# Model for Managing Software Development Projects by Fixing Some of the Six Project Management Constraints

[1]Boris Todorović, [2]Miroslav Matić

[1]*Information technology project manager, Republic of Srpska Securities Commission, Banja Luka*

*Email: boris.todorovic@gmail.com*

[2]*CEO, East Code d.o.o., Banja Luka, Email: eastcode@blic.net*

**General survay**

DOI: **10.7251/JIT1101033T**

UDC **007:004]:005.334**

**Abstract:** This study is focused on the software development process, viewed from perspective of information technology project manager. Main goal of this research is to identify challenges in managing such projects and provide a model for delivering software solutions that satisfies client's expectations. Project management theory describes six constraints or variables in every project, which project managers can use to better control the project and its outputs. Fixing some of the six project management constraints (scope, cost, time, risks, resources or quality) will allow project manager to focus on most important project aspects, rather than being drawn between all of the variables.This paper is aimed at information technology project managers and portfolio managers, as it describes the practical application of this model on a software development project. Findings of this research support the theory that, by applying good project management practice and focusing on project/business-critical requirements, will enable project managers to complete projects successfully and within tolerance limits. Results show that by identifying key business constraints, project managers can create good balance of six constraints and focus on the most important ones, while allowing other constraints to move between limits imposed by clients and stakeholders.

**Keywords:** software development, project management, PMBOK, six project constraints, fixed project constraints, risk management, quality management, project scope management

## Introduction

Information technology project management and software development processes have been around for several decades, but have begun maturing only at the brink of 21$^{st}$ century. Since the early 90's, when majority of software-based corporations started expanding rapidly, until today, whole process of software development and project management has been constantly challenged. The *Chaos Report study* [17] suggests that most information technology projects even today do indeed fail, or are heavily challenged – notproducing quality software, not conforming to business scope and cost requirements and even going over budget. Costs of software development have steadily been brought to a more acceptable level by adopting modern software development methodologies such as Unified Process and eXtreme Programming, which provided a new set of tools, methods and techniques for project managers and team members.

On the side of the project management, most influential framework today is *Project Management Body of Knowledge* or **PMBOK** [12], which proposes set of six constraints or variables, which are used to evaluate project success. By controlling projects scope, cost, time, quality, resources and risks using this framework, project managers can indeed efficiently manage projects. However, in the real world situations, it is not entirely possible to control all of these constraints.

*Main hypothesis* of this paper is that it is more effective and realistic to fix some, but not all of the constraints. In other words, project manager must set most important aspects of the project with the client and stakeholders. Top priority constraints must be fixed, while others will be monitored to be within acceptable limits.

## INFORMATION TECHNOLOGY PROJECT MANAGEMENT IN THEORY

Software development is not just an activity in which specific software is written in a programming language, but a whole *set of processes and activities*, with clearly defined structure and rules. In theory [14] [2], software development process consists of several phases: user requirements definition, system analysis, system design, implementation (programming), software testing (quality control) and installation in production environment. Schwalbe (2006, p. 46) suggest that these phases are not sufficient in the perspective of project manager, so two additional phases are added to software development lifecycle: project initiation and project planning. These two phases are actually starting points for any software development project, as they are not initiated by the project team but by senior management, board of directors, technical directors or prospective clients. The software development lifecycle can be then illustrated as follows:

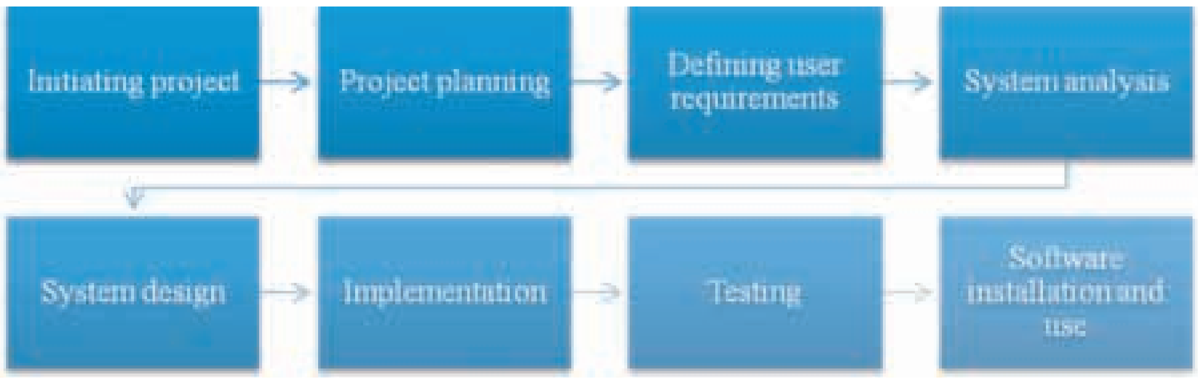
Some of the modern software development methodologies, such as widely accepted IBM's Unified Process, propose a mix of software development and project management processes by including such disciplines as project change management, general project management and environment management. However, a clear separation must be made between software development methodology, which has to do with controlling the software building process itself; and project management methodology, which in essence provides a set of tools, methods and techniques for managing project, financial assets, human resources, time, communication, etc. Goals of project management include not only creating software and proving highest level of quality possible (which is primary goal of software development process), but creating software within budget, timeframe, with acceptable level of risk and available human resources (Nicholas & Steyn, 2008, p. 4).

### Project Management Body of Knowledge (PMBOK)

Project management body of knowledge is a project management methodology written by one of the largest international project management professional organizations, The Project Management Institute. First edition of "*A Guide to Project Management Body of Knowledge*" was published in 1987 and the latest edition in 2010. Since 1987, it has become number one standard in project management worldwide [1]. The standard itself comprises of five process groups [12].

1. Project initiation
2. Project planning

DIAGRAM 1–MIXED SOFTWARE DEVELOPMENT PROJECT LIFECYCLE(SATZINGER, JACKSON, & BURD, 2004, P. 64)



3. Project execution
4. Project monitoring and controlling
5. Project closing

Each of the process groups deals with the specific timeframe of the project. Project initiation process group deals with portfolio management, or precisely speaking, a process of initiating new project using proven and reliable methods [16]. Project planning is all about analyzing project requirements and allocating resources and time in order to complete project successfully and within budget. Project execution deals with the time in project when majority of work is being done, and is focused on maintaining team, leveraging resource usage and optimizing the process in order to satisfy limitations and client's expectations. Project monitoring and controlling is about monitoring team's progress and controlling any unwanted situations by constantly applying good practice and leadership skills of a project manager. Finally, project closing deals with the final phase of the project, when team is delivering the product and writing closure reports, evaluating team work in lessons learned report and generally getting acceptance by a client.

On the other side, PMBOK describes the *nineknowledge areas*, or project manager's key competencies: [12]

1. Project integration management
2. Project scope management
3. Project time management
4. Project cost management
5. Project quality management
6. Project human resources management
7. Project communication management
8. Project risk management
9. Project procurement management

These nine knowledge areas are the primary focus of the standard, as they provide a necessary set of techniques, tools and methods for project managers to follow. For example, project time management describes usage of critical path analysis, PERT technique, Gantt and network diagrams, in order to create preliminary and final project time frameworks. PMBOK also suggests best practices for applying these techniques, tools and methods, as well as workflows in projects with clearly identified input and outputs (list of project documentation).

Some of the knowledge areas are not directly linked with the project requirements, such as project communication management, integration management and procurement management. Each of them is dealing with intra-project issues. However, six others reflect the real project requirements set by the client.

**Project constraints (project management triangle)**

Most projects have defined certain financial or schedule limits, such as what is the definite project budget or what is the due date of system being fully operational. These are not imposed by the development team, but by management, clients or stakeholders. Project manager must work with these non-technical project requirements and a framework for managing them is actually contained within PMBOK. PMBOK proposes concept of using a *project management triangle*, or managing and evaluating project success through three variables, or constraints: **time**, **cost** and **scope** (as illustrated in the diagram below).

Diagram 2 - Three variables of project management triangle(Project Management Institute, 2000)


Time constraint presents *schedule*, or allocated time for project team needed for successful completion of project. Cost constraint is a *budget*, or financial assets allocated for human resources, hardware, software, or other incurring costs such as consulting services and goods.Finally, scope constraint represents *realized project goals*, or in the information technology terms, functionality of the finalized software. In order for project to be kept on track and within limits, project manager must continually evaluate these three constraints and delegate project team activities accordingly.

The three constraints are in fact interconnected, as Schwalbe [15] and Haugan [4] notes. Increasing the scope of the project has direct impact on time and costs, as more work will be done, and more work means spending more money on resources. Fixing all of the constraints is impossible, but fixing one or two is them is possible. For example, fixing scope and time will mean that project will be done on schedule with all functionality, but project manager may manipulate with cost variable, by having various software development contractors join the project in order to actually meet previous two constraints. Fixing the third constraint is, therefore, very hard, in theory considered impossible [5].

The outcome of this triple-constraint model is software quality. That is, if project satisfied acceptable limits of three constraints, the produced software has got enough level of quality. This statement, however, was disputed in modern project management and software development theory by Haugan [4] and Hamilton et al [3], among others. A project could, in fact, satisfy the acceptable limits of the three constraints, but the quality may be unacceptable for the client. This is why there was a need for change of triple-constraint model.

### Evolution of project constraint management in PMBOK 2010 (six project constraints)

Triple constraint model was changed in PMBOK's 4thedition [12] in order to better cover all possible variables that are affecting success of projects. First of all, quality was removed from being a goal of the project, to being a constraint. Secondly, project risks and (human) resources are added, forming a final **six-constraint model**. The main goal of the model is project itself – *the successful completion of a project*, as illustrated on the diagram below:

DIAGRAM 3 - SIX CONSTRAINTS OF PROJECT MANAGEMENT (PROJECT MANAGEMENT INSTITUTE, 2010)


When arguing that quality is a constraint, not a goal [10], suggests the quality itself can projected. In other words, project team and client can agree on what level of quality software will have. Project manager can than make tradeoffs based on the agreement and can balance between quality, risks, costs or any other constraint. Finally, all six variables illustrate real world scenario, where project manager has to find a balance between different requirements. They indeed provide an excellent overview of all potential issues of a project.

## RESEARCH METHODOLOGY

This research proposes a new model of managing software development projects by utilizing existing best practice in the field of project management. The six-constraint model, described in PMBOK [12] was modified by the author in order to achieve greater level of project control. Since all projects have certain, specific requirements, a project manager can agree with the client on their importance. For example, if it is a critical for a project to be released on the specific date and with all the functionality completed, project manager can fix those constraints (time and scope), while leaving others negotiable. This model uses method of **fixing certain constraints** in order to prioritize tasks and project success factors.

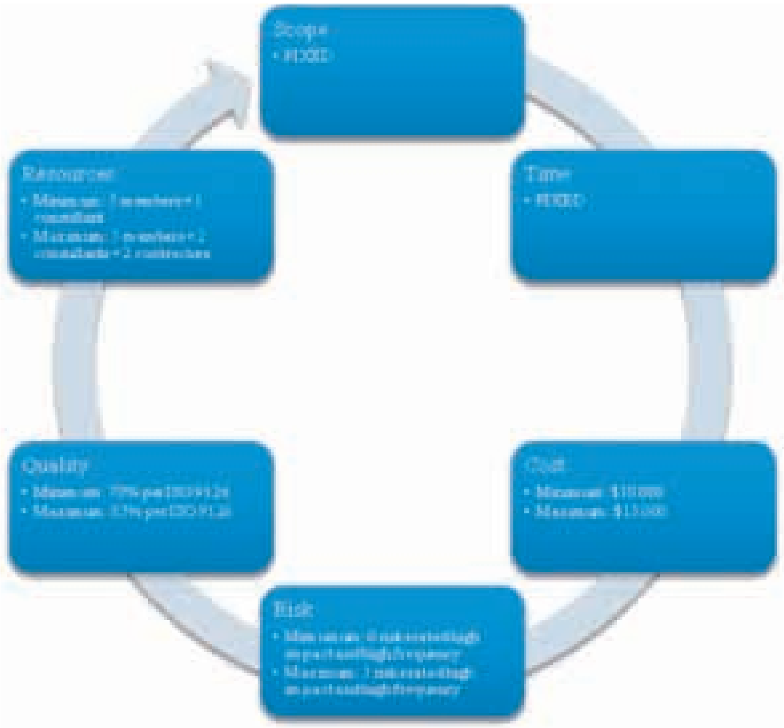
While agreement with client can be made on priorities, project manager must also set *control limits for rest of the six constraints*. If this is not set, costs, for example, could skyrocket, eventually spending all of the financial resources on getting the work done on time and within scope. This would pose a great risk for the project, and although time and scope constraints would be satisfied, other constraints would get out of control.Diagram below illustrates usage of such model, with clearly defined constraints that are fixed for the project and with defined maximum and minimum control limits for others.

Since this paper seeks adequate model for project constraints management, adequate research subject was chosen. Experimental research in this paper was conducted at a software development company in Bosnia and Herzegovina. Research will, therefore, be conducted on one software development project, in order to confirm research hypothesis.The software development project chosen for the purpose of research was the creation of *centralized, Web based information system*. System featured following requirements:

- Web-based application with centralized storage mechanisms based on relational database management systems (SQL-like)
- multiuser environment with authentication and encryption capabilities
- adequate security mechanisms
- efficient and effective data entry interfaces and report creation
- document management system integrated with Microsoft Office SharePoint technologies

The requirements regarding the usage of this software were not well formed at the beginning of the project, so the **project scope was not defined entirely**. However, the company was given **fixed amounts of financial assets** and was given a **very strict timeframe** in which software had to be fully operational. This was an excellent opportunity to test real word situation in which three out of six constraints were fixed.

Diagram 4–Example of model in which two out of six constraints are fixed (source: author)




While company could easily shift focus from time and costs to scope, risks, resources or quality, this model was used to in fact control the most important aspects of the project. This research will set goals and limits for each of constraints and evaluate them after completion of the project. After the evaluation, we will present projected and real values for each constraint and will inspect how the project manager succeeded in applying the model and controlling the project's success.

In order to hide sensitive financial data, project costs will be enumerated using *relative weights*. For example value 1.0 will indicate initial costs, while value 1.5 will indicate 50% greater costs relative to initial project costs. Project time constraint will be measured using *total working hours* (w/hrs.), which is the only true measurement of time it took to produce working software solution. Project scope will be measured by number of *use cases implemented* (uc/i) by project team in the final software build. Resource usage will be measured by indicating *number of persons* were active during project lifecycle, but making a clear difference between their roles (e.g. project manager, team member, contractor or consultants). Level of quality will be evaluated using ISO 9126 standard its "quality in use metrics", then summarizing the result using weighted averages for the entire software in order to provide single *quantitative software quality level* [6]. Risk constraint will be measured by extracting *number of major risks* with combined value (possibility of occurrence * impact) information from risk register, a part of risk management documentation.

## ANALYSIS OF RESEARCH RESULTS

### Presentation of research results

The following table represents comparison between planned and actual values for all six constraints, measured during experimental research on a chosen software development project:

Measured results for the project were compared to initially planned values, in order to create a **performance index** for each characteristic. *Green-colored performance index result* indicates a positive result, meaning that planned value was completely met. *Yellow-colored performance index result* indicates a satisfying result, where actual values did not meet the plan, but are within minimum and maximum control limits. *Red-colored performance index result* indicated result that was on the limit, or out of minimum and maximum control limits.

### Discussion and analysis

Using the suggested model, project manager was able to fix two of six constraints and to control them throughout the lifecycle of the project. As presented in research results, *cost* performance index was at 100%, meaning that there was no cost overrun. *Time* constraint's performance index was at 103%, just slightly over the planned value, but within the control limit. Although this was a case of project team working behind original schedule, we can conclude that this constraint was very much within control limits and almost entirely met. Real-world situations, such as changing business environments don't always allow for complete satisfaction of all plans. Since costs were fixed, *resources* were also limited, so performance index for this constraint was also 100%. Finally, all of the fixed constraints have been successfully managed using the model, which results in their nearly perfect performance index, which was the primary goal of this research.

Since the project was managed in a way to satisfy budget and schedule, other constraints did not achieve planned values. *Scope* constraint had performance index of 89%, meaning that project team did not produce all of the software modules by the end

TABLE 1 - COMPARISON OF PROJECTED AND ACTUAL VALUES FOR SIX CONSTRAINTS (SOURCE: AUTHOR)

| | Planned values | Minimum control limits | Maximum control limits | Actual values | Performance index |
|---|---|---|---|---|---|
| Cost | 1.0 | 1.0 | 1.1 | 1.0 | 100% |
| Time | 2.900 w/hrs. | 2.900 w/hrs. | 3.100 w/hrs. | 2.980 w/hrs. | 103% |
| Scope | 123 uc/i | 102 uc/i | 123 uc/i | 110 uc/i | 89% |
| Resources | 1 project manager (PM), 3 project team members (PTMs), 2 contractors (CTRs) | 1 PM, 2 PTMs | 1 PMs, 3 PTMs, 4 CTRs | 1 PM, 3 PTMs, 2 CTRs | 100% |
| Quality | 90% by ISO 9126 | 80% | 95% | 83.45% | 92% |
| Risks | 10 major risks | 5 m/r | 14 m/r | 14m/r | 140% |

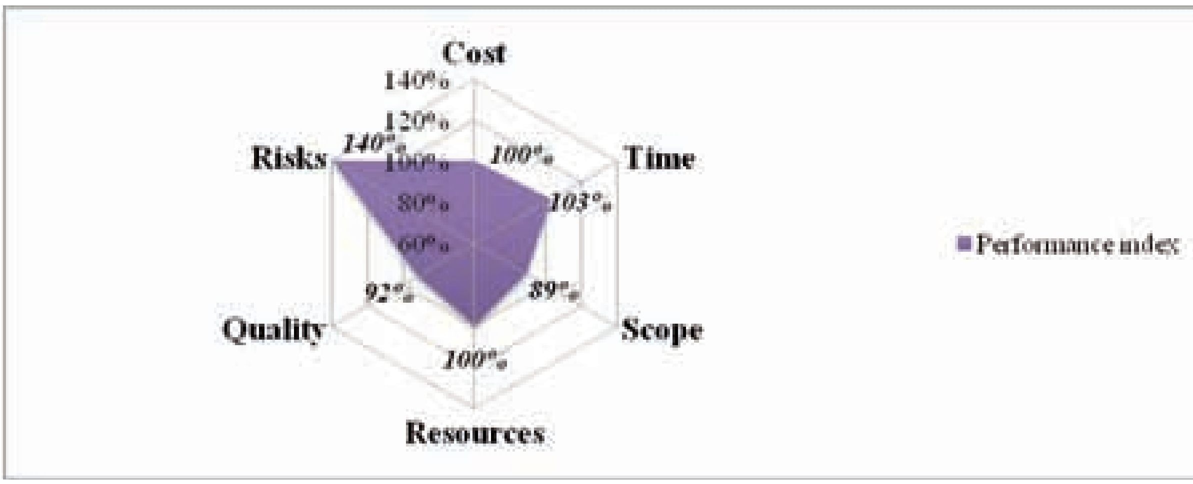
of the project. However, performance index was within control limits. Same thing happened with *quality* and *risk* constraints, whose performance indices were at 92% and 140% respectively. Quality was within control limits, as well as risks.

We must note that risks were at the highest level of maximum control limit, which was due to the project team focusing on delivering as much quality as possible. This, in essence, means that management of risks was of tertiary priority (primary priority being fixed cost and time constraints, and second being quality and scope). Also, risks were higher due to usage of relatively new technology - Microsoft SharePoint 2010. Since the development of modules based on this technology was not a priority, risk management process was focused on other project goals and issues, leaving this as a major risk though the end of the project.

## Conclusion

This research presented the practical, experimental results, which support the main hypothesis: by fixing some of the project management constraints, project managers can more effectively control success and outcomes of projects. Focusing on important aspects of the project, such in this case, satisfaction of budget and schedule, other constraints can be well balanced and kept within minimum and maximum control limits.

Diagram 5 - Graphical representation of performance indexes (source: author)




Although the model's effectiveness was be measured by comparing projected and actually achieved performance indices, it should be used on day-to-day basis by project manager as a mean of continually controlling project performance.

This paper opens a new research direction in information technology project management by suggesting a new and practical model for controlling project's success. Further research directions for this model include evaluations of different mixes of fixed project management constraints, as well as application of this model to small, medium and large project teams.

## References:

[1] Blokdijk, G. (2007). *Project Management 100 Success Secrets*. Lulu.com.

[2] Gibbs, R. D. (2006). *Project Management with the IBM® Rational Unified Process®: Lessons from the Trenches*. Upper Saddle River: IBM Press.

[3] Hamilton, G., Byatt, G., & Hodgkinson, J. (2010). The Unspoken Additional Constraint of Project Management. *American Society for the Advancement of Project Management*.

[4] Haugan, G. T. (2011). *Project Management Fundamentals: Key Concepts and Methodology*. Vienna, VI: Management Concepts Inc.

[5] Heldman, K., Baca, C. M., & Jansen, P. M. (2007). *PMP Project Management Professional Exam Study Guide*. Indianapolis, IN: John Wiley and Sons.

[6] ISO/IEC. (2002). *ISO 9126-2: Software engineering – Product quality – Part 1 (External metrics)*. ISO/IEC.

[7] McConnell, S. (1997). *Software Project Survival Guide*. Microsoft Press.

[8] Newell, M. W., & Grashina, M. N. (2004). *The project management question and answer book*. AMACOM Div American Mgmt Assn.

[9] Nicholas, J. M., & Steyn, H. (2008). *Project management for business, engineering, and technology: principles and practice*. Butterworth-Heinemann.

[10] Pressman, R. S. (2001). *Software Engineering: A Practitioner's Approach*. Boston: McGraw-Hill.

[11] Project Management Institute. (2000). *A Guide to the Project Management Body of Knowledge*. Newton Square, PA: Project Management Institute.

[12] Project Management Institute. (2010). *A Guide to the Project Management Body of Knowledge*. Newton Square, PA: Project Management Institute.

[13] Royce, W. (1998). *Software Project Management: A Unified Framework*. Reading, MA: Addison-Wesley.

[14] Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2004). *System Analysis and Design in a Changing World*. Boston: Thomson Course Technology.

[15] Schwalbe, K. (2006). *Information Technology Project Management*. Boston: Thomson Course Technology.

[16] Springer, M. (2005). *Program Management*. Purdue University Press.

[17] The Standish Group. (1995). *The Standish Group Report - CHAOS*. The Standish Group.

Submitted: February 20, 2011
Accepted: May 30, 2011