UDC: 681.32.06
Original scientific paper

THE CORRELATION BETWEEN BOOTSTRAP MATURITY LEVELS AND THE CHARACTERISTICS OF THE SOFTWARE DEVELOPMENT PROCESS

Zdravko Krakar

University of Zagreb, Faculty of Organisation and Informatics, Varaždin, Croatia
Croatian Information Technology Agency, Zagreb, Croatia
E-mail: zkrakar@open.hr

Danko Vukadinović

City of Zagreb's Municipal Data Processing Centre, Zagreb, Croatia
E-mail:danko.vukadinovic@gzaop.tel.hr

This paper analyses the influence of process metrics on the increase of so-called maturity levels according to the Bootstrap model. The basic characteristics of this model are described and those metrics that need to be implemented in the software development process in order to change its features are considered and these features can be quantitatively interpreted.

The most important characteristics of the software process were measured by using the metric of function points. Also, the influence of a maturity level on achieved productivity was analysed, and a strong correlation was obtained. The results that were obtained illustrate the importance of the application of metrics when one wants to increase the quality of the software development process.

Keywords: metrics, function points, CMM, Bootstrap, software, maturity level, productivity, correlation.

1. INTRODUCTION

The software industry is constantly having big problems. The characteristics of software as a product mean low productivity in development, low certainty in view of the quality of the final product and high maintenance expenses. As a consequence of the pressure to change this situation, several methods have been developed with the aim of improving the way software is produced. The people who suffered because of this pressure were initially sophisticated users like the military, state administration, banks, insurance companies and those companies/industries that are ecologically sensitive.

These methods include the CMM (Capacity Maturity Model) and Bootstrap – the European version of the American CMM model. Their mutual characteristic is the so-called managing maturity development within software development. These methods have the same three starting points: Wats Hemphry's maturity development model, an increase in the maturity of the process of software development via key areas and a process in which the situation is continuously improved.

There are 5 basic maturity levels. They are the initial level, the repeatable level, the defined level, the managed level and the optimising level and they can be expressed numerically from 1 to 5. The Bootstrap method has a finer gradation of 0.25 within each level. The goal here is to achieve as high a maturity level as possible since its increase improves the process, decreases expenses and increases product quality.

A scheme for Wats Hemphry's maturity development management model is given in Fig.1.

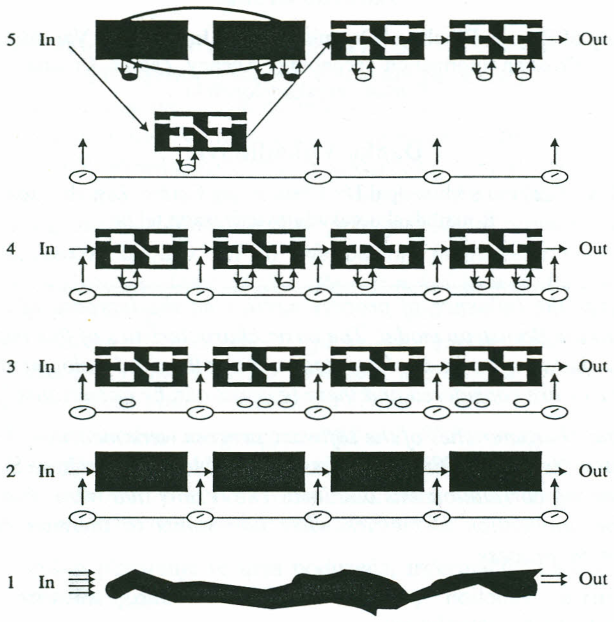

Figure 1. Wats Hemphry's maturity development management model

In the Bootstrap method there are 18 key areas which need to be implemented into the process of software development gradually and in a specific order. Table 1 shows the connection between these key areas, the focus of the process and the maturity levels. By measuring around 180 features that belong to the aforementioned 18 key areas and by using an appropriate algorithm, one can quantitatively determine the momentary maturity level. By using the resulting maturity profile, a programme for improving the situation is created – software engineering methods are implemented into the existing methods of work. New measurements show an improved maturity level and this type of iterative procedure is employed to manage maturity development.

Table 1. The basic criteria for each maturity level according to the Bootstrap method

Level	Focus	Key areas
5 Optimising	Continuous process improvement	• Prevention of errors • Managing technological changes • Managing development changes
4 Managed	Process and product quality	• Process measurement and analyses • Process management
3 Defined	Process engineering	• Process organisation • Process definition • Process revision • Education programme • Group co-ordination • Product engineering • Integration management
2 Repeatable	Project management	• Project planning • The possibility of reconstructing the course of development • Supplier supervision • Quality assurance • Configuration management • Request management
1 Initial	-	-

The characteristics of each maturity level are as follows:

1. The Initial Level

The development process is unstable, it happens ad hoc and can often be chaotic. Success is totally dependent on the individual capabilities of the team members. Outside requests are not known well enough, they are unstable and unforeseeable. The planned time and expenses are not met and they are regularly surpassed. Any practice is subject to the priority of the activities. The way in which the team work could best be described as nervously.

The whole development process resembles a black box and management has great difficulties in monitoring the project's progress. As far as they are concerned software system development represents an uncontrolled process. User demands are not foreseeable and they appear unexpectedly. No metrics are applied in the process.

2. The Repeatable Level

In order to meet deadlines and monitor expenses and results, the basic project management functions have already been established. The process is disciplined in that success can usually be repeated in similar projects.

User demands and working procedures are relatively controlled and they are established by project management. At this level the development process can be viewed as an array of black boxes and an insight into the project state is possible at their connecting points. Although the management does not know the details of what goes on in each of the "boxes", they are familiar with the situation at the control points and by using these points they supervise the process. Management deals with problems when they occur. Certain metrics are applied in the process.

3. The Defined Level

At level 3 the development process is entirely standardized, documented and integrated. All info system development and maintenance projects use a extremely detailed version of the standard process, which is adapted to each individual case.

The internal structure of each black box can be discerned which means that all the procedures in the development process have been recognised and as such well defined. Both the management and the designers know their jobs and duties well. Management actively prepares for the risks that may occur. A reliable report on the project state can be obtained promptly from the process at any time. Various metrics are applied in the process, and they have a very important role to play.

4. The Managed Level

A characteristic of the managed level is a detailed knowledge of the causes and consequences in the entire development process. By collecting the results of a series of process and product quality measurements, there are no more unknown quantities in its realisation and the process and products are intensively measured. Statistical supervision and process analyses methods are applied.

Management can monitor progress and observe problems and they the have an objective, quantitative basis for making decisions. The possibility of foreseeing results is improving because the process variability is decreasing due to the intensive use of metrics.

5. The Optimising Level

At level 5 the quantitative information in the process, as the well as favourable conditions for innovative ideas, new technologies, the prevention of errors and the managing of development changes, enable continuous process improvement. Management can monitor with precision the influence and effect of changes in the process and thus influence chronic losses that are already minimal, but are decreasing further. The application of a wide spectrum of various metrics is standard practice.

2. THE INFLUENCE OF METRICS ON THE INCREASE OF MATURITY LEVELS

In the previous section one could observe a connection between the maturity levels and the growing need for intensive use of metrics. The higher the maturity level, the greater the need for their existence.

How to organise the implementation of metrics in the software development process is not so straightforward. The most commonly used models are the SEI (Software Engineering Institute Pittsburgh) model, the SEL (Software Engineering Laboratory NASA) model and the AMI (Application of Metrics in Industry) model of the European Strategic Programme for Research and Development of Information Technology (ESPRIT). The mutual characteristic of all of these models is that they start with the goal of improving the organisation that produces software by using the Goal–Questions–Metrics (GQM) approach [1, 4-73].

Table 2. shows the connection between maturity levels and the activities necessary in order to reach the subsequent level.

Table 2. The connection between maturity levels according to the CMM/Bootstrap model and the activities that represent areas of the primary goals for the application of software engineering metrics [1,3-47]

Maturity level	**Activities that need to be carried out in order to reach the subsequent level**
1. INITIAL	PROJECT SUPERVISION • Project management – elaboration of plans, defining a schedule (of priorities), resource assessment, elaboration of development specifications. • Monitoring by management ▪ review and approval of all major plans ▪ elaboration of periodical reports on the project status. • Quality assurance – independent validation and verification of software development • Change supervision – change management and successive implementation
2.REPEATABLE	DEVELOPMENT PROCESS SUPERVISION • Establishing a development process group responsible for defining the process, identifying technological needs and monitoring the status of the development process. • Establishing the development process architecture – defining the life cycle, technological and management duties and check points. • Implementing development process technology – software engineering methods and technologies, prototyping, new programme languages.
3. DEFINED	IMPLEMENTING PROCESS MEASUREMENT • Establishing detailed process metrics – the basic set of process measurements for identifying expenses and profits. • Creating a process metrics database – of data collected during process measurement in all projects. • Resource assurance – for collecting, assessing, entering and maintaining data in the process metrics base as well as assessing methods of analyses and result interpretation. • Management for quality – an independent group for ensuring quality and monitoring each project with regard to the set plan and goals, and warning the management of aberrations in the process.
4. MANAGED	MEASUREMENT AUTOMATISATION AND PROCESS OPTIMISATION • Automatic data collection aiming to eliminate errors and inprecisions caused by manual data collection. • Process adjustment (improvement) – aimed at problem prevention and efficiency improvement.

The basic set of metrics according to the AMI model, forms an array of the following 17 metrics [1,153]: the effort needed for development, the development time, the size of the system, the size of the original code, the cyclomatic number, the testing coverage, the number of deficiencies in the process, the number of product deficiencies, the number of observed errors, the number of changes, the number of documentation pages, the structural metrics, the use of resources, slippage metrics, the personnel profile, the maturity level and project type. These metrics should be gradually implemented into the development process.

The most common software engineering metric used today (from which other metrics are deduced) is the Function Points Analysis (FPA) and with this it is possible to determine - the size of the application, the project duration, the working effort and productivity in the software development process [2].

It has already been stated that at the initial level there are still no defined demands for the use of metrics in practice. The quality of the development process and the finished product depend entirely on the quality of the designer, not the organisation. The repeatable level represents the first level at which the application of metrics is possible and necessary. Key areas such as project planning, the possibility of reconstructing the course of development, product quality assurance and supplier supervision, all generate the need for obligatory use of the appropriate metrics. Some of them are the number of deficiencies per product unit, or effort, or the number of errors etc.

A characteristic of the defined level is a stable and standardized production process. In such conditions it is necessary to apply productivity metrics, time metrics, expenses metrics, success and result metrics. Such notions continue at the managed level where both the production process and the course of development of the software product are continuously measured in order to achieve optimal management at each moment. At the optimising level one would expect an automatic system for the collection of the most varied data.

So, the thesis is that the influence of metrics grows with the increase of the maturity level, i.e. that increasing the intensity in which metrics are used contributes to the increase in maturity levels. An attempt shall be made to prove this thesis via real measurements.

3. MEASURING THE CHARACTERISTICS OF THE SOFTWARE DEVELOPMENT PROCESS

3.1. The relationship between the characteristics of the software development process and maturity level management

Managing maturity levels in software development, as a manner of fulfilment of the prerequisites for software organisation success, enables a reduction in the influence of outside interference. Managing maturity levels, based on Watts-Hemphry's model, hope to reduce the entropy of the system itself, increasing the stability of the software development process and raising the efficiency of the whole organisation.

When addressing the influence of organisation maturity management on the software development process, two aspects should be distinguished, and they are manifested through:

- Managing user requirements that come from the business environment. The business environment influences user requirements, and a better quality organisation will reduce the influence of interference that is entering the software organisation
- Managing internal organisation establishment, thereby ensuring the realization of demands by optimally using software organisation resources.

Such a view of the influence source does not imply the same division of characteristics, but a demarcation of the possible sources into those that come from the environment and those that come from the software organisation itself. In other words, the influences on the same characteristics of the software development process can come from different sources, and maturity management should ensure a reduction of the negative effects and an increase in the positive ones. In this context the consideration of the influence of maturity level increase l_i on the characteristics c_i of the software development process can be divided into three groups:

1. Reducing the characteristics c_i that imply expenditure of resources such as time, working effort and expenses, the number of changes, etc., $(l_1 < l_2 \Rightarrow c_1 > c_2)$,
2. Increasing the characteristics c_i that imply results and effects such as productivity, quality, effectiveness, user satisfaction, etc., $(l_1 < l_2 \Rightarrow c_1 < c_2)$ and
3. Reducing the aberration Δc in planned/foreseen (c_{ip}) and realised values/proportions of characteristics (c_{ir}) $(l_1 < l_2 \Rightarrow \Delta c_1 > \Delta c_2,, \Delta c_i = c_{ip} - c_{ir})$.

By applying this approach to internal attributes of the software development process it can be observed that a higher maturity level influences a shortening of the development time, a reduction in working effort, expenses and the number of changes and an increase in productivity.

Development expenses, as an internal attribute of the software development process belong to the group in which maturity management has a reducing effect when viewed per product unit for the same or similar characteristics of the entering demand (group 1). Expenses can be reduced by management, by the organisation and by technological interventions in the process. Since methods such as the CMM and Bootstrap method measure the fulfilment of demands in these areas, a hypothesis can say that at a higher maturity level the expenses per unit of software product are reduced.

In addition to a reduction in expenses per product unit, the influence of a maturity level can also be seen in the reduction in the aberration between planned and actual expenses, i.e. their elimination, which is in accordance with the 3rd group of ways in which maturity management influences the characteristics of the software development process.

On the other hand, productivity is an illustrative example of the influence of maturity levels on the improvement of internal process attributes, thereby ensuring the project is shorter and/or a reduction in working effort. As in the case of expenses, productivity can be influenced by management, by the organisation and by technological interventions in the process structure. The methods of maturity aforementioned level measurement measure the fulfilment of demands in these areas, and it is to be expected that with a higher maturity level, productivity will also be higher and the project will be shorter, and that which is show in Fig 2. was obtained by an interpolation of the actual, measured results and these are shown in detail in the subsequent items.

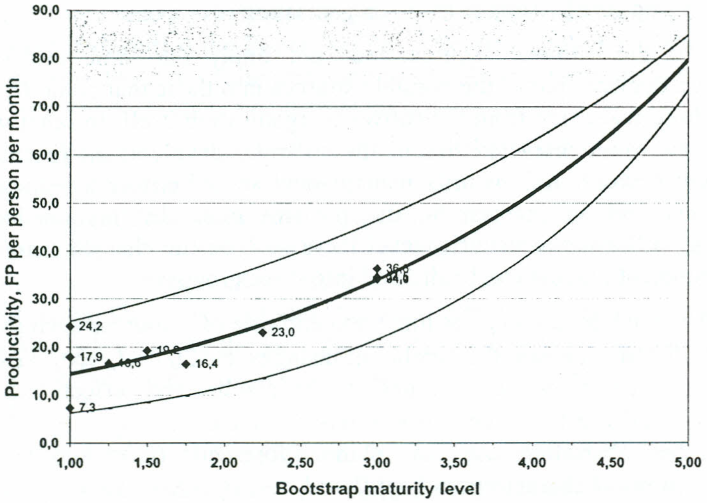

Figure 2. The expected increase in productivity depending on the maturity level

With the increase in productivity because of a higher maturity level, it is also to be expected that the aberration between planned and realised productivity will be reduced, while at the same time influencing a reduction in the aberration in other process characteristics such as development time, working effort, expenses, etc.

If the aforementioned approach were applied to external processes, it is to be expected that with a higher maturity level there would be an increase in product quality, process stability and effectiveness and user satisfaction and a simultaneous reduction in expenses.

As an example of the influence of maturity levels on external process attributes consider product quality viewed in accordance with the ISO/IEC 9126 standard [3]. With a maturity level increase it is to be expected that product quality will also increase. The Bootstrap method of maturity level measurement intrinsically includes quality system demands according to the ISO 9001 standard, thereby connecting process characteristics with the level to which this standard is met [4]. A reduction in the aberration between the projected and realised results of the process quality improvement programme can also be connected to the maturity level. This has already been expressed at the second maturity level with quality assurance as one of the key areas but has also been expressed through other key areas, right up to the optimising level.

3.2. Measuring the characteristics of the software process in a particular case

The characteristics of the software development process can be seen through appropriate attributes. According to [6,73] they can be divided into internal and external ones. Internal attributes include time, effort, expenses, the number of changes, productivity, etc. External attributes include attributes of quality, stability, expenses, effectiveness, reliability, user satisfaction, etc.

In attempting to investigate the influence of maturity levels on some of the aforementioned characteristics of the software process in a particular software organisation, research was carried out on a sample of 11 projects. The processed data and measured entities for 6 project characteristics are given in Table 3. The metrics used to calculate productivity were the Function Points Analysis (FPA) [2] and the Metric of Working Effort, while the maturity level was assessed by the Bootstrap method [7]. Productivity is expressed as the number of function points per person month.[1]

Table 3. The measured characteristics of the projects included in the research

Project Code	Project size, function points	Duration, months	Effort, person month	**Productivity, FP per person month**	**Maturity level**	Team size
K05/Y2K	75	8.0	10.2	**7.3**	**1.00**	12
K17/Napl*	206	15.0	11.5	**17.9**	**1.00**	3
K09/Osigur	69	8.0	2.8	**24.2**	**1.00**	2
M04/Reversi*	48	7.5	2.9	**16.6**	**1.25**	1
L04/Dozvole*	305	15.3	15.9	**19.2**	**1.50**	4
F52/Pot	265	8.0	11.5	**23.0**	**2.25**	5
B05/JCI*	65	4.0	4.0	**16.4**	**3.00**	2
B07/PrUv	299	5.0	8.8	**34.0**	**3.00**	5
B07/Sukc	256	5.0	7.5	**34.1**	**3.00**	5
B05/Bjam	226	3.5	6.5	**34.5**	**3.00**	5
B07/Prlz	288	5.0	8.0	**36.3**	**3.00**	5

* - These projects were pilot projects.

The processed results for 11 observed projects show a range with characteristics of 1:4 for a project duration, and 1:12 for the number of team members. With given data, there is a noticeable range in the measured maturity levels from 1.00 to 3.00 according to the Bootstrap method.

From the results it can be seen that the relation between project duration and working effort is higher in projects performed in organisation units with a higher maturity level and is in the range of 0.36 at maturity level 1.00 to 1.87 at maturity level

[1] Productivity was calculated on the basis of the functional entity of the application delivered to the user, and the total working effort expressed in working months. The standard working month was set at 176 working hours (22 working days).

3.00. The average duration of the project was 7.7 months, and the average working effort 8.2 months, as in Fig. 3.

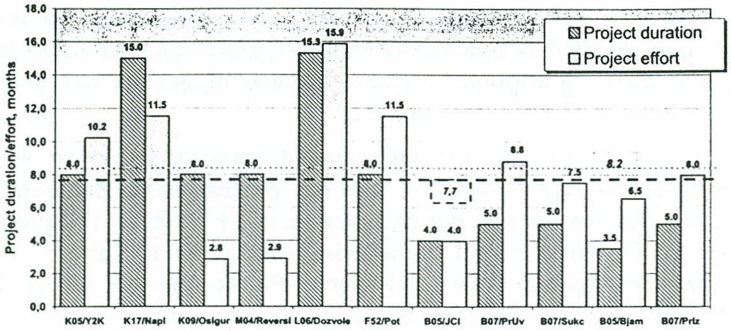

Figure 3. Project duration and effort in months

Regardless of the relatively small sample of measured projects and the number of their characteristics, it can be observed that there is a large aberration between projects carried out in the same organisation, especially with productivity and maturity levels. Therefore, an attempt was made to study in detail the relationship between these two characteristics in order to determine a statistically significant dependence between them, and this was performed as in item 4.

4. STUDYING THE CORRELATION BETWEEN MATURITY LEVELS AND PRODUCTIVITY

4.1. The influence of maturity level on productivity

Within the context of software engineering metrics productivity is the capability of a development environment to deliver a specific number of product units per unit of time. Productivity represents the productive capacity, i.e. the software organisations potential when viewed within the context of plans and projections for the forthcoming period of time.

The possible factors influencing this productivity can be divided into several groups. These are -

- the scope/size and complexity of the problem area and environment
- the size and complexity of those products that need to be delivered
- the characteristics of the human resources (team size, experience, knowledge, skills, etc...)
- the potential of the technology that is currently available
- the architecture, i.e. the life cycle of the software process
- the organisation's characteristics and these include software process management and supervision.

Maturity level management includes the supervision, management and improvement of all aspects that influence the quality of the software product and process, i.e. the success of the software organisation. Since productivity is an internal attribute of the software process, one would expect that maturity management influences productivity.

By taking into account the influence of the aforementioned factors in general, their contribution to the productivity level can be expressed as the following:

$$P = \frac{S(f,c,...)}{TC} \times \eta_h(k,s,m,...) \times \eta_o(m,o,e,..)$$

with denotations:

P	- the productivity for defined performance conditions
$S(f,c,...)$	- the physical size of the system that depends on the required functionality, complexity, etc.
TC	- the potential of the applied technology (software and hardware)
$\eta_h(k,s,m,...)$	- the influence of human factors and these depend on knowledge, skills, motivation, etc.,
$\eta_o(m,o,e,..)$	- the influence of organisational factors such as management levels, the organisation itself, applied engineering methods, etc.

With the exception of $S(f,c,...)$, the physical size of the system as a characteristic of the product itself, which for the most part is determined by parameters outside the software organisation, all other variables in this expression are the subject of maturity management, i.e. their contribution to the maturity level is measured by applying an appropriate method (CMM, Bootstrap).

4.2. Research results

Throughout the conducted research, the results were measured from 11 projects. They showed a range in productivity per project from 7.3 to 36.3 FT/month per team member, see Fig. 4, and in this way significant differences in productivity, that were achieved within the same software organisation, were observed. These differences are within the ratio of 1:5 and this implies a great inhomogeneity between individual development teams.

The differences that have been calculated can be explained primarily by the technological conditions between individual projects, but can also be explained by the influence of other factors. The possible causes of these aberrations can be found by comparing some of the characteristics of those projects with the highest and lowest measured productivity. Some of the characteristics are shown in Table 4, such as the applied CASE and development tools and quality assurance programme implementations, and these in turn can be used as indicators of the maturity level of the project environment.

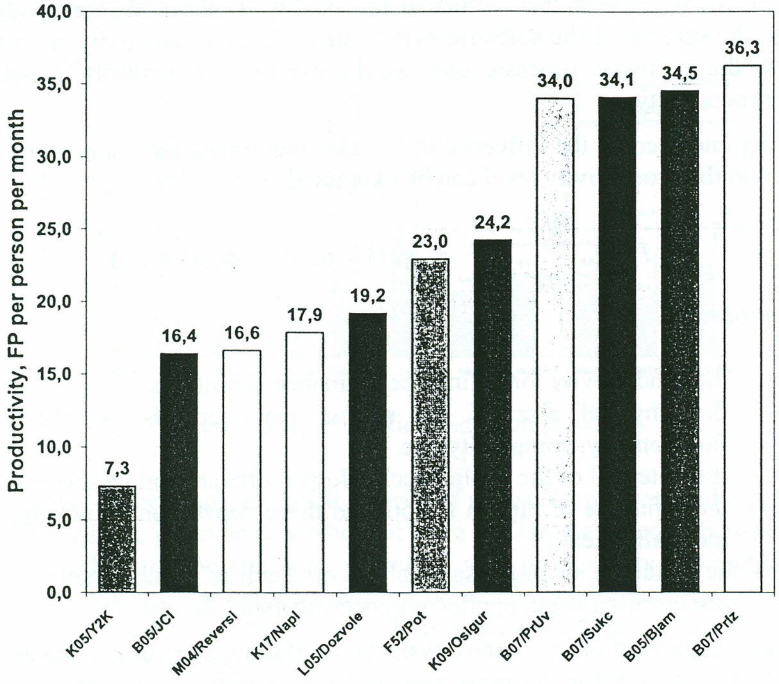

Figure 4. Project productivity in function points per person per month

Table 4. A comparison of the characteristics of projects with the lowest and highest productivity

Project Characteristics	**Project K05/Y2K**	**Project B07/Prlz**
Software project type	Enhancements	Development
Project size in adjusted function points	75	288
Team size	12	5
CASE Tools	No	Yes
Language/Development tool	PL/I (3GL)	VisualAge (4GL)
Implemented QA program	No	Yes
Productivity in adjusted function points per person month	7.3	36.1
CMM/Bootstrap level	1.00	3.00

In order to check the measured productivity a comparison was made with industrial averages for the given project characteristics, see Table 5. Only the data for projects with the highest and lowest measured productivity are shown.

Table 5. A comparison between the measured projects with the highest and lowest productivity and industrial averages for those same project characteristics

Project	Project characteristics[2]	Productivity, FP per person month[3]	Industrial averages[4]		
			Min	Median	Max
B07/Priz	Experienced staff Structured methods CASE tools High-level languages	27.1	20.0	40.0	100.0
K05/Y2K	Experienced staff Unstructured methods Ordinary tools High-level languages	5.5	5.0	10.0	15,0

The data given in this table shows that they are within the expected range of aberrations of industrial averages for the stated software project characteristics. In other words, the two projects shown, along with all the other projects and in accordance with their organisation, their methodological and technical potential, have all fallen within industrial averages.

While analysing the connection between productivity and maturity levels, it was observed coefficient that this connection is very strong. Fig. 5. shows the results that were obtained. A significant level of correlation is noticeable, and this can be based on the curve regression. The correlation coefficient r is in the range between 0.675 and 0.736.

The interpretation of this dependency is not so simple because the manner in which the maturity level is measured includes a whole series of hierarchically placed, influencing parameters, starting with the organisation, the methodology and the technology and these are the highest factors. The statistical significance of this dependency was tested using the t-test, see Table 6, and its existence was determined for each type of curve dependency.

[2] The characteristics that are described are taken from the tables in [5, 247-248]

[3] For comparison with world averages, productivity has been calculated (recalculated) on the basis of a 132-hour working month (instead of the 176-hour one)

[4] Also taken from the tables in [5, 247-248]

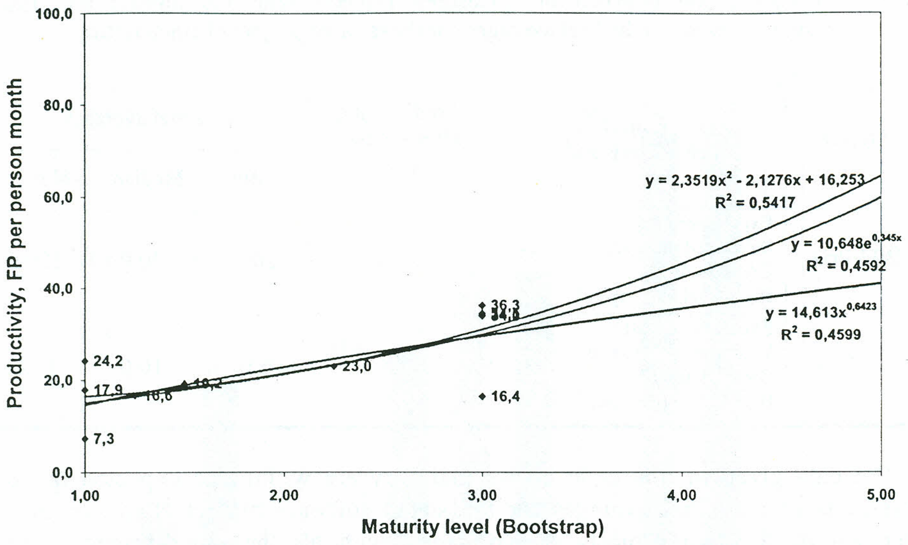

Figure 5. The correlation between the maturity level and productivity in FP per person per month

Table 6. The statistical significance of the correlation between productivity and software organisation maturity

Type of correlation		r^2	r	t	$t_{\alpha=0.05}$	Significance
Polynomial	$y = 2.3519\,x^2 – 2.1276x + 16.253$	0.5417	0.736	3.261563		Yes
Exponential	$y = 10.648\,e^{0,345x}$	0.4592	0.6776	2.764419	1.833	Yes
Potential	$y = 14.613\,x^{0,6423}$	0.4599	0.6782	2.768317		Yes

It should be pointed out that one would expect the determined correlation co-efficient to be more accurate because in the observed case the maturity level was measured for the organisation unit (SPU maturity level), and the productivity was measured for the project level. Measuring the maturity level of each project and cor relation with the corresponding productivities would give greater precision.

5. CONCLUSION

The Bootstrap method measures the so-called maturity level in software development, and this level can be between 1 and 5. Maturity level management demands the implementation of a series of key areas in the software development process. One of these areas is metrics. By analysing 11 projects within the same software producer, a series of observations were obtained. It was proven that productivity expressed in function points grows with an increase in the maturity level.

There is a very strong correlation between these two dependent variables, which are in the range 0.678 to 0.736. It was also the case that very large differences were noticed between individual project teams, (expressed in terms of the productivity that was achieved). This was in a range of 7.3 to 36.3 FT/month within the same organisation. This implied a significant inhomogenity in the workplace and in practice this would need to be eliminated. The results showed that the expected project duration was 7.7 months, and the realised average working effort was 8.2 months. A statistical significance was determined between the maturity levels and productivity.

The results that were obtained show that it is possible to confirm the hypothesis that a higher maturity level has a positive influence on the characteristics of the software development process, i.e. that the Bootstrap method can be used to foresee the range of characteristics of the software process and these characteristics could include productivity, the level of expenses, user satisfaction, etc.

By using all the appropriate information management could initiate numerous improvements in operational work, which would simply not be possible without the use of metrics.

REFERENCES

[1] A. Kuntzmann-Combelles, P. Comer, J. Holdsworth, S. Shirlaw. *The AMI Handbook: A Quantitative Approach to Software Management*. The AMI Consortium. South Bank Polytechnic, London, 1992.

[2] Garmus, D., Herron, D., Measuring the Software Process: *A practical Guide to Functional Measurements*. Yourdon Press, Prentice Hall PTR, Upper Saddle River, New Jersey, 1996.

[3] ISO/IEC FDIS 9126-1:2000 Information technology – *Software product quality* – Part 1: Quality characteristics, 1991.

[4] ISO 9001:2000 *Quality Management Systems* – Requirements.

[5] Jones, Capers. *Applied Software Measurement: Assuring Productivity and Quality*. McGraw-Hill, 1996

[6] Z. Krakar. *Metrike u informacijskim sustavima, bilješke s predavanja na poslije-diplomskoj nastavi*. Fakultet organizacije i informatike, Varaždin, 1997,

[7] *Procjena aplikativnih sektora GZAOP-a metodom Bootstrap* (Gradski zavod za automatsku obradu podataka). Zavod za informatiku Hrvatske, Centar za kvalitetu, Zagreb, 1999.

Received: 13 February 2000
Accepted: 10 April 2000

Zdravko Krakar
Danko Vukadinović

KORELACIJA IZMEĐU BOOTSTRAP RAZINA ZRELOSTI I KARAKTERISTIKA PROCESA U RAZVOJU PROGRAMSKE OPREME

Sažetak

U radu se analizira utjecaj mjerenja karakteristika procesa u razvoju programske opreme na povećanje njegove zrelosti prema Bootstrap modelu. Opisane su temeljne značajke ovog modela kao i metrike koje je nužno uključiti u proces nastanaka programske opreme kako bi se njihovom primjenom utjecale na povećanje razina zrelosti ovog procesa.

Provedeno je mjerenje najvažnijih karakteristika procesa u proizvodnji programske opreme što je načinjeno putem funkcijskih točaka. Također, analiziran je utjecaj razina zrelosti ovog procesa na dostignute produktivnosti i ustanovljena je visoka korelacija. Dobiveni rezultati ukazuju na značaj primjene metrika na povećanje kvalitete procesa razvoja programske opreme.

Ključne riječi: metrika, funkcijska točka, CMM, Bootstrap, programska oprema, razina zrelosti, produktivnost, korelacija.