Measurement, Modelling and Management Approach to the Software Project Dynamics Optimisation

Vasyl Soloshchuk

PhD student, The Chair of Computer Mathematics and Mathematical Modelling, National Technical University “Kharkiv Polytechnic Institute”

Abstract: In this paper the approach to the software project dynamics research and optimisation is presented. This approach is based on the Measurement, Modelling and Management of the software development process (the MMM approach). We propose to collect the statistics data of such software project parameters as software product size, requirements size, effort, duration, staff number, costs, etc., in order to forecast these parameters for future projects and optimise the whole software development process and its management. The chief practical impact of this research work is the project management system development based on the MMM approach.

The research is supported by the IBM Faculty Awards program (2006-2007).

1 Introduction

The optimisation of the software development process is one of the main problems in the IT industry [Br95]. The connection of mathematical methods and industrial standards of management are of great importance for the optimisation of this process [AM91, FSS03, La04]. There are the following problems in the software project management [Sw04, Sm02, FSS03]: software projects parameters estimation (costs, duration, effort, etc.), lack of mathematical approaches which support the unified ways of requirements descriptions and estimation of its size, software projects parameters measurement and collection of statistics data during the project lifecycle. These problems could be solved by using the successful software development methodologies (RUP, MSF, XP, etc.), modern standards of the quality assurance (SEI CMM/CMMI, ISO), computer-aided tools in all the processes of a software project.

2 Approach

The main idea of our research consists of two parts. The first is connected with development of the mathematical model of the software project. The aim of the second part is to develop the Project Management System (PMS) on the basis of the proposed
mathematical model for the analysis and estimation of the software project parameters. The management cycle of the software project includes the following important processes: planning, controlling, analysis and estimation. Different tools in the market are used for the project planning and controlling (MS Project, GanttProject, JIRA, Bugzilla, Teamwork, Open Workbench, Primavera, etc.). There are some approaches to the modelling of the software development process and software project costs estimation based on the definite formal methods (Function Points Analysis, COCOMO, COCOMO II, and COCOMO II 2000 [Bo04]), but there is a lack of project management systems that uses these approaches in real practice for the analysing and estimating the parameters of the software project. The maturity dependence of standardized methodologies and computer-aided tools of software project management processes is shown in Fig. 1.

Lack of tools and practical methodologies for supporting the process

<table>
<thead>
<tr>
<th>Planning</th>
<th>Controlling</th>
<th>Analysis</th>
<th>Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition of tasks,</td>
<td>controlling duration,</td>
<td>statistics data collection</td>
<td>reuse of statistics data</td>
</tr>
<tr>
<td>roles, resources,</td>
<td>costs, effort, schedule</td>
<td>of software project</td>
<td>for the estimation of the</td>
</tr>
<tr>
<td>effort, etc.</td>
<td>progress, etc.</td>
<td>parameters from different</td>
<td>project parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sources of computer-aided</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tools</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1 Project management processes

Our approach is based on the three basic concepts:

- measurement of the software development process:
  - integration and adoption of different approaches for the software artefacts size measurement;
  - development of tools for the software development process measurement and for statistics data collection for the further analysis and modelling on the basis of mathematical models;
- modelling and simulation of the software development process:
  - development of the software project mathematical model. The RUP is used as the main methodology for the description of the software project lifecycle;
  - development of tools for the functional dependences identification of the software project parameters on the basis of the mathematical model;
  - development of tools for the software project simulation based on the identified parameters of the mathematical model for estimating effort, costs, duration, optimal staff number, etc.;
- management of the software development process:
  - development and integration of the PMS in the software development process.

It is planned that our PMS should include the following components:

- software project planner (work breakdown structure, Gantt diagram, resources, some other basic project management things);
• project metrics measurement (based on Function Points Analysis, project effort tracking, etc.; it will measure size of products, effort, staff number, costs, etc.);
• code measurement tools (it will use CVS as code repository);
• modeller for configuring software project development process (possibly some reuse of IBM Rational Business Modelling tools);
• software project analysis and simulator tool based on the mathematical model;
• estimator for planning projects and work reusing analysis and simulation results.

We are using the Java family for PMS development: J2EE, JSF/JSP, Hibernate, JDBC, Eclipse IDE. The PMS is planned to be developed as Web-oriented system and Eclipse IDE plugin. The general architecture of the PMS is shown in Fig. 2.

![Fig. 2 Architecture of the PMS based on the MMM approach](image)

At present the research in this area has already been done, for instance, the development of the software application for measurement of such parameters as functional requirements size, effort, time, source code size and some other parameters. The basic dynamic mathematical model for simulating the software processes has been created. Statistics data collected with the help of the developed measurement tools is used for this simulation [So06].

### 3 Measurement Metrics and Mathematical Models

The main input parameters for the analysis with the help of mathematical model should be [FP95, Eb05, Ca97, Sw04]: size of software programs, quality of software programs, effort, staff number, and durations of a project, project processes and tasks. Effort, staff number and durations could be measured easily. In practice of project management the
size of tasks are estimated directly in time terms (hours, days). But for sizing the tasks some units of real size are needed. In our approach Function Point (FP) as the most specified functional metric is used as a unified metric of the software product size [Bo04, IF94]. IFPUG Function Points measure the functional size of software projects and software applications based on the Functional User Requirements (FUR). FUR represent the processes and procedures performed by the software and can be sized by IFPUG Function Points. The standard of the Functional Size measuring is ISO/IEC 20926. The main idea of FP metric is to count the number of base logical components: internal logical Files (ILF), external interface files (EIF), external inputs (EI), external Outputs (EO), external Queries (EQ). The module of the proposed PMS for counting FPs is called Functionality Measurer.

The static mathematical model describes the duration of the total work which depends on the staff number and work size. According to Brooks [Br95] we have 4 dependences of duration from staff for different task types. They can be united in one dependence as it is shown in Fig. 3.

![Diagram](image)

**Fig. 3 Dependences of work duration on staff number**

We assume that the total duration of the work is the sum of the simultaneous tasks development duration and the duration of all tasks integration. The other assumption is that the duration increases non-linearly depending on work size. This assumption is the main one in costs models like COCOMO, COCOMO II, COCOMO II 2000 [Bo04]. According to all the assumptions we’ve got the following duration and effort model:

\[
D = D(N, F) = D_s(N, F) + D_t(N, F) = F \left( \frac{1}{p_s(N)} N + \frac{1}{p_t(N)} \right) = F \left( \frac{1}{P_s e^{-\lambda_s N}} + \frac{1}{P_t e^{-\lambda_t N}} \right)
\]

where \(D\) – work duration, \(N\) – staff number, \(F\) – work size,

\(D_s(N, F) = F / p_s(N)\) – duration of simultaneous tasks development,

\(p_s(F) = P_s e^{-\lambda_s F}\) – staff productivity in simultaneous tasks development,

\(D_t(N, F) = F / p_t(N)\) – duration of tasks integration by the team,

\(p_t(F) = P_t e^{-\lambda_t F}\) – team productivity in tasks integration,
\[ E \] total effort (work costs).

The total productivity of a development team for this type of work is the following:

\[ p = \frac{p_s(F)p_r(N)}{p_s(F)N + p_r(N)}N. \] (2)

The example of the model is shown in Fig. 4. We have four statistical points in it. The line shows the optimal staff number for achieving the minimal duration of the work.

![Fig. 4 The dependence of the process duration on the work size and staff number](image)

The mathematical model is used in our PMS for parameters estimation. Work size is the size of software products. The staff number is the total number of developers, analytics, testers and other process specific roles involved in the current process. Thus, we solve two general problems of mathematical modelling [Pe75]:

- parameters identification of the model (1) (analysis in the PM): the statistics data of different projects such as work durations, staff number, work size are used to identify the parameters of the model (1);
- estimation of the work parameters (estimation in the PM): model (1) is used for estimating the duration setting different values of staff number and work size.

The dynamic mathematical model is constructed according to the assumption that the main software product is the functionality and it could be represented by different artefacts in each software project process. We use states of functional requirements according to which artefacts have already been developed during the production process as it is shown in Fig. 5. The size of each product is counted according to the status of each functional requirement. Some specific artefacts are developed in each process. But the main product is the functionality of the software. Thus, the main idea lies in the fact that each artefact reflects the development stage of functionality. For example, according to the RUP we have six main artefacts: business models, requirements, design, source code, test cases and released functionality produced in a software project.
In this way we connect together the size of software product, effort of development, staff number and the process of development.

4 Conclusion and Ongoing Work

In this paper the chief goal is the demonstration of the unity of the theoretical research on software development processes and the real industrial practice of using those approaches. The statistics data of software projects are used for estimating such parameters as functional requirements size, effort, duration, staff number, costs, etc. The further research will be concentrated on such factors as including the minimal size of the simple task assigned for a developer in the model as well as the measuring and modelling of software product quality aspects and the further development of the PMS.

Bibliography


