Developing Small Web-Based Systems

CARLOS J. COSTA (*) MANUELA APARICIO (**),
(*) carlos.costa@iscte.pt Information Science and Technology Department, ISCTE, Lisboa, PORTUGAL
(**)manuela@lusocredito.com, Lusocredito, Lisboa, PORTUGAL

Abstract: In this paper it is proposed a website engineering process. The proposed process is largely supported in the concept of usability. But it incorporates concepts of software engineering, specifically simplified. This simplification results from the need of applying to small site development. The approach was applied to a site used by an accounting team.

Key words: software engineering, information systems development, application development, web engineering

1 Introduction

The development of small web based systems, like the development of other software, presents some difficulties and challenges. In order to tackle those challenges, either researchers or practitioners have proposed development methodologies. First, those approaches began structure each one of the phases of the development process. Then, it was demonstrated that more sophisticated approaches should have been used, either incorporating the user contribution in different steps of development (like in prototyping) or even integrating them tightly in the development process (like in joint application development or participatory design). But, the sophistication of development process is also related to the incorporation of development techniques and languages. In this context, it is important to identify not only programming techniques and languages, and its corresponding paradigms (structured languages, object-oriented languages), but also the use of design techniques and languages, often incorporating the same paradigms of the languages.

In the following section, some of the concepts here identified are explored and situated in the literature. Then, it is proposed an approach, and then this approach is illustrated in an example. Some of the issues related to the application of the approach are then discussed.

2 The Software Engineering Process

Programmers began by noticing that in the development of complex information systems, they could divide the process in several phases. In this context, several models appeared more or less ad-hoc, with the purpose of surpassing the resulting problems of the little importance attributed to the development process [5].

A great step was given when in the seventies the **model waterfall** was disclosed [9] [2]. In this model, the several phases are clearly defined, constituting tight blocks, in which the results of a phase will correspond to the inputs of the following phase. It is in the context of this model that was disclosed the succession of the following steps in the information system development process: conception of the system, analysis, design, coding, implementation and maintenance.

In each phases the information systems development process, team must define the corresponding objectives and final products.

Among the several models that constitute development of the waterfall it is of highlighting the **V-model**. It is an approach that evidences the top-down design and the bottom up implementation.

The problems of the waterfall took the one that if they developed methods that supplied faster results, needing less initial information at the same time that had larger flexibility. With the iterative development, the project is divided in smaller components. This allows that development teams demonstrate results earlier and that could obtain feedback of the users. Frequently, each one of the iterations is in fact a mini waterfall process.

With the **iterative approach**, the development team can avoid great projects and have the certainty that something with value is given to the users in relatively small intervals of time. For example, as soon as the report of requirements is produced, the design may be initiated and programming, test and implementation may be done perhaps in less than two weeks.

Very frequently, it is difficult to know all the requirements of the system in the beginning. Typically the users know which are many of the objectives that they want to see solved with the system but they do not know which the nuances nor every detail of the system and possible capacities. The use of prototypes comes to help significantly. In fact, when a **prototyping model** [21] is used, development team builds a simplified version of the proposed system and introduces it to the user so that they criticize it. When giving its feedback, the user will contribute to incorporate in the system the proposed readjustments. Sometimes, the prototype is laid out as soon as the development team identifies

requirements, being developed new programs. Typically, the phases of the prototyping model consist in defining and picking up of requirements (identical but fewer deepened than in the waterfall), designing, creating or modifying the prototype, evaluation, refinement of the prototype and implementation of the system.

The prototyping consists of the creation of a model that works, instead of creating the drawing in the paper. In fact, when confronted with the drawing in the paper (screen layouts, structured language, flowchart, data flow diagrams), the users can look at the design of the project and not noticing the real impact of all those outlines.

The prototyping can be used basically at three levels: design of the interface with the user, performance design and determination of the functional requirements.

The prototyping is used in most of the cases as helping way of conception of the interface with the user. This process is costlier than the traditional method of conception of the layout of the screen in the paper, however it is also much more effective.

The second more frequent use of the prototyping is the modeling of the performance. The modeling of the performance is accomplished in a phase more assault of the development cycle than the conception of the interface with the user, consisting of the accomplishment of a model to evaluate the performance of the system. This use of the prototyping was especially useful before the beginning of the use the relational model in databases. With the use of data bases little alterations to the implementation remains. On the other hand, with the use of machines with powerful processing capacities, the evaluation of the performance of the system started to have smaller importance. This prototyping type is still important for situations in that the performance is reputed of great importance.

The functional prototyping is more recent than the other types. This application of the prototyping is concerned with the iterative development of the specification. This approach becomes difficult to distinguish from the iterative development. The fundamental difference results from the fact that iterative development consists in production of a piece of the whole system while the prototype is developed with the purpose of not using the prototype in the end. This way the code developed for the prototypes doesn't present the same quality. However, this can be discussed, and in a lot of situations the prototypes can be incorporate in the final system.

Gib [11]suggests that successful large systems started out as successful small systems that growth incrementally. Based in this premises, it was proposed the incremental approach. An **incremental approach** performs some initial analysis to scope the problem and identify major requirements. Those requirements that will deliver most benefits to the client are selected to be the focus of a first increment of development and delivery. The installation of each increment provides feedback to development team and informs the development of subsequent increments.

The **spiral model** [1]was created to include the best characteristics of the waterfall and prototyping, at the same time that introduces a new component of evaluation of the risk. This model can be viewed as supporting incremental delivery, although Gilb [11]arguing that spiral model does not fully support his view of incremental development, as there are aspects of systems development that does not emphasize or include.

Intimately related with the iterative processes is the participants' involvement in the development process. It is in this picture that new approaches appeared:

- Joint applications development (JAD),
- Participatory development (PD)
- Socio-technical systems theory (STS).

IBM originally presented JAD in the ends of the seventies, having received attention of the industry some years later. The JAD was related to the BSP (Business Systems Planning) IBM Methodology. This methodology emphasizes structure and agenda. Everything is explained in detail: "to do" lists are included, as are masters of useful forms [27]. While JAD has as goal to improve system, PD main goal is the improvement of workplace. Consequently, criteria for validation are also different: while JAD uses quantitative criteria like performance index or time saving, PD has qualitative criteria based in democratic concepts, like mutual learning, mutual education or conflict resolution [17], [6].

A variant of the prototyping model is RAD or **rapid application development** [19]. RAD introduces very narrow limits of time in each development phase, leaning on strong tools of fast development, what allows more efficiency.

Some researchers support that user involvement is especially important in prototyping development, especially in CSCW applications [13].

Some researcher identified two types of system development: a customer development environment and a package development environment. The first one has as goal the software development for internal use and usually not for sale. The package development has as goal software development for external use and typically for sale. Customer participation may be possible in both perspectives but in order to make it possible it is necessary to establish customer development links. As customer developer links can be

used: support lines, survey, user interface prototyping, requirements prototyping, or interviews [16].
The basic premises of the **reuse model** [27] are that the system should be constituted using existent components, in opposition to the systems based on new components. Booch [3] suggests an iterative incremental approach to object-oriented software development. It is in this scope that models supported in objects are incorporated. During the 90s, the advantage of objects orientation paradigm became obvious for many software engineering academics; they started to develop methodologies that integrated techniques and languages supported in this paradigm. But the large number of methodologies that had been proposed reminded the negative experience that had already occurred with the structured methodologies. Consequently, the idea of creating a unified methodology seemed to be the most adequate to the case.
During the 1980 and 1990 countless proposals had appeared. These had been called object-guided methodologies (Table 1).

Methodology	Paradigm
SSADM[26]	Structured
Yourdon System Method [27]	Structured
Engineer Information[19]	Structured
Stradis- Structured Analysis, Design and Implementation of Information System[10]	Structured
Booch[3]	Object Oriented
OMT-Object Modelling Technique[24]	Object Oriented
OOSE - Object Oriented Software Engineering: [15]	Object Oriented
OOAD - Object Oriented Analysis and Design: [7]	Object Oriented
RUP [23]	Object Oriented

Table 1– Methodologies used in the information systems development

With the enormous growth of web software development, several methodologies were tested. But, generally did not deal adequately. Consequently, new methodologies for the development in the web appeared. Those methodologies were either supported in the structured paradigm or supported in the objects oriented paradigm.

Methodology	Paradigm
RMM Relationship Management Mode [14]	Structured
ERMIA – Entity Relationship Modelling for Information Artefact[12]s	Structured
RUP - Rational Unified Process[23]	Object Oriented
OOHDM – Object-Oriented Hypermedia Design Method[25]	Object Oriented

Table 2– Methodologies used in the WEB development

Other methodologies or methods for development appear as an alternative more adjusted to the web environment. Some of those approaches are listed in the Table 2, but others may complete this list like [20], [8] or [18],
More recently, some authors consider the Usability as a methodology of site construction with rose potential of success [22][4].

3 Proposal

It is proposed here an approach based in [4], to witch it was added another phase: maintenance. According to theses authors, the process of web site development may be decomposed in the following phases: requirement analysis, conceptual design, prototype and mockup, production, launch and maintenance.
The approach identified here also incorporates some of the design techniques related to the UML.

Phases	User Intervention	Product
1. Requirements analysis		
- Needs analysis	Survey	Report
- Use case	Interview	UC diagram
- Use case description	Interview	UC description
2. Conceptual Design		
- Use case model	Users	UC Diagram
- Sequence model	Users	Seq. Diagram
- Navigation diagram	Users	Navigation diagram
- Class diagram	Users	Class Diagram
3. Prototype and mockup		
- Screen prototypes	User testing/ Participatory Design	Prototype
4. Production		
- Writing	User testing/ Participatory design	SW
- Multimedia files	User testing/ Participatory design	SW
- Screen design	User testing/ Participatory design	
- Software production (Business layer)	Interview/ Participatory design	SW
- Database development	Users	SW
5. Launch		
- Communication Plan		
6. Maintenance		

Table 3– Phases, user intervention and products of the proposed approach.

The phases and activities identified in the last table may form a process, as it is presented in the following figure.

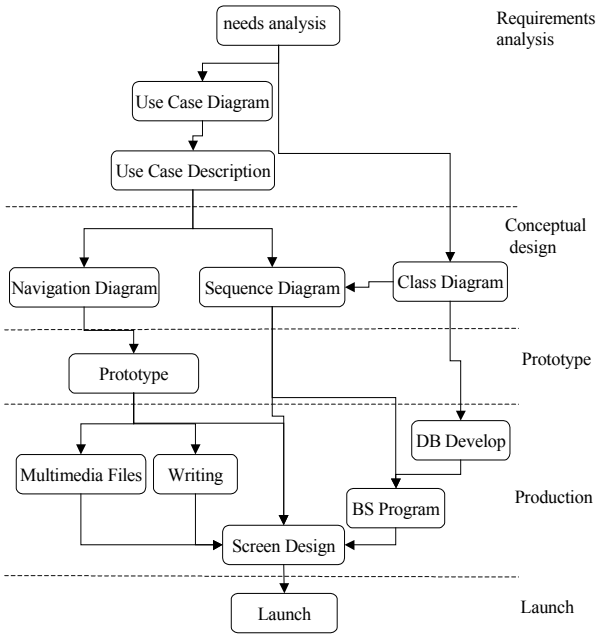

Figure 1 - Small web based systems development process

4 Case

I. Requirements

Business (but also any Organization) is a going concern. Accounting theory must be applied in those organizations that are open and dynamic systems. Those systems are not based in natural or in immutable rules. It changes, adapts and develops according to changes in economic, financial and cash environment. For this reason, in accounting a set of rules should be followed:

- Expenses for a period should be matched against related income, rather than cash payments compared to cash receipts
- Revenue and costs are noted when they are incurred and not when cash is received or paid,
- Present accounts are drawn up following the same principles as the previous accounts,
- Revenue or costs are only recorded if it is certain that they will be incurred.

For a much clear illustration of this case it is defined here what are the basic tasks of accountancy functions:
Accountant creates accounts and records transactions. He may also prints profit and loss statement, balance sheet, extract of accounts and prints accounts.
Creating an account is a process that includes titling and grouping in the correspondent class, linking the new account with other accounts that are related with.
Recording a transaction is the task that corresponds to the registry of an economic, financial or cash change. In other words, most of the transactions are based on documents, those documents cab be internal or external. Summarizing, a transaction consists of registering facts that are related to the everyday life of a business. Printing extracts, balance sheet or profit and loss statement is having an outcome of the processed information, on with analysis is required so that decisions can be taken.
Double-entry bookkeeping is a method of recording the transactions of a business in a set of accounts. It is a system of bookkeeping where both debit and credit entries are recorded in the accounts at the same time (e.g., as a sale is credited to the sales account the purchaser's debt is debited to the debtors account).

II. Conceptual design

In the conceptual design phase, the functionalities of the application are worked out. The navigation diagram allows identifying either the Windows Menu either the Web pages.

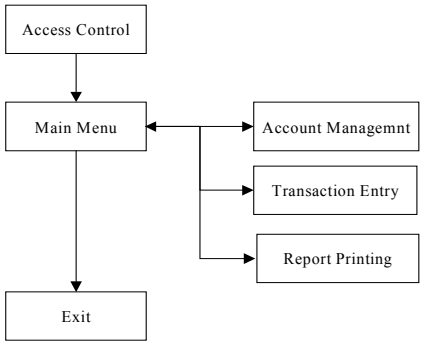

Figure 2 - Navigation Model

The following classes were identified: User, Account T, Account E, Transaction and Entry.

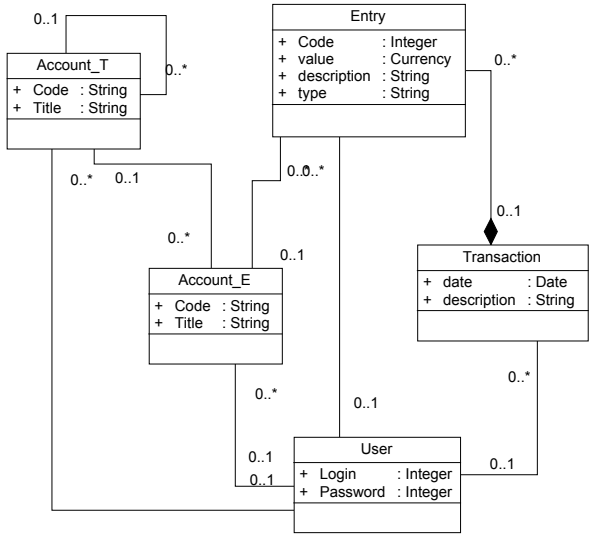

Figure 3 - Class Diagram

User – Users are the users that have permission of using application. The data held about users is its login and password.

Account T – "Account T" may be decomposed in "Account T" or in "Account E" but cannot be used in entry. Each account is composed of Accounts Entry.
Account E - An account is a named segment of a ledger recording transactions relevant to the person or matter named.
Transaction – A transaction is composed of a set of Entries. Each transaction must have a set of entries that have a debit value equals to the value in credit.
Entry – An entry corresponds to the recording of each account in a transaction.

User (Login, Password)
Account_T (Code, *Acc_Code*, *Login*, Title)
Account_E (Code, Title, *Login*, *Code_S*)
Transaction (TransactionID, Date, Description, *Login*)
Entry (Code, *TransactionID*, *Acc_Code*, Value, Description, Type)

III. Prototype

The development of the prototype was performed using tools employed to produce mock-up, like Visio. In order to evaluate some of the functionalities, it was also developed a prototype with Microsoft Access.

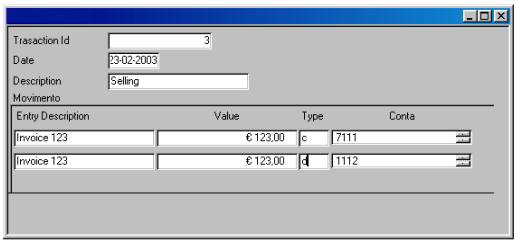

Figure 4 - Mock-up for the Windows

IV. Production

The production phase includes the choice of programming languages. For example, in order to produce Webepages, the following languages may be used: Webpages: Javascript, Flash, Java for the Applets. The use of CSS, forms, forms or tables it is also decided in this phase. On the other hand, the use of server-side languages must be analyzed. The use of Perl, PHP, ASP or Java depends on the platform used and also the knowledge of the development team.
In this case, it was decided to use Perl to produce CGI.
In order to perform the database creation, it was used a relational database. In the following lines, it is possible to see the SQL corresponding to the tables that were created.

```
create table User (
Login                INTEGER           not null,
Password             INTEGER,
primary key (Login)
);
create table Account_T (
Code                 CHAR(254)         not null,
Acc_Code             CHAR(254),
Login                INTEGER,
Title                CHAR(254),
primary key (Code),
foreign key (Login)
      references User (Login),
foreign key (Acc_Code)
      references Account_T (Code)
);
create table Transaction (
Login                INTEGER,
date                 DATE,
description          CHAR(254),
transationId         CHAR(254)         not null,
primary key (transationId),
foreign key (Login)
      references User (Login)
);
create table Account_E (
Code                 CHAR(254)         not null,
Login                INTEGER,
Title                CHAR(254),
Code_S               CHAR(254),
primary key (Code),
foreign key (Code)
      references Account_T (Code),
foreign key (Login)
      references User (Login)
);
create table Entry (
Login                INTEGER,
Acc_Code             CHAR(254),
TransationId         CHAR(254),
Code                 INTEGER,
Value                REAL,
Description          CHAR(254),
Type                 CHAR(254),
User                 CHAR(254),
primary key (Code)
foreign key (transationId)
      references Transaction (transationId),
foreign key (Acc_Code)
      references Account_E (Code),
foreign key (Login)
      references User (Login)
);
```

SQL was also used to develop the reports. In the following lines, it is possible to see the code that was used for list accounts.

```
SELECT Account_E.Code, Account_E.Title,
(SELECT Sum(Value)
from Entry
where type="d" and Entry.code= Account_E.code) AS Debit,
(SELECT Sum(Value)
from Entry
where type="c" and Entry.code=Account_E.code) AS Credit,
FROM Account_E, Entry
WHERE Entry.code = Account_E.code
GROUP BY Account_E.Code, Account_E.Title
```

V. Launch and Maintenance

The form of advertising the website depends on the target audience. In this case, the target audience is a very narrow group of users.

In opposition, maintenance is a very important process. In fact, it is important to clearly identify how maintenance may be performed in order to allow a clear segregation between development and maintenance and updating and preservation of accounting sensitive information.

5 Discussion

The system is being implemented, but some of the advantages of the process adopted are already clear:

- Using a simple and clear process is something that people evolved considered adequate;
- The involvement of users in the development process by using tools like MS-Access is an adequate form of contributing to a more accurate requirement analysis.
- In this particular situation, the launch phase does not seam to be important.

The identification of the class diagram was a tricky process. In fact, the analysis process was difficult as long as the accounting system has some difficulties that are not easily understood by system engineers.

6 Conclusion

In this paper it is proposed a website engineering process, mostly supported in the concept of usability. It also incorporates concepts of software engineering, specifically simplified in order to answer to the need of applying to small site development. The approach was applied to a site used by an accounting team. From the application of the approach several advantages were identified, like the advantage of using a clear process, and the possibility of cooperation between users and specialists in prototype development.

References

[1] Boehm, B. "A spiral model of software development and enhacment"; Computer; May 1988.

[2] Boehm, B. Software engineering Economics; prentice-Hall; Englewood Cliffs; NJ; 1981

[3] Booch, G. Object-Oriented Analysis and Design with Applications, 2nd. Edition. Addison Wesley. 1994

[4] Brinck, T., Gergle, D. & Wood, S. Usability for the WEB - designing WEBsites that works. San Francisco, Morgan Kaufmann Publishers. 2002.

[5] Canning, R.G.; Electonic Data Processing for Business and Industry; John Wiley & Sons, NY; 1956.

[6] Carmel, E.; Whitaker, R. & George, J. ; "PD and Joint application Design: a transatlantic Comparision"; Communication of the ACM, June 1993; Vol. 36; No.4.

[7] Coad, P. & Yourdon, E.. Object-Oriented Analysis, 2ª edição. Yourdon Press, 1991.

[8] Diniz, E. & Vieira, C., Sugestões para a Criação de Documentos WEB – Visando a Usabilidade. Anais/ WorkShop sobre Fatores Humanos em Sistemas Computacionais. Florianópolis, UFSC, SBC, 2001.

[9] Enger, N. "Classical and Structured Systems Life Cycle Phases and Documentation" in Cotterman, W., Conger, J, Enger, N. Harold, F. (Ed) System Analysis and Design: A foundation to th 1980s; Elsevier North Holland; NY; 1981.

[10] Gane, T & Sarson, C. Structured Systems Analysis. McDonnell Douglas, 1982.

[11] Gilb, T. Principles of Software Enegeneering Management, Vokingham, Addison-Wesly, 1988.

[12] Green, T. & Benyon, D., The skull beneth the skin: entity-relationship models of information artefacts, International Journal of Human-Computer Studies, 44, 6 pp. 801-828, 1996.

[13] Gronbaek, K.; King, M. & Mogenen, P.; "CSCW Challenges: Cooperative Design in Engineering projects" Communications of the ACM; Jun 1993; Vol. 36; No. 4; pp 67-77.

[14] Isakowitz, T., Stohr, E. & Balasubramanian, P. "RMM: A Methodology for Structured Hypermedia Design"', Communications of the ACM, Vol. 38, No. 8, August 1995, pp. 34-44.

[15] Jacobson, I. Object-Oriented Software Engineering: A Use Case Approach. Addison Wesley, 1992.

[16] Keil, M & Carmel, E.; "Customer-Development Links in Software Development"; Communications of the ACM; May 1995; Vol. 38; No 5; pp. 33-44.

[17] Kensing, F.& Munk-Madsen, A; "PD: Structure in the toolbox"; Communication of the ACM; jJun 1993; Vol. 36; No. 4; pp78-84.

[18] Lynch, P. & Horton, S.. WEB Style Guide. Yale University School of Medicine's Center for Advanced Instructional Media (CAIM). 1997.

[19] Martin, J.; Rapid Application Development; MacMillan; NY; 1991.

[20] Martins, C., Price, A. & Pimenta, M.. Sistematização do desenvolvimento de aplicações hipermídia na WEB: comparação de metodologias. Anais/VII Simpósio Brasileiro de Sistemas Multimídia e Hipermídia. Florianópolis, UFSC, SBC, 2001.

[21] Naumann, J. & Jenkins A. "Prototyping: the new paradigm for systems development", MIS Quaterly; 6 – 3; September 1982, pp. 29-44.

[22] Nelson, J. Designing WEB Usability: The Practice of Simplicity. Indianapolis, New Riders Publishing, 2000.

[23] Rational. Rational Unified Process, Best Practices for Software Development Teams. A Whitepaper, 1998.

[24] Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy & W. Lorenzen. Object-Oriented Modeling and Design. Prentice Hall, 1991.

[25] Schwabe, D. & Rossi, G. "An Object Oriented Approach to WEB-Based Application Design", Theory and Practice of Object Systems Vol. 4, No. 4. Wiley and Sons, New York. 1988.

[26] Weaver, Ph., Lambrou, N & Walkley M. Practical SSADM Version 4+, 2ª edição. Financial Times Prentice Hall, 1998.

[27] Woods, J. & Silver D.; Joint Application Design: How to Design Quality Systems in 40% Less Time; Wiley, NY; 1989.

[28] Yourdon, E.; object oriented systems design: an integrated approach; Prentice-Hall International edition; 1994.