

Towards a Reference Process for Developing Wireless Internet Services

Alexis Ocampo, Daniela Boggio, Jürgen Münch, Gino Palladino

Abstract—Wireless Internet services such as mobile Web applications promise an enormous market potential. The field is characterized by extreme time-to-market pressure and insufficient knowledge about development procedures and technical constraints. This results in insufficient guidance for project managers and software developers on selecting appropriate development processes, techniques, methods and tools. In addition, there is an enormous lack of knowledge about the effects (such as effort consumption, defect injection) of such technologies that hinders the transfer of innovative technologies into practice. This article describes an initial reference process by summarizing essential technologies for the development of wireless Internet services and experience with these technologies on the levels of life cycle processes, engineering processes, and managerial processes. The reference process is based on a comprehensive literature survey and the execution of development projects for wireless Internet services. The goal of the article is to provide domain-specific guidance for project managers and software developers with accompanying lessons learned from the past.

Index Terms— Mobile Services, Wireless Internet Services, Software Process Modeling, Process Elicitation, Reference Process Model, Software Project Management.

1 INTRODUCTION

The lack of guidance on selecting technologies (i.e., processes, methods, techniques, tools) for the development of wireless Internet services makes accurate project planning very difficult and, in many cases, impossible. This is a serious problem, especially for time-to-market critical projects. Usually, for new application domains (such as wireless Internet services), no explicitly described guidance is available yet. Furthermore, the design and introduction of technologies is very risky, because typically there exists no previous experience about which technologies are suitable and executable in the environment of the developing organization. To remain competitive in the market, an appropriate and piloted reference guide is needed very quickly.

Early results from a multi-case study conducted at Nokia Mobile Phones [7] and experience from the NASA Software Engineering Laboratory (SEL) [14] clearly show the importance of a stable and well-implemented infrastructure for the deployment of technologies. A prerequisite for this is an explicitly defined reference process for the application domain that is tailorable to specific project contexts.

This article describes an initial reference process by summarizing essential technologies for the development of wireless Internet services and experience with these technologies on the levels of life cycle processes, engineering processes, and managerial processes. The reference process is based on a comprehensive literature survey and the execution of two completed development

projects for wireless Internet services. The goal of the article is to provide domain-specific guidance for project managers and software developers with accompanying lessons learned from the past. However, the reader has to adapt the conclusions to his or her context.

The article is structured as follows: Section 2 gives background information on how the reference model was developed. Section 3 characterizes the wireless Internet services domain and describes the reference process by accentuating the specifics of the domain, listing essential development technologies for different process levels and phases, and giving initial experience with these technologies. Finally, Section 4 summarizes the article and discusses experience and open issues.

2 BACKGROUND

Since software development projects are unique regarding their combination of specific goals and characteristics, providing 'ideal' and, at the same time, universal development technologies is not a suitable solution for real life [12]. Instead, effective and efficient software development technologies tailored to the particularities of the application domain and project constraints are required.

The wireless Internet services domain is an application domain that can be characterized as follows from the business point of view: quickly evolving technology, upcoming new devices and communication protocols, together with the need for new business and billing models that will fit in with the completely new services portfolio. From the technical viewpoint, the following essential characteristics can be seen: volatility of requirements, heterogeneity of the terminals and the service providers, difficulty in testing with all types of terminals, limited bandwidth, terminals, screens, computing power, memory, battery, discontinuity of service, and location awareness. Most of the characteristics also apply to at least one other application domain. Nevertheless, the

- A. Ocampo and J. Münch are with the Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, 67661 Kaiserslautern, Germany. Email: {ocampo, muench}@iese.fhg.de).
- D. Boggio is with the Motorola Technology Centre, 10147 Italy Turin. E-mail: ADB005@email.mot.com.
- G. Palladino is Investnet Italia Spa, 20125 Italy Milano. E-mail: g.palladino@investbv.com.

Manuscript received (xx, 2003). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

xxxx-xxxx/0x/\$xx.00 © 200x IEEE

© 2003 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

DOI: 10.1109/TSE.2003.1265526, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1265526>

combination of characteristics can be seen as typical for the wireless Internet services domain. Examples of new wireless Internet services can be expected in the domain of Mobile Entertainment, Telemedicine, Travel Services, Tracking and Monitoring Services, or Mobile Trading Services.

At the moment, there is very little experience in developing software for such services systematically. Therefore, designing processes for this domain implicates several difficulties: 1) Whereas several standards exist for conventional software development, no such standards are available for wireless Internet services that could be used as reference. 2) The wireless Internet services domain lacks specific experience on particular technologies, their applicability and constraints. 3) The variations of the applications and, as a consequence, possible variations of the development technologies, are not sufficiently understood.

There are several ways towards solving this problem: one widely accepted idea in the software engineering community is the execution of pilot projects (i.e., prototype developments) and the parallel descriptive modeling of development processes, which leads to the explicit definition of process models, product models, and resource models [18]. Descriptive software process modeling captures processes as they take place during development. For example, the initial descriptive process model elicited from a software organization that faces a new project on wireless Internet services could be a practice adopted from a similar domain. Then, establishing baselines (e.g., an effort baseline), and collecting and using measurement data may further enhance the understanding and control of software development processes, products, and relationships between them [28].

The reference process presented in this article was obtained during the development of two pilot projects. Based on market demands (such as the need to adapt existing services for the Internet towards wireless Internet services or to create new services) and companies' interests, initially two target contexts for two projects were defined: the development of a wireless Internet service for mobile online trading (Pilot 1) and the development of a service for mobile entertainment (Pilot 2). An excerpt of the description of the project contexts is shown in Table 1.

The overall method used as a basis for designing the reference process for developing wireless Internet services is depicted in Figure 1 and is described in detail in [29]. The method consists of the following steps: set up pilots, perform pilots, elicit and model processes, search and evaluate processes and practices from related fields, analyze commonalities and differences, and create comprehensive process model. We call the resulting comprehensive process model in this article reference process, because it is intended to be used as a reference for developers and managers that provides initial orientation in selecting appropriate technologies.

Creating a reference process solely from observing software projects limits the level of detail and precision to the abilities of the respective project organizations in which the projects were performed. Therefore, we enhanced the developed reference process carefully with knowledge from other sources (e.g., literature, experience reports from similar projects). The creation of the reference process is related to *Situational Method Engineering*, an area of research that deals with how to build project-specific methods called *situational methods* from existing method parts called *method fragments* in a consistent way [54]. Brinkkemper et al. [54] provides a list of requirements, guidelines, and con-

TABLE 2
PILOT PROJECTS CONTEXT

Context	Characteristic	Project 1	Project 2
Internet Computing			11
Software characteristics	Application type	Information system	Computation intensive system
International Conference Proceedings	Business area	Mobile online trading	Mobile online entertainment
Computers Communications	Business area	Mobile online trading	Mobile online entertainment
IEEE Transactions on Software Engineering	Business area	Mobile online trading	Mobile online entertainment
Information and Software Technology	Business area	Mobile online trading	Mobile online entertainment
Development characteristics	Project type	Client System adaptation	Client New development Server New development
	Transport protocol	GSM/GPRS/UMTS	GSM/GPRS/UMTS
	Implementation language	WML	Client: J2ME Server: J2EE
Enterprise characteristics	Organizational context	Investnet-Italy	Motorola MTCI
	Role	Service provider, content provider, service developer	Technology provider, service developer

straints for assembling methods, so that the resulting method is meaningful and consistent.

2.1 Search Scope

The literature search scope was oriented towards technologies that, regardless of the complexity of the environment, are means for producing wireless Internet services of high quality and on time. The criteria to select the articles and include them in the results were: 1) The article presents an approach viable to be integrated in the projects. 2) The article gives a clue for creating a flexible process model that creates products of high quality, on time. 3) The article explicitly describes techniques, tools, guidelines, steps, procedures or methods. A combination of different methods was used in the literature search in order to find as much information as possible. It is important to note that there is no perfect method for finding all the information related to a specific subject [17]. Keyword searches were performed in the following databases: INSPECT, THEMA, COMPENDEX, COMPUSCIENCE, and NTIS. A manual search was performed on the editions between 1999 and 2002 of the following journals: IEEE Internet Computing, IEEE Transactions on Software Engineering, IEEE Software. Other sources of information were international conference publications, software engineering related journals, software engineering consultancy firm reports, software engineering institute reports, and wireless Internet services industrial key player reports. Table 2 presents an overview of the articles found.

3 REFERENCE PROCESS

In this section, an initial version of the reference process for developing wireless Internet services is presented. The characteristics of the wireless Internet services domain are relevant, for example, for choosing the right life cycle process model to follow, or the most appropriate technique for doing requirements, design, etc. Thus, the added value of the reference process relies on the following: 1) descriptions of the domain characteristics; 2) allocation of the domain characteristics in the reference process; 3) descriptions of the experience of real projects on addressing these characteristics

Below, technologies are described for the following process levels and phases: Life cycle processes, requirements, design, implementation, test, and managerial processes. For each subsection, a description of the characteristics of wireless Internet services is given, a set of suitable practices found in literature is presented, and the experience from real projects is described.

3.1 Life Cycle Processes

3.1.1 Wireless Internet Services Characteristics

Flexible Processes

The market of wireless Internet services is expected to grow in accordance with technology, and to be limited by time constraints of the market. Software development organizations must have the possibility to change their strategy in a given circumstance, in a given point of time during development. Strict, inflexible process models like the waterfall model are not suitable for such a context. Organizations must react to the context in the most appropriate manner, and that is only possible through flexible processes.

3.1.2 Practices

Suitable life cycle processes for the wireless Internet services domain are the throwaway prototype model and the incremental development model [38], which were found suitable for domains of similar characteristics like Internet and mobile phone [6], [10], [26], [19], [3], [31]. Through these models, essential operational functions are provided initially, and then more capable versions of the system. Increments are usually defined as an agreement between the customer and the development organization. This allows development organizations to get feedback from the final customer during the development of the increments until the final version of the solution is delivered. Additionally, monitoring and controlling the project plan can be done more precisely, and the quality of increments can be assured with the established verification and validation activities.

Agile development practices and techniques aims at finding a balance between flexibility and structure in the actual business environment, where volatility and uncertainty increase [55].

One of them, adaptive software development (ASD), proposes to face uncertainty with short delivery iterations; new requirements and technical information with intensive collaboration among managers, customers and developers; and process improvement with reviews after each iteration and project retrospectives. The dynamic systems development method (DSDM) [56] contains three major phases: functional model iteration, design and build iteration, and implementation. All three of them are iterative. DSDM is similar to ASD because it considers collaboration, learning and allows to introduce new functionality (new

requirements) in the project as new things are learned. Additionally, prototypes built for each feature are preferred over long documents as documentation.

Another agile approach, *Extreme programming*, is proposed by [35] as suitable for Web-based projects where time to market plays an important role. *Extreme programming* focuses on producing source code and test drivers, avoiding documentation, and handling the volatility of requirements through small releases. *Extreme programming* reflects an incremental development model. Development cycles are short and based on requirements that will really generate business value for the customer. One risk of agile approaches is that they rely on the tacit knowledge of developers [37]. This should be carefully considered, especially because developers are still learning due to the immaturity of the wireless Internet services domain. Additionally, issues like scalability and performance have to be carefully designed.

The spiral model [36] assumes risks as the driving force of software projects. This model proposes ongoing refinement of the system specification into source code components. Refinements are made through cycles, and each cycle is risk assessed. A risk assessment determines if a project continues or is cancelled. The nature of the spiral model seems reasonable to apply in a convoluted domain like wireless, but the real cost of identifying, analyzing and maintaining risks can be high, which is not so suitable for small and medium-sized companies.

Boehm [37] proposes a combination of agile and plan-driven methods through risk-driven spiral methods, which are intended to balance flexibility and discipline. The rationale behind that is that although the market changes constantly and puts pressure on development organizations to deliver their products rapidly, increasing dependability of systems and applications demands high quality products.

3.1.3 Experience

A comprehensive process model from the real experience of two pilots is the result of overlapping their original process descriptions (Fig. 1). Although a graphical notation for representing the reference process model is still the subject of research, this exercise helps to identify common points and variations between the pilots. The model seems like a waterfall model but in practice it is a combination of the incremental model and the V-Model. Once the requirements specification is ready, subsets of requirements are designed, implemented, and tested incrementally.

3.2 Requirements

3.2.1 Wireless Internet Services Characteristics

High Volatility of Requirements

Wireless Internet services will be deployed across heterogeneous networks, heterogeneous clients, and for all types of users. The business cases are not clear yet. The stakeholders (i.e., network operators, service providers, application providers, technology owners) are still researching possible business cases that promise revenues in this new growing market. All this uncertainty is reflected in high volatility of the requirements. This phenomenon was also seen in the Internet domain. A flexible and agile mechanism to specify requirements early or late in development is needed.

Usability

creating a prototype for user analysis. The prototype consists of user interfaces that will be discussed with the users and then implemented according to the feedback. In a heterogeneous market like wireless Internet services, this approach could be of great help, because it forces development organizations to consider a wider spectrum of possible profiles.

Combinations of the previously mentioned approaches could be applied depending on the context of the software development organization. For instance, a small software development organization has fewer resources for looking at the requirements using the market division than a large organization. Certainly organizations that want to enter the wireless Internet services domain must know that one real concern and possible factor of success is the usability of the application.

Regarding device independence, a good starting point for clarifying the concepts is given by [24]. It is a survey that presents a classification of available technologies, and their relationship with device independency in the context of wireless Internet services. For example, device attributes like output, input, processor, memory, multimedia objects, application language, or browser language influence the degree of independency of an application. Devices receive content as multimedia objects, application languages or browser languages. Depending on the underlying hardware, devices are able to use different types of content. Therefore, in order to achieve device independency, the content must be sent in a compatible format for a given device.

There are technologies that can be used to adapt the content or application according to the device capabilities. Content adaption can be done in the server, proxy or client browser. Some examples of these technologies are: HTTP request header files, CC/PP composite capability preferences profile, WAP UAPROF, SyncML, and Universal plug and play.

The mentioned survey is a good reference for understanding how each of the mentioned technologies can help when trying to deliver a device independent application. The bad news is that at the moment, there is no dominant/unique standard, therefore, choosing a specific technology can imply high risks.

In order to mitigate this risk, the W3C consortium is working on an initiative focusing on device independency and standardization. The idea is that web content and applications are accessible anyhow and anytime. Anytime refers to many access mechanisms (i.e., heterogeneous clients that can provide access anytime), and anyhow refers to many modes of use (i.e., audio, voice, touch, among others). One product of this effort is the Device Independency Principles document [32]. At the moment, the principles are general, but they will be specialized with guidelines and requirements to obtain device independency as well as to concentrate all standardization efforts in one place.

Use cases are proposed by [6], for describing the functionality of the application. This is a more traditional approach where, after some interviews with the customer, the developer describes the functionality of the application. No recommendations are given in order to address usability, device independence, or high volatility of requirements, even though the model proposes accepting requirements late in development.

3.2.3 Experience

In the observed projects, the requirements were written down in special meetings with the customer. The marketing people informally communicated usability requirements. Informal meetings

were held between developers and marketing personnel, and based on existing knowledge, discussions about best probable usability experience were held. Open questions usually triggered a feasibility study. In the feasibility study, prototypes were developed with the purpose of testing the actual technology (i.e., devices and networks), and understanding its real capability.

The projects used similar templates to write the requirements. Use cases were used for describing, at a high level, the functions of the service. Client and server functional requirements refined use cases, and they are specified as informal text. Performance and usability requirements were specified as nonfunctional requirements.

In order to refine technical requirements and decide on the technology to be used, a feasibility study was needed. The feasibility study consisted of building a prototype in order to test network and mobile devices features. During the feasibility study, different communications protocols were tested to verify the data exchange rate. HTTP turned out to support too slow of a data rate, therefore, UDP was chosen as the communication protocol.

3.3 Design

3.3.1 Wireless Internet Services Characteristics

Scalability

Usually, if an Internet site is successful, high consumption of primary systems resources (CPU, memory, file system bandwidth, and network bandwidth) is expected. Wireless Internet services will run on top of Internet sites. The same behavior is thus expected for successful wireless Internet services. Therefore, wireless Internet services sites should be designed to be scalable.

Seamless Mobile Services

Aiming at seamless/transparent mobile services requires mechanisms to hide heterogeneous and changing contexts.

Wireless Internet services can be provided on weakly interconnected low-speed networks such as GSM or high-speed networks such as UMTS [27]. As the user moves, changes from a faster network to a slower network are obvious. Mobility of the terminals has consequences on the product model (a component to follow and track is needed in the architecture). Thus, while designing a service for mobile devices, which requires data exchange over the network, the current location of the user, and the bandwidth available on the wireless network, are two issues to be considered, among others.

Charging and Billing Models

The evolution of mobile networks and the transformation of mobile devices into Internet terminals have created the need for a new billing and charging model, where the subscribers are charged not only for the time and quality of the telephone usage, but also for the Internet services they use (i.e., email, Internet browsing, multimedia messages). The need for very well defined components, interfaces and communication mechanisms between components is important in order to create charging and billing mechanisms capable of supporting actual or future business models.

Usability

As already mentioned in the requirements phase, physical attributes of mobile devices, especially, are a challenge for creating usable applications, but the structural design of a wireless Internet service site, which determines how users will navigate within it,

is also an important activity. The system should be designed to allow the users to find what they are looking for quickly. The complexity of this task depends on the size of the site and its relationships with other sites.

Device Independence

A discussion was already introduced on this subject in the requirements section. Findings oriented towards designing for device independence are presented in this section.

3.3.2 Practices

A survey performed within 25 organizations in the UK [26] revealed that formalized design web techniques like hierarchy charts, site flow charts, and storyboards were used in the web domain. Hierarchy charts were used to relate web pages of a site. Site flow charts sketched the decisions to reach certain functionality, and storyboards contain the sequence of web pages that a user will encounter within a web site. These techniques were used basically to design the navigation of the structure.

Some of the studied companies had developed website layout standards for using video, animation, graphics, colors, and navigational standards such as where to place the back button, and the use of banners and menus. Standards for designing web site content were found, as, for example, the use of specific keywords.

One of the major conclusions of the study is that few organizations have guidelines or standards for web site development and ad hoc practices are dominant. A similar scenario can be expected in the wireless Internet services domain. Ad hoc approaches can lead the development of complex and unmaintainable sites. Structured techniques/guidelines for design are necessary sooner or later.

Examples of the use of structured techniques are given by [6], [27], [8], and [2]. They have in common the use of object-oriented principles to design static and dynamic views of a wireless Internet service application. Patterns like the MODEL-VIEW-CONTROLLER are recommended for use in wireless Internet service applications by [27], where the logic is concentrated on the server and none or a minimum of the business logic is revealed on the client side. The use of this pattern can have additional benefits such as: All the components are defined logically, each component has a function, interfaces are defined between components, each component can be implemented as another pattern, high reusability, high flexibility, reduced cost, and higher quality. Other techniques use patterns to customize the role of a user, and the structure, behavior and links of a page [8].

Regarding device independence, Giannetti [13] provides the Device Independence Web Application Framework (*DIWAF*). The framework is based on the “single authoring” principle, which consists of designing for the most capable device and automatically adapting content to different device classes. Content, layout and style are separated for reuse whenever possible.

ScalableWeb is a technique presented by [5] that allows authors to build a device-independent presentation model at design time. *ScalableWeb* is also based on the single authoring technique, where authors can produce the layout specification for the largest screen size of a given device, and then a rendering system renders the device presentation model into device-specific presentations [39].

Mori et al. [15] present an *XML-based approach* oriented to design applications that are device independent. The tool TERESA (Transformation Environment for Interactive Systems Representations)

provides a semiautomatic environment supporting the presented method and transformations.

One commonality between *ScalableWeb* and the *XML-Based approach* is the specification of a task model as input for creating the device presentation model, or the abstract user interface, that are later transformed into the device specifics with the help of an automatic/semiautomatic tool.

All of the approaches agree on the need of a high level description of content, style and interaction that allows adaptation.

At the moment, topics of research in the device independency area are, for example, how to balance independency with the usability of the application. One application may appear as expected in the devices, but the usability experience might not be as satisfying for all of them. What are the steps to create abstract, general presentation models? Should the generation of the specific presentation models be totally automatic, or just partially, so the authors could manipulate it?

Regarding scalability, [4] introduces what they call “Scalability design process” based on a set of strategies useful when designing scalable Internet sites. The strategies are based on the design principles of a scalable architecture: divide and conquer, asynchrony, encapsulation, concurrency, and parsimony. The paper contributes with a set of guidelines for system partitioning, i.e., dividing the system into components with a well-defined interface and functionality. The message is clear. Successful wireless Internet services need to be scalable, but scalability demands a detailed architecture, a detailed design, and, once implemented, requires monitoring and maintenance.

In order to build seamless mobile services, Friday et al. [2] propose techniques that can be used to adapt the system and improve the quality of service of the network (QoS) at different levels (i.e., user, application, middleware, and transport). For example, the system can allow the user to change from synchronous to asynchronous tasks (user level), or through proxy services the application can use local substitute services based on cache information (application level). At the middleware level the information can be fetched only when needed (on demand), and finally at the transport level, data can be prioritized, reordered, and exchanged according to the bandwidth situation. The adaptation techniques were validated through the development of a mobile collaborative system. One of the final conclusions of the study was that mobile systems must have the support of adaptation techniques at all levels, in order to be effective, but architecture to propagate QoS information through the system is still required.

A discussion of billing infrastructure and charging models for the actual and future Internet, and how they could be modified for being used in wireless Internet services, is presented in [33]. One interesting example is the Paris-Metro charging model proposed by [34]. This model supposes that the subscriber defines a travel class as an association between cost and network traffic. For example, the subscriber could define that he will use the network in first class or second class according to the association network traffic-cost. The network could also detect that the first class is full (i.e., high traffic), therefore all the subscribers who want to use the network will have to use only the second class. If the subscriber would like to use the first class for a given service, then he will have to pay the correspondent penalty. According to [33] this model introduces complexity to the network behavior, overhead to the subscriber, and what is most important for developers, changes to the software application and extensions to the commu-

nication protocols. Therefore, developers should ask themselves during the conception of the application's design how much the model of charging and billing impacts the system's architecture.

The usage-centered engineering [22] approach presented in the requirements phase that addresses user interface usability does continue in design. Designers must produce a role model, a task model, and an abstract model. The role model groups the common characteristics of user interaction with the system in roles. These characteristics are related to the purpose, duration, attitude toward the system, and information exchange between the user and the system. A task model is a set of task cases and their relationships. A task case lists the steps of the system to provide the desired functionality without assumptions about the user interface. Finally, the abstract model describes the user interface with interaction contexts. The abstract model does not contain details about the look and behavior of the user interface. Designers use these models to create a comprehensive user interface.

Nerurkar [30] suggests merging the GUI methodologies used for designing traditional systems with the new Web design techniques [41], in order to improve Web design methodologies. Nerurkar defends the fact that the essentials of user-centered interface can be applied for Web interface design.

More specific guidelines for designing user interfaces can be found for devices or families of devices. The big mobile device producers or programming platform providers offer them, for example, the *MIDP style guide* offered by Sun Microsystems, Inc [45].

The previous approaches gave guidelines for producing usable sites, but how could that be measured? The *Card Sorts* technique is proposed by [23] for eliciting quality measures of web pages. It is a technique based on a personal construct theory, whose objective is to elicit and ensure the validity of a measure for a fuzzy attribute like *quality* in a new field such as the Internet. It provides a systematic way to elicit quality measures that the stakeholders consider important. In a new domain like wireless Internet, this can be of great help, because it minimizes the suppositions about the stakeholder's usability preferences.

3.3.3. Experience

The observed projects performed both high-level and low-level design. The high-level design is represented in the conceptual level of abstraction, and the low-level design in the concrete level of abstraction. Additionally, for each level of abstraction, four viewpoints are represented: structural, behavior, deployment and development. The structural viewpoint covers the concerns related to the composition of information and architectural components, whereas the behavior viewpoint considers the dynamic aspects of the architecture. The deployment viewpoint shows the allocation of architectural components to physical nodes of computing and network environments. The development viewpoint shows work organization and choice of technologies mapped to services and components.

The notation used for describing the design was UML. Regarding usability, developers recommend to design a user interface avoiding graphical controls that can only be used by a mouse device, or touch screen. That is, design the application thinking that it has to be suitable for the most limited and less limited device. Also, design for different input types as touch screens, keyboards, and mouse.

The postmortem analysis of the observed projects showed that the

effort spent in the design phase was larger than that spent in the other phases. After interviews with the developers, it was found that one of the reasons was that designing usable interfaces required many attempts due to the heterogeneity of devices.

Although technical features like scalability, navigability, QoS, and billing models, were addressed in an ad-hoc manner, developers envision that they could be achieved by creating components, or agents responsible for such issues. Therefore, a parallel project was dedicated to designing a service management component, which provides services for authentication, authorization, user profile management, provisioning (restricted to self-subscription and client code deployment), and billing (restricted to accounting and mediation). Another project was given the task to design and implement an agent that negotiates the preferences of a client with the limitations of the server to provide the service. Network bandwidth limitations were addressed since the very beginning of the project. Developers designed trying to optimize the data exchange rate, and to reduce the amount of data sent over the network to the minimum.

3.4 Implementation

3.4.1 Wireless Internet Services Characteristics

Programming languages and protocols

There is a diversity of platforms, programming languages, and protocols already available for developing mobile applications and Internet services [27], [24]. Software developers have to answer questions like: Is WML suitable for implementing a wireless Internet game? If yes, which are the preconditions, or technical requirements? If not, what is the best suitable technology?

Technical constraints: Power, memory storage, and security

It is not enough to have a good design in order to optimize resources. It is also important to look at programming techniques, programming standards, or tools for tuning and optimizing the produced code.

Flexible communication channels between developers

Lack of knowledge and experience must be addressed in software development organizations with communications mechanisms that allow developers to resolve problems quickly and efficiently. The experience of developers or the will to work collaboratively is not enough.

3.4.2 Practices

There are plenty of Internet services for providing financial, weather, or sport information to clients, i.e., services that require little user interaction. These services can in theory be deployed in wireless Devices using the Wireless Application Protocol (WAP). At the moment the WAP 1.x and 2.0 standards are available [43]. WAP 1.x uses the Wireless Markup Language (WML) for document formatting. WML is a language similar to HTML, specially designed for small clients with small screens and low bandwidth. A web site developed with HTML does not need a complete architecture rework of the service in order to be translated into WML, but maintenance can be a heavy duty because every modification to the desktop version should also be made in the mobile version.

WAP 2.0 uses the extensible hypertext markup language (XHTML) for formatting the document. In theory, WAP2.0 allows developers to create richer applications that handle multimedia and animation, among other features. XHTML can be displayed by almost all available browsers, but not all HTML features

can be converted into XHTML.

cHTML is the content development language for i-mode (NTT Docomo's Wireless Service). cHTML is also similar to HTML, but is optimized for wireless networks and devices.

Wireless Internet services that demand interaction of the users, like games, need a more flexible programming platform. Today some wireless devices can be programmed using some sort of C-like language, but C is not a cross-platform language and therefore, portability among different hardware architectures is lost. A dedicated client should be deployed for every possible platform, slowing down time-to-market of the service and increasing costs. JAVA, on the other hand, is commonly used because of its portability. Especially after the release of J2ME, Java can now be deployed on many wireless devices providing a common ground for developers.

Regarding device resources, although J2ME has been optimized, it still demands considerable capacity from the processor. In order to optimize the use of the device power, tips, guidelines and techniques can be found in the J2ME/WAP developer discussion groups. One example can be found in [44]. Memory storage is another constraint that tends to improve with technology evolution. Techniques used to reduce the size of compiled code to be stored in the device memory are welcome. Obfuscation, for example, is a technique to protect software and optimize its execution that can be useful [50].

Security for WAP applications can be assured through the Wireless Transport Layer Security (WTLS). In the case of J2ME applications, power and memory constraints make security a challenge.

Pair programming can be seen as difficult to implement in an industrial context, but in a new domain such as the wireless Internet, it could perhaps have a good effect in order to resolve new and uncertain problems thanks to its person-to-person communication mechanism. Pair programming is an extreme programming technique where two developers produce code in one machine [35]. One person concentrates on the strategy to produce the code and the other on whether the approach could work, how it could be simplified, and has the control of the computer.

Zettel et al. [19] propose the *LIPE* process model to develop e-commerce services on time to market, based on other extreme programming techniques like *refactoring*, *realize scenario*, and *rework code*.

Although extreme programming techniques seem to be flexible enough, there are some drawbacks. In the case of *refactoring*, small teams formed by great developers can be successful [36]. But what if the developers are not so experienced? It has been concluded by [41] that agile practices are not intended for larger teams. Although success cases for larger teams have been presented [42], this assumption must be considered especially in the wireless Internet Services Domain, where projects can become large and complex.

3.4.3 Experience

The observed projects confirm that WML has important limitations concerning presentation. The look and feel of WML sites is expected to be very simple and not sophisticated. In the case of J2ME, some devices showed performance problems when the graphics were stored on the client. A proposal for solving this problem is to store the graphics on the server and upload them to the device on demand. This implies the use of caching algorithms,

which optimize performance. Additionally, developers obfuscate the code in order to optimize the storage of bytecode in memory. Although obfuscation is a technique for protecting the java code from reengineering, obfuscators can reduce the size of bytecode by renaming classes, methods and variables in a shorter string. The observed projects assured security through the mentioned service management component. It is part of the plans to secure the communication channel between the server and the mobile device when using J2ME.

3.5 Test

3.5.1 Wireless Internet Services Characteristics

Testing in a realistic environment

A considerable amount of effort is required in order to prepare realistic conditions for a wireless Internet service to be tested. Consider a service provider that wants to test its new weather information service. The service should run on laptops, mobile phones and PDAs. Additionally, the service provides wheather location related information. This scenario already implies important assumptions to be made while testing the application. Different devices, different wireless networks, and different user profiles. There are also many context factors that influence on the results of the test, e.g., network traffic, user location, and mobility. Recreating scenarios with typical situations requires effort and money for setting up the appropriate hardware and software.

3.5.2 Practice

Finding mechanisms or tools for simulation or emulation of scenarios can help to solve the problem and reduce costs for setting up a realistic testing environment. Technology providers provide emulators for many mobile devices on the market. Emulators approximate the functionality of the real device. In practice, using network emulators can be of great help for creating tests, collecting the results, and repeating them under different network conditions.

One distributed network emulator system, EMPOWER [46], provides a mechanism to emulate the mobility of a wireless network in a wireline network. The preliminary results of emulating node mobility of wireless networks using EMPOWER are encouraging [44]. EMPOWER allows the user to define packet latency and bandwidth as parameters and test a given topology wireless network. There are plans to include a physical layer simulator in order to improve the emulator capabilities of wireless networks.

A description of the Model Based Testing technique, and its application for testing a Pocket PC application can be found in [47]. The technique uses finite state machines and directed graphs or state transition diagrams as a basis for testing the functionality of the application. Benefits of model-based testing are the possibility to automate, and the fact that the structure of states and transitions is written, which gives a general understanding to all team members on how the application should work. It is still a topic of research to find out if model-based testing is suitable for finding faults, due to the fact that the effort invested by developers on building the model is not depreciable.

Simulation can also be done to test characteristics like usability. One visualization-based approach to improve the usability of a web site, and a predictive model to locate problem areas, is introduced by [9]. The approach underlines the importance of using visualization techniques to understand the behavior of users on a web site and to identify unreachable places. Visualization tech-

niques can also be used to analyze the past behavior of a site, and to understand the impact of new changes.

3.5.3 Experience

The developers who participated in the projects performed their unit tests using emulators for WAP and J2ME. One fact noted after testing was that the performance of emulators is still far from the performance of the real device. Developers had to program new code in order to replace functionality provided by the emulator but inexistent in the real device. Thus, it was suggested that future projects should try to introduce the use of real devices early in the development process.

The assumption at the beginning of the projects that setting up a real test of the application, i.e., the right device(s), server, and network, was a complex task led to the inclusion of an activity in the process for this purpose only. The fact is that the effort for setting up the testing environment of a wireless Internet service is considerable, and should not be underestimated.

3.6 Managerial Processes

3.6.1 Wireless Internet Services Characteristics

Small increment planning

As already mentioned, the incremental development model is suitable for domains where high volatility of requirements, lack of experience and time to market are success factors. What is the best increment to be delivered? What is a realistic time interval for each increment? How to select a consistent set of requirements for the increment? These are questions that have already been addressed in the area of requirements engineering and applied in areas like Internet or Mobile. Regarding wireless Internet services, a software development organization can have more than one role, for example as application provider, service provider, and integrator of services. This characteristic demands vision and discipline for releasing the product, and impacts the procedure to determine the best suitable or most profitable increment.

3.6.2 Practices

One planning method for increments called *Construction Planning* is presented by [11]. It is a nine-step process used for the development of radio systems, which allows project managers to model and plan the functionality of increments, track their evolution, and update the project plan. The method uses as basis good and bad increments planning experiences from real projects. *Construction Planning* helps to have control of increments, and receive constant feedback from the customer on the quality of the products.

A planned release or increment also determines which customer will get special features and what will be the quality in a given point of time.

A tool for understanding the nature of planning releases was developed by [48], and tested in three case studies. Based on the assumption that cost and value were the most important factors for deciding what is a release, developers were allowed to assign cost and value to each requirement, and to prioritize them. The tool would find a set of possible releases, which were reviewed and validated by the developers. One important statement is the definition of the release planning activity as a *wicked* problem [49]. A wicked problem is a problem that stops when there is no more time, no money or the solution is good enough. It is a prob-

lem with no optimal solution, unique and irrepeteable, therefore no measures of success are possible. This was confirmed in the case studies. It was seen that the number of variables influencing the definition of a suitable release could be very large, and that every release was influenced by new variables, or the context changed the values of the existing ones. Software organizations could benefit from this area of research as it addresses one key activity for the development of wireless Internet services.

The usability of mobile devices as a medium to deploy WAP applications has been criticized because of their physical limitations by [52], and [53]. Their usability test and studies reveal that bigger and more capable user displays improve user interfaces, and therefore acceptance by users. An interesting study done by [51] was focused on detecting usability problems through testing, and afterwards improving the application. In this study, the users tested alternatives of a user interface with the same functionality. Users were gathered, and performed the same activities within a wireless Internet service. Meanwhile the time and number of interactions were measured. Finally, users were interviewed about their experiences and suggestions. The measures and information were taken as basis for deciding which was the most suitable interface and for creating guidelines to be used in future projects.

3.6.3 Experience

As already mentioned, the observed projects follow an incremental development model. The set of requirements to be delivered in every cycle was selected based on the following criteria: available technology, available resources, and system architecture.

Initially, the design and experience of developers were the most important factors for taking the decision about the most appropriated release. But this changed as technology problems were discovered, forcing extra development of new libraries. The releases were replanned, and resources reallocated, keeping consistency with the architecture. The problem was worked around, but the technology still does not provide the desired support.

3.7 Summary of Practices

Table 3 presents the results of the literature search. A result should be read in the table as follows: Source **Hammar [25]** taken from the **Internet Services** domain contributes with a **Process**, to be used during **Requirements, Design, Implementation**, and enhances **time to market, quality and usability**.

A list of possible values for each column is given in the following:

Domain: These are the related domains where the literature was found, like mobile, agile, telematic, wireless, Internet, and Wireless Internet.

Contribution: For example, guidelines, a technique, methods, processes, models, roles, and tools.

Phase: These are the spots in the process where the contribution are presumed to be helpful for the development of wireless Internet services

Possible success factors: Based on justifications found in literature [20], [2], [16] and on the objectives of the observed projects, four possible success factors were used: Time-to-market, interface usability, device independence, and quality. Contributions found in literature are presumed to add value to the software development process by enhancing the previous possible success factors.

4 SUMMARY AND OUTLOOK

This article has presented an initial reference process for developing wireless Internet applications that comprises applicable technologies and experiences and is tailored for the specifics of the domain. The lack of knowledge about wireless technologies, the unavoidable growth of this type of applications in the coming years, and the need of a systematic approach for developing these applications are important reasons to justify the creation of such a reference process. This process has been descriptively elicited in a systematic way through the development of pilot projects and literature study.

The reference process model presented in this article does not significantly differ from traditional iterative process models on the life cycle level, but it includes domain-specific and experience-based guidance on the level of engineering processes. At the moment it is difficult to provide more concrete decision support for project planning due to inexperience and to the recentness of the domain, but recording experience is considered to be part of the future work of the authors in order to understand the complexity of the domain.

ACKNOWLEDGMENT

The work has been funded by the European Commission in the context of the WISE project (No. IST-2000-30028). We would like to thank the WISE consortium, especially the coordinator Maurizio Morisio, for the fruitful cooperation. We would like to thank Sonnhild Namingha from the Fraunhofer Institute for Experimental Software Engineering (IESE) for reviewing the first version of the article. Additionally, we would like to thank Fabio Bella from IESE for his cooperation in the WISE project and the anonymous reviewers for their helpful comments on this article.

REFERENCES

- [1] Cockburn, A.: *Agile Software Development*: Addison-Wesley Pub Co; ISBN: 0201699699; 1st edition (2001).
- [2] Friday, A., Davies, N., Blair, G.S., Cheverest, K.W.J.: *Developing Adaptive Applications: The MOST Experience*. *Integrated Computer Aided Engineering*: ICAE, vol 6, No 2, pp. 143-158 (1999).
- [3] Nilsson, A., Anselmsson, M., Olsson, K., Johansson, Erik.: *Impacts of Measurement on an SPI Program*. Q-Labs (http://www.q-labs.com/files/Papers/SPI99_Imp_of_Meas_on_SPI.pdf).
- [4] Roe, C., Gonik, S.: *Server-Side Design Principles for Scalable Internet Systems*. *IEEE Software*, vol.19, No. 2, pp. 34-41 (2002).
- [5] Wong, C., Chu, H., Katagiri M.: *A Single Authoring Technique for Building Device Independent Presentations*. In: *W3C Device Independent Authoring Techniques Workshop*, (2002).
- [6] Adamopoulos, D.X., Pavlou, G., Papandreou, C.A.: *An Integrated and Systematic Approach for the Development of Telematic Services in Heterogeneous Distributed Platforms*. *Computer Communications*, vol. 24, pp. 294-315 (2001).
- [7] Raffo, D., Kaltio, T., Partridge, D., Phalp, K., Ramil, J.F.: *Empirical Studies Applied to Software Process Models*. In: *International Journal on Empirical Software Engineering*, vol. 4, No. 4 (1999).
- [8] Schwabe, D., Mattos, G.R., Rossi, G.: *Cohesive Design of Personalized Web Applications*. *IEEE Internet Computing* vol. 6, No 2, pp 34-43 (2002).
- [9] Chi, E.: *Improving Web Usability Through Visualization*. *IEEE Internet Computing*, vol. 6, No 2, pp. 64-71 (2002).
- [10] Karlsson, E., Taxen, L.: *Incremental Development for AXE 10*. *ACM SIGSOFT Software Engineering Notes*, vol. 22, No. 6 (1997).
- [11] Karlsson, E.: *A Construction Planning Process*. Q-Labs, LD/QLS 96:0381, Lund Sweden (1999).
- [12] Brooks, F.P. Jr.: *The Mythical Man-Month*. *Essays on Software Engineering*, Anniversary edition. Addison Wesley. Reading MA (1995).
- [13] Giannetti, F.: *Device Independency Web Application Framework*. In: *W3C Device Independent Authoring Techniques Workshop*, 25-26 (2002).
- [14] McGarry, F., Pajerski, R., Page, G., Waligora, S., Basili, V.R., Zelkowitz, M.V.: *An Overview of the Software Engineering Laboratory*. *Software Engineering Laboratory Series Report, SEL-94-005*, Greenbelt MD USA (1994).
- [15] Mori, G., Paterno, F., Santono, C.: *An XML Based Approach for Designing Nomadic Applications*. In: *W3C Device Independent Authoring Techniques Workshop*, (2002).
- [16] Omidyar, G.: *Internet Services over Mobile and Wireless Networks Architectures and Protocols*. In: *Conference Proceedings of the First European Conference on Universal Multiservice Networks*, pp. 1-4, France (2000).
- [17] Cooper, H.M.: *Scientific Guidelines for Conducting Integrative Research Views*. *Review of Educational Research*, vol. 52, No. 2, pp.291-302.
- [18] Rombach, H.D., Verlage, M.: *Directions in Software Process Research*. *Advances in Computers*, vol. 41, pp. 1-63 (1995).
- [19] Zettel, J., Maurer, M., Münch, J., Wong, L.: *LIPE: A Lightweight Process for E-Business Startup Companies based on Extreme Programming*. *Proceedings of the Third International Conference on Product-Focused Software Processes Improvement (PROFES)*, pp. 255-270, (2001).
- [20] Basu, K., Hee, Lee.: *Challenge of Universal Mobility and Wireless Internet*. *Conference Records of the 2000 IEEE Wireless Communications and Networking Conference*, pp. 1563-1568 (2000).
- [21] Beck, K.: *Extreme Programming Explained: Embrace Change*. Addison Wesley, (2000).
- [22] Constantine, L.L., Lockwood, A.D.L.: *Usage-Centered Engineering for Web Applications*. *IEEE Software*, vol. 19, No. 2, pp.42-50 (2002).
- [23] Upchurch, L., Rugg, G., Kitchenham, B.: *Using Card Sorts to Elicit Web Page Quality Attributes*. *IEEE Software*, vol. 18, No. 4, pp. 84-89 (2002).
- [24] Buttler, M.H.: *Current Technologies for Device Independence*. HP Laboratories, HP-2001-83, Bristol., (2001).
- [25] Hammar, C.M.: *Designing User-Centered Web Applications in Web Time*. *IEEE Software*, vol. 18, No. 1, pp. 62-69 (2001).
- [26] Taylor, M.J., McWilliam, J., Forsyth, H., Wade, S.: *Methodologies and Website Development: A Survey of Practice*. *Information and Software Technology*, vol. 44, No. 6, pp. 381-391 (2002).
- [27] Kovari, P., Acker, V.B., Marino, A., Ryan, J., Tang, L.K., Weiss, C.: *Mobile Applications with Websphere Everyplace Access Design and Development*. IBM SG24-6259-00 (2001).
- [28] Becker-Kornstaedt, U., Hamann, D., Kempkens, R., Rösch, P., Verlage, M., Webby, R., Zettel, J.: *Support for the Process Engineer: The Spearmint Approach to Software Process Definition and Process Guidance*. *Proceedings of the Eleventh Conference on Advanced Information Systems Engineering (CAISE '99)*, pp. 119-133. *Lecture Notes in Computer Science*, Springer-Verlag. Berlin Heidelberg New York (1999).
- [29] Becker-Kornstaedt, U., Boggio, D., Muench, J., Ocampo, A., Palladino, Gino.: *Empirically Driven Design of Software Development Processes for Wireless Internet Services*. *Proceedings of the Fourth International Conference on Product-Focused Software Processes Improvement (PROFES)*, (2002).
- [30] Nerurkar, U.: *Web User Interface Design: Forgotten Lessons*. *IEEE Software*, vol. 18, No. 6, pp. 69-71 (2001).
- [31] Yau, V.: *Project Management Strategies and Practices for Wireless CDMA Software Development*. *Proceedings of the IEEE International Conference on Industrial Technology*, (1996).
- [32] <http://www.w3.org/TR/2001/WD-di-princ-20010918/>
- [33] Cushnie, J., Hutchison, D., Oliver, H.: *Evolution of Charging and Billing Models for GSM and Future Mobile Internet Services*. *Lecture Notes in Computer Science*, vol. 1922, pp. 312-323 (2000).
- [34] Odlyzko, A.: *Paris Metro Pricing: The Minimalist Differentiated Services Solution*. AT&T Laboratories Research, April 1999.
- [35] Maurer, F., Martel, S.: *Rapid Development for Web-Based Applications*. *IEEE Internet Computing*, vol 6, No 1, pp. 86-90 (2002).
- [36] Boehm, B.W.: *A Spiral Model for Software Development and Enhancement*, *IEEE Computer*, vol 21, No 5, pp. 61-72 (1988).
- [37] Boehm, B.W.: *Get Ready for Agile Methods, with Care*, *IEEE Computer*, vol 35, No 1, pp. 64-69 (2002).
- [38] McDermid, J.A., Rook, P.: *Software Development Process Models*, *Software Engineer's Reference Book*, Ed., Boca Raton, FL: CRC Press, pp. 15.26 – 15.28. (1994).
- [39] Wong, C., Chu, H., Katagiri, M.: *GUI Migration Across Heterogeneous Java Profiles*. *Proceedings of ACM SIGCHI-NZ'02*. (2002).

[40] Rosenfeld, L., Morville, P.: Information Architecture for the World Wide Web, O'Reilly, New York, (1998)

[41] Highsmith, J., Cockburn, A.: Agile Software Development: The Business of Innovation, Computer, vol 34, No 9, pp. 120-122. (2001).

[42] Cockburn, A., Highsmith, J.: Agile Software Development: The People Factor, Computer, vol 34, No 11, pp. 131-133. (2001).

[43] Read, K., Maurer, F.: Developing Mobile Wireless Applications. IEEE Internet Computing, vol 7, No 1, pp. 81-86. (2003).

[44] <http://wireless.java.sun.com/midp/tips/appsize/>

[45] <http://java.sun.com/j2me/docs/alt-html/midp-style-guide7/preface.html>

[46] Zheng, P, Ni, M.N: EMPOWER: A Network Emulator for Wireless and Wireline Networks. In Proceedings of IEEE INFOCOM (2003).

[47] El-Far, I.K., Thompson, H.H., Mottay, F.E.: Experiences in Testing Pocket PC Applications. In Proceedings of the Fifth International Software and Internet Quality Week Europe, (2002).

[48] Carlshamare, P.: Release Planning in Market-Driven Software Product Development: Provoking and Understanding. Requirements Engineering, No. 7, pp. 139-151, (2002).

[49] Ritel, H., Webber, M.: Planning problems are wicked problems. In: Cross N (ed). Developments in Design Methodology. Wiley, Chichester, pp 135-144, (1984).

[50] Colberg, C.S., Thomborson, C.: Watermarking, Tamperproofing and Obfuscation- Tools for Software Protection. IEEE Transactions on Software Engineering, vol. 28, No. 8, pp. 735-746, (2002).

[51] Buchanan, G., Farrant, S., Jones, M., Thimbleby, H., Marsden, G., Paz-zani, M.J.: Improving Mobile Internet Usability. In proceedings World Wide Web 10, pp. 673-680, (2001).

[52] Nielsen, J.: Graceful Degradation of Scalable Internet Services, WAP: Wrong Approach to Portability, Alertbox 31/10/1999 at <http://www.useit.com/alertbox/991031.html>

[53] Nielsen, J.: WAP Backlash, Alertbox 09/07/2000 at <http://www.useit.com/alertbox/20000709.html>

[54] Brinkkemper, S., Harmsen, F., Oei, H.: Configuration of Situational Process Models: An Information Systems Engineering Perspective. In W. Schaefer (ed.) Proceedings of the European Workshop on Software Process Technology (EWSPT'95), LNCS 913. Berlin: Springer-Verlag, 193-196. (1995)

[55] Highsmith, J.: What is Agile Software Development? The Journal of Defense Software Engineering. October, (2002).

[56] DSDM Consortium: Dynamic Systems Development Method. Version 3, <http://www.dsdm.org>.

tute SFB 501 "Development of Large Systems with Generic Methods" funded by the German Research Foundation (DFG). Dr. Münch's research interests in software engineering include: (1) modeling and measurement of software processes and resulting products, (2) software quality assurance and control, (3) technology evaluation through experimental means and simulation, (4) generic methods for the development of large systems, (5) technology transfer methods. He has been teaching and training in both university and industry environments, and also has significant R&D project management experience. Jürgen Münch is a member of the IEEE Computer Society and the German Computer Society (GI).

Daniela Boggio received her MS in Electronical Engineering (specialization: Telecommunications Devices) from the Politecnico of Turin, Italy, in December 1998. MS Thesis: "Receiver Design for the CDMA IS-95 Mobile Radio System". She developed her MS thesis at the École Nationale Supérieure des Télécommunications in Paris, thanks to a six months grant in the frame of the ERASMUS project. She joined Motorola Technology Center in August 1999 as research engineer in the area of analytical availability/performance modelling. Since July 2000 she is working as Project Leader in the development of services and applications for the 2.5G/3G. She was responsible for the organization and management of the Engineering Induction Training Program held at MTCL in May 2000 and October 2000. Other work experiences: from March 1999 till July 1999 she was with ALENIA AEROSPAZIO, Aeronautics Division, in the Design Department, Communications Team.

Gino Paladino received his undergraduate degree in Electronic Engineering in 1987. From 1989 to 1999 he worked at ITALTEL, an Italy voice and data telecommunication network manufacturer, (www.italtel.com) as hardware designer then as software developer network designer. He joined Investnet in the beginning of 2000, as developers team coordinator.

Alexis Ocampo received his MSc degree in Computer Science from Los Andes University, Colombia, in 1999 and his title as Systems Engineer from Industrial University of Santander, Colombia, in 1997. Since February 2002, Alexis Ocampo has been a research scientist at the Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, in the department of Quality and Process Engineering. Before that, he worked for 5 years as a research-developer on new technologies and methodologies with the software company Heinsohn Associates, Bogota, Colombia. His master thesis entitled "Implementation of PSP in the Colombian Industry: A case study" was developed within this company. He also worked as an instructor at the University of Los Andes in the Department of Systems and Computation. Alexis Ocampo's research interests in software engineering include: (1) modeling and measurement of software processes and resulting products, (2) software quality assurance and control, (3) technology transfer methods.

Jürgen Münch received his PhD degree (Dr. rer. nat.) in Computer Science from the University of Kaiserslautern, Germany. Dr. Münch is Department Head and Competence Manager for Quality and Process Engineering at the Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern. Since November 2001, Dr. Münch has been an executive board member of the temporary research insti-

TABLE 3
LITERATURE SEARCH RESULTS

Source	taken from the domain	contributes with a	to be used during					and enhances
			R	D	I	T	M	
Hammar [25]	Internet Services	Process	x	x	x			Time to market, usability, quality

Constantine et.al. [22]	Internet Services	Process	x	x	x			Usability, quality
Beck [21]	Agile processes	Process	x		x	x		Time to market
Cockburn [1]	Agile processes	Process	x		x	x		Time to market
Adamopoulos et.al. [6]	Telematic Services	Process	x	x	x	x		Quality
Taylor et.al. [26]	Internet Services	Processes		x				Time to market
Kovari et.al. [27]	Wireless Internet Services	Techniques		x	x			Usability, quality
Schwabe et.al. [8]	Internet Services	Process		x				Time to market, usability, quality
Friday et.al [2]	Mobile Services	Techniques		x				Usability, quality
Cushnie et.al. [33]	Wireless Internet Services	Desing Model		x				Usability, quality
Maurer et.al. [35]	Internet Services	Process	x		x	x		Time to market
Nerurkar [30]	Internet Services	Process		x				Usability, quality
Rosenfeld et.al. [41]	Internet Services	Techniques		x				Usability, quality
Sun Microsystems	Wireless Internet Services	Guidelines		x				Usability, quality
Upchurch et.al. [23]	Internet Services	Process		x		x		Usability
Buttler [24]	Wireless Internet Services	Technologies		x	x			Device independence, time to market
W3C [32]	Internet Services	Guidelines		x				Device independence, time to market
Gianneti [13]	Internet Services	Process		x				Device independence, time to market
Wong et.al. [5]	Internet Services	Technique		x				Device independence, time to market
Wong et.al. [39]	Internet Services	Tool			x	x		Device independence, time to market
Mori et.al. [15]	Internet Services	Technique		x	x			Device independence, time to market
Roe et.al. [4]	Internet Services	Technique		x				Quality, time to market
Zettel et.al. [19]	Internet Services	Process	x		x	x	x	Time to market, usability, quality
Sun Microsystems [44]	Wireless Internet Services	Guidelines			x			Quality
Zheng et.al. [46]	Wireless Internet Services	Tool				x		Quality
El-Far et.al. [47]	Wireless Internet Services	Process				x		Quality
Chi [9]	Wireless Internet Services	Tool				x		Quality
Karlsson [11]	Mobile Services	Process					x	Time to market, quality
Carlshamare [48]	Market driven domains	Process					x	Time to market, quality

R = Requirements, D = Design, I = Implementation, T = Testing, M = Managment