams and designer's experience
Both the mosaic plot in figure 7 and the chi-square independence test (X-squared = 14.777, df = 18, p-value = 0.6772) support the idea that diagrams used in the project do not relate to the experience of the modelers.

The next investigated question was if there is a correlation between the platform chosen for the application logic (programming languages) and the type of diagrams?

![Figure 8. Association between types of diagrams and platform for the application logic](image)

Figure 8 shows no large variation of the diagrams structure along different platforms for application logic used in the project. Both the chi-square independence test (X-squared = 15.406, df = 20, p-value = 0.7527) and Fisher's Exact Test for Count Data (p-value = 0.6826) confirmed the lack of relationship between two variables.

The next investigated group of questions was: Is there any association the use of diagram (no matter what type/ category and) and the development method category in use within the project? What are the diagrams and tools used in the agile methods? Are there differences between agile and non-agile development methods in terms of types of diagrams in use? This group will offer the answer for main hypothesis of our research.

The mosaic plot in figure 9 on the left shows almost no difference between agile and non-agile methodologies in terms of using at least one type of diagram. The graphic is confirmed
statistically as the both the chi-squared test of independence and Fisher's Exact Test return: X-squared = 0, df = 1, p-value = 1.

**Figure 9. Diagram usage vs. category of development method (left) and association between types of diagrams and development method category (right)**

For the question: *Is there any association between the type of diagram and the development method category in use within the project?*, the mosaic plot in figure 9 (right) displays no association. Also the test of independence (H0: Diagram and Method Category are independent) returned X-squared = 8.3949, df = 12, p-value = 0.753. We fail to reject the H0. Apparently, the diagrams in use do not vary significantly on the development method category (agile vs. non-agile vs. no method).

The next question of interest was *whether there is a statically significant difference between projects using agile methodologies vs. projects using non-agile methodologies in terms of number of diagrams in use?*

As graphics in figure 10 shows, there seems to be no difference among categories of development method (agile vs. non-agile), and the number of types of diagrams used in the project in both cases of all projects (left) and of projects using at least one type of diagram (right).
Based on these results, the answer to the main question could not be determined unambiguously because there are no notable differences between diagrams using in the agile methodology versus non-agile ones.

5. Discussion

This study aimed to identify the ways modeling (diagrams and modeling tools) is influenced by development methods used (agile versus non-agile versus no method at all). A series of successive questions was developed. Thus the question about the most popular diagrams (figure 1 left) confirms what was revealed in Dobing et al. (2008), Petre (2013), Gorscheck et al. (2014), and Davies et al. (2006). Class diagrams, sequence diagrams, activity, state machine diagrams were mentioned in these studies, usage percentages depending on the survey year. Thus, Davies et al. (2006) noted the most popular diagrams are entity-relationship diagrams, data flow diagrams and in lesser extent the UML diagrams.

Regarding modeling (and CASE) tools, our study found that the most popular were Enterprise Architect, Visio and Visible Analyst, while in Davies et al. (2006) the most frequent were Visio and Rational Rose. In Gorscheck et al. (2014), it was found that most developers employed paper and board for making models (nearly 12%), while only 1% mentioned the use of a computerized tool. Although our study did not identify such a correlation, which could be a subject of a further research, apparently using a tool or another is dependent on the type of tools taught during studies and training of the developers. This is supported by Gorscheck et al. (2014) and Grossman et al. (2005), in which more than 70%, respectively 80% of those using modeling hold at least a bachelor’s degree.
In the literature, only Petre (2013) and Grossman et al. (2005) related IS/software modeling to the *industries for which the software was developed*, and variables were considered with respect of respondent demographics, and less about the relationship industries-diagrams. However, it can be concluded that the relationship between the industry of the software development and the use of diagrams is not necessarily significant, as confirmed by our study in figure 4 and the independence test.

Regarding the question on the link between type of diagrams and the budget of the project, the results of our study (figure 5) shows there is no significant association between the two variables. This is confirmed also in Grossman et al. (2005) and Dobing (2008). The result could have an explanation based on a subjective interpretation of the budget amount (that would include only the wages and cost of equipment purchases, only wages and profits, etc.) and the real information held by respondents. On the other hand, Davies et al. (2006) found a notable association concerning the size of the project and the diagram usage, but the size of the project is somehow assimilated with the size of the organization which would develop the software.

Besides the budget, we have pursued an analysis of association between the project duration and the diagrams used. According to the results (see also figure 6), apparently, the length of the project does not determine in a significant manner the type of diagrams used.

As for relationship between diagrams and modelers’ experience (figure 7), our study shows that diagrams used in the project do not relate significantly with the experience of the modelers. In Dobing et al. (2008) medium experienced developers use diagrams most frequently and the more experienced used diagrams in a lesser extent. Similar results were found by Gorschek et al. (2014) and Dumitriu and Mesnita (2002). This difference might be explained by the smaller number of our respondents. Also, we were interested mainly in analysts or designers (modelers), and not so much in developer categories.

For questions concerning the relationship between the types of diagram and the platform used for the application logic (figure 8), our results show no significant correlation. Gorschek et al. (2014) find for C++ and C# developers a positive dependency, i.e. developers employing these programming languages have a higher interest in modeling than respondents using other languages. Somewhat unexpected was that for Java developers no dependency could be established, and Python or Ruby programmers are the “rare” category. Gorschek et al. (2014) find a possible reason for this situation: developers using C++and C# paid more attention for documentation and updating documentation, while developers using Python and Ruby have no such interest. But this explanation does not cover the difference for Java.
About diagrams and tools used in agile methods, we had expected that diagrams are not used on a large scale in projects using agile method. But figures 1 and 2 show that there is no correlation between the method and the type of diagrams and tools used. We could suppose this situation is due to the small number of projects analyzed by our study. Many respondents answered with “None” for the question regarding the development method. Also, the method used does not affect the number of diagrams in the project. Therefore, this issue requires a more detailed analysis, in terms of diagram types and tools used in agile projects, as well as the motivation and perception of developers to use them.

6. Conclusions and future direction

The main target pursued in this paper was to find a link between the type of diagrams built within the IT projects and some other variables, such as: applied methodology (agile vs. non-agile), project size, experience of developers, programming languages, industries and CASE features in the IT companies in the Romanian market. This approach could prove to be important for both companies and academia, from Romania as well as from other countries. In the latest 10-15 years Romania has become a software development center. Such kind of study can help in the adoption of techniques, tools and methods according to the project requirements, and also modelers experience.

Therefore, our research followed to answer to basic research question "Do the developers use modeling in their work whether they use agile or non-agile methods?"

Similar to the findings of other studies, our research shows that UML diagrams are the most popular among developers. We fail to identify a strong association between the types of diagrams and the CASE features employs in the project, but there seems to be a strong correlation between the number of CASE features and the number of diagram types used in projects.

Regarding the influence of project industry for the modeling, further analysis is needed for investigating the relationship between the area (industry) of the software project and the decision to use or not certain diagrams and tools.

Although there are important factors in the success of a project, the budget and the duration do not appear to have significantly influence in diagrams usage. On the other hand, there are evidences that budget has great influence on the methodology, tools and models used.

Finally, there seems to be no correlation between the method category (agile vs. non-agile) and the type of diagram. We might suppose this situation is due to the small number of projects analyzed by our study and small number of respondents that answered the question.
related to the development method. Also, the method does not seem to affect the number of diagrams within the project. Therefore, this issue requires a more detailed analysis, in terms of diagram types and tools in agile projects, as well as the motivation and perception of developers to use.

Even if the subject of models and modeling tools in software development is important for the company, there are no many detailed studies to determine if and for what purpose the developers, customers and managers use and decide to use the modeling and specific tools. Consequently, for a future direction of the research a new observational analysis will be conducted about diagrams and CASE features in Romanian company, with focus on association between level and type of training and the CASE features used. Also, more attention must be paid for the relationship between type of projects (Gorschek et al., 2014), industries of projects, diagrams and tools used. It would also be interesting to further investigate the influence of the platforms for application logic (programming languages) on type of diagrams, given that many of these platforms now have important CASE features.

References


Information and Management Sciences, Book Series: Series of Information and Management Sciences, Vol. 1, pp. 66-75


