Saint-Petersburg Russia

Microsoft Technologies in Theory and Practice of Programming

Contest – Conference

Project Metrics

Project:	Web-SynDic
Customer:	Department of Computer Science, Head of the Department Yury A. Bogoyavlensky Petrozavodsk State University
Developers:	Dmitry G. Korzun (senior manager, PhD) Kirill A. Kulakov (technical manager, master student, BSc) Andrey Y. Salo (senior student) Andrey A. Ananin (junior student) Mikhail A. Kryshen (junior student)
Product:	Web System for Demonstrating and Testing the Syntactic Algorithms for Solving Linear Diophantine Equations in Nonnegative Integers

Contents

1 Introduction

The Web-SynDic project[1] is a student software engineering (SE) project of the Petrozavodsk State University (PetrSU), Department of Computer Science (CSDept).

The project is related to the research done at CSDept of PetrSU in development of a new type of algorithms for efficient solving some classes of nonnegative linear Diophantine equations (NLDE) by syntactic (parsing) methods [15, 16, 17]. These syntactic algorithms seem to be promising tool for solving some classes of NLDE system; more exactly a class of NLDE system, associated with formal grammars (ANLDE systems). For this class the syntactic algorithms allow efficient (polynomial and pseudo-polynomial) computations comparing with the general NLDE case when the same computational problems are NP-complete or even overNP [18].

The general goal of the project is to develop a full function web system[2] for visual demonstrating and testing the syntactic algorithms via the Internet. This allows researchers to input ANLDE systems (manually or automatically generated), search their Hilbert bases, test the correctness of the found solution, estimate the resource consumption, and compare the efficiency with available solvers, different from syntactic.

1.1 Document overview

In this document the key metrics of the Web-SynDic Project are introduced. They make a summary quantitative view on the project size and its other characteristics.

The project schedule is presented in section 2. This presents overall division of the project process into phases and the corresponding time resources.

Human resource is described in section 3. This presents how much human-hours were spent for the project.

Section 4 gives the size values of the developed software. Standard metrics in LOC and Kb are used.

According to SE standards the project produced several important documents. The size of documentation in pages is given in section 5.

The important testing metrics are presented in section 6. This include the division of the testing phase into subphases, number of executed tests and ratio of uncovered errors.

Section 7 describes the size of the project CVS repository, which stores all project-related data.

[1]The original, more detailed document set of Web-SynDic Project can be found at [1].

[2]The Web-SynDic Server is published at [3].

1.2 References

[1] Web-SynDic Project development and maintenance site.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/` (English)
`http://zeta.cs.karelia.ru/Web-SynDic/doc/rus/` (Russian)

[2] The Web-SynDic CVS repository, located at
`zeta.cs.karelia.ru/usr/local/cvsroot/Web-SynDic/`

[3] The primary Web-SynDic Server is installed at
`http://zeta.cs.karelia.ru:8080/Web-SynDic/main.jsp`

[4] Web-SynDic: Requirements Specification. The original version.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/requirements/`

[5] Web-SynDic: Design Specification. The original version.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/design/`

[6] Web-SynDic: Test Plan. The original version.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/testing/`

[7] Web-SynDic: Implementation Document. The original version.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/implementation/`

[8] Web-SynDic: Test Execution Document. The original version.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/test-exec/`

[9] Web-SynDic: User Guide. The original version.
`http://zeta.cs.karelia.ru/Web-SynDic/doc/eng/man/`
`http://zeta.cs.karelia.ru:8080/Web-SynDic/docs/UserGuide/index.html`

[10] Web-SynDic: Software Requirements Specification. Version for the Microsoft Conference, March 2004.

[11] Web-SynDic: High-Level Design Description. Version for the Microsoft Conference, March 2004.

[12] Web-SynDic: Test Execution Log. Version for the Microsoft Conference, March 2004.

[13] Roger S. Pressman. *Software Engineering. A Practitioner's Approach.* European adapt., 5th ed. McGraw-Hill, 2000. 915 p.

[14] Ian Sommerville. *Software Engineering.* 6th ed. Addison-Wesley, 2000.

[15] Yury A. Bogoyavlensky, Dmitry G. Korzun, *Obshchiy vid resheniya sistemy lineynih diophantovih uravneniy, associirovannoy s kontextno-svobodnoy grammatikoy.* Trudy Petrozavodskogo gosudarstvennogo universiteta. Ser. "Prikladnaya matematika i informatika". Vol. 6. Petrozavodsk, 1998. pp. 79–94. (in Russian)

[16] Dmitry G. Korzun. *Syntactic Algorithms for Solving Nonnegative Linear Diophantine Equations and their Application for Modelling of Internet Link Workload Structure.* PhD Thesis, Department of Computer Science, University of Petrozavodsk, 2002. 185 p. (in Russian)

[17] Dmitry G. Korzun. *Grammar-Based Algorithms for Solving Certain Classes of Nonnegative Linear Diophantine Systems.* Proceedings of Annual international Finnish Data Processing Week at the University of Petrozavodsk (FDPW'2000): Advances in Methods of Modern Information Technology. Vol. 3. Petrozavodsk, 2001, pp. 52–67.

[18] Schrijver A. *Theory of linear and integer programming.* Wiley, Chichester, 1986.

2 Project schedule

The waterfall model [13, 14] was chosen for the software process. The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. The first release of the required web system is planned be finished in December, 2003.

The waterfall model defines the following phases of the process:

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one). The main artifacts:

- Version 1.00 of the User Requirements [1, 10],
- Version 1.00 of the Project Plan [1].

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements. The main artifacts: Requirements Specification [4, 10]:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).

- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

2. Design (26.08–27.10.2003). The main artifacts:
Design Specification [5, 11]:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems design (25.10.2003, analysis & inspection 25-27.10).

Test plan & test cases design [6] (29.10.2003, analysis & inspection 29-31.10).

3. Unit implementation (30.10–06.12.2003). also including unit testing.
The main artifacts:

- Implementation document [7],
- Test execution document [8, 12].

4. Integration implementation (03.12–20.12.2003). also includes integration testing.
The main artifacts:

- **The Web-SynDic System [3].**

5. Validation testing (15.12–20.12.2003). See Test execution document [8, 12].

The first release (December, 2003). user manual completion [9], customer validation.

In 2004 the alpha- and beta testing phase is started. The alpha-testing consists of three stages. The beta-testing is testing of Web-SynDic in real Internet environent after publishing the system for public access. It starts at 12.01.2004 and will be completed at April–May 2004.

- Preparation to alpha-testing (12–15.01.2004).
- 1st stage of alpha testing with 3rd year students of Mathematical faculty (16-18.01.2004).
- Analysis of discovered errors and debugging, the next version of Web-SynDic. Start to form the to-do list (16.01–10.02.2004).

- 3rd stage of alpha testing: Microsoft conference in SPb (16.02–5.03.2004).
- Analysis of discovered errors and debugging, the next version of Web-SynDic. Start to form the to-do list (5–14.03.2004).
- 3nd stage of alpha-testing with 2nd&3rd year students of Mathematical faculty (10–20.03.2004)
- Analysis of the results of alpha-testing. Work according with the to-do list, the next version of Web-SynDic (March–April 2004).
- Publication of Web-SynDic, beta-testing (April–May 2004).

Figure 1 sums up the Web-SynDic project schedule as Gantt diagramm.

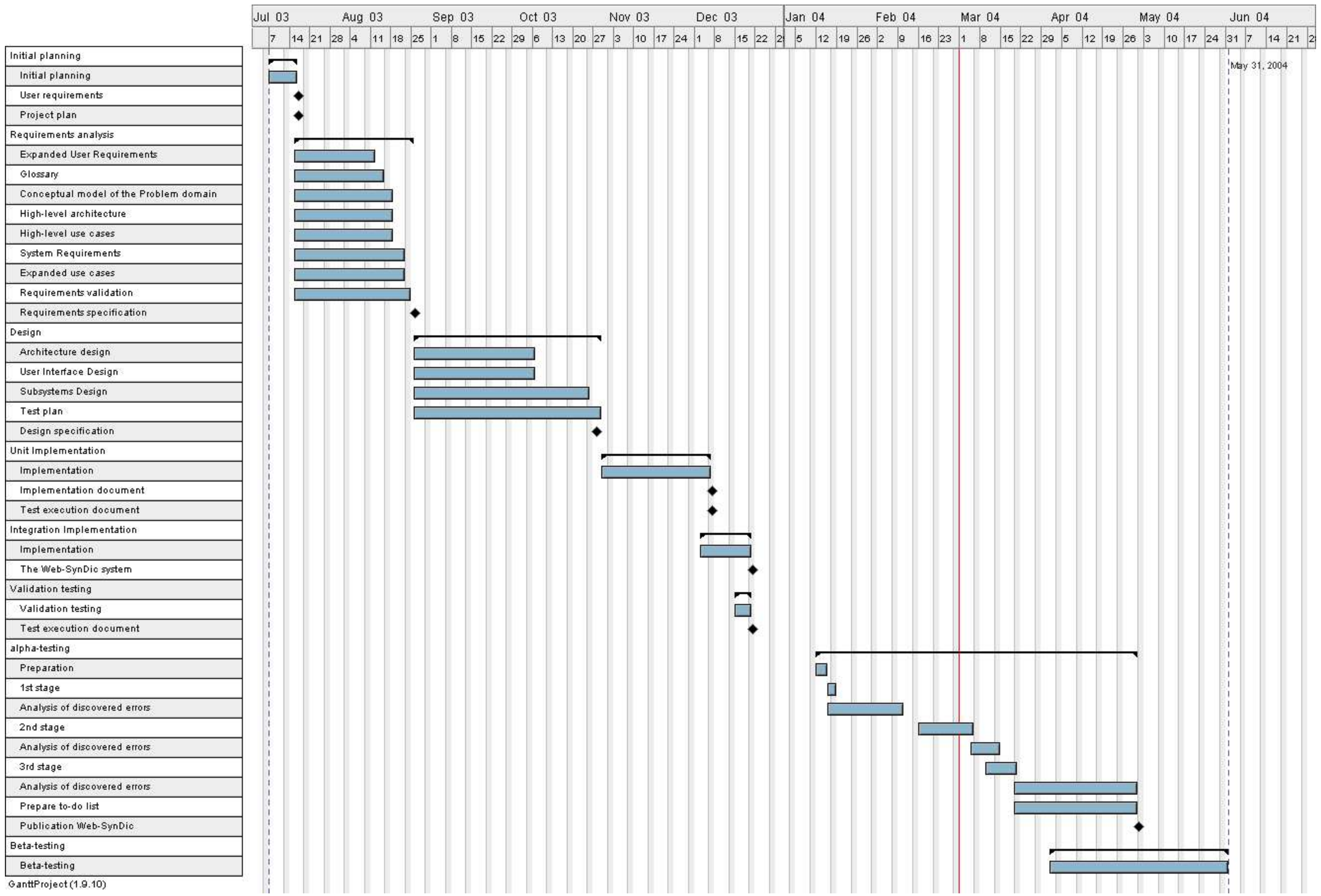

Figure 1: Gantt diagram of the project schedule

3 Human Resource

This section describes human resources (in working hours) spent for each process phase of the Web-SynDic project. The main phases are Requirements analysis, Design, Implementation, and Testing.

There were 6 developers at the beginning of the project. From November till recent time there are 4 developers.

The human resources are calculated using personal working hours of each developer [1]. The sum consumption of human resources in hours is shown in Figure 2. The testing phase does not include working hours spent by external testers for alpha-testing (students, researches, and other volunteers).

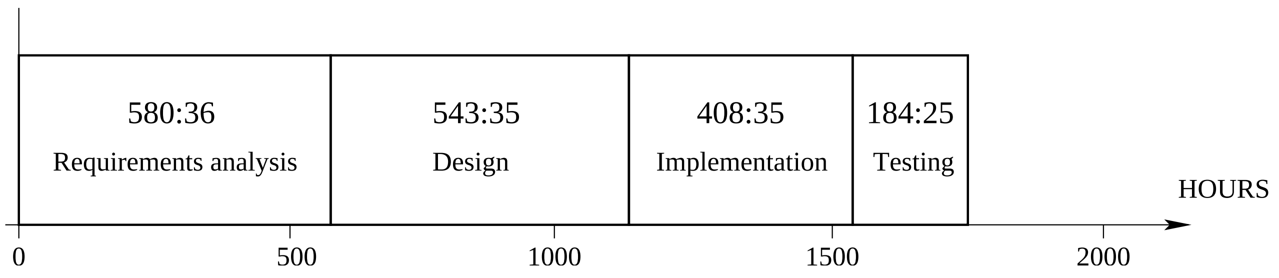

Figure 2: Working hours summary

4 Software Size

This section present size metrics of the produced software. Table 1 contains size values for each subsystem in lines of code and its relative contribution (%) as well as the total size of the whole software system.

Totally the software consists of 49 Java files or 195 Kb, 26 JSP files or 68 Kb.

In particular, this table shows that the most complex subsystems are the web and algorithm servers. Obviously, these subsystems are the most important for the application and their sizes just confirm this fact.

Table 1: Size of code and relative effort metrics for the Web-SynDic system

Subsystem	**Programming language**	**LOC**	**%**
Web-server and Session processing	Java + JSP	1800 + 2000	19 + 21
Algorithm server	Java	3600	38
Data store	Java	450	6
Management	Java	640	7
Statistics	Java	870	9
Total size		**9360**	**100%**

5 Documentation Size

The documentation size produced by the project and the corresponding time consumption are shown in Figure 3. The X-axis shows the time spent for the phases, the Y-axis corresponds to the amount of produced document pages during a phase.

Note that the documentation, produced with in the alpha-testing phase, is counted in the Test Execution document [8]. This computation does not include documents, produced for the Microsoft Contest–Conference.

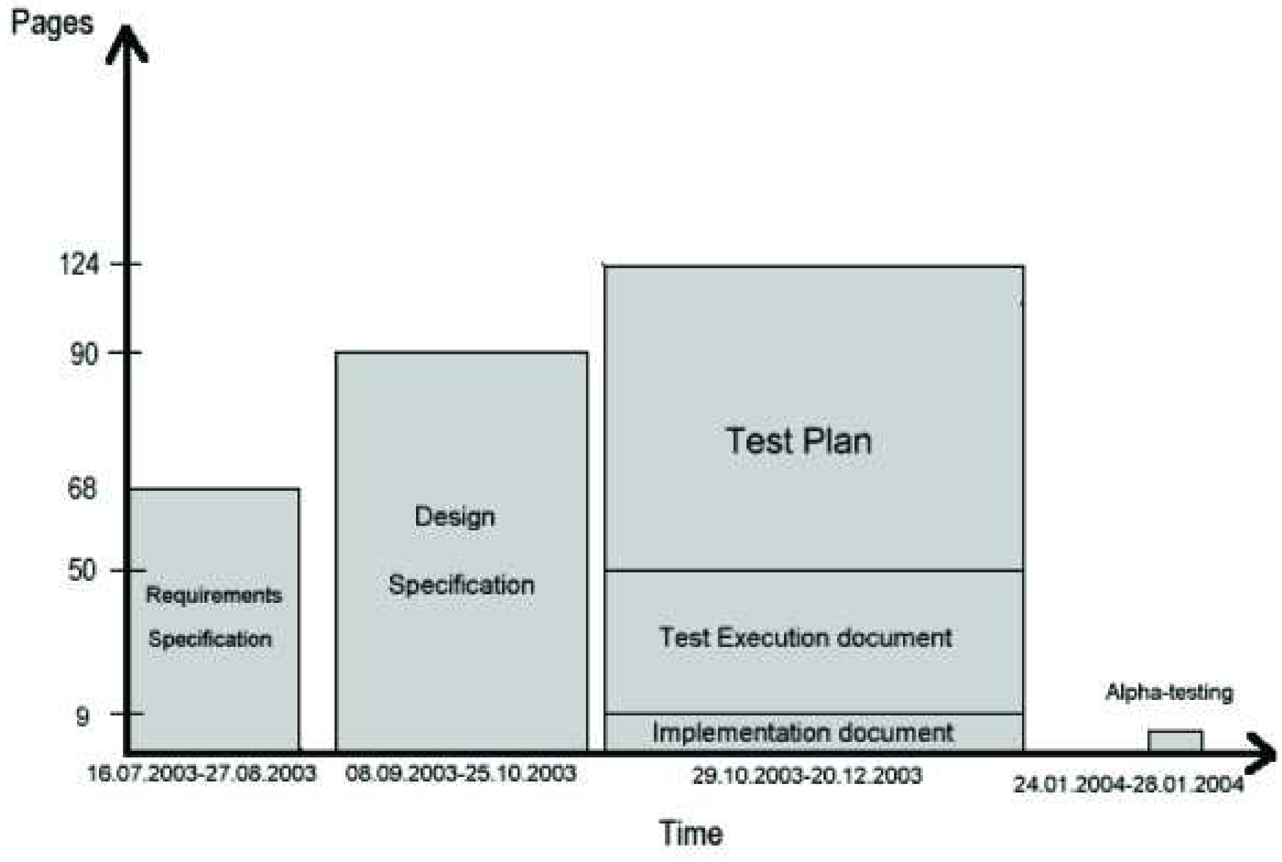

Figure 3: Metrics for documentation size

The Implementation Document [7] is very short. This means that there were a few changes to the design.

The Test Plan [6] is longer than the Test Execution Document [8] because the test plan describes test cases in detail yet the test execution only reports; majority of input/output test data among with supplementary code are located separtely from the test plan and test execution document.

6 Testing Size

In this section metrics for main testing subphases are introduced. This includes the number of executed tests and the number of found error/defects in the software. The metrics are shown in Table 2.

Table 2: Metrics for the testing phase

Testing subphase	**Number of tests**	**Errors found**	**Errors/Tests**
Unit testing	217	37	17%
Integration testing	117	25	21%
Total: unit & integration	334	62	19%
Alpha	58 students $\times$ 1.5 hours = 15 flaws found		

The low percent of uncovered error shows the design was developed sufficiently and the inspection process was perfomed on an appropriate level.

7 Project CVS Repository Size

The Control Version System (CVS) was used to store all data related to the project [2].

The total size of the repository is 43 Mb. This includes all versions of documentation (English and Russian branches), code (Java and JSP), executable external algorithms, test supplementary data (input, output, drivers, stubs), and CVS management data.