Title	Web-based Systems Development: An Empirically-Grounded Conceptual Framework
Author(s)	Lang, Michael
Publication Date	2007
Publication Information	Lang, M. (2007) Web-based Systems Development: An Empirically-Grounded Conceptual Framework. In Wrycza, S. (ed.), Proceedings of 2nd AIS SIGSAND European Symposium on Systems Analysis & Design: Education and Practice, Gdansk, Poland, June 5, 2007, pp. 80-91.
Item record	http://hdl.handle.net/10379/475

Web-based Systems Development: An Empirically-Grounded Conceptual Framework

Michael Lang

Business Information Systems Group

Cairnes School of Business & Public Policy

National University of Ireland, Galway

Michael.Lang@nuigalway.ie

ABSTRACT

This paper encapsulates the main findings of an in-depth study of Web development practices in Ireland. The essential research objective was to build a richer understanding of the modern context of Web development and of how that context influences design practices. At the outset, a conceptual framework was derived through a synthesis of issues in the literature and an analysis of existing models of IS development. Data was then gathered through a dual-mode (Web and postal) quantitative survey which yielded 165 usable responses, and later through a series of 14 semi-structured qualitative interviews in a follow-up field study. Following an interpretive approach, elementary statistics and grounded theory were used to iteratively analyse the data until a reasonably comprehensive and stable explanation emerged. This is presented in the form of an elaborated conceptual framework of Web-based systems development as "situated action".

Keywords

Web-based systems development; situated action; empirical studies of systems analysis & design; systems development methods and techniques

INTRODUCTION

The latter years of the 1990s saw a frenetic surge in activity on the World Wide Web, driven by improvements in networking and communications technologies, enhanced browser capabilities, more advanced server-side and client-side functionality, increased sophistication of visual user interfaces, and the rise of electronic commerce. This sudden and spectacular growth caused quite a degree of apprehension amongst the academic research community because the apparently "out of control" Internet technological upheaval was progressing at such a chaotic pace that the state-of-theory was left lagging some distance behind the state-of-practice [1]. Whereas the Web a few short years previously was predominantly a publishing medium, it was metamorphosing so quickly into an applications development environment that serious doubts hung over the readiness of the incumbent generation of Web designers, many of whom were self-trained and from backgrounds other than "proper" software engineering.

On such a premise, Murugesan & Deshpande [2] called for a "new concept and discipline of Web Engineering" and affirmed that there was a "pressing need for new methods and tools" [3]. In similar vein, Oinas-Kukkonen et al [4] claimed that "systematic analysis and design methodologies for developing Web information systems are necessary and urgently needed among practitioners". Speculation was rife of an imminent "Web crisis" on foot of a prevalent view that industry development practices in general were unsystematic and unreliable. Whether these remarks were well-founded or mere "exception reporting" [5] is arguable, for the software industry has supposedly been chronically afflicted by a "crisis" as long as it has existed [6,7].

This research project was initiated at a point (c. 2001) when there was much sensational talk in the academic literature of an imminent "Web crisis". Quite a number of empiricial studies of Web development, mostly of the nature of descriptive surveys or narrow experience reports, were published about that time. Though useful and interesting, those studies are now a little dated. Setting aside general HCI research on the effectiveness/usability of Web sites and the mainly experimental contributions of the Web Engineering community, remarkably few studies of actual industry *practice* have since appeared. Following the post-Y2K implosion of the "dot.com" bubble, the Web design industry went through an upheaval whereby firms engaging in haphazard practices were forced to either reform (if they were capable of so doing) or perish (as very many of them did). Development technologies have advanced remarkably in recent years, and many Web development firms originally established in the mid- to late-1990s have at this stage settled down and attained process maturity. The objective of this research project was therefore to contribute towards a richer and updated understanding of the "real-world" context of Web-based systems development, and of how that context influences design practices. The empirical phases of this study were conducted between Winter 2002 and Summer 2006. Specifically, the research questions were as follows:

RQ 1.   What is the profile of a typical Web-based systems development project?

RQ 2.   What are the main challenges being experienced by Web-based systems designers in practice?

RQ 3.   What development practices are being engaged to address these challenges?

RQ 4.   What situational factors influence the enactment of development practices?

RQ 5.   Where formalised design guidance is in place, what is its nature and from where is it derived?

RESEARCH APPROACH

A three-phase research approach was taken, as shown in Figure 1. At the outset, a number of informal meetings were held with a few experienced Web developers to help solidify the research objectives, assess the salience and relevance of certain aspects raised by the literature, and uncover any major topical issues of which the researcher was unaware.

The second phase consisted of a dual-mode (postal and Web-based) survey of 438 organisations. The sampling frame included organisations engaged in bespoke software application development; those specialising in Web or interactive multimedia systems design; companies from traditional media that had branched into "new media"; and large organisations with internal IT departments. The survey received an overall response rate of 52%, ultimately yielding 165 usable responses.
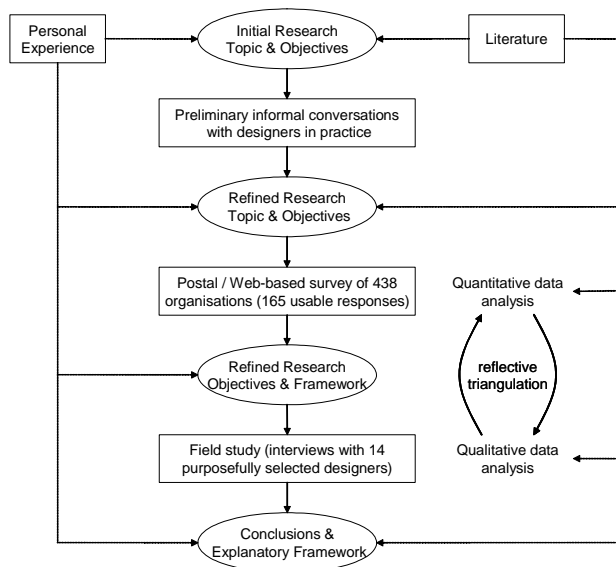

Figure 1. Overview of research process

The third and final phase was a follow-up field study, consisting of semi-structured qualitative interviews with 14 Web developers. The selection of interviewees was theoretically driven, chosen so as to seek out similarities and dissimilarities, looking at both typical and atypical cases. They varied according to organisational size, organisational type, application domains, client location (in-house versus external Web development houses), and the interviewee's professional background. Many of the interviewees had recently won or been nominated for awards at prestigious national ceremonies. It was assumed that award winners would be more forthcoming, knowledgeable and insightful, and also that they exemplify best practice. In most of the organisations visited, one personal interview was conducted with the team leader, typically convened during the mid-day break so as not to encroach upon busy work schedules. In one organisation two developers were separately interviewed, and in another the managing director brought five staff members into the meeting room. Where available,

secondary data sources were also consulted. Data gathering continued until a point of reasonable "theoretical saturation" was reached.

The survey data was mainly analysed using descriptive and enumerative statistics, such as frequency distributions, averages/medians, and cross-tabulations. Because an interpretive approach was taken in this research project, no formal hypotheses were set out. Instead, some theoretical propositions based on posited relationships in the conceptual framework were explored by means of simple difference/correlation tests. The qualitative data gathered in the field study was analysed using a hybrid method, mainly based on the procedures of grounded theory [8-10], but also informed by the principles laid down by Miles & Huberman [11]. Although data gathering for the survey and field study phases was done in chronological sequence, data analysis was an iterative and parallel activity, involving both inductive and deductive reasoning in a grounded, reflective process. Through this triangulation of methods and data, the inherent weaknesses of individual methods are reduced, strengthening the validity and reliability of findings.

Limitations of the Survey

The survey element of this research project is limited by a number of shortcomings, some of which relate to the inherent weaknesses of questionnaires and are compensated for by the field study:

- The survey questionnaire comprised mostly fixed-format questions that captured quantitative data, and responses to the few open-ended questions were scant. For this reason, a qualitative follow-up field study was conducted to elucidate upon the survey findings.

- As is generally the case with survey research, there remains the possibility that findings may be skewed because of reliability and validity issues. Numerous measures to counteract and alleviate potential bias were taken, but it is very difficult to fully eradicate the possibility of contamination.

- The survey was conducted in a small geographical region (the island of Ireland), so caution must be exercised in generalising findings to wider international populations. To test for regional bias, the survey could be replicated in another area and it would be interesting to conduct a cross-national comparison of Web development practices. The option of so doing was originally envisaged at the outset of this project, but to rigorously and successfully perform such a study would involve considerable procedural and methodological challenges [12], necessitating collaboration between a distributed team of international partners. For that reason, it was decided not to pursue this option for now, but to defer it as a future possibility. Indeed, parts of the questionnaire used in this study have since been replicated in surveys of Web development

practices in Korea and in Croatia. It is hoped at a future date to compare findings.

Limitations of the Field Study

While the combination of a quantitative survey in conjunction with a qualitative field study helps to redress some of the shortcomings of either used in isolation, there remains a number of intractable problems with the field study which mainly have to do with the shortcomings of interpretivism, grounded theory, and qualitative interviews. In brief, these are:

- Interviews can be intrusive and atypical; by his very presence as a "foreigner" in an organisational setting, a researcher may introduce bias into that setting. Though interviewees were generally relaxed, forthcoming, and willing to be recorded by means of a digital audio device, there remains the possibility that some unnatural behaviour was caused by the intrusion of the interviewer.

- With data gathered from field studies, only "analytical" generalisability is possible. Logical inferences can be drawn, but statistical inferences can not. This shortcoming was redressed by the combination of a field study and a large-scale survey in this research.

- Qualitative data is prone to subjective and conflicting interpretations. Because this field study was based on interviews personally conducted by the author, he has the advantage above all others of having a first-hand "feel" for the data and is therefore best placed to draw conclusions. That said, the opinions of a number of academic colleagues and peers were sought in order to assess the plausibility of interpretations. The author's knowledge of relevant background literature was also an important point of reference in the interpretive process, as was his own professional experience in the area.

- Because the resultant explanatory framework is a deliberate simplification and is grounded in a limited number of observations, it cannot be expected to account for all possible variations that might be encountered. Of course, no explanation can ever be said to be complete so it is necessary to decide when to stop. In the logic of grounded theory this happens at "theoretical saturation", the point of diminishing returns beyond which analysis is necessarily delimited. At this juncture there typically remain data fragments which have not been fully exhausted, but the conceptual model is bounded and deemed to be "good enough" because, while accepted as being incomplete, it accounts for most of the observed variations in the recorded data incidents [10].

- Again, because the field study was based on a restricted sample of interviewees, it is limited to the extent that this sample is broadly representative of the general population. The interviewees in this study were purposefully selected in order that comparisons and differences might be drawn between cases, but it should be noted that they were mainly award-winning companies. As such, they may be argued to be unrepresentative of industry as a whole, but it was decided that it would be preferable to capture a description of best practices (i.e. award winners) rather than general practices.

- For reasons of limited access, just one person was interviewed in most of the organisations visited. For issues where the unit of analysis is the organisation (e.g. the use of processes and procedures) as opposed to the individual (e.g. the influence of one's professional background), the reliance on a single organisational spokesperson is clearly not ideal for it can be prone to rather personal and biased interpretations. It might have been better, for the sake of reliability, to have spoken with a number of persons within each organisation, in different roles and at different levels of the organisational hierarchy. The unfortunate reality is that with small businesses, such access is often difficult to negotiate, particularly in the industry of Web development where pressing deadlines and multiple concurrent projects are the norm. Furthermore, the participants in this field study were distributed geographically across Ireland which placed a constraint on the feasibility of multiple return site visits. As it turned out, there were indeed possible reliability issues with some of the interview data because: (i) at times, the received impression was that the interviewee was self-convinced that initiatives they pushed for are "working"; (ii) some interviewees were a little opinionated; (iii) in a few cases, it seemed that the interviewee was trying to impress the interviewer, either endeavouring to provide the "correct" answer or even veering towards a "sales pitch". Ultimately, it was necessary to use a degree of personal judgement to separate credible statements from the ones which seemed likely to be exaggerated. Where possible, interview transcripts were cross-checked against survey data and other secondary data to look for anomalies which cast doubts over reliability. A few discrepancies were found between survey and interview responses, but these were all readily explained by changes in organisational practices that had been implemented in the interim period between the execution of the survey and the conduct of interviews. In spite of the shortcoming of having interviewed just one person in most of the organisations visited, the researcher is of the opinion that interviewees for the greater part were frank, forthright, and representative of the general views that pertain within their organisations.
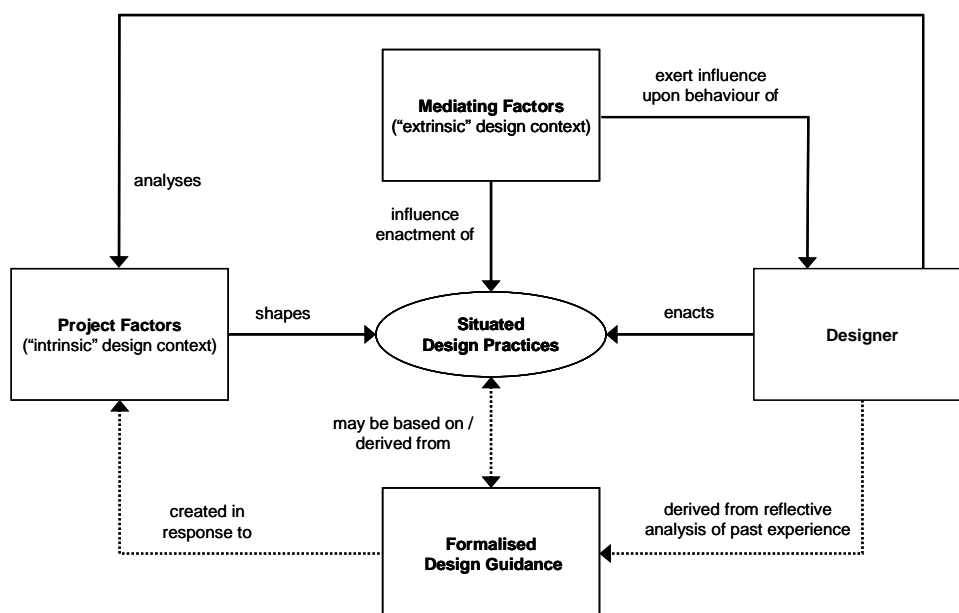
OVERVIEW OF THE CONCEPTUAL FRAMEWORK

Anselm Strauss, one of the original advocates of grounded theory (GT), has affirmed that it can be used not merely to build new theories, but also to extend existing theory by filling in gaps [13]. Accordingly, the framework derived by this study used GT to produce an extended variant of the "Method-in-Action" model, given that the application of this model to Web-based systems development has not yet been investigated in depth [14]. Elements were also adapted from other models, including NIMSAD [15], Multiview/WISDM [16,17], Kumar & Bjørn-Andersen's model of designer values [18], and Gasson's social action model of ISD [19]. The iterative GT technique of "constant comparison" was used firstly to synthesise the main concepts of these models into a coherent unified framework, and then to mould this initial framework into the empirically-grounded model which emerged as the sense-making tasks of data gathering and analysis progressed. Simply put, the resultant framework came together in a manner that was both top-down and bottom-up. Conceptual categories were initially derived from a review of literature and other models, then the content of these categories was filled in by a grounded analysis of empirical data.
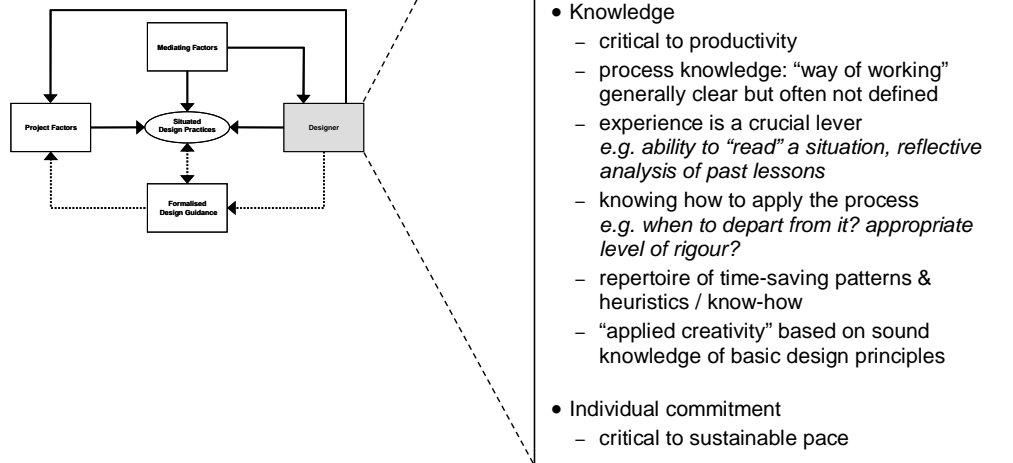
As the research project unfolded and the conceptual framework began to take shape, it became the nucleus of all efforts, providing reference links to the background literature and research questions, informing the research design and philosophical perspective, and guiding the elicitation and reflective analysis of data. The refined conceptual framework which eventually emerged is presented above. At its heart, design practices are regarded as situated actions, purposefully enacted by knowledgable actors who analyse the design context and act accordingly, drawing upon their own experiences to choose an appropriate method. The foundation of the "situated action" view of design is that, "rather than attempting to abstract action away from its circumstances and represent it as a rational plan, the approach is to study how people use their circumstances to achieve

intelligent action" [20]. It rejects the "technical rationalist" assertion that formalised design methods can be executed objectively. Rather, design methods must always be uniquely interpreted; as Essinck [21] puts it, "in a real life project one has to puzzle together one's own specific method, tuned to the problem at hand and the situation the designer is in".

Because of space constraints, it is not possible here to report the full details of empirical findings as they relate to the various categorical headings of the conceptual framework (a completed report is available from the author upon request). The following sections therefore just briefly explain the elements of the framework as they apply to the practice of Web-based systems development.

Designer-encapsulated Factors

A designer's professional training and education can shape his problem-solving orientation and world view by indoctrinating certain values and conditioning him to think and behave in certain ways [22]. An analogy can be drawn here with Kuhn's notion of a "scientific community" which he defines as "the practitioners of a particular specialty … [who] have undergone similar educations and professional initiations" [23]. Kuhn makes the point that these communities, or "schools", may "approach the same subject from incompatible viewpoints". "Incommensurable" [23] or "incongruent" [24] viewpoints can cause people to work at cross-purposes, which has been seen to lead to disappointing outcomes in ISD projects [25]. A number of authors have mentioned that it would be interesting to investigate the practices of Web designers from backgrounds other than software development, so as to build a broader, richer understanding [26,27]. However, this issue has received very little attention thus far. In view of this gap in the literature, a comparison of the methods and approaches used by designers from different professional backgrounds was one of the main concentrations of this study.

Designer-encapsulated Factors

- Professional background
  - problem-solving orientation & priorities
  - historical practices of native discipline
  - cross-skilling / pluralism

- Knowledge
  - critical to productivity
  - process knowledge: "way of working" generally clear but often not defined
  - experience is a crucial lever
    e.g. ability to "read" a situation, reflective analysis of past lessons
  - knowing how to apply the process
    e.g. when to depart from it? appropriate level of rigour?
  - repertoire of time-saving patterns & heuristics / know-how
  - "applied creativity" based on sound knowledge of basic design principles

- Individual commitment
  - critical to sustainable pace

In the survey phase of this research, the cover letter attached to the questionnaire simply requested that it be completed by someone in a design role, the rationale being to capture a cross-section of respondents across the various disciplines that contribute to Web-based systems development. As expected, two dominant disciplinary groupings emerged: computer-based systems development (CBSD), and visual design (VD). Differences in priorities and preferences were observed, apparently influenced by the historical practices (e.g. software specifications versus graphic design "briefs") and orientations (e.g. functional/transactional versus informational/promotional) in each field. For example, the VD group were considerably more lax than the CBSD group as regards requirements documentation, and were also generally very loose concerning the use of "approaches" and "methods". Indeed, the notion of a design "method" seemed to be alien to many of the VD group. On the other hand, the CBSD group were mostly comfortable with the idea of a systematic process for Web-based systems development, such processes mainly being adaptations of traditional software development methods and techniques.

In the follow-up field study, the influence of professional background on design practices was probed in greater depth. Interestingly, a number of different problem-solving perspectives were discovered, each clearly shaped by the various priorities and orientations of the respective disciplines. The perspectives identified were: Web-based systems development as the design of a functional software application (emphasis on back-end functionality); as the design of an interactive tool (emphasis on ergonomics); as the design of a directed communicational dialogue (emphasis on audience engagement); and as an extension of branded graphic design (emphasis on visual presentation).

The framework therefore recognises that a designer's professional background and education can shape his "world view" by conditioning him to think and behave in certain ways. While different perspectives and orientations were found to exist, it would seem that, at least in the field of practice, there is a growing degree of pluralism, as evidenced by a substantial degree of cross-skilling and cross-pollination of techniques.

Though some tasks and stages of Web-based systems development may be formalised and codified, or even automated, there remains a critical need for creative human intervention and the exercise of judgement. Many authors argue that software design is essentially a highly skilled craft [28-30]. It is inaccurate to conceive of design as merely following some pre-defined "cookbook" method; rather, design requires creative thinking and draws upon the skills and experiences of talented individuals [31-33]. Rumbaugh [34] puts it as follows:

> "You can't expect a method to tell you everything to do. Writing software is a creative process, like painting or writing or architecture. There are principles of painting, for example, that give guidelines on composition, color selection, and perspective, but they won't make you a Picasso … Some methods claim to fully automate the [software development] process, to tell you every step to follow so that software design is painless and faultless. They are wrong. It can't be done. What can be done is to supply a framework that tells you how to go about it and identifies the places where creativity is needed."

Continuing with the analogy between painting and software design, it is interesting to read the following extract from Leonardo da Vinci's *Trattato della Pittura* (Treatise on Painting) of 1651:

> "These rules will enable you to possess a free and good judgement, since good judgement is born of good understanding, and good understanding derives from reason expounded through good rules, and good rules are the daughters of good experience – the common mother of all the sciences and arts" [35]

This relationship between method, understanding, experience and judgement, which of course is not specific to painting, can also be seen in the writings of Schön [36] and, within the ISD literature, in the work of Introna & Whitley [37,38]. Accordingly, like the Method-in-Action model, the conceptual framework derived by this study recognises the vital contribution played by creative, talented individuals in the successful execution of the design process. Designers interpret the design context and use their judgement to decide what actions to take in a particular situation.

A strong theme which emerged from the field study was the role of knowledge and experience as a crucial lever in the determination of how Web development processes and guidelines are tailored to meet the needs of the particular situation at hand. Furthermore, knowledge is a critical asset in a development environment characterised by high-speed work practices because it contributes to productivity. More knowledgeable employees are able to work faster because they are equipped with a repertoire of time-efficient "tricks", heuristics, and patterns acquired along the downward traverse of the learning curve. It was found that most of the award-winning companies interviewed have mechanisms in place to facilitate and encourage the management of Web design knowledge, with rewards and bonuses accruing to employees who use slack time to gain and exchange useful knowledge. A number of companies schedule regular time slots for innovative research activity, setting aside normal development work.

The other main designer-encapsulated factor which emerged in this study was individual commitment. Again, like knowledge, this is critical in order to be able to sustain a continuous pace of high-speed delivery. Such issues as organisational culture, appropriate reward mechanisms, and the adoption of practices to eliminate morale-sapping overtime were found to be important in this regard.
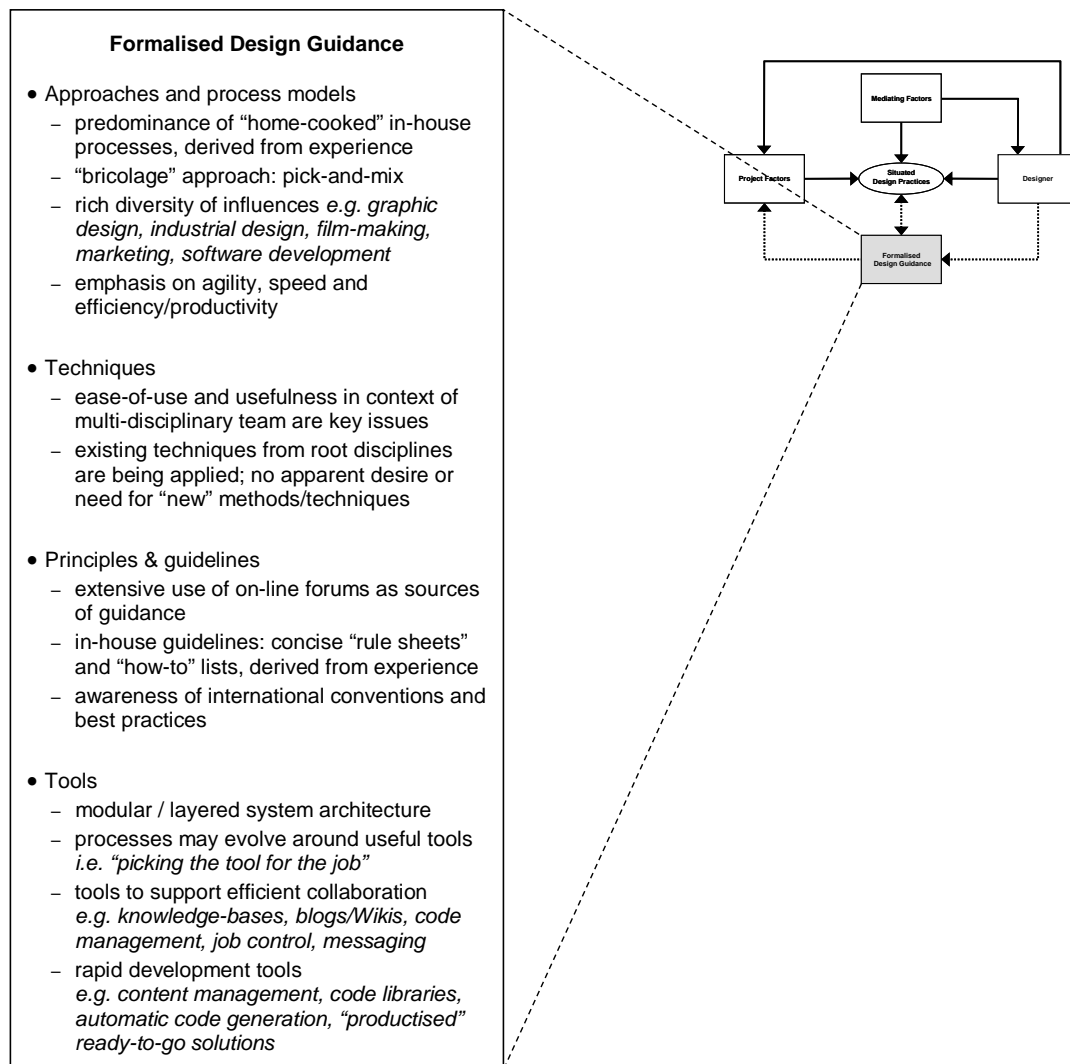
Formalised Design Guidance

Departing slightly from the original Method-in-Action framework, the term "formalised design guidance" is used here in preference to "formalised method" because this study found that, even where Web developers have process documentation in place, it is usually not at the comprehensive level of "method" but more often seems to be simply a collected body of concise procedures, rules of good practice, heuristics and guidelines, or "how-to"

memoranda (e.g. intranet-based "Wiki's" and "blogs").

Though 83% of survey respondents have a clearly understood way of working, in very many cases development processes are not explicitly documented. A similar pattern emerged during follow-up interviews. It would seem that design know-how is best transmitted and acquired by working "on the job", rather than from perusal of formalised procedures or attending training programmes. Most organisations use a "home-cooked" in-house development process that is founded on research, experimentation and reflective analysis of past experience. On the basis of interview findings, these in-house "methods" seem not to be complete end-to-end solutions, but more of a high-level process model within which there is a pick-and-mix selection of low-level techniques to support phase tasks. They are mainly hybrids and custom-tailored variants, based on combinations of internally devised guidelines and public domain methods, informed by an awareness of best industry practice as gleaned from handbooks or on-line forums, and supported by or based around useful tools. This is consistent with the concept of "bricolage" whereby Web designers, rather than shunning method, judiciously assemble fragments of methods and distil the most useful elements into a flexible custom-made approach. Though the same high-level process model may be applied across all projects, tailoring occurs at the level of within-phase tasks, depending on the needs of the particular situation at hand.

Ironically, while there is a vast and ever-growing "jungle" of academically-produced Web-based systems development methods in the literature, none of which are being used to any significant extent in actuality (as evidenced by the survey results), the findings of the field study suggest that out in the real world a single generic high-level process dominates, it resembling a derivative of the traditional "Waterfall" software development model wedded to an amalgam of sub-processes inherited from the fields of graphic design, HCI, strategic marketing / brand design, and industrial design. On the basis of the interview data gathered in this research project, it can be concluded that what differentiates one company from the next is not the overall shape or format of their development process, – notwithstanding the fact that many companies do indeed present their process as a unique selling point, – but rather the way in which the finer points of that process are uniquely interpreted by their design team in the specific context of a particular project.

In addition to the form of the generic Web development process model, – which represents a fusion of approaches drawn from a variety of sources, – the influence of multiple disciplinary fields on the practice of Web-based systems development is evidenced by the finding that all interviewees, regardless of their professional backgrounds, found that the same methods and techniques they had formerly used in their "native" discipline transferred across to Web design. This suggests that wholly new methods and techniques for Web-based systems development are neither necessary nor appropriate. It was also generally found that ease-of-use, usefulness and representational capabilities are important

Formalised Design Guidance

- Approaches and process models
  - predominance of "home-cooked" in-house processes, derived from experience
  - "bricolage" approach: pick-and-mix
  - rich diversity of influences *e.g. graphic design, industrial design, film-making, marketing, software development*
  - emphasis on agility, speed and efficiency/productivity

- Techniques
  - ease-of-use and usefulness in context of multi-disciplinary team are key issues
  - existing techniques from root disciplines are being applied; no apparent desire or need for "new" methods/techniques

- Principles & guidelines
  - extensive use of on-line forums as sources of guidance
  - in-house guidelines: concise "rule sheets" and "how-to" lists, derived from experience
  - awareness of international conventions and best practices

- Tools
  - modular / layered system architecture
  - processes may evolve around useful tools *i.e. "picking the tool for the job"*
  - tools to support efficient collaboration *e.g. knowledge-bases, blogs/Wikis, code management, job control, messaging*
  - rapid development tools *e.g. content management, code libraries, automatic code generation, "productised" ready-to-go solutions*

factors which affect the choice of conceptual modelling techniques for Web design. Whereas the emphasis of traditional software development techniques was on back-end functionality (e.g. ERDs for database-driven applications), there is now also an essential need for front-end design techniques drawn from the field of visual communications, such as storyboards and "mood boards".
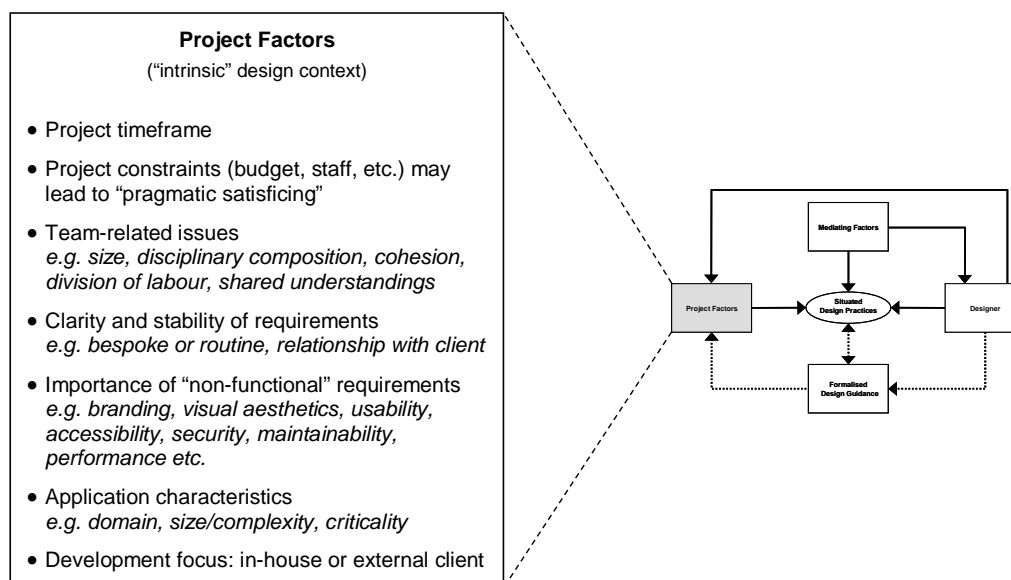
Given the high-speed nature of Web-based systems development, the emphasis of formalised design guidance is very much on agility, speed, efficiency and productivity. Streamlined processes are necessary in order to maximise throughput, and also to sustain a continual pace by eradicating the need for ongoing overtime (which has fatiguing and demoralising effects). Interestingly, the Web developers interviewed have evolved practices that are markedly similar to those of the "agile" methods family, such as: collective code ownership; an emphasis on simplicity; the use of regular informal team briefings; insistence on a close working relationship with the client; the pursuit of continuous process improvement through reflective evaluation; and a general emphasis on people, communication, and working software over processes, documentation, and adherence to a plan. Processes and procedures are

therefore treated as flexible frameworks to guide and assist the essentially creative tasks of analysis and design.

The central role of tools in the formalisation of work practices also emerged as an important factor. For example, the use of automatic code generation, re-usable components (both code and graphical elements), enhanced RAD tools, modular tiered systems architectures, and "productised" software solutions greatly speeds up Web development without subverting cost or quality. Additionally, the store of in-house knowledge, which is an important factor in productivity, can be more effectively leveraged through the advantageous use of collaborative forums such as intranets, "Wiki's", and "blogs".

Project Factors (intrinsic design context)

Whitley [38] makes the point that "in order to be able to use a method appropriately, it is necessary to have an understanding of the context in which it is being used". There is a significant body of literature on the notion of situation-specific "method engineering" [39-43], and while there are considerable issues surrounding the feasibility of such an approach in practice [44-46], it is nevertheless generally accepted that different situations

Project Factors
("intrinsic" design context)

- Project timeframe

- Project constraints (budget, staff, etc.) may lead to "pragmatic satisficing"

- Team-related issues
  e.g. size, disciplinary composition, cohesion, division of labour, shared understandings

- Clarity and stability of requirements
  e.g. bespoke or routine, relationship with client

- Importance of "non-functional" requirements
  e.g. branding, visual aesthetics, usability, accessibility, security, maintainability, performance etc.

- Application characteristics
  e.g. domain, size/complexity, criticality

- Development focus: in-house or external client

warrant different approaches [21,47-51]. All of the aforementioned conceptual frameworks recognise that design practices may be affected by the specific circumstances of the design context, which is variously referred to as the "problem situation"/"methodology context" (NIMSAD), "situation" (Multiview), "context" (Kumar & Bjørn-Andersen's model), and "business/development context" (Method-in-Action). Here, the design context is represented by the categories labelled "Project Factors: intrinsic design context" and "Mediating Factors: extrinsic design context".

It was found that the duration of a typical Web development project is of the order of 2 to 3 months. Such short delivery cycles, until recently at least, were unprecedented in traditional software development and are made possible in Web-based systems development by a combination of factors. Firstly, the Web is an immediate delivery medium which is not impeded by production, distribution and installation delays. Secondly, as evidence by the interview data, there have been dramatic gains in recent years in developer productivity, coupled with ever more efficient and refined development processes. This has been achieved through the use of high-speed rapid application tools, templates and wizards for automatic code-generation, plug-and-play database connectivity, and libraries of pre-fabricated components and applets. Web programming is now advanced to a point where most development time is invested into the ongoing evolution of an out-of-the-box solution. Code production for a project has moved from crude cut-and-paste re-use to instant automatic generation, meaning that most of the standard back-end functionality required for any given project can be up and running within a day or two. The visual design of the GUI front-end, like the traditional production process for commercial art, can also be done within a very short timeframe. A fully-proven working prototype can therefore be very quickly launched, which can later be modified and enhanced in such a manner that end-users may be largely oblivious to the ongoing changes. As such, rapid/agile and evolutionary/incremental

development approaches are a natural fit to the Web environment.

Consistent with the previous work of Baskerville & Pries-Heje [52,53], this study found, as one would expect, that time pressure is the central determinant of design practices. However, there are discrepancies between this research and that of Baskerville & Pries-Heje, most notably with their finding that developers may resort to the practices of "coding your way out" and "negotiated quality" because of the pressures of high-speed development environments. Whereas in Baskerville & Pries-Heje's study such practices were endemic, in this research hardly any such incidents were discovered. This can be explained in a number of ways. Firstly, the interviewed companies were mostly award-winners, a likely indicator that they make special efforts to strive for excellence and quality. Secondly, the marketplace has become more competitive in recent years and users are much less tolerant of unprofessional standards of work, meaning that expectation levels have risen. Thirdly, as already mentioned, the use of pre-fabricated "productised" solutions that are already fully tested means that robust systems can be rapidly delivered without compromising cost or quality. Even in the worst case scenario for a development team, where they face the dreaded "backs-to-the-wall" combination of acute time and resource constraints, a tactic herein coined as "pragmatic satisficing" is engaged, meaning that a tried-and-tested solution is re-used, albeit it may not be the best possible outcome.

It was found, initially in the survey and later in the follow-up interviews, that most Web development teams are small, typically comprising about 5 to 10 members for any given project. This affords the advantage that communication problems are minimal and that cohesion can more easily be achieved, both of which are important for timely delivery. As teams grow in size, knowledge becomes fragmented. There consequently arises a need to formalise and standardise working methods (e.g. conventions for collective code ownership) because

otherwise wasteful inefficiencies due to "re-inventing the wheel" can occur. In both the survey and the follow-up interviews, it was found that larger teams tend to make more use of documented guidelines and procedures.

Conflict between Web designers from different professional backgrounds was not found to be much of a problem in practice. This is because the once rival factions of software engineering and graphic design have over time come to gain an appreciation of each others' perspectives and priorities (as evidenced by a considerable degree of cross-skilling), and it is now easier to separate front-end and back-end Web design into different layers than it was a few years ago.

The clarity and stability of requirements is an age-old issue in systems development, but in high-speed environments it is important to "nail" a prioritised list as quickly as possible. In comparison with traditional software development, it was found during the field study that a greater weighting of time in Web-based systems development is spent on analysis and design as opposed to coding. Requirements analysis is the most time-consuming phase of all in Web development, whereas coding can actually be very quick. Though most of the functional requirements for a Web-based system are typically standard and can therefore be readily described, the bespoke elements take time to specify, as does a considered analysis of the fine details of the overall package including the "non-functional" requirements (usability, accessibility, security, performance levels, etc.). As initially revealed by the survey and later substantiated by follow-up interviews, it is common practice to produce and sign-off a detailed requirements specification before commencing full scale production, the purpose of which is to keep feature creep in check and compel clients to make firm decisions.

From the interviews, it seems that most organisations use largely the same development process for all types of applications, regardless of delivery platform or application domain. While the general process may be very similar across all projects, the rigour with which its sub-tasks are executed varies, as one would expect, in accordance with application size/complexity and application criticality. Some evidence was found in the survey that in highly specialised areas such as interactive e-learning/CBT applications, a proprietary method might be used, and also that in some sectors (e.g. Financial Services) there is a greater emphasis on processes and documented procedures (e.g. detailed functional specifications, formalised organisational guidelines). However, a shortcoming of this study is that insufficient data was gathered to analyse the influence of specialised application domains on the finer details of Web development processes and procedures (e.g. security is a concern in the development of e-banking systems, but this was only incidentally touched upon in this research).

The focus of systems development activity (i.e. in-house versus external client) was also found to impact development practices. Whereas Web design agencies can agree plans with clients and negotiate with them over who pays for subsequent over-runs, in-house development teams are in a "hands tied" situation,

meaning that project planning is necessarily done very differently.
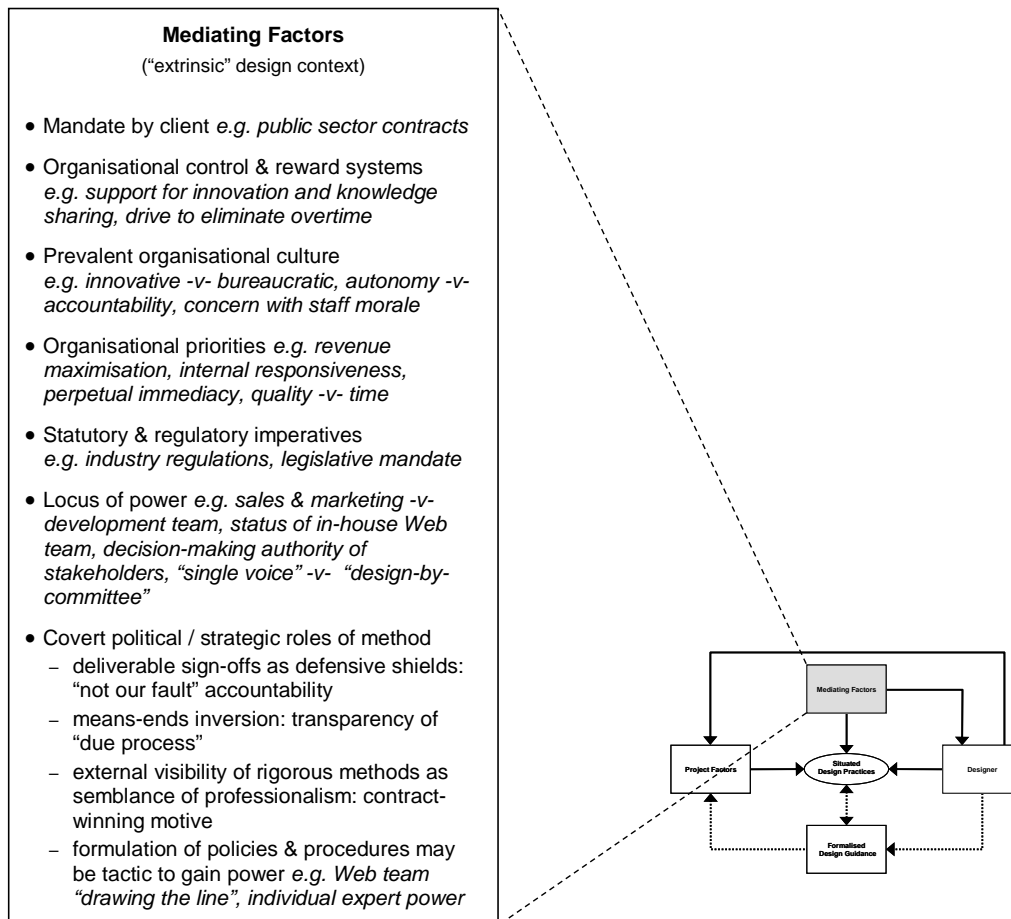
Mediating Factors (extrinsic design context)

Design practices can sometimes be affected by the intervention of extraneous factors, the influence of which may be to cause designers to pursue a course of action they might not otherwise take. For example, it was found during interviews that there may be a mandate by the client that certain procedures are to be rigidly followed (e.g. because of statutory requirements to comply with certain standards, or the existence of binding protocols for procurement sor software testing), or not to be followed (e.g. political pressure to complete, "just do it!").

As was previously observed by Powell et al [54], this study found that the locus of power within organisations can significantly influence the development approach. For example, fledgling in-house Web development units often have to resort to "pragmatic satisficing" behaviour because they are under-resourced. In Web design agencies, a typical cause of conflict is the competing motives of the sales team (revenue maximisation) and the development team (quality optimisation), – this argument is usually won by the sales team, and programmers might end up being coerced into taking shortcuts to meet targets. The locus of power is also a common issue for client organisations, where the politics, indecision, and communicative difficulties arising from the "design-by-committee" syndrome can frustrate even the best laid project plans.

Associated with the concept of reward and control systems are two closely related other concepts: organisational priorities and organisational culture. Prerogatives such as perpetual immediacy, statutory and regulatory imperatives, a commercial desire to maximise revenue/throughput, a need to be internally flexible with schedules and requirements, or a focus on quality above time and cost considerations can impact development processes by directing priorities. Similarly, the culture of an organisation, as reinforced by control and reward mechanisms, is also a relevant issue (e.g. emphasis on individual *accountability* as opposed to responsible *autonomy*).

As with the original Method-in-Action model, it was again found in this study that development methods may fulfil covert political roles. These included: establishing a power-base for method champions (e.g. the XP, WAI, or BS7799 "expert"), maintaining a transparent and accountable audit trail of the development process as a protective fallback (e.g. the in-house "blame game", or negotiating responsibility for change requests or delays with clients), providing assurance that correct and "proper" practices are being followed (e.g. public-sector tenders), and helping to raise the status of in-house Web development departments (e.g. the creation of internal policies to "legitimise" or "professionalise" operations).

Mediating Factors

("extrinsic" design context)

- Mandate by client *e.g. public sector contracts*

- Organisational control & reward systems
  e.g. support for innovation and knowledge sharing, drive to eliminate overtime

- Prevalent organisational culture
  e.g. innovative -v- bureaucratic, autonomy -v- accountability, concern with staff morale

- Organisational priorities *e.g. revenue maximisation, internal responsiveness, perpetual immediacy, quality -v- time*

- Statutory & regulatory imperatives
  e.g. industry regulations, legislative mandate

- Locus of power *e.g. sales & marketing -v- development team, status of in-house Web team, decision-making authority of stakeholders, "single voice" -v- "design-by-committee"*

- Covert political / strategic roles of method
  - deliverable sign-offs as defensive shields: "not our fault" accountability
  - means-ends inversion: transparency of "due process"
  - external visibility of rigorous methods as semblance of professionalism: contract-winning motive
  - formulation of policies & procedures may be tactic to gain power *e.g. Web team "drawing the line", individual expert power*

CONCLUSIONS

The framework presented in this paper provides a macro-level overview of the context of Web-based systems development and the various inter-related issues therein. A criticism that can be made of much "Web engineering" research, particularly that which concentrates on design methods, is that problems are often investigated in isolation, without due consideration of their "natural" context in the real-world environment of practice. For example, there is a vast array of academically-produced Web/hypermedia design methods in the literature, but very few of these are being used in industry. There are many reasons why this may be so, but the long-standing criticism [55] remains that many of these methods have only been validated in restricted experimental settings or pilot studies as opposed to industrial-strength projects. The framework is helpful in this regard by providing academic researchers and method developers with a view of the over-arching context of Web-based systems development, thereby encouraging systemic thinking and "big picture" problem-solving, which ultimately should lead to research products that are more attuned and adaptable to the demands of practice.

As regards implications for education, IS/IT graduate programmes historically placed substantial emphasis on formalised design methods and techniques as described in standard textbooks, neglecting or entirely ignoring the factors which impact the use of those methods and techniques in practice. This limited one-dimensional perspective meant that perplexed graduates straight out of college often found themselves at a loss to understand how so much of the material they had diligently studied seemed to be irrelevant in the "real world". The conceptual framework derived by this research is therefore potentially valuable for educators because it constitutes the outline for a revised and extended curriculum which treats Web-based systems development as a situated contextually-sensitive activity.

REFERENCES

[1] Cusumano, M. A. & Yoffie, D. B. (1999) Software Development on Internet Time. *IEEE Computer.* 32(10) October, 60-69.

[2] Murugesan, S. & Deshpande, Y. (1999) Preface to ICSE'99 Workshop on Web Engineering. In *21st International Conference on Software Engineering (ICSE)*, Los Angeles, California, USA, May 16-22, pp. 693-694.

[3] Murugesan, S., Deshpande, Y., Hansen, S. & Ginige, A. (1999) Web Engineering: A New Discipline for Development of Web-based Systems. In *1st ICSE Workshop on Web Engineering*, Los Angeles, California, USA, May 16-17, pp. 1-9. ACM Press.

[4] Oinas-Kukkonen, H., Alatalo, T., Kaasila, J., Kivelä, H. & Sivunen, S. (2001) Requirements for Web Engineering Methodologies. In Rossi, M. & Siau, K. (eds), *Information Modeling in the New*

Millennium, pp. 360-382. Hershey, PA: Idea Group Publishing.

[5] Glass, R. L. (1998) Is There Really a Software Crisis? *IEEE Software.* 15(1) January-February, 104-105.

[6] Naur, P. & Randell, B. (eds) (1969) *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 October 1968.* Brussels: Scientific Affairs Division, NATO.

[7] Gibbs, W. W. (1994) Software's Chronic Crisis. *Scientific American.* September, 72-81.

[8] Glaser, B. G. & Strauss, A. L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research.* New York: Aldine de Gruyter.

[9] Strauss, A. & Corbin, J. (1998) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd Edition. Thousand Oaks, CA: Sage. ISBN 0-8039-5939-7.

[10] Locke, K. (2001) *Grounded Theory in Management Research.* London: Sage.

[11] Miles, M. B. & Huberman, A. M. (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd Edition. Thousand Oaks, CA: Sage.

[12] Lang, M. (2002) The Use of Web-based International Surveys in Information Systems Research. In Remenyi, D. (ed), *European Conference on Research Methodology for Business and Management Studies (ECRM 2002)*, Reading, England, April 29-30, pp. 187-196. MCIL: Reading.

[13] Strauss, A. L. (1970) Discovering new theory from previous theory. In Shibutani, T. (ed), *Human Nature and Collective Theory,* pp. 46-53. Englewood Cliffs, NJ: Prentice Hall.

[14] Fitzgerald, B., Russo, N. L. & Stolterman, E. (2002) *Information Systems Development: Methods in Action.* London: McGraw-Hill.

[15] Jayaratna, N. (1994) *Understanding and Evaluating Methodologies, NIMSAD: A Systemic Framework.* London: McGraw-Hill.

[16] Avison, D. E., Wood-Harper, A. T., Vidgen, R. T. & Wood, J. R. G. (1998) A further exploration into information systems development: the evolution of Multiview2. *Information Technology & People.* 11(2), 124-139.

[17] Vidgen, R., Avison, D., Wood, B. & Wood-Harper, T. (2002) *Developing Web information systems: from strategy to implementation.* Oxford: Butterworth Heinemann. ISBN 0-7506-5763-4.

[18] Kumar, K. & Bjørn-Andersen, N. (1990) A Cross-Cultural Comparison of IS Designer Values. *Communications of the ACM.* 33(5) May, 528-538.

[19] Gasson, S. (1999) A Social Action Model of Situated Information Systems Design. *DATA BASE (ACM SIGMIS).* 30(2) Spring, 82-97.

[20] Suchman, L. A. (1987) *Plans and Situated Actions: The Problem of Human-Machine Communication.* Cambridge: Cambridge University Press.

[21] Essinck, L. J. B. (1988) A conceptual framework for information systems development methodologies. In Bullinger, H.-J. et al. (eds), *Information Technology for Organisational Systems,* pp. 354-362. Amsterdam: North-Holland.

[22] Sahraoui, S. (1998) Is Information Systems Education Value Neutral? *Journal of Computer Information Systems.* 38(3) Spring, 105-109.

[23] Kuhn, T. S. (1996) *The Structure of Scientific Revolutions*, 3rd Edition. Chicago: University of Chicago Press.

[24] Orlikowski, W. J. & Gash, D. C. (1994) Technological Frames: Making Sense of Information Technology in Organizations. *ACM Transactions on Information Systems.* 12(2) April, 669-702.

[25] Bostrom, R. P. & Heinen, J. S. (1977) MIS Problems and Failures: A Socio-Technical Perspective, Part 1: The Causes. *MIS Quarterly.* 1(3) September, 17-32.

[26] Jonasson, I. (2000) *Developing the Information Systems of Tomorrow - competencies and methodologies.* M.Sc. Dissertation, Department of Computer Science, University of Skövde, Sweden.

[27] Russo, N. L. & Graham, B. R. (1999) A First Step in Developing a Web Application Design Methodology: Understanding the Environment. In Wood-Harper, A. T. et al. (eds), *Methodologies for Developing and Managing Emerging Technology Based Information Systems: 6th International BCS Information Systems Methodologies Conference,* pp. 24-33. London: Springer.

[28] Wroblewski, D. A. (1991) The Construction of Human-Computer Interfaces Considered as a Craft. In Karat, J. (ed), *Taking Software Design Seriously: Practical Techniques for Human-Computer Interaction Design,* pp. 1-19. Academic Press.

[29] McBreen, P. (2002) *Software Craftsmanship: The New Imperative.* Boston: Addison Wesley.

[30] Taylor, P. R. (2004) Vernacularism in software design practice: Does craftmanship have a place in software engineering? *Australasian Journal of Information Systems.* 11(12), 14-25.

[31] Glass, R. L. (1995) *Software Creativity.* Englewood Cliffs, NJ: Prentice Hall.

[32] Shaw, M. & Garlan, D. (1996) *Software Architecture: Perspectives on an Emerging Discipline.* Prentice Hall.

[33] Stolterman, E. & Russo, N. (1997) The Paradox of Information Systems Methods: Public and Private Rationality. In *5th British Computer Society Conference on Information Systems Methodologies*, Lancaster, England, pp. 351-362.

[34] Rumbaugh, J. (1995) What is a method? *Journal of Object Oriented Programming.* 8(6) October, 10-16;26.

[35] White, M. (2000) *Leonardo: The First Scientist.* London: Little, Brown & Company.

[36] Schön, D. A. (1983) *The reflective practitioner: how professionals think in action.* London: Temple Smith.

[37] Introna, L. D. & Whitley, E. A. (1997) Against method-ism: exploring the limits of method. *Information Technology & People.* 10(1), 31-45.

[38] Whitley, E. A. (1998) Method-ism in Practice: Investigating the Relationship between Method and Understanding in Web Page Design. In *19th International Conference on Information Systems (ICIS)*, Helsinki, Finland, December 13-16, pp. 68-75.

[39] Kumar, K. & Welke, R. J. (1992) Methodology Engineering: A Proposal for Situation-Specific Methodology Construction. In Cotterman, W. W. & Senn, J. A. (eds), *Challenges and Strategies for Research in Systems Development,* pp. 257-269. John Wiley & Sons.

[40] Brinkkemper, S. (1996) Method engineering: engineering of information systems development methods and tools. *Information and Software Technology.* 38(4), 275-280.

[41] Hidding, G. (1996) Method Engineering: Experiences in Practice. In Brinkkemper, S. et al. (eds), *Method Engineering: Principles of Method Construction and Tool Support.* London: Chapman & Hall.

[42] Song, X. (1997) Systematic Integration of Design Methods. *IEEE Software.* 14(2) March-April, 107-117.

[43] Gnatz, M., Marschall, F., Popp, G., Rausch, A. & Schwerin, W. (2003) The Living Software Development Process. *Software Quality Professional.* 5(3), 4-16.

[44] Ter Hofstedt, A. H. M. & Verhoef, T. F. (1997) On the Feasibility of Situational Method Engineering. *Information Systems.* 22(6-7), 401-422.

[45] Fitzgerald, B., Russo, N. L. & O'Kane, T. (2003) Software Development Method Tailoring at Motorola. *Communications of the ACM.* 46(4) April, 65-70.

[46] Truex, D. & Avison, D. (2003) Method Engineering: Reflections on the Past and Ways Forward. In *9th Americas Conference on Information Systems (AMCIS)*, Tampa, Florida, USA, August 4-6, pp. 508-514.

[47] Malouin, J.-L. & Landry, M. (1983) The Mirage of Universal Methods in Systems Design. *Journal of Applied Systems Analysis.* 10, 47-62.

[48] Kraemer, K. L. & Dutton, W. H. (1991) Survey Research in the Study of Management Information Systems. In Kraemer, K. L. (ed), *The Information Systems Research Challenge: Survey Research Methods. Volume 3,* pp. 3-58. Boston, Massachusetts: Harvard Business School.

[49] Ratbe, D., King, W. R. & Kim, Y.-G. (1999) The fit between project characteristics and application development methodologies: a contingency approach. *Journal of Computer Information Systems.* 40(2) Winter 1999-2000, 26-33.

[50] Cockburn, A. (2000) Selecting a Project's Methodology. *IEEE Software.* 17(4) July-August, 64-71.

[51] Jackson, M. (2000) The Origins of JSP and JSD: A Personal Recollection. *IEEE Annals of Software Engineering.* 22(2) April-June, 61-63.

[52] Baskerville, R. & Pries-Heje, J. (2001) Racing the E-Bomb: How the Internet is Redefining Information Systems Development Methodology. In Russo, N. L. et al. (eds), *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective. IFIP WG8.2 Conference, Boise, Idaho, USA, 27-29 July 2001,* pp. 49-68. Boston: Kluwer Academic Publishers.

[53] Baskerville, R. & Pries-Heje, J. (2004) Short cycle time systems development. *Information Systems Journal.* 14(3), 237-264.

[54] Powell, T. A., Jones, D. L. & Cutts, D. C. (1998) *Web Site Engineering: Beyond Web Page Design.* Upper Saddle River: Prentice Hall.

[55] Fitzgerald, G. (1991) Validating New Information Systems Techniques: A Retrospective Analysis. In Nissen, H.-E. et al. (eds), *Information Systems Research: Contemporary Approaches and Emergent Traditions,* pp. 657-672. Elsevier Science Publishers B.V. (North-Holland).