*Krešimir Lukač**

UDK 621.391:65.01.012(497.5)
Prethodno priopćenje

SOFTWARE PROJECT MANAGEMENT AT REPUBLIC OF CROATIA

Autor u članku prikazuje glavne značajke i probleme globalne industrije software-a, ciljeve i načela tehnike software-a i definira ulogu projekta menadžmenta u razvojnom procesu software-a. Prikazuje i ciljeve i metode menadžmenta softwarskog projekta u Republici Hrvatskoj. Predlaže i ključna područja i sljedeće korake za daljnje poboljšanje softwarskih procesa u hrvatskim organizacijama.

Introduction

Software presents corner stone of today´s information age. It has essential importance for economic development, scientific researche, business development, communication as well as for all other aspects of human society affecting our daily way of living. Explosive development of information technology has created huge demand for new software making software industry the fastest growing segment of knowledge based information and communication industry of today.

Software systems present one of the most complicated human products. However, resources necessary for software development haven´t kept pace with such growing demand creating today´s so called "software vacuum". This is the consequence of increasing complexity and the fact that software development depends on people whose work quality varies as well as shortage of adequate experts, immaturity of methodology and technology beeing unable to provide robust systems

* K. Lukač, Master of Science, Swisscom, Zürich. Članak primljen u uredništvu: 04. 01. 2002.

- reliable, fault tolerant, secure, adaptable and cost effective. According to research of software projects conducted by Standish Group 73% of projects are late, significantly increase budget, are abandoned before delivery or are totally unusable upon completion[1].

In order to get insight into software development process in Croatian organisations in such global environment the survey on the basis of CMM v1.1 standard has been conducted. Thus Croatian software industry has been positioned in comparison to the global software industry, the key areas have been identified and next steps for its improvement have been suggested.

Project Management in Software Engineering

Management is key element of software development process. In order to be successful it is necessary to understand volume of the required work, assess risks, necessary resources, time, costs as well as to control and manage changes. Management activities begin before technical implementation, resume during software implementation and end after software is out of operational work.

The main goal is to create software system that will meet user requirements within given cost, time and people restraints. In order to achieve it principles of software engineering have to be used during all phases of software development and maintenance[2].

Six main goals of software engineering are[3]:

(1) Functionality - software ability to satisfy user needs and solve his problems. This is primary goal. If it is not reached all other goals are missed.

(2) Supportability - ability to maintain, enhance or otherwise change the software. These changes appear during operational work, due to change of user requirements or environment in which software operates. This is especially important during software design.

(3) Reliability - probability that the software system will perform without failure under specified conditions or use. The goal of optimum system is to have 100%

[1] STSC: Guidelines for Successful Acquisition and Management of Software Intensive Systems, Software Technology Support Center, Hill AFB, 2000., pp. 2-6., according James G.: IT Fiascos…and How to Avoid Them, Datamation, November 1997.

[2] Boehm B.W.: Software Engineering Economics, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981., pp. 17

[3] STSC, see 1, pp. 9-22.

reliability. If the software can be repaired instantaneously after error occurrs without system disruption it has 100% reliability as well.

(4) Safety - a guarantee that software system will not fail under stressed operational conditions.

(5) Efficiency - refers to the best use of critical resources: processor cycles and memory locations.

(6) Understandability - link between stated problem and the corresponding solution. Well-engineered software is readable i.e. code can be easily read and there is respective documentation. Data structure (objects) and algorithms should be easily distinguished.

In order to achieve goals mentioned above it is necessary to apply the following software engineering principles[4]:

(1) Abstraction - separating essential process characteristics from all unnecessary details. It is important for managing software complexity. Actually, the whole process of software engineering is based upon this principle. Each level in the software development process presents lower abstraction level of the final product.

(2) Information hiding - hiding details that don´t effect other parts of software system. It is usually accomplished by modular design where all information contained in one module is hidden from other moduls if they don´t need this information.

(3) Modularity - organizing software applications into discrete modules connected with appropriate interfaces i.e. it presents physical architecture of the software system. Coupling is the measure that denotes how tight the modules are mutually connected. On the other hand, cohesion describes how tight the internal modul elements are coupled.

(4) Localization - physical location of resources within module with strong cohesion and loose coopling.

(5) Uniformity - usage of consistent notation for all modules without unnecessary differences.

(6) Completeness - including all important system elements. Abstraction and completeness ensure that modules are necessary and sufficient.

(7) Confirmability - possibility of sotware decomposition that enables better testing and makes modifications easier to implement.

[4] Ibid., pp. 9-25.

Software Project Characteristics

Project can be defined as "temporary attempt undertaken to create unique product or service"[5]. However, software projects are significantly different from other project types. Their common characteristics are: (a) invisibility - progress during software development can´t be seen instantly; (b) complexity - in comparison to other projects software projects are more complexed; (c) flexibility - software can be relatively easily changed. On one hand it is software advantage but on the other it is potential source for its instability.

Software systems can be clasified as socio-technical systems since they require technical and people organization. Therefore software project managers are expected to be technically competent as well as capable for organizing and managing human resources.

Project management can be analysed through five basic management functions[6]:

(1) project organization - definition of project teams and project management structure;

(2) project planning - creation of project plan i.e. definition of requirements that project has to meet, activity plan and assignement of ressources;

(3) human resources management - how to use people involved in the project in the most effective way;

(4) leadership - creating vision and strategy for vision achievment;

(5) project controlling - project manager has to track project status continuously, analyse gathered information and conduct corrective actions. The senior management has to be always informed about project status.

Achieving efficient software process capability requires continuous improvement of software quality. Among others it is stimulated by increased customer expectations, requests for decreasing software development costs, productivity improvement and respective life cycle duration.

Quality of software products can‘t be increased if only end product has been controlled. The basic foundations for software improvement should be in the software process itself. Therefore, it is important to have properly defined software development process i.e. project management.

---

[5] Duncan, R.W.: "A Guide to the Project Management Body of Knowledge", Project Management Institute, Sylva, 1996., pp. 4.

[6] Lukač K.: "Project Management at Software Development Life Cycle", Faculty of Economics, Master Thesis, Zagreb, 2001., pp. 15.

There are several standards related to software quality and software process. One of the most widespread standards today is Capability Maturity Model (CMM). It was developed by Software Engineering Institute in cooperation with MITRE Corporation. The initial release was published in 1991.

CMM - Capability Maturity Model

CMM presents sets of recommendations in several key process areas that have been proved to enhance software process capability. It is the guidance on "how to gain control of software organizations processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence"[7]. It helps in determinig current process maturity and the most critical issues important for software quality and process improvement. The purpose is to help organizations to be focused on a limited number of activities and to work intensive on them thus increasing software process capability. The emphasis is on undertaking many small, evolutionary steps rather than revolutionary innovations. Popularity and success of CMM makes it one of the most preferred standards in USA and other countries. Some of the organizations using CMM are Citicorp, Hewlett-Packard, IBM, Motorola, U.S. Air Force and so on. Empirical results have shown that in average $5.7 return on investment in software process improvement is saved for every $1 spent[8].

The Five Levels of Software Process Maturity

CMM defines following terms[9]:

(1) Software process - set of activities, methods, practices and transformations that people use to develop and maintain software and the associated products (e.g. project plans, design documents, code, test cases and user manuals).

(2) Software process capability - the range of expected results that can be achieved by following a software process.

---

[7] Paulk M.C., Curtis B., Chrisis M.B., Weber C.V.: "Capability Maturity Model for Software", Version 1.1, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1993., pp. 5.

[8] Curtis B., Hefley W.E., Miller S.: "Overview of the People Capability Maturity Model", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, September, 1995., pp. 7.

[9] Paulk M.C., Curtis B., Chrisis M.B., Weber C.V., see 5, pp. 3.

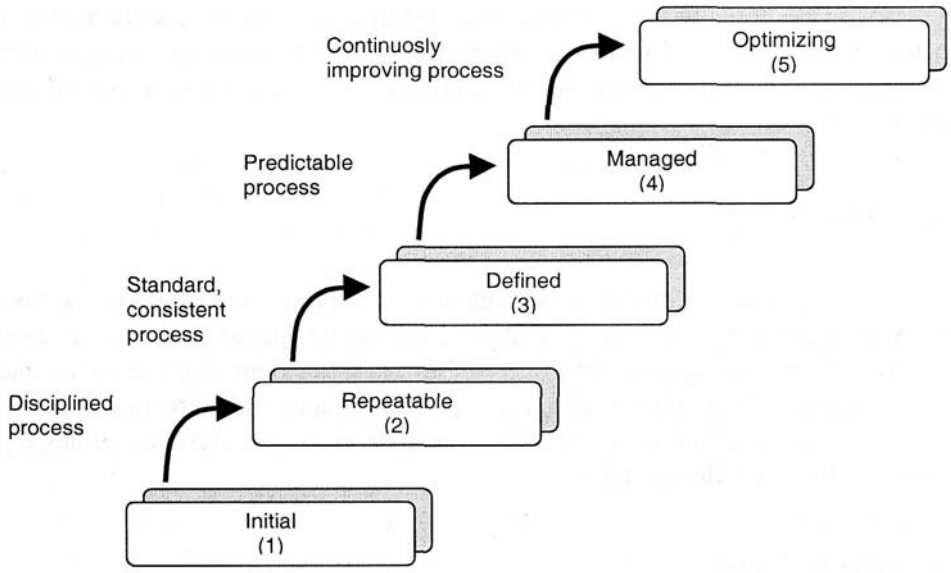
(3) Software process performance - the actual results achieved by following a software process.

(4) Software process maturity - the extent to which a specific process is explicitly defined, managed, measured, controlled and how effective it is.

As a software organization gains in software process maturity it institutionalizes its software process via policies, standards and organizational structure. Institutionalization entails building the infrastructure and the corporate culture that supports the methods, practices and procedures of the business which will last even after those who originally defined them have gone.

Figure 1.

THE FIVE LEVELS OF SOFTWARE PROCESS MATURITY[10]



CMM ranks software organizations into five maturity levels as shown in Figure 1.

Focusing on improvement of key process areas (KPA) inside each maturity level, organization can continuosly enhance its software process maturity.

[10] Ibid., pp. 8.

Levels Description

Initial Level

At this level there is no stable environment for software development. People are overloaded, budget is often exceeded, there are exeptional individuals or "heroes" which are capable to stand pressure and in the same time to meet requirements. During crises plans are often abanded. If the key people leave organization capability which they bring is also gone.

Repeatable Level

Organizations on this level have defined policy for managing software projects and procedures to implement it. Basic principles of project management have been applied and it is based on the experience from previous projects. Systematic approach to requirement analysis, planning, controlling, quality assurance, configuration management has been applied. Software process is stable since previous success can be repeated.

Defined Level

Standard software process is documented and applied in whole organization. Software engineering and management processes are integrated. Processes are used and adapted enhancing efficient praxis of software development. Education and training program is implemented. Specific projects adapt standard software process in order to satisfy unique characteristics of respective project. Activities, roles and responsibilities are clearly defined.

Managed Level

Organization sets quantitative quality goals regarding software products and processes. Productivity and quality are measured in all projects. Collected data are analysed on organizational level in order to improve processes. Projects fall within acceptable quantitative boundaries i.e it is possible to predict quality and performances. In case of significant variations corrective actions are being taken to solve the problem.

Optimizing Level

Organization is focused on continuous software process improvement. It is possible to identify strengths and weaknesses of applied processes. Focus is on error prevention. Cost benefit analysis is applied for new technologies and proposed process changes. Improvements are result of process enhancement and innovations using new technologies and methods.

These five CMM levels represent a way for improvement of software processes whereby the focus is on software engineering and management activities inside specific key process areas.

Survey of Software Project Management at Republic of Croatia

The goals of the software project management survey at Republic Croatia are: (1) introducing CMM standard to project managers as one of the most widespread standard in the field of software engineering; (2)getting information about existing software development processes at Republic Croatia on the basis of specific CMM key process areas; (3) determing advantages and disadvantages of the software projects investigated; (4) identify activities necessary for enhancing software processes; (5) getting preliminary results about maturity of organizations at Republic Croatia regarding implementation of software processes in relation to five CMM maturity levels.

The survey has been conducted on the basis of Maturity Questionnaire based on CMM v.1.1 standard. Questionnaire has been organized according to the key process areas (KPA) and comprises all 18 KPAs. It contains structured and unstructured questions with unhidden goals since the assumption that respondent is willing to give answer has been used. The advantage of using the questionnaire is the possibility of systematic usage of research method as well as relative speed of getting data. Disadvantage of the research method applied in getting these primary data is the human factor at persons examined i.e. possible bias.

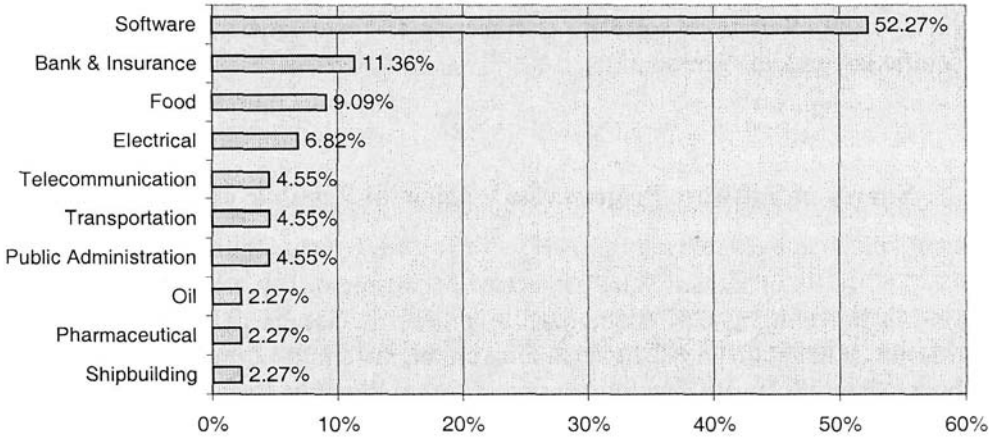
53 software projects from 44 organizations located at different parts of Republic Croatia have been comprised within the survey.

Organization Characteristics

Figure 2. shows organizations researched classified per specific industry.

Figure 2.

ORGANIZATIONS PER INDUSTRY



Companies from software industry dominate (52%) i.e. software development presents core business here. They are followed by bank and insurance companies (11%) and companies from other industries where software development is not core business but presents one of the main factores of strategic competition advantages and makes essential part of their business.

Researched organizations are located at the following regions: Zagreb (52%), Međimurje (11%), Primorsko-goranska (11%), Istarska (9%) and Osječko-baranjska (4%). Other organizations are equally distributed between Požeško-slavonska, Varaždinska, Šibensko-kninska, Zadarska, Dubrovačko-neretvanska and Koprivničko-križevačka regions.

Figure 3.

TOTAL EMPLOYEE NUMBER DISTRIBUTION

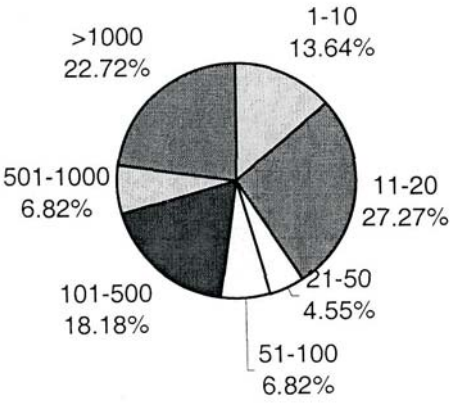
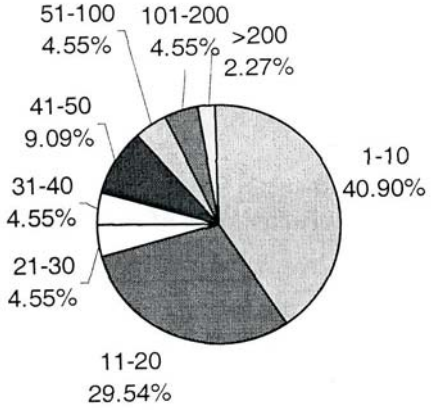


Figure 4.

EMPLOYEE NUMBER AT SOFTWARE DEVELOPMENT AND MAINTENANCE



Average number of employees in software industry is 19. Hence, if we take into account the number of employees as one of the criteria of organization size it can be concluded that small companies prevail in the software industry.

Figure 5.

ORGANIZATION SIZE OF GLOBAL SOFTWARE INDUSTRY[11]

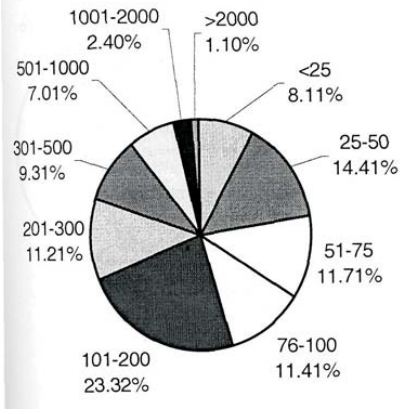
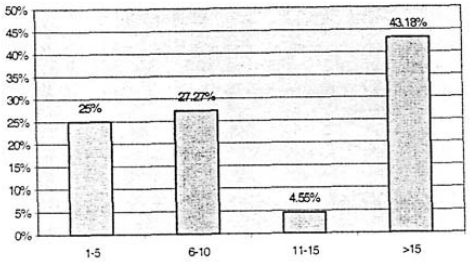


Figure 6.

BUSINESS YEARS



[11] http:/www.sei.cmu.edu/sema

Figure 3. shows distribution of total number of employees in researched organizations. In contrast to the global software industry where less than 8% of organizations have up to 20 people engaged in software development and maintenance, this percentage goes up to 70% in Croatian organizations.

Taking into account number of business years and number of employees as well as organization life cycle concept, it follows that over 50% of organizations are in the starting or development phase. Therefore it can be concluded that Croatian software industry is relatively young.

Characteristics of Respondents

From the total number of respondents 87% are men and 13% women. Figure 7. shows age distribution. Average age is 33 years.

Figure 7.

RESPONDENTS AGE DISTRIBUTION

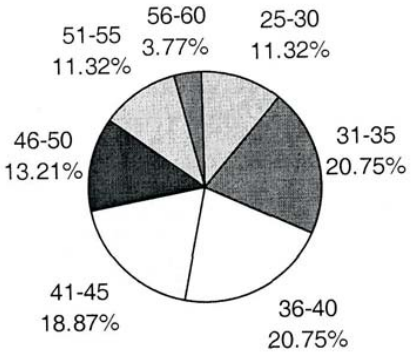
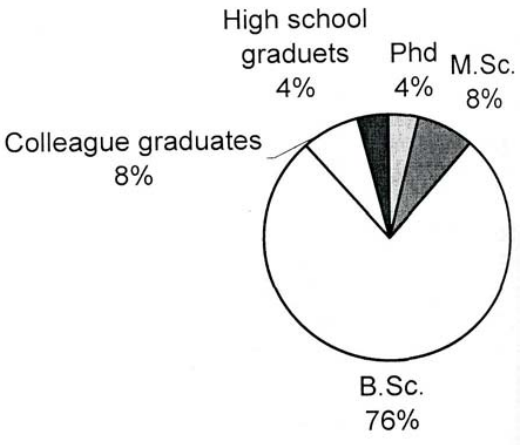


Figure 8.

QUALIFICATIONS



71% of respondents are technical graduates, 21% economical and 8% other. These results could be expected as well.

Figure 9.

RESPONDENT POSITIONS

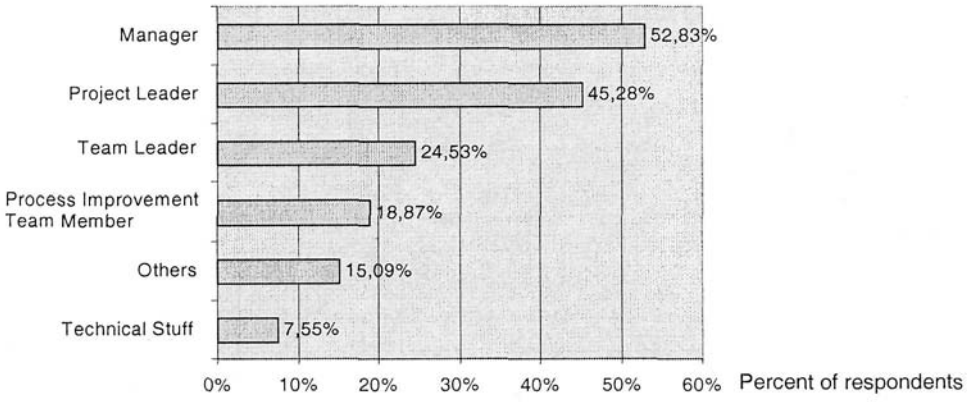
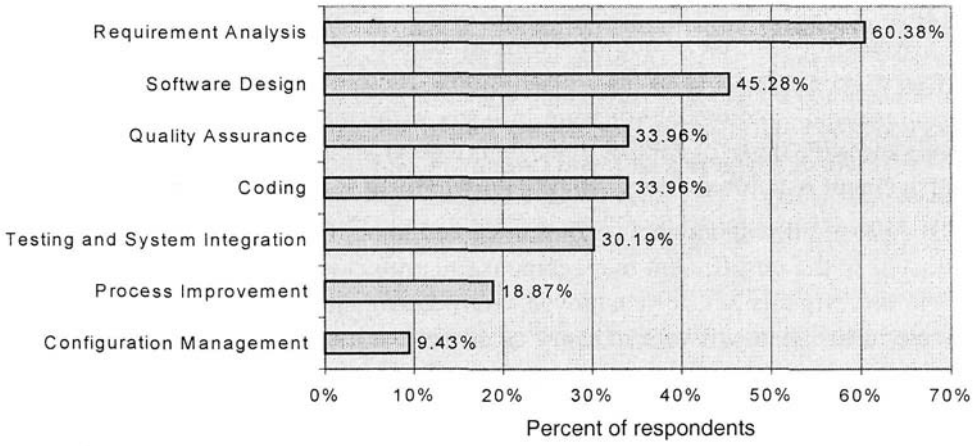


Figure 9. shows that majority of respondents are managers (53%), project leaders (45%) and team leaders (25%).

Figure 10.

RESPONDENT ACIVITIES



Majority of respondents work on requirement analysis (60%), software design (45%) and quality assurance (34%). These activities correspond to KPA Requirement Management, Project Planning and Quality Assurance from the second CMM maturity level. This data are indicative for preliminary results of the maturity levels of the researched organizations.

Figure 11.
NUMBER OF EMPLOYEES ON THE PROJECT

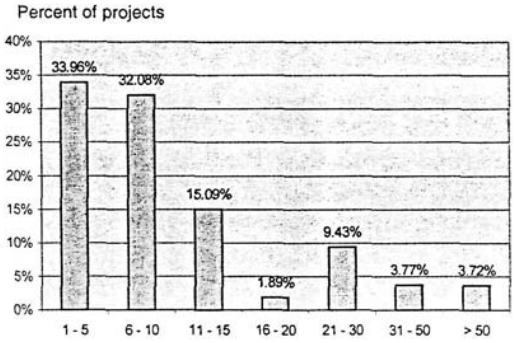
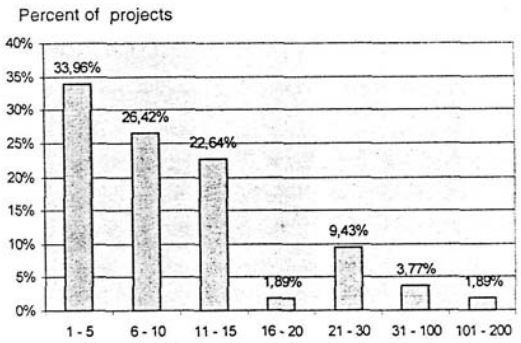


Figure 12.
RANGE OF CONTROL



Project teams working on software development have up tu 10 people in more than 66% of all cases while a number of projects with more than 50 people involved is small (3,77 %). This confirms former statement about starting and development phase of Croatian software industry.

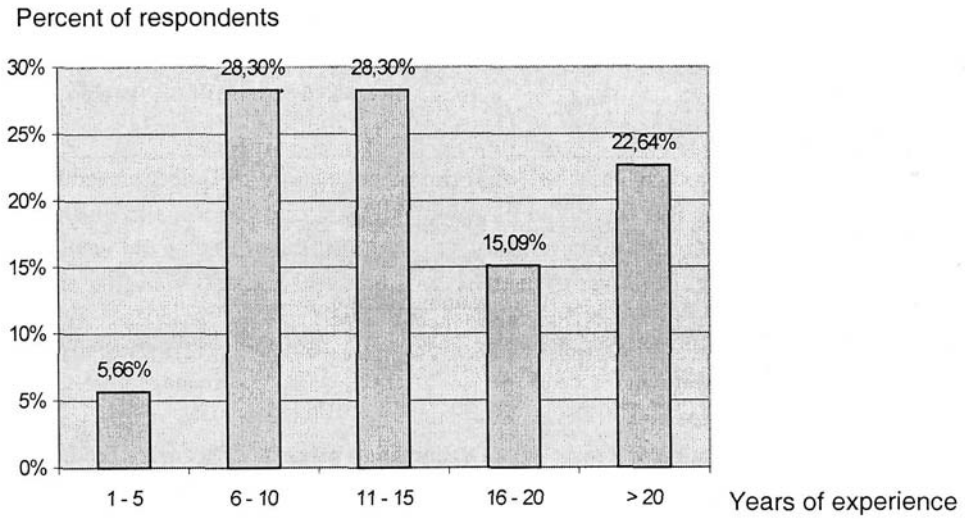
Range of control as one of the main aspects of organizational structure is between 1:10 at more than 50% of all cases. Therefore we can speak about low organizational structure of software projects with small number of management levels and wide range of control. It presents a good assumption for fast and efficient management reaction on the changing environment requirements.

74% of all respondents are not informed about CMM standard. This is the indicator of the unsufficient management and project leaders education. Thus one of the survey goals has been achieved - introducing the CMM standard to the key people in Croatian software industry as one of the most relevant standard in this segment and stimulate them on changes in order to increase efficiency of software processes.

Respondents experience in software development is ranging mainly between 6 and 15 years (57%) as shown in Figure 13. This indicates that management and project leaders as majority of respondents inquired have enough experience in software project implementation.

Figure 13.

RESPONDENTS EXPERIENCE



However, a very bad indicator is that even 77% of respondents haven´t participated in software development estimation using any standard (CMM, ISO 9000/9001 or some other standard) before. It means that senior management doesn´t pay enough attention to the efficiency estimation of software processes in their organizations. If the management doesn´t know the status of their processes it can´t make adequate decisions for its improvement. Therefore it can be concluded that most initiatives for enhancement of software development come from project teams and not from the management for which this should be one of the main tasks.

Software Development Praxis

As already mentioned survey comprises all 18 key process areas of CMM model. The table below shows distribution of KPA according to respective categories of software process.

Table 1.

THE KEY PROCESS AREAS BY MATURITY LEVEL[12]

Process Category Level	Management	Organization	Engineering
5. Optimizing		Technology change management Process change management	Defect prevention
4. Managed	Quantitative process management		Software quality management
3. Defined	Integrated software management Intergroup coordination	Organization process focus Organization process definition Training program	Software product engineering Peer reviews
2. Repeatable	Requirements management Software project planning Software project tracking and oversight Software subcontract management Software quality assurance Software configuration management		
1. Initial	Ad hoc proces		

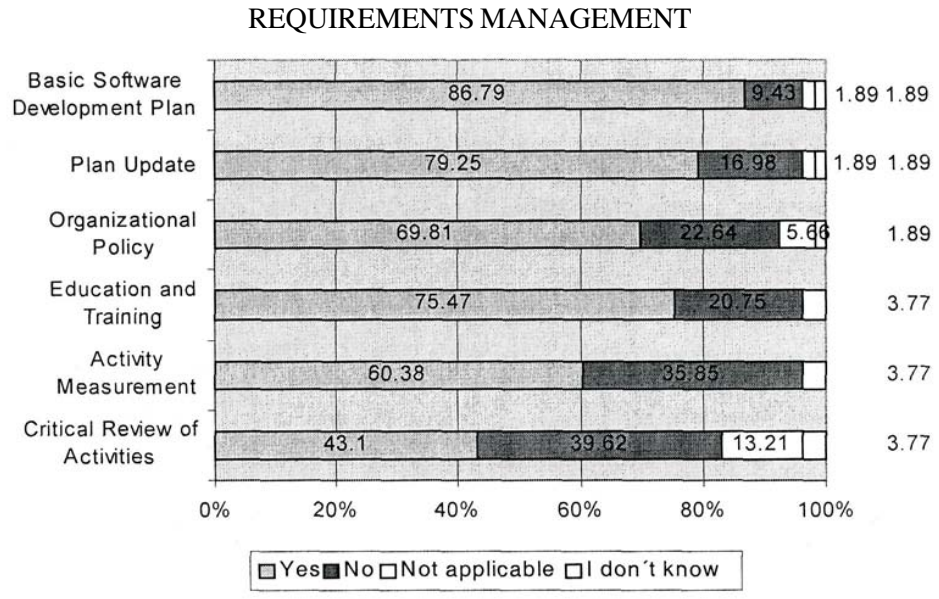
[12] Paulk M.C., Weber C.V., Garcia S.M., Chrisis M.B., Bush M.: Key Practices of the Capability Maturity Model, Version 1.1, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1993., pp. 0-25.

Survey results of some of the most significant key process areas for Croatian software industry are presented below.

Requirements Management

Goal of this KPA is to reach agreement between organization and client regarding client´s requirements. It covers all technical and nontechnical issues and serves as basis for estimation, planing, implementation and controlling activities on the software development project. In case of requirements changes respective project deliverables (process description, plans, working procedure, code, documentation) and activities are updated to keep consistency.

Figure 14.



In almost 87% of projects surveyed there is baseline of software development defined and it presents foundation for software engineering and management. In case of requirements changes, project plans, working procedure and documentation are updated in 79% of all projects. In 70% of all cases organizational policy is implemented (basic principles established by senior management which have to be applied during decision making). Project stuff is educated and trained for relevant working procedures in 75% of all projects. Measures for varifying activity status

(for example total number of suggested, opened, approved or accepted changes) have been applied in 60% of all projects. In 43% of all cases critical reviews from quality assurance department have been conducted.

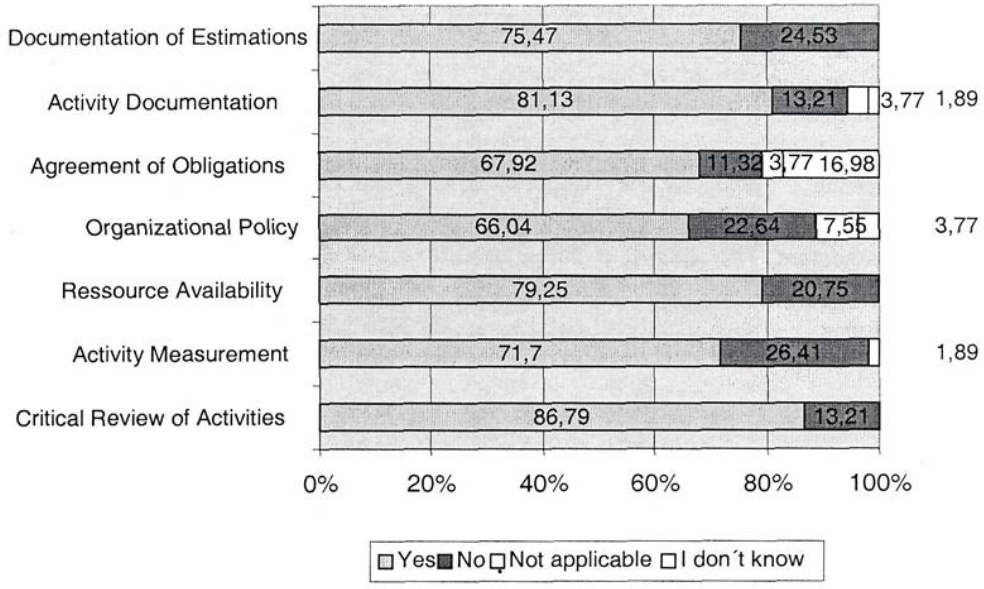
Activities mentioned above have been applied in the great majority of the projects researched. However, the last activity is one of the critical ones for success of software process since it has direct influence on software functionality and in this case it shows insufficient management involvement in the software process.

Software Project Planning

Plans for software engineering activities are created here as well as plans for project managing. Also, estimation of necessary work, responsibilities, time and activity sequence has been determined.

Figure 15.

SOFTWARE PROJECT PLANNING



In 75% of all projects, estimation of necessary time, costs and activity plan are documented in order to be able to track software project. Activities that will be performed as well as obligations have been documented in 81% of projects.

Agreement regarding obligation fulfilment from relevant project stuff has been requested in 68% of all cases. Organizational policy for software project planning has been applied in 66% of all cases while in 79% of all projects corresponding planning resources already exist (e.g. experienced individuals, money ressources). In 72% of all cases activities on the project planning have been measured (e.g. comparison of reached milestones with planned dates). In 87% of all cases management conducts critical reviews of those activities. Taking into account data above it can be concluded that all activities from this KPA have been applied in the great majority of projects researched.

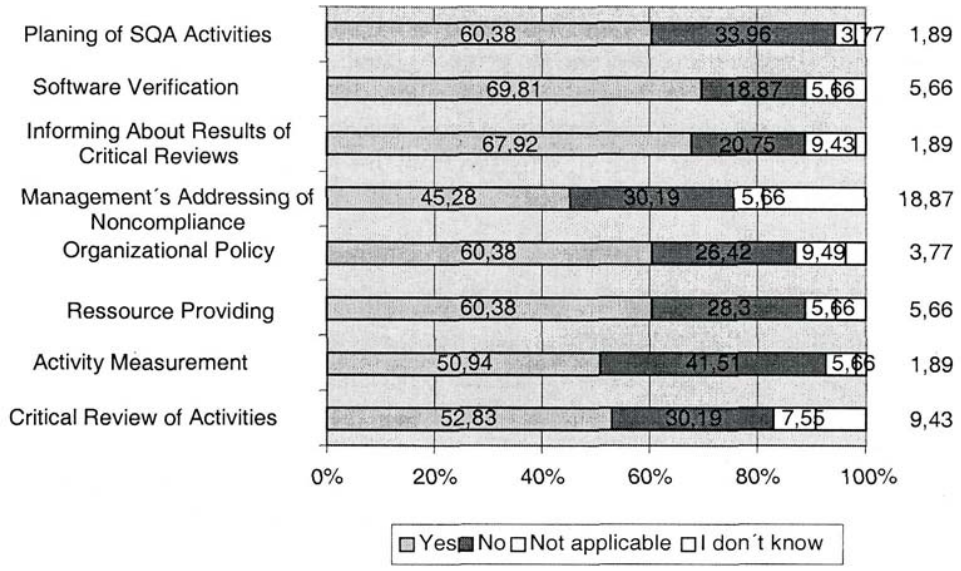
Software Quality Assurance

Activities on software quality assurance help management to get insight into the software process applied to specific project. Critical reviews and audit verify whether software products and activities have been done according to relevant procedures and standards. It presents valuable source of information for management.

Software quality assurance activities have been planned in 60% of projects. In 70% of all cases software verification in relation to respective standards, procedures and requirements has been done. Results of critical reviews regarding quality assurance have been proceeded to the management and relevant groups in 68% of all cases while senior management discusses noncompliance issues in relation to relevant standards in 45% of all cases.

Figure 16.

SOFTWARE QUALITY ASSURANCE



In 60% of all projects organizational policy for implementing software quality assurance has been applied. In 60% of cases ressources necessary for implementation of these activities have been assured (e.g. money, managers for tracking and solving deviations). In 51% of all projects costs and status of quality assurance activities have been measured. Activities of this KPA have been criticaly reviewed from senior management in 53% of all projects.

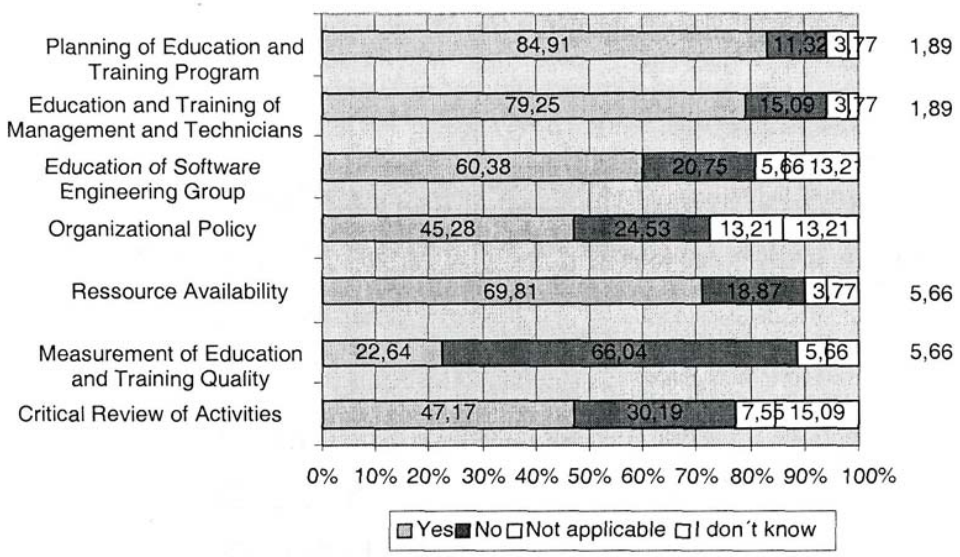
Generally, all activities of this KPA have been implemented in almost half of all surveyed projects. However, it is necessary to focus more on noncompliance issues as well as on measurement and critical review of quality assurance activities.

Training Program

People as individuals, their skills, abilities and knowledge represent the key for the project success. Therefore objective of education and training program is developing people´s skills and extending knowledge in order to make them able to efficiently satisfy specific requirements. First, these needs have to be clearly identified accross organization and after that education and training program has to be implemented. Methods could be formal (i.e. courses) or unformal (i.e. training on the job).

Figure 17.

EDUCATION AND TRAINING PROGRAM



Education and training activities for management and other project stuff are planned in 85% of all projects while they are really implemented in 79% of all projects. Software Engineering Group has been educated in 60% of cases. Organization policy is applied in 45% of all cases. Ressources necessary for implementation of these programs have been assured in 70% of all cases. However, even in 66% of all projects, measurement of program quality has not been done. Senior management periodically reviews these activities in 47% of all cases.

Generally, it is very positive that managers and project leaders are aware of importance of getting new knowledge and developing individual skills in order to be successful in their job. However, quality of these programs has not been sufficiently measured.

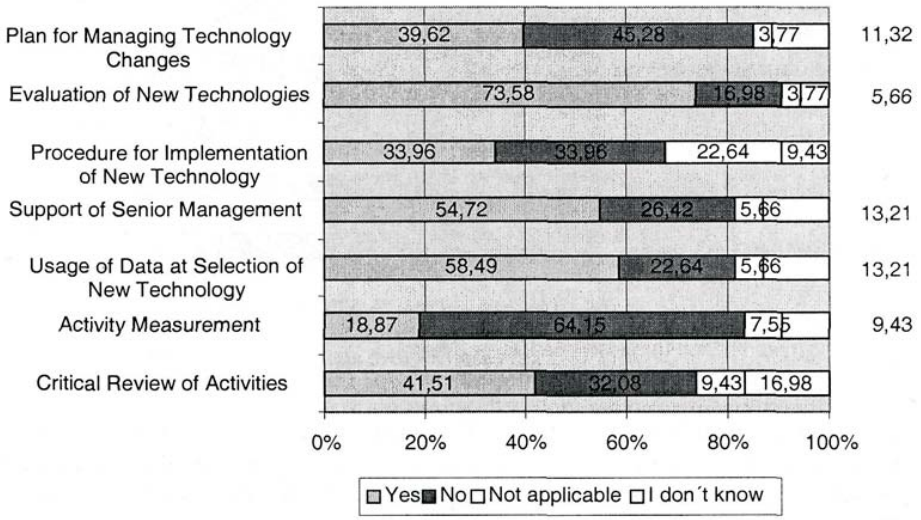
Technology Change Management

Technology change management comprises idenitification, selection, evaluation and implementation of effective new technologies into organization. The goal is to improve software quality, increase productivity and decrease the cycle time for software development. Usually a specific group is established (such

as group for software process engineering) which takes part in the project and helps in implementation and evaluation of new technologies as well as in managing changes of existing technologies. After that selected technologies are incorporated in standard software process inside organization.

Figure 18.

TECHNOLOGY CHANGE MANAGEMENT



In the majority of the projects researched (45%) plan for managing technology changes doesn´t exist. However, in 74% of all projects new technologies are evaluated in order to determine their effect on quality and productivity. In the majority of organizations documented procedure for incorporation of new technologies in standard software development process inside organization doesn't exist i.e. it is not followed. Senior management sponsors these activities in 55% of all projects.

Data that can help in the selection of new technology exist in 58% of all cases. However, at even 64% of all cases measurement used to determine status of these activities has not been done. Senior management periodically reviews organization´s activities for managing technology change in 32% of projects surveyed.

Data above indicate that activities and objectives of this KPA have not been sufficiently covered. Hence, it has direct influence on software quality, productivity and time duration of software life cycle in Croatian organizations. The most critical is the measurement of the status of these activities as well as non existence of procedures for implementation of new technologies at software process.

Maturity Levels of Croatian Software Processes

An algorithm for preliminary ranking of maturity levels takes into account a number of "yes" and "no" answers. Answers "not applicable" or "I don´t know" have not been taken into account whereby answer "I don´t know" indicates that respective activity has not been institualized inside organization. Key process area can satisfy specific level only if percentage of "yes" answers conforms with percentage shown in Table 5. Bootstrap methodology has been used with the assumption that organization satisfies specific maturity level only if at least 80% of all goals from key process areas for respective level have been covered as well as 100% of all goals from all lower levels. For example, organization on the third maturity level has to cover all goals from the key process areas on the second maturity level and at least 80% of all goals from the key process areas on the third maturity level.

Table 2.

ALGORITHM FOR MATURITY LEVEL RANKING

Maturity Level Estimation	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
KPA on 2nd Level	<80%	80%	100%	100%	100%
KPA on 3rd Level	-	-	80%	100%	100%
KPA on 4th Level	-	-	-	80%	100%
KPA on 5th Level	-	-	-	-	80%

Preliminary results of the software process maturity at Croatian organizations are shown in Figure 19.

Figure 19.

CROATIAN MATURITY PROFILE

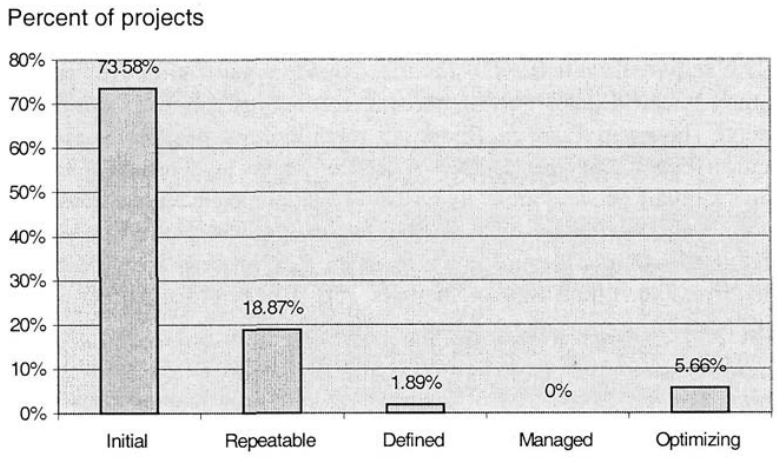
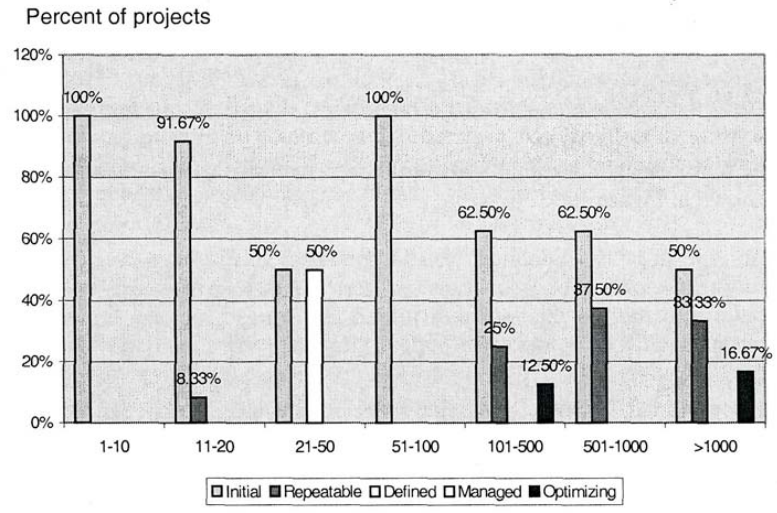


Figure 20.

MATURITY PROFILE BY ORGANIZATION SIZE



74% of all processes researched are on the first maturity level. These are organizations without stable environment for software development and maintenance. There is no appropriate management and implementation of good engineering practise. Therefore we could conclude that majority of Croatian organizations with software development praxis are characterized with passive instead of proactive behaviour.

On the second CMM maturity level there are 19% of projects surveyed. In these organizations, policy regarding management of software projects is defined. Software development process is documented, measured and can be enhanced. Respective standards have been defined and organization is trying to meet them.

On the third maturity level there are about 2% of projects researched. In these organizations standard software development process has been defined. Education and training program for management and other project stuff has been conducted in order to meet requirements. Criteria, standards and working procedure are defined. Since these software processes have been appropriately defined management has good insight into the project development.

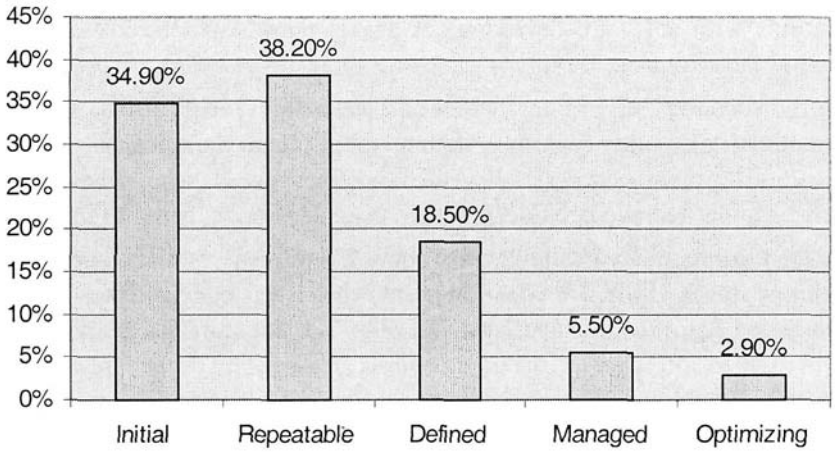
On the forth level there is no organization while on the fifth level there are 5,66% of projects researched. These are big Croatian companies with long time experience of software development projects in specific field of interest with 30 to 100 people involved. They are characterized by proactive acting and continous enhancement of software processes with focus on error prevention.

As shown in Figure 20. there is positive correlation between organization size and software process maturity.

The distribution of maturity levels of global software industry is shown in Figure 21. It has been done on the basis of 1018 organization estimations conducted since 1997.

Figure 21.

MATURITY PROFILE OF GLOBAL SOFTWARE INDUSTRY[13]



As shown, global software industry has been moved from Initial level towards Repeatable level in comparison to the Croatian software industry that is still mainly at Initial level of software process implementation.

Conclusion

Croatian software industry is relatively young industry with over 50% of organizations at the starting or development phase. Organizational structures of software projects are low with small number of management levels and wide range of control. At more than 70% of all projects, standard software development process is defined inside organization. Thus it presents a good assumption for fast and efficient management reaction. However, even 74% of managers and project leaders are not familiar with CMM standard and 77% haven´t participated at software process estimation before. It indicates their insufficient education. That is also one of the reasons why 73,58% of projects surveyed are positioned at the first CMM maturity level, 18,87% at the second, 1,89% at the third, 0% at the fourth and 5,66% at the fifth level.

[13] http://www.sei.cmu.edu/sema

Activities from the following KPA are the weakest covered: Quantitative Process Management from the fourth level as well as Technology Change Management and Process Change Management from the fifth CMM level.

Hence, Croatian software industry is still mainly at ad hoc way of software project implementation in comparison to the global software industry that has moved from Initial to Repeatable level.

The next objective of Croatian software industry should be moving from Initial to Repeatable CMM level. Organizations should be focused on critical activities of KPA(s) on the second CMM level such as creation of specific plans, measurement of activity status, reviewing activities from senior management as well as defining organizational policy for related KPA(s).

LITERATURE:

1. *Boehm, B.W.: "Software Engineering Economics"*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981.
2. STSC: *"Guidelines for Successful Acquisition and Management of Software Intensive Systems"*, Software Technology Support Center, Hill AFB, 2000.
3. *Duncan, R.W.: "A Guide to the Project Management Body of Knowledge"*, Project Management Institute, Sylva, 1996.
4. *Paulk, M.C., Curtis, B., Chrisis, M.B., Weber, C.V.: "Capability Maturity Model for Software", Version 1.1*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1993.
5. *Curtis, B., Hefley, W.E., Miller, S.: "Overview of the People Capability Maturity Model"*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, September, 1995.
6. *Lukač, K.: "Project Management at Software Development Life Cycle"*, Faculty of Economics, Master Thesis, Zagreb, 2001.
7. *Paulk, M.C., Weber, C.V., Garcia, S.M., Chrisis, M.B., Bush, M.: "Key Practices of the Capability Maturity Model", Version 1.1*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1993.
8. *Gilb T.: "Principles of Software Engineering Management"*, Addison Wesley, 1988.

PROJEKT SOFTWARSKOG MENADŽMENTA U REPUBLICI HRVATSKOJ

Sažetak

Članak na početku prikazuje glavne značajke i probleme globalne industrije software-a. Nakon toga prikazani su glavni ciljevi i načela tehnike software-a. Definira se uloga projekta menadžmenta u razvojnom procesu software-a. Opisuju se zajedničke karakteristike softwarskih projekata. Također je prikazan CAP (capability maturity model) kao jedan od najraširenijih standarda software-a, kao i razine zrelosti koje specificiraju razmjer u kojem se proces software-a eksplicitno definira, vodi, mjeri, nadgleda, i koliko je uspješan u pojedinoj organizaciji.

Nadalje članak prikazuje ciljeve i metode menadžmenta softwarskog projekta u Republici Hrvatskoj. Prvo se prikazuju značajke organizacije i onih koji odgovaraju. Nakon toga se daju ključna područja procesa obuhvaćena pregledom. Ilustrirani su detalji nekih najznačajnijih ključnih područja hrvatske industrije software-a. To uključuje menadžment uvjeta, sigurnost kvalitete software-a, program usavršavanja kao i menadžment tehnoloških promjena.

Prikazuje se metodologija vlastitih sila u procjeni razina zrelosti. Daju se preliminarni rezultati zrelosti softwarskih procesa u hrvatskim organizacijama. Rezultati su uspoređeni s profilom zrelosti globalne industrije software-a. Također se predlažu ključna područja i sljedeći koraci za daljnje poboljšanje.