Journal of Web Engineering, Vol. 6, No.1 (2007) 073-096
© Rinton Press

INDUSTRIAL ACCEPTABILITY OF WEB DESIGN METHODS: AN EMPIRICAL STUDY

FRANCA GARZOTTO

Department of Electronics and Information, Politecnico di Milano (Italy)

VITO PERRONE

Department of Computer Science, University College London (UK)
{garzotto@elet.polimi.it , v.perrone@cs.ucl.ac.uk}

Received May 15, 2006
Revised July 25, 2006

In this work we present the results of a study that has aimed at identifying the requirements for Web design methods that may influence the industrial acceptability, that is, the characteristics that prevent, or contribute to, the adoption of design methods in a business environment. The empirical study involved (by way of focus groups and surveys), over 100 potential users of Web design methods including project managers, analysts, information architects, visual designers, implementers, recruited from companies and non academic institutions intensively involved in the development of Web based applications. Our study has gathered qualitative and quantitative information that highlight expectations and needs of stakeholders of Web design methods. It has highlighted that usability, modularity, scalability, customizability, support to fast prototyping and incremental development, support to design-related activities (training, project management, design documentation delivery) are critical requirements for a design method to be adopted in the industrial practice. To define our study, we have adopted a holistic perspective. We have investigated requirements looking at design methods as to engineering products that should work within the overall development process in which design occurs, and within the organizational context in which this process takes place.

Key words: design method requirements, Web design, industrial acceptability, technology transfer, requirements engineering, survey, focus group
Communicated by: D Lowe & O Pastor

1 Introduction

Over the last decade, we have witnessed an increased proliferation of design methods in the Web application domain, most of which developed in academic contexts. Existing methods differ by level of abstractions, notations and primitives, but all of them aim to help developers master the complexity of Web design and improve the quality and cost effectiveness of the overall development process. However, some studies [3] highlights that most methods have been nearly neglected outside the academia. In the attempt of understanding this phenomenon, we carried on an empirical study aimed to discover the needs, expectations, and current practices of the “real” users of Web design methods. By analyzing the results of such a study we expect that requirements contributing to the industrial acceptability of Web design methods might be unveiled. Retuning our methods according to what has emerged from the practitioners' inputs might increase our (and in general the community's) chances to overcome the hurdle of acceptability.

In our study we have adopted two well known techniques for user research - focus groups and survey – which are known to complement each other, and, combined together, potentially achieve more reliable results [22]. Focus groups are collective interviews and discussions involving a small set of target representatives per session (usually from six to twelve persons). Focus groups allow collecting qualitative information which are difficult to reduce to quantitative data for systematic quantitative analysis. On the other hand, they are acknowledged as a good tool to get a general intuition of why an "audience" behaves as it does, to uncover desires and thinking processes, to collect anecdotes, and to investigate group reactions to ideas. Our focus group sessions had mainly "exploratory" objectives, aiming to capture the participants' general desires concerning Web design methods, to unveil their expectations, to understand the motivations underlying the adoption or rejection of a new method. We used the focus groups' outcomes to formulate a set of hypothesis that have been then tested on a larger sample of users, by way of a survey, whose results have been quantified, analysed, and compared using statistical techniques. The survey questions focused on the characteristics of Web design methods that emerged from the focus group sessions as relevant factors that may prevent, or contribute to, the adoption of a design method by industry. Respondents were asked to judge the relevance of various features and to prioritize them, as well as to identify some aspects of their current design practice.

Given the very nature of Web applications, heterogeneous types of professionals are usually involved in the various stages of the Web design process, and are potentially exposed, at different degrees, to design methods. Addressing such heterogeneity, the participants of our study included persons with different profiles: project managers, analysts, information architects, implementation designers. Participants were recruited from small, medium and large sized firms and public and private institutions which are involved in Web application development.

The remaining of this paper is organized as it follows. Section 2 maps out the current scenario for Web design methods and contextualize our study; Section 3 describes the focus group design and its qualitative findings. Section 4 presents the survey and the statistical analysis of its results. In section 5 we summarize our reflections drawn on the achieved results and in section 6 we draw the conclusions.

2 Current Scenario and previous studies: lack of non functional requirements

In literature several methods have been proposed for supporting the design of Web applications. Most of them – OO-HDM [26], WebML [8], HDM [15], W2000 [2], UWE [17], WSDM [10], OO-H [16], and many others – have been proposed by the academic community, whilst the most known proposal in the industrial community is the WAE (Web Application Extension) proposed by Jim Conallen [9] in Rational (now IBM). The validity of such methods and ideas has been clearly confirmed by numerous successful publications in world-class conferences and journals. Moreover, their inventors have usually assessed the practical applicability of their proposals on the field by using them in applied research projects and reporting high level results.

Although to some extent these methods have been successful, our experience and on-the-field studies like [3] [5] [12] highlight that practitioners in industry are not using them. However, while such studies clearly show the poor adoption of design methods for Web applications, no study has investigated, as far as our knowledge is concerned, what may encourage their adoption or, conversely, what obstacles are hindering their diffusion.

Adopting a holistic approach, we have considered design methods as engineering products with target market and stakeholders. As any other engineering product, their success is strictly related to the degree by which their requirements are accomplished. Traditionally, requirements are divided between functional and non-functional ones. Functional requirements for a design method can be considered all those dictating what design aspects a method should support and may be either domain dependent or domain independent. Investigating functional requirements is out of the scope of this work, even because we believe that, given the high quality results reported in literature, this category of requirements has been extensively analyzed and most of current methods accomplish them [25]. Non functional requirements are those concerning quality aspects like usability, performance, accuracy, flexibility, etc., process requirements like costs, time etc., and interfaces requirements. This requirements category has not been explicitly investigated so far but lack of support for such requirements might be one of the main causes of the fact that existing design methods have only achieved a partial success. In this light, our investigation can be seen as an attempt to identify non functional characteristics of design methods since it does not address specific design aspects but rather the qualities that might influence their acceptability in the industrial setting.

3 Focus Groups

3.1 Focus group organization

We recruited the focus groups participants from companies or governmental institutions who (at the time of the study) were partners, with us, of large multi-team projects at national or international level, in different domains such as e-commerce, e-banking, e-publishing, e-culture, e-learning.

We organized six focus group sessions, each one running for approximately two hours and involving six-eight persons (as well as one moderator and his or her assistants). For each focus group we prepared a set of general issues to loosely guide the discussion. These issues were elaborated on the basis of our experience in building and using our own design models (HDM [15] and W2000 [2]), on studies reported in literature (see previous section), and, as the study proceeded, on our understandings from previous focus groups. However, we opted for a free-flow conversation, instead of constraining it in a question/answer schema, to foster uncovering of unconsidered opinions, desires, experiences, and feelings. To stimulate participants to focus on experiences and problems, rather than jumping to solutions, we suggested a scenario-based approach [7]. We invited participants to talk about design stories, episodes, or anecdotes occurred during their design activities, sometimes interplaying their narrations with questions such as “what was the most frustrating (or satisfying) situation you experimented here, and why?”, “how easy/difficult it is for you to share design decisions with other team members, and why”, “if you had this feature would you use it?” By way of this “story telling” work, and the discussions that arose, we could shed light on the complex experiences of the interviewees; the way they carry on design tasks; the organizational contexts which they work in; the problem they face and how they solve them; the individual (sometime shared) idiosyncrasies; the parameters which they base their choices and take their decisions on, the erroneous expectations and disappointments, the prejudices and mental models they have.

3.2 Focus groups findings

During the focus groups we collected a lot of raw information, recorded by means of annotations, on-the-fly "transcripts", visual sketches, post-session debriefing notes, which needed a first elaboration to be further analyzed. Here the collected data was interpreted, cross-related among the various interviews, filtered and clustered. The main findings of this first part of our study can be described as it follows (in brackets we quote the participants' words as recorded during the various sessions).

"The concepts of design and design method may be ambiguous"

The term "design method" has different meanings and is used in many different ways (which oftentimes happens also in the scientific literature, especially when the term is used in different communities). In some cases, the notion of design method has an informal sense: a set of in-house practical guidelines on how to carry on the design activities. For some participants, design method is synonymous of design modelling language i.e., a set of concepts, primitives, and notations to describe the design solutions concerning a system under development, or some aspects of it. For other participants, a design method is "a kind of methodology", providing modelling features plus "something concerning how to use them throughout the design activities". A design method may address "how to do design" at different degrees of complexity, rigour, and granularity, ranging from heuristic guidelines to a more formal process [23], expressed by means of tasks, activities, and phases, their inputs, outputs, constraints, mutual relationships and dependencies, resources and needed expertise, etc. In the rest of this paper we will use the term design method in the third sense, and design model as a shortcut for design modelling language.

"The organizational scenario is a critical issue"

Project managers look to software development from "business oriented" and "process driven" perspectives. They have in mind the current practice, the organizational and budget constraints of their institution and they are strongly concerned both by the cost of introducing a new method and by the organizational changes it may require in the current workflow (a phenomenon sometime referred as "organizational inertia") [20]. They would like to be able to introduce it partially and to integrate it in the current practice progressively. In addition, managers look for processes that are well structured, systematic, easy to monitor. They look for tools that support managerial activities related to design, such as planning and monitoring design activities in terms of time, resources, expected outputs, milestones, etc. Finally, managers are concerned about training issues, as they may face situations of frequent turn-over. Newcomers must become productive in the use of a method as soon as possible, and availability of good documentation, training modules, a "reference workflow" for the design process, are all considered important incentives to the adoption of a new method.

“We want to learn a method fast and to use it easily”

The majority of participants agreed that the facility to understand and to use concepts and notations of a design method is a key criterion for its adoption. Only few persons, involved in the advanced phases of design, declare that they can sacrifice usability to the richness of a design method and to good tools for fast prototyping or code generation (discussed in the follow). The availability of high quality, usable documentation is considered an important issue for both improving ease of use and achieving good learnability rates. One of the key criticisms on academic methods is the lack of documentation that is targeted to practitioner and that offers real examples and complete real case studies.

“We need live support”

Most of the participants stressed that applying a method to the design of complex applications is much more effective if there is a proper “mediation” by experts. An approach that turned out to be effective in some participants’ experience is to be trained on a design method in a short and intensive course, and then carrying on together (trainers and trainees, side by side) some design activities in at least a small part of a real project.

“We keep reusing portions of our previous designs in different contexts”

Expert designers work a lot “by reuse” in the different projects and constantly apply (consciously or unconsciously) design patterns [14]. There is a general agreement about the value of patterns as a means for improving the effectiveness of design, sharing design experience, and improving communication. Some participants indicate that they use patterns (either offered by some pattern book or defined inside the organization) in their team as a jargon to describe in a few words design solutions that would otherwise require a lot of words or graphical notations to be communicated.

“We speak different languages... but we need to cooperate”

In the different stages of the development cycle of a significantly complex interactive system, the design involves different tasks and heterogeneous knowledge and skills. A Web design team usually includes domain experts, economic analysts, communication experts, information architects, graphical designers, software architecture designers, security experts, and so on. In this context, several participants highlighted two conflicting requirements. On the one hand, there is the need of different design languages and guidelines for addressing the different design tasks. During the very early stage of design, designers need to sketch solutions and to use a design method mainly as a tool for brainstorming and reasoning, or for discussing ideas with developers, customers, users. Moving towards implementation, designers need more formal and detailed modelling languages, closer to the business logic and the software architecture of the system-to-be. On the other hand, designers need means by which the different kinds of people involved in the design can communicate, understand each other, discuss their opinions, and ultimately cooperate.

“We want to be free”

Participants’ stories show that expert designers feel uncomfortable within the constraints of a rigid, sequential design workflow. Being requirements definition a fluid and evolving task, design hypothesis may change rapidly, partial design solutions are continuously produced, and often need to be reworked. Thus focus group discussions confirm that in complex applications a design process is intrinsically iterative and incremental [19]: application design specifications are produced through a set of progressively improved versions, iterating the same design task several times. In addition, the reported experiences highlight that designers use the same method in multiple ways, according to the different practices used in a company or the individual mental model and style. Expert participants like to customize the design notations and primitives and the design process, to adapt them according to the different situations of use induced by different application fields, project constraints, working context, and so forth. They regarded the customization capability of UML [4] as an appealing feature of this method, although only a few of them actually used it.

“Writing design documentation requires too much effort”

Delivering a good quality design documentation is crucial both for managerial reasons (being sometime the contractual basis for discussing the development follow-up with the customer) and for implementation effectiveness (to avoid misunderstandings with the implementers). Most of the participants complain about the effort needed to work by hand when writing design specifications and pack them into good design documents. While some participants were more in favour of a “generic” design tool, other invoked a tool strongly tailored to the specific design method they adopt. Several of them claim that the availability of specification tools like Rational Rose™ is a key reason of the UML success. Among their desiderata for a support tool, they mention features like versioning, consistency check (an activity where machines are more effective than humans), and support to clustering and multiple views of design specifications.

“We need to move to implementation, after all”

Many participants were concerned with the mapping from design to early prototyping and to implementation. They mentioned the sense of frustration arising when, after the effort needed to produce accurate design specifications, they have to translate them by hand and from scratch into a mock up prototype or more mature implementation structures. Many participants consider the code generation facility of Rational Rose™ (which generates object oriented code skeletons starting from UML specifications) one of the main appealing features of this tool.

“Standards are not a panacea”

The issue of “being a standard” for process, model, or method, received divergent opinions, ranging from someone who very much appreciate standards to others who, more cynically, do not consider standards as an issue at all.

4 The survey

On the basis of results gathered in the focus group sessions, we identified a significant number of hypotheses which needed a further validation by way of quantitative data analysis. A survey was thus built addressing three main goals:

1) Drawing a picture of the current design practice in industry and to analyze whether the current practice may influence the users’ opinions about design method issues

2) Validating (or invalidating) desires, opinions, and trends discovered during the focus groups on the basis of a larger sample and by means quantitative data suitable for statistical analysis

3) Ranking the relevance of different requirements about several aspects of design methods and support documentation and tools

4.1 Recruitment

Target companies and organizations were recruited among those ones we had some direct or indirect (by means of our partners or colleagues) relations with. This strategy was motivated by two main reasons. First, we wanted to avoid a low response rate[a] that might cause a non-response bias on the survey results [6]. Second, since the questionnaire required about 20-30 minutes to be filled up, having exploited such connections made us more confident of the achieved responses.

The survey was sent by e-mail messages, addressed to our (or our partners/colleagues') contact persons within the company, with the instruction of distributing the survey package to from 1 up to 5 his/her colleagues (depending on the company size, 1-2 for small, 1-3 for medium and 1-5 for large). In the instruction we also requested to involve persons who had not worked with us (or our partners/colleagues) and with specific profiles: professionals with at least 5 years of experience, covering one of the following roles:

- Project managers (PM): addressing those members of a development team who usually have a strong technical background built in several years working in the field and are in charge of the whole development process of a system. They should have a global view on which the required characteristics of a design method are in all the project phases. We also expressed, in the e-mail, a preference about this kind of profile because we supposed it should have a more comprehensive view on requirements and, in some cases (mostly in small and medium companies) could influence the decision of adopting or not a new method.
- Senior analysts/designers (SA): addressing those members of the project team who are in charge of, respectively, acquiring and analyzing the system requirements and designing the early solution. We grouped together these two roles because, in our experience, in many project concerning Web systems the same persons may cover both these roles in the early phase of the project. Moreover, in medium companies these roles may be undistinguished.
- Senior Developers (SD): by this role we intend those members of the development team who are in charge of defining in detail the software design and leading the implementation activities.

In synthesis, in this investigation we mainly addressed those professionals who use (or should use) design methods to support their daily work, or at least are exposed to design specifications and, thus, involved in the decision of whether adopting or not a new design method. This means that this sample is a purposive, non-probabilistic one. Our selection of the sample was driven by the the same argumentations used by Fitzgerald [13] in his survey. He states that: "…researchers have criticised the lack of use of random sampling strategies in Information System research [21] (cited in [13]). However, Mason ([24] cited in [13]) has pointed out that in IS research, access is often one of the

[a] In similar surveys, Barry & Lang (2001) received only 10% of usable responses and Fitzgerald (1998) 21%.

problematic issues. He suggests that rather than criticising the lack of true random sampling, researchers should strive to construct samples that allow the most powerful inferences to be made. Therefore, one of the principles guiding the sample selection was that of ensuring that participants would be likely to be significantly and directly involved in the phenomenon of interest. This strategy is in keeping with that recommended in [11] which recommends that samples be chosen for theoretical reasons so that the phenomenon of interest is more likely to be present."

The questionnaire was sent to 11 organizations in North and South America (5 and 6 respectively), and 70 in Europe (from 8 different countries) in two sessions 6 months apart one from the other. Due to the short distance between the two sections and considering that we targeted the same kinds of companies, we consider them as a unique sample for the sake of this study.

4.2 Questionnaire structure

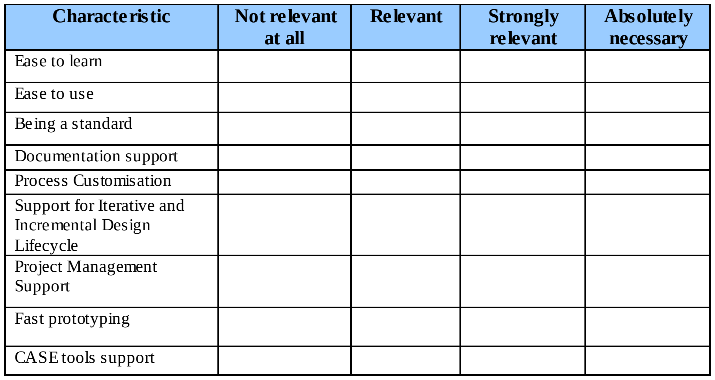
The questionnaire is organized in four sections: 1. Profile and Current Practice; 2. General requirements for a design method; 3. Requirements for Support Documentation; 4. Requirements for Support Tools. Each section includes several groups of questions, definitions of the terms and is complemented by explanations of the questions when needed. Moreover, the questionnaire has a companion glossary where each non trivial term used to formulate the questions is clearly explained.

Section 1 - "Profile and Current Practice": aims to identify the respondent' profile (according to the categories listed above) and explores methods and processes currently adopted within the respondent's company in projects where the respondent has been involved. This part investigates three main aspects of the current practice:

- **Business Logic design**: what business logic design methods (Object Oriented, RAD, etc.) have been used so far
- **Web design**: focusing on Web systems development, what Web engineering design methods have been used to design specific aspects of Web applications like navigation, presentation, etc. Besides specific Web engineering design methods (HDM, OOHDM, WebML, etc.) and the usual "others" and "none" options, we also include the generic "UML" option to investigate whether UML is also used to this purpose, despite the fact it lacks of a proper expressive power with this respect [12]
- **Design Process**: what design processes have been adopted in most projects

Section 2 – General Requirements of a Design Method: addresses general requirements of a design method. Firstly, respondents are asked to rank (using a Likert scale: *Not Desired at all*, *Desired*, *Strongly Desired*, *Absolutely Necessary*) the relevance of each characteristic (see Figure1), then to mark which specific characteristics of each individual aspect they consider important for a method (see some samples of questions in figure 2).

Section 3 – "Requirements for Design Documentation": addresses requirements of support documentation about the design method and aims to identify what kinds of documentation practitioners would wish to be provided with.



Characteristic	Not relevant at all	Relevant	Strongly relevant	Absolutely necessary
Ease to learn				
Ease to use				
Being a standard				
Documentation support				
Process Customisation				
Support for Iterative and Incremental Design Lifecycle				
Project Management Support				
Fast prototyping				
CASE tools support				

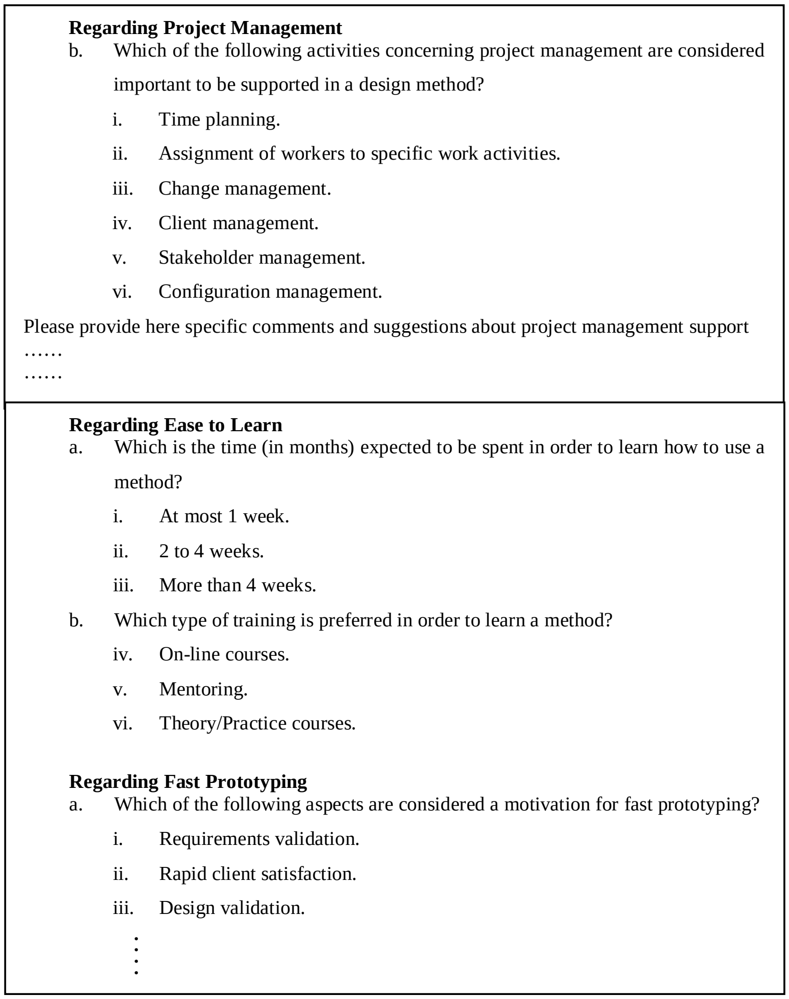
Figure 1: Investigating general requirements on a design method

Section 4 – “Requirements for Support Tools”: addresses requirements focused on tools that support the design specification and other activities related to design (documentation delivery, fast prototyping, code derivation, etc.). This section includes several questions where respondents are asked to judge the relevance of some specific tool characteristics using a four-point Likert scale as shown in section 2. In particular, the investigated features are been:

- **Flexible models management:** features that allow designers to easily switch back and forth among different design tasks
- **Model Versioning:** feature that helps designers manage different versions of their design specifications (e.g., produced by different authors or at different design stages)
- **Code derivation:** feature that allows the generation of code skeleton from design specification
- **Semi-Automatic generation of prototype:** features that allow the generation of a prototype application from the design models. Responses were also cross-checked with those acquired in section one where we asked what kind of prototype is is desired and what purpose prototyping should serve
- **Integration with a design method:** questions that aims to explore the desired degree of adherence of a CASE tool to a specific design method, investigating whether respondent need a system that is strongly tailored to a specific design method, or rather prefer a general purpose CASE tool, which can be personalized by designers according to the method they adopt
- **Multiple view of the same design artifact:** feature that allows deriving multiple “views” of the same design artifact, clustering and restructuring a set of design specifications to select only those relevant for a given set of users, looking at specifications at different levels of abstractions and so forth
- **Consistency Check:** feature that allows detecting the consistency of design specifications with respect to the model constraints, reporting possible violations (e.g., a missing cardinality in a relation, a missing attribute in an information structure, a dangling link, and similar)

- **Semi-automatic derivation of documentation:** feature that allows producing design documentation exploiting the information collected during the model drawing

- **MS-Windows look and feel:** question that aims to discover whether a standard MS Windows-like look and feel is required or rather different interface paradigms for the design tools interface are accepted as well.



Regarding Project Management

b. Which of the following activities concerning project management are considered important to be supported in a design method?

i. Time planning.
ii. Assignment of workers to specific work activities.
iii. Change management.
iv. Client management.
v. Stakeholder management.
vi. Configuration management.

Please provide here specific comments and suggestions about project management support
……
……

Regarding Ease to Learn

a. Which is the time (in months) expected to be spent in order to learn how to use a method?

i. At most 1 week.
ii. 2 to 4 weeks.
iii. More than 4 weeks.

b. Which type of training is preferred in order to learn a method?

iv. On-line courses.
v. Mentoring.
vi. Theory/Practice courses.

Regarding Fast Prototyping

a. Which of the following aspects are considered a motivation for fast prototyping?

i. Requirements validation.
ii. Rapid client satisfaction.
iii. Design validation.

⋮

Figure 2: Sample of detailed questions about individual requirements

4.3 Analysis of survey responses

A total of 82 filled questionnaires were received giving an overall response rate of 44 percent. This level of response rate is twice the rate achieved in similar studies – in [3] it was 10 percent and 22 percent in [13]. To analyze the questionnaire data we have used a software package, specially suited for statistical analysis of survey, named StatPac (www.statpac.com). Depending on the kind of question, the acquired data falls in one of the following categories:

- **Option Rating:** includes all the questions where respondents are asked to rank the importance of various possible characteristics of either method, documentation or tool using a four-level Likert scaleb ("Not Relevant", "Relevant", "Strongly Relevant", "Absolutely Necessary"). Descriptive statistic techniques (mean, median, mode, standard deviation, significance coefficient) have been used to analyze the sample's tendency expressed by such a kind of data
- **Multi-response:** includes all the questions concerning either the desired attributes/features/properties of design methods or respondent's professional experience. The collected data represents the number of respondents who have expressed a preference for one of the proposed options. Percent frequency distribution and confidence interval (calculated by first computing the estimated standard errorc of the proportion, and then using the t distribution to find the actual interval) have been used to analyze the sample's tendency. Findings falling in this category are described adopting the notation "observed frequency" ± "expected variation" within the stated Confidence Interval (usually CI=95%). Thus, referring to the variable X, the value 65±10,3% with CI=95% means that X has been chosen by 65% of our respondents and that in a similar population we could expect a variation of ±10,3% with a confidence interval of 95%

Besides the analysis of each single question, we have also used various cross-variable statistic analysis techniques to evaluate eventual mutual influences among variables. Typically, the cross analysis has involved information representing the respondents' profile and data representing their requirements about method, documentation and tools. The Mann-Whitney test (more suitable respect to the T-Test in case of ordinal data) [1] has been used to uncover eventual significant differences among different groups of practitioners with different professional background or role (exploiting data acquired in the first section) for any variable where a Likert scale has been adopted.

Finally, correlation coefficients have been calculated for variables potentially related with one another, on the basis of observations collected in the focus group sessions, in order to uncover eventual relationships among the respective tendencies. The correlation coefficient has been calculated using the Spearman's rank-difference technique in case of ordinal data and the Pearson's product-moment formula in case of interval or ratio-type data.

In the following paragraphs the result of the questionnaire analysis is reported organized according to the questionnaire structure. Eventual relationships between respondents' background and findings, and between various variable tendencies are pointed out and commented only where these have shown a statistical relevance.

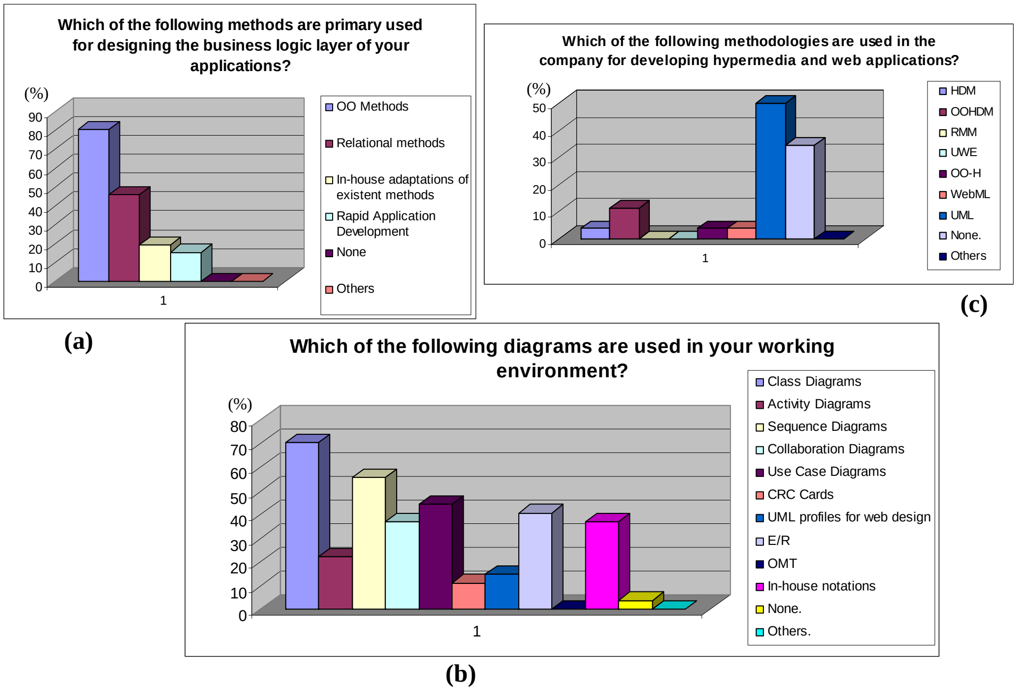
[b] Collected data are ordinal since they have an inherent order and sequence. However, it cannot be assumed that the respondent means that the difference between "Not Relevant" and "Relevant" is the same as between "Absolutely Necessary" and "Strongly Relevant". Adopted statistic technique takes into account such consideration, that is, they are those typically used to analyze data organized according to a Likert scale.

[c] In particular, since the analyzed sample is quite reduced with the respect of the whole population (say less than ten percent), we have not used the finite population correction factor (1-n/N).

4.4 Respondents' profile and current practice

In this paragraph we describe the respondents' profile and current practice on the basis of the data collected in the first section of the questionnaire (*Profile and Current Practice*). This information should provide readers with a clearer picture about the professional background of our respondents as well as an idea of the current practice in the investigated field. This information is also used, by means of cross-tabulation techniques, to assess eventual influences of the respondents' background on their desiderata concerning design methods.

As regard to the respondent role, most of respondents were Project Manager (77% of the total, meaning that respondents have followed our request concerning the preferred profiles) followed by Senior Developers (15%) and Senior Analysts/Designers (8%). Cross-tabulating respondent roles with company size, we can only point out that PM were almost equally distributed across all the three sizes, while most of SD belonged to small companies (which are expected to be less structured) and most of the SA belonged to large ones (where it is easier to find professionals who cover this specific role).



(a) **(b)** **(c)**

Figure 3: Adopted methods for design activities

Concerning design methods, Figure3a shows how *object oriented methods* are the most used (74%) along with *relational methods* (40,7%), to design the business layer of an application. Interestingly, nobody has answered "no design methods" are used, even if some of the respondents declare that *in-house adaptations* (15%) of current methods are used. Focusing on object oriented methods and referring to the UML environment (Figure3b), we notice that that almost all UML "models" are quite popular. *Class diagrams* are the most used (70%) together with *sequence* (55%) and *collaboration* (37%) diagrams. More than one third of the respondents uses *in-house notations* for representing such diagrams, and only *7%* of respondents use the *UML profile* for Web applications design (the WAE – Web Applications Extension proposed in [9]). The generic questions on design methods have been complemented with a question focusing on the specific Web applications domain. Here we asked which design methods are used for designing Web applications, citing the most known academic methodologies like HDM [15], OO-HDM [26], RMM [18], UWE [17], OO-H [16], WebML [8], as well as the generic UML again (explaining that we do not refer to any specific UML Web extension but to UML models in general). Responses to this question, shown in Figure3c, highlight that in practice *no Web specific academic method* is actually used and respondents are about to be split in those who use in some way *UML* (48%) and those who do *not use any methodological support* (44%) for developing Web applications. Comparing this finding with the results of Figure 3b, we might argue that some of respondents who claim to use UML but not UML profile might use in-house customized UML when addressing Web application design.

Finally, concerning the adoption of design processes, findings of Figure 4 tell us that the *traditional cascade lifecycle* is still the most used (40,7%), while *RUP*, *XP* but also *none* and *others* have received almost the same score (20%).

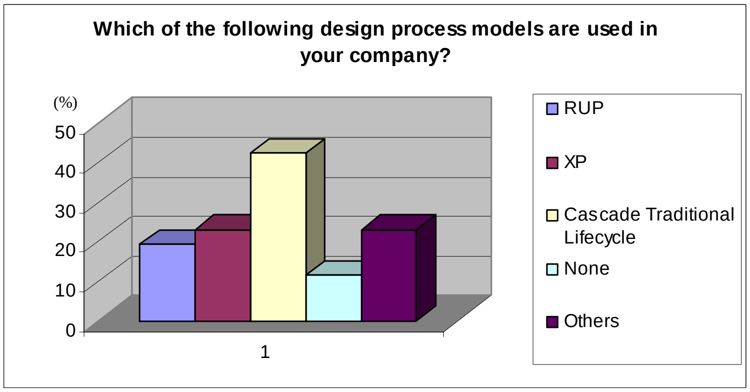


Figure 4: Adopted design processes

4.5 General Requirements for a Design Method

Table 1 summarizes the information acquired in Section 2 of the questionnaire shown in figure 1. Looking to the statistical analysis of the acquired data, a number of considerations about the population requirements can be drawn. *Easy to Use* has the highest mean and smallest variance and is considered *strongly relevant* or *absolutely necessary by 80%* of the respondents, resulting definitively the most desired requirement for a design method. *Easy to Learn* is also considered quite important (42,3% *relevant* and 37,5% *strongly relevant*) but mean and variance are similar to other requirements. Being *Ease to Learn* and *ease to use* typical factors contributing to perceived usability, we argue that one of the lacks of current method is being usable. Preferred learning activities and time have been also investigated and respondents' opinions are reported in the following of this section. *Support for Iterative and Incremental Design Lifecycle* is considered the second most important trait of a design method showing as respondents look for methods which can support them along the overall development process. Concerning *Case Tools* and *Documentation,* respondents consider both of them desirable but their means and variances show that respondents do not consider them being fundamental.

Matching the statistics on the acquired data with the information collected in the focus group sessions, we have drawn some additional considerations on the observed phenomena.

	Mean & SD	Not Relevant at all	Relevant	Strongly Relevant	Absolutely Necessary
Project Management Support	2,77 0,85	3,8%	38,5%	34,6%	23,1%
Process Customization	2,80 0,94	4,0%	44,0%	20,0%	32,0%
Ease to Learn	2,65 0,78	3,8%	42,3%	38,5%	15,4%
Ease to Use	3,22 0,63	0,0%	20,1%	48,1%	32,8%
Fast Prototyping	2,84 0,67	0,0%	32,0%	52,0%	16,0%
CASE Tools Support	2,38 1,00	23,1%	30,8%	30,8%	15,4%
Support for Iterative and Incremental Design Lifecycle	3,19 0,88	3,8%	19,2%	30,8%	46,2%
Documentation Support	2,92 0,73	0,0%	30,8%	46,2%	23,1%
Be a Standard	1,50 0,80	65,4%	23,1%	7,7%	3,8%

Table 1: General requirements for a design method (see also figure 1): Summary results

Case Tools Support presents the highest variance. A possible explanation for this finding can be found in the considerations reported in “We speak different languages... but we need to cooperate” paragraph in the paper section 3. During the analysis activities, practitioners desire a design method enabling them supporting early reasoning about the system to be by means of un-formal, lightweight, partial, fast to draw and non-technical models. In this perspective, practitioners may consider “paper and pencil” (expression used by some of our interviewees) the most suitable design tool, thus a software tool (as defined in our questionnaire) is not needed. Moving towards implementation, models usually become more formal and complete so that large design documents are usually produced to feed the implementation activities. From this perspective, a support tool may be considered a necessary complement of a design method. However, neither the respondent roles not their background appear to influence their opinion, since no statistical relevance has been found applying the Mann-Whitney U Test (two-tailed probability above 0,6)

Process Customization and *Fast Prototyping* are both considered desirable features but respondents' opinions are less uniform with the respect of the other ones. Finally, concerning negative answers, interesting is to point out that the investigated segment of practitioners does not consider *Being a Standard* an important trait of design method as it is clear by the fact that 64,5% of respondents consider it *Not relevant at all* and by its overall mean and variance.

Documentation Support is considered a *strongly relevant* complement by most of the practitioners but in the focus groups we noticed uncertainness about its actual purpose. In particular, respondents are divided among those who would use it to learn the method and those who consider it as a reference document to be used run-time once the method's basic elements have been learnt. Yet, most of them would prefer to avoid referring to the documentation in order to use the method revealing it as a component that is needed but not necessary.

An additional set of useful indications emerges from the analysis of the answers to detailed questions (see sample in figure 2) concerning each specific characteristic. No significant relationships have been found in general between these results and the respondent's role and background as shown by the Mann-Whitney U Test that presents a two-tailed probability usually above 0,8.

Project management support: among the project management activities considered *important to be supported by a design method*, respondents answer, on an average of three preferences per case, that *Time Planning* (73,1% ±9,6% with CI=95%), *Change Management (76.9%* ±9% with CI=95%*)* and *Configuration Management* (76.9% ±9% with CI=95%) are the most relevant.

Easiness of learning: the gathered responses show that the *69,2%* (±10% with CI=95%) expects to spend *between two and four weeks* (on the whole) learning a design method, *19,2%* prefers spending *at most one week*, and *11,5%* can afford spending *more than four weeks*. These data confirm our hypothesis that respondents expect that a considerable amount of time may be needed to learn a method and they are conscious that they need to invest such a time. Concerning the desired type of training, most of respondents (*50*±10% with CI=95%) desire courses with *theory and practical* components provided as either *mentoring courses* (*46,2±10,8%* with CI=95%), that is, courses involving side-by-side expert support or *online Web courses* (*33,33±10,2%* with CI=95%).

Easiness of use: among the characteristics that make a design method easy to use, most of respondents consider of primary importance the *availability of guidelines and patterns* (*70±9,2%* with CI=95%). *Flexibility* is also considered a high priority for a design method (*65±10,3%* with CI=95%) reflecting the fact that, in the Web field, practitioners generally face projects with very different characteristics.

Fast Prototyping: almost all the respondents consider fast prototyping as a powerful means for *validating requirements* (*88,5±6,9%* with CI=95%) and half of them (*51±10,8%* with CI=95%) considers it useful for *design validation*, while a minor number of respondents believe it can be used to gain a *rapid client satisfaction* (*40±12%* with CI=95%). Respondents were also asked *"After each iteration, which type of "delivery" is considered more effective"* between *Evolutionary* and *Throw-away prototypes.* The former results the preferred type of delivery (*81,6±8,4%* with CI=95%) pointing out that practitioners look at effectiveness mostly from the costs point of view. Cross-tabulating this outcome with the previous one it comes out that all (*100%*) the respondents who consider more effective a throw-away prototype vote *“validating requirements”* as the unique motivation for fast prototyping. Moreover, this segment is also the majority (*78,6±8,9%* with CI=95%) of those who consider *“validating requirements”* as the sole motivation. On the basis of these results, we can argue that for the most part of our interviewees a throw-away prototype is considered effective when it is used to support the requirements activities, while an evolutionary one is preferred when moving towards implementation.

4.6 Requirements for Design Documentation

Table 2 summarizes the main findings concerning documentation support.

	Mean & SD	Not Relevant at all	Relevant	Strongly Relevant	Absolutely Necessary
HandBook	2,54 0,93	11,5%	42,3%	26,9%	19,2%
User Guide - Manual	2,85 0,91	7,7%	26,9%	38,5%	26,9%
Cookbook	2,27 0,98	23,1%	42,3%	19,2%	15,4%
Book	2,76 1,07	16,0%	24,0%	28,0%	32,0%
On-line Hypermedia	2,88 0,97	11,5%	19,2%	38,5%	30,8%

Table 2: Summary results on desired documentation types.

The results highlight that *online hypermedia* and *User Guide* are the preferred forms of documentation, presenting the highest means and being marked as *absolutely necessary or strongly desired* by two third of our practitioners. In contrast, *cookbook* has received the highest percentage of *not-desired* marks and lower values for *strongly desired* and *absolutely necessary* than the other forms of documentation. *Book* presents the highest variance although *absolutely necessary* has collected more votes than the other ranks. We do not have a clear explanation for these statistics, we can only argue that it may depend on the personal experience of the respondent with using book for learning/using design methods.

Regarding the expected sizes, in terms of amount of pages for each proposed documentation type, responders’ preferences are:

Handbook: 5-10 pages 19%; 10-20 pages 70%; more than 20 pages 11%.

User Guide Manual: 40-50 pages 33%; 50-80 pages 37%; more than 80 30%

Cook Book: 10-20 pages 35%; 20-40 pages 42%; more than 40 pages 42%

Book: 70-100 pages 33%; more than 100 pages 67%

On-line Hypermedia: 20-40 pages 59%; more than 40 pages 41%

A further consideration can be done by cross-tabulating these data with the previous documentation preferences. Concerning User Guide/Manual, respondents who answer they do not desire this kind of documentation generally expect its size being more than 80 pages pointing out that huge manuals may (not surprisingly) scare practitioners (correlation 0.5, p<0.01). On the other hand, no significant correlation (<0.2) emerges between the expected size of *book, cookbook* and *handbook* and respondents' opinions about their usefulness.

4.7 Requirements for Support Tools

Support tools are considered important complements for a method as shown by the results acquired in section 2 where only 23% of respondents consider it *not relevant*. However, the focus group sessions have clearly pointed out that different practitioners have different expectations about tools features and characteristics. Section 4 investigated what features and aspects of a CASE tool are considered more important by our respondents. Table 3 summarizes the main findings regarding this investigation.

	Mean & SD	Not Relevant at all	Relevant	Strongly Relevant	Absolutely Necessary
Flexible models management	2,65 0,73	3,8%	38,5%	46,2%	11,5%
Code derivation	2,54 0,89	19,2%	15,4%	57,7%	7,7%
Model versioning	3,16 0,73	4,0%	8,0%	56,0%	32,0%
Semiautomatic generation of prototypes	2,23 0,80	19,2%	42,3%	34,6%	3,8%
Integration with methodology design activities	2,81 0,68	0,0%	34,6%	50,0%	15,4%
Multiple view of the same design artifact	2,36 1,02	20,0%	44,0%	16,0%	20,0%
MS-Windows look and feel	1,96 0,94	42,3%	23,1%	30,8%	3,8%
Semi-automatic derivation of documentation	2,77 0,93	11,5%	23,1%	42,3%	23,1%
Consistency check	3,31 0,91	3,8%	19,2%	19,2%	57,7%

Table 3: Requirements for a design tool

From the gathered responses, the most desired features for a tool are definitively *Consistence check* and *Model Versioning*. It is clear that users considered them important tasks in the design activities, and, drawing on the discussion achieved in the focus groups, it is clear that practitioners highly expect these tasks being almost completely carried by a software tool. Practitioners consider them both tedious and highly sensitive and a software tool is viewed as more reliable for executing them. Looking to *not relevant aspects*, it can be noticed as *MS-Windows Look And Feel* receives the highest percent for the *Not relevant at all* rank and the lowest one for the *Absolutely necessary* rank. *Semi-automatic generation of documentation* and *prototype* have collected variable opinions likely for the same considerations reported in the paper section 4.5 and concerning the use of design methods with respect to the analysis and implementation phases. *Multiple views of artifacts* is the feature with highest standard deviation probably because it can be considered the most sophisticated among the investigated set so that some respondents might have not been familiar with it.

Finally, cross-tabulating respondents' opinions about *motivations for fast prototyping* with their ranks concerning these two features some further considerations can be argued. Most of respondents (*71.3±9.2%* with CI=95%) who rank *Semi-automatic derivation of prototype* between 1 and 2, in the equivalent Likert scale, would like to use the prototype for requirements validation, while a significant percent (68*.1±9.4%* with CI=95%) of those who rank it between 2 and 4 consider *design validation* the most important motivation for fast prototyping. Moreover, in both cases the Mann-Whitney U Test presents a two-tailed probability p<0.1 strengthening the evidence of a relationship between respondents' orientation concerning design methods (and their tools) and prototyping. Combining these results with the related ones discussed in paragraph 3.5, it can be argued that respondents are divided between those who consider design methods mostly as a reasoning tool to be used in the early stage of the development process, and those who expect to produce models that can be used throughout the overall development process up to the implementation activities where models should be transformed in software skeleton (evolutionary prototype).

5 Lessons learned

Findings reported in sections 3 and 4 mostly represent the point of view of the practitioners involved in the field. We have tried to keep the analysis as much objective as possible although some of the provided interpretations may have been influenced by our experience and background. On the other hand, in this section we summarize our reflections on the results achieved in our study expressing them in terms of lessons learnt.

Lesson 1: A holistic view for design methods is needed

The key lesson emerged from our study is that in order to be accepted and used in an industrial environment, a design method per se is not enough. We should adopt a "holistic" view of design methods, looking at them within the *organizational context* in which they have to work, and in the context of the *overall development cycle*. In this perspective, a design method should be integrated with features that support the various activities directly or indirectly related to design. Thus the ideal companions of a design method are an accurate *documentation,* a *training* strategy, and a set of *tools*. Tools should provided the needed functionalities to support authoring of design specifications, mapping from design to *prototyping and implementation, design documentation delivery*, and *project management.*

Lesson 2: A design method should be easy to learn

One of the most important characteristics that industry people have identified for a design method is *learnability* - a fundamental component for the non-functional-requirement usability. Professionals do not have time and resources to invest for learning new cumbersome methods. Our findings suggest that practitioners require learning and starting up to use a method in no more than 4 weeks. One of the success factors of the "Entity Relationship" method (probably the most successful conceptual design method ever) stems from the fact that it is very easy to transmit its basic concepts. Our study also suggests that theory and practical courses combined with mentoring courses (better if including hands-on activities carried on side by side with an expert) are the preferred training activities.

Lesson 3: A design method should compromise between richness and simplicity

To improve learnability (and, more generally, *usability*) design methods should find a compromise between richness and simplicity, trying to balance completeness and expressive power of their modelling primitives with intuitiveness, and supporting some evidence of their utility for representing different features of real applications. A possible way to achieve this compromise could be to deliver *multi-version* design methods, made of a "basic kit" and an "advanced kit". The basic kit should provide a set of basic, easily understandable modelling primitives and a "reference" design process loose enough to be easily integrated in different contexts. It should be learned relatively easily (2-5 days) and almost immediately applied for designing relatively simple applications. The advanced version should address more sophisticated design needs and offer more sophisticated tools for customizing the model and the process (see also Lesson 5). The advanced version should be learnt after the basic features are fully digested, and used to design complex application features. The survey results also show that it is unlikely that practitioners fully adopt a method as it is (see paragraph 4.4). In this light, the "basic kit" should address the key aspects of Web design and convince practitioners of the potential advantages of adopting the method even in case they would use a customized version of it.

Lesson 4: A design method should be "multi-lingual"

Moving from the early design phases to design for implementation, different design problems are faced and different professional profiles are involved. Accordingly, the purpose of a design method varies progressively from a user oriented reasoning tool to a system oriented specification language. A design method aiming to address multiple phases should therefore offer a variety of languages – different concepts, notations and guidelines – tailored to address different targets and different design issues. Each language should allow designers with different backgrounds to "speak their own language", that is, the one that better fits with the way they currently (in the context where the method should be introduced) represent and talk about design solutions. On the other hand, a design model should enhance the communication among members of the multidisciplinary design team throughout the various development phases. In this respect, guidelines on how to pass from a "version" of the design to another one should be provided. Moreover, tools should embody functionalities to support such a translation and to keep the different models coherent with one another.

Lesson 5: A design method should be modular and scalable

Design methods must face the so called "organizational inertia [20], i.e., the tendency of organizations to stick to tried-and-true methods. If a design method requires to be adopted "all or nothing", it may impose drastic changes in the proven practice of a company and it may seriously compromise its chances of adoption. In contrast, if a design method can demonstrate its advantages even if partially employed, it may have better chances of acceptability. A smooth, progressive introduction of a method can convince practitioners of the value of the overall method and create appetite for a complete adoption. For these reasons, a design method should be *scalable* and *modular*. It should include different autonomous components (see also lesson 3) that can be adopted individually (i.e., independently from the others) and easily mapped to the current development practice of an organization.

Lesson 6: A design method should be flexible and customizable

No matter if a design method is fully or only partially adopted (see lesson 4), designers should be allowed to use it in multiple possible ways. A design method should offer means to customize its modelling primitives and to personalize the design process, according to the current style of work, the design concepts already in use in an organization, or the specific characteristics of the application being developed. The need of easily customizable methods has been also confirmed by the survey whereby the fact that most of practitioners use ad-hoc methods or adapt existing languages – e.g. UML – to address their needs. We believe that, referring to lesson 3, practitioners may only adopt a core set of concepts and ideas integrating them with their current practice and customizing them on the basis of their company or even project specific needs. Methods should thus be enough flexible to accommodate this custom.

Lesson 7: A design method should provide "patterns"

By offering pre-packed modelling solutions or activities skeletons for recurrent design situations, patters help designers to understand how to solve specific design problems, how to use the design concepts of a method, or how to carry on a design task, thus making the design activity easier and more effective (especially for newcomers). In addition, a design method should include some guidelines on how to map design solutions at a given level to design specifications at a lower level of abstraction while moving from analysis towards implementation. The "rules" of the E-R model that describe how to map E-R diagrams into relational tables are a good example where this principle is applied. Tools should support such mapping as much as possible relieving designers from this tedious task and ensuring consistence among the different levels.

Lesson 8: A design method should be complemented by high quality documentation

Accurate documentation is crucial for making a design method easy to learn (see lesson 2) and easy to use in an industrial setting. A recurrent deterrent for adopting a new methodology is the lack of business-oriented documentation being most of academic methods documented only through scientific papers. Our study points out that the preferred supports are well structured online documentation (probably because it can be easily accessed), followed by agile user guides and manuals, and that richness of examples, case studies, and lesson learned are perceived as an important content in all forms of documentations.

Lesson 9: A design method should be complemented by various kinds of support tools

An industrial project is unlikely to adopt a method that lacks of appropriate support tools. Our study highlights that industry demands *four* main categories of functionality and have some specific requirements for each of them. *Authoring tools* are needed to support not only design specification but also brainstorming and reasoning activities. Some desired features of an authoring tool are: support for multiple views of the specification schemas (e.g., at different levels of details, across different design perspectives), versioning, support for switching among different design schemas, consistency check. *Specification derivation tools* are needed to produce high quality professional documentation (which usually requires more information than just a set of diagrams or formal specifications) by providing specific facilities like configurable documentation templates. *Code derivation tools* allow the production of code skeletons and portions of the final code, speeding up the implementation process. *Fast prototyping* tools are needed to create (automatically or semi-automatically) “quick&dirty” or evolutionary prototypes from a set of design specifications. In addition, prototyping tools should support traceability, helping to take under control the impact of requirements changes into design solutions.

Lesson 10: What kind of prototype better suits practitioners depends on what development phase it has to be used in

Derivation of prototypes from design models is a highly desired requirement. Tangible, interactive, visual artefacts create a “look&feel” effect which can be impressive for customers, and useful for discussing requirements and preliminary design decisions with non technical stakeholders. In this phase, when the change rate is still high, practitioners require to derive prototypes from design specification quickly and with little effort. To this end, throw-away prototype are considered more suitable since they do not require a rich and complete design being defined, do not need technology decisions being taken, and requires less effort to be produced. Moving towards implementation, design models are naturally enriched with details, implementation decision are taken and the chance rate lowers. In this light, having prototypes that can evolve towards implementation skeletons is considered more appealing even if it may require some extra effort.

6 Conclusions and Final Remarks

In this paper we have discussed the results of an empirical study aimed at gaining a deeper understanding of the factors, obstacles and incentives that prevent, or contribute to, the acceptability and adoption of design methods by the industrial world. As discussed in several field studies and in various scientific events, the problem of overcoming the hurdle of acceptability is a priority that the Web engineering community still needs to tackle. This issue is even more important considering that, from the technical point of view, most of the existing methods have achieved high level results and have demonstrated to embody the needed functionalities to build high quality Web systems. A thorough analysis of non-functional requirements of design methods per-se including usability, modularity, scalability, and customizability, and of support documentation and tools may help to uncover the issues that are hampering current methods in overcoming the hurdle of acceptability by the practitioner community. Although we do not pretend to consider the achieved results an exhaustive analysis of all the possible non-functional requirements, they provide a significant collection of information and data that scientists or firms, involved in the definition of design methods, can use to get their works closer to the practitioners' expectations. They provide guidelines on how to improve the quality, as perceived by the practitioners, of design methods and related support documentation and tools and ultimately may contribute to improve the chances of achieving a successful "technology transfer" for design methods.

Some of the results of our study may be generalized to refer the broader field of software development methods in general. In this attempt the reader should consider that in our study we have targeted practitioners involved in the development of Web systems thus the results reflect the inclination of this specific sector. The specific characteristics of a Web system development process [27,28] may have influenced the respondents' opinions. However, we believe that, having our analysis focused on "non functional" aspects of design methods, most of the factors lowering the acceptability chances for academic design methods in general can be related to the reasons we discuss in the paper.

Finally, from the several discussions we had with practitioners and from some data gathered from the questionnaire, a lack of motivation for using design methods specific for Web applications arose. A reflection about this aspect led us to a further consideration the Web engineering community should probably take into account for defining the future research and dissemination strategies. The lack of awareness about web development methods surfaced in the study may be explained by the fact that most of the practitioners are not aware that developing a web application is different from developing other software products. This lack leads them to the adoption of existing development methods ad-hoc adapted to cope with the Web system's peculiarities. Thus practitioners do not see the advantage of learning and using Web specific design methods versus using a generic method they already know. We may conclude that, for achieving a larger acceptability of its methodological results, the web engineering community should not only try to improve the quality of their methods along the directions we have identified, but may also need to communicate more effectively that developing web applications has some peculiarities which make Web specific methods potentially more effective than adaptation of generic methods.

Acknowledgments

The authors are grateful to Mauricio Sansano, who carried on the preliminary version of the survey, the partners of projects UWA, V-NET5, OPENDRAMA, MODA-ML, MEDINA, DICE and VICE for the time they spent with us in the focus group sessions, and all the practitioners who participated to the survey.

References

1. Blalock, H., Social Statistics (2nd Edition), McGraw-Hill, New York, 1972
2. Baresi, L., Garzotto, F., Paolini, P., Perrone V., Hypermedia and Operation Design". UWA project Deliverable D7, 2001, available at www.uwa-project.org or at www.elet.polimi.it (Perrone's web page)
3. Barry, C., Lang, M., A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE MultiMedia, v.8 n.2, 52-60, 2001.
4. Booch, G., Jacobson, I., Rumbaugh, J., The Unified Modeling Language User Guide. The Addison-Wesley Object Technology Series, 1998.

5. Britton, C., Jones, S., Myers, M. and Sharif M., A Survey of Current Practice in the Development of Multimedia Systems. Information and Software Technology, vol. 39, no. 10, pp. 695-705, 1997.
6. Brown G., Low Response Rate and their Effects on Survey Results. In proceedings of the 28th National Conference of the Market Research Society, Adelaide, Australia, 3-6 October, 1999.
7. Carrol, John M., Scenario-Based Design; New York, John Wiley & Sons, 1995.
8. Ceri, S., Fraternali, P., Bongio, A., Web Modeling Language (WebML): a modeling language for designing Web sites. In proc. of the 9th World Wide Web Conference (WWW9), Amsterdam, 2000
9. Conallen, J., Modeling Web Application Architectures with UML. Communications of the ACM, 42:10, pp. 63-70, 1999.
10. De Troyer, O., Leune, C., WSDM: A User-Centered Design Method for Web Sites. In proceedings of the 7th International World Wide Web Conference, Elsevier, 1998.
11. Eisenhardt, K., Building theory from case study research, Academy of Management Review, 14, 4, 532-550, 1989.
12. Engels, G., Groenewegen, L., Object-oriented modelling: a roadmap. In A. Finkelstein, editor, "The Future of Software Engineering", Special Volume published in conjunction with ICSE, 2000.
13. Fitzgerald, B., An Empirical Investigation into the Adoption of Systems Development Methodologies. Information & Management, Vol. 34, pp. 317-328, 1998.
14. Gamma, E., Johnson, R., Helm, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, 1995.
15. Garzotto, F., Paolini, P., Schwabe D., HDM - A Model-Based Approach to Hypertext Application Design. ACM Transactions on Information Systems, Vol. 11, No. 1, pp 1-26, 1995.
16. Gómez, J., Cachero, C., Pastor, O., On Conceptual Modeling of Device-Independent Web Applications: Towards a Web Engineering Approach. IEEE Multimedia 8(2), 2001.
17. Hennicker, R., Koch, N., A UML-based Methodology for Hypermedia Design. Lecture Notes in Computer Science, Springer Verlag, Volume 1939, 2000.
18. Isakowitz T, Stohr EA., Balasubramanian P., RMM: A design Methodology for Structured Hypermedia Design. Communications of the ACM Vol.38 No8 August pp 34-44, 1995.
19. Jacobson, I., Booch, G., Rumbaugh, J., The unified software development process. Addison-Wesley, Reading, MA, 1999.
20. Kaindl, H. et al., Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda. Requirements Engineering journal, 7(3), pp 113-123, 2002.
21. Kraemer, K., Dutton, W., Survey research on information systems, in Kraemer, K. (ed) The Information Systems Research Challenge: Survey Research Methods, Publishing Division,Harvard Business School, Boston, 3-57, 1991.
22. Kuniavsky, M., Observing the user experience: A Practitioner's Guide to User Research. Morgan Kaufmann Series in Interactive Technologies, 2003.
23. Lowe, D., & Webby, R., Utilisation of Process Modeling in Improving the Hypermedia Development Process. The New Review of Hypermedia and Multimedia, vol. 5, 1999.
24. Mason, R., Commentary on survey research on information systems, in Kraemer, K. (ed) The Information Systems Research Challenge: Survey Research Methods, Publishing Division, Harvard Business School, Boston, 59-65, 1991.
25. Perrone, V., Conceptual Modelling for Multi-Channel Web Applications. PhD thesis, Department of Electronic and Information, Politecnico di Milano, Italy (www.elet.polimi.it), 2004.
26. Schwabe, D., Rossi, G., An Object Oriented Approach to Web-Based Application Design. Theory and Practice of Object Systems, 4 (4), J. Wiley, 1998.

27. Henderson-Sellers, B., Lowe, D., Haire, B., OPEN Process Support for Web Development. Annals of Software Engineering, 13, 163-202, 2002.
28. Lowe, D., Henderson-Sellers, B., Web Development: Addressing Process Differences. Cutter IT Journal, 14(7), 11-17, 2001.