Feedback from Users on a Software Product to Improve Its Quality in Engineering Applications

Barbara Begier[1], Jacek Wdowicki[2]

[1] Institute of Control and Information Engineering, Poznan University of Technology,
Pl. M. Sklodowskiej-Curie 5, 60-965 Poznan, Poland
[2] Institute of Structural Engineering, Poznan University of Technology
ul. Piotrowo 5, 60-965 Poznan, Poland

Abstract. Users' involvement in a software process is one of strategies for achieving an improvement of software quality. The described research is referred to the software system applied in civil engineering. The continual feedback from users makes possible to learn the user's point of view and to improve a product just according to her/his notes and expectations. The applied method based on a questionnaire survey has been presented − the layout of questionnaires is the original solution. Four iterations of the collaborative development, including software quality assessment, took place. The level of users' satisfaction from a product is currently better than that at the beginning.

1 Introduction

Users assess the quality of software in practice, after the product is delivered. But software quality is born in the software process which is several times repeated till the withdraw of the considered product. Users' involvement in the process is one of possible strategies which support software quality improvement, like the QAW Method [2], for instance. In general, a cooperation with users is the base of agile methodologies [1]. It is not in contradiction with other trends in software engineering, like the MDA (Model Driven Architecture) recommended by the OMG, for example.

The described research is referred to the class of software products which support calculations in an engineering discipline. General characteristics of the considered products are given in the next section. According to the ISO 9000:2000 the *level of a customer satisfaction* from a product is one of recommended measures of its quality. The level of user's satisfaction from a software product derives from its quality, specified using accepted criteria and particular measures of its assessment.

The presented approach is based on a feedback from users in the product life cycle (see section 3). That feedback must respect accepted methodologies and various constraints [7]. The own experience shows that many users are eager to present their suggestions to improve a product [5, 6]. The presented method has been applied to the software system BW applied in civil engineering − it is briefly described in the section 4. Users of the described class of software are highly qualified experts in their discipline who are able to give valuable feedback to software developers. The assessment of software quality takes place after the next version of the product has been introduced. The questionnaire survey is here the recommended form of a software assessment by its users. The content of a questionnaire must be comprehensive and clear for its respondents. The practice shows that one edition of a questionnaire is far not enough to obtain the successful result.

The described method to provide the valuable questionnaire survey has been developed (see section 5), applied several times in practice and successively improved to learn the user's point of view, including notes given to particular metrics and his/her suggestions to improve the product (see section 6). The obtained results make possible to develop the software product exactly to its users' expectations.

2 The Class of Software Supporting Engineering Calculations

Software products supporting calculations of various constructions in civil engineering represent the class of numerical data processing − the numerical input describe a geometry of a construction, expected loads, coefficients of equations, and parameters which control a running software. Data and results are transformed to graphical forms. Generated technical drawings, charts, and maps of stresses contain technical terms and physical units which must be in accordance with technical terminology and notation. Particular programs of calculations often cooperate with the well known AutoCAD system. Authors of this kind of software focus first of all on a correct software construction [8], starting from mathematical and physical models, then programming numerical calculations providing the required accuracy and visualization of results, including image processing.

The strength of wall constructions is calculated each time in a design process of a building − so the supporting software is applied when such a need arises. The progress in civil engineering (new types of constructions, different environmental conditions including various kinds of loads and applied materials) involves the development of a supporting software. Software products required for engineering purposes are maintained many years and evolve till their withdraw. Applied algorithms are based on stable and repeatedly proven principles of mechanics.

Users of the described class of software products are widely regarded as experts in their domain. High qualified users expect an added value − precise results of computations, according to the requirements and the technical standards. The notions of functionality, reliability and safety constitute the canon of their work. Professionals in civil engineering are able to cooperate with software authors.

The following quality elements, to be considered in a production of that class of software, have been specified [3, 4]:

1. Quality of goals of the software project − the specified level of an automation of designer's work is required in the given area; the specification of the object of calculations and sources of loads should be clearly developed.
2. Quality of requirements − their specification includes the correctness of data describing types of analyzed constructions and all possible loads.
3. Theoretical grounds of accepted technical solutions come from the domain analysis, namely the civil engineering in the presented case.

4. Quality of the software process − specification of the software development cycle is worked out, including the required documentation, and then followed.
5. Quality of software construction providing its modifiability and portability.
6. Quality of interfaces between program modules and the provided proper access to their common data.
7. Specification of software quality criteria and their particular measures to provide a product assessment by its users.
8. Quality of testing − provided test cases assure the correctness of results and make them reliable (although it is not easy to provide the real life data).
9. Quality of the user interface including graphical forms of presentations.
10. Expected verifications of the input data.
11. Required cooperation with the AutoCAD system.
12. Structure, content and language of the documentation, including documents of all changes, test cases, and instructions for users.

All quality elements should be developed, discussed, assessed, and modified by software designers, domain experts and potential users. The domain experts participate obligatory in the software process to provide the quality elements denoted at 1−3 and 8. They also may cooperate with quality engineers and software designers to provide the other quality elements, especially those 4−7 here. In any engineering calculations the correctness of the input data play the more important role than in other fields – any mistake here may result in a construction disaster. At least the elements denoted above from 9 to 12 should be assessed, in the authors' opinion, by all users of the given software product. This statement is developed in the next sections.

3 Software Life Cycle Including a Feedback from Users

Software products applied in civil engineering are developed and improved in their entire life cycle. In the described approach the main source of ideas to improve the product comes from its users, as shown in the Figure 1. The following steps are recommended to provide a regular feedback from users on the given product:

1. Get learn and specify the users' profiles.
2. Specify and plan the forms of users' involvement in the process, mainly in a quality assessment of a product.
3. Specify several dozen of quality criteria and measures.
4. Design the questionnaires with some place left for suggestions of changes.
5. Carry out a survey.
6. Collect results and process them statistically.
7. Record and analyze carefully all suggestions and proposals.
8. Compare current values of given notes with the previous results.
9. Select measures to be improved, choose some suggestions and start the next development cycle to build a new version.

Thus the provided answers and their analysis constitute a basis of software modifications and improvement − one may ***control those features*** which *are specified*, then *implemented*, *measured*, *assessed*, and possibly *improved* in the next iteration.

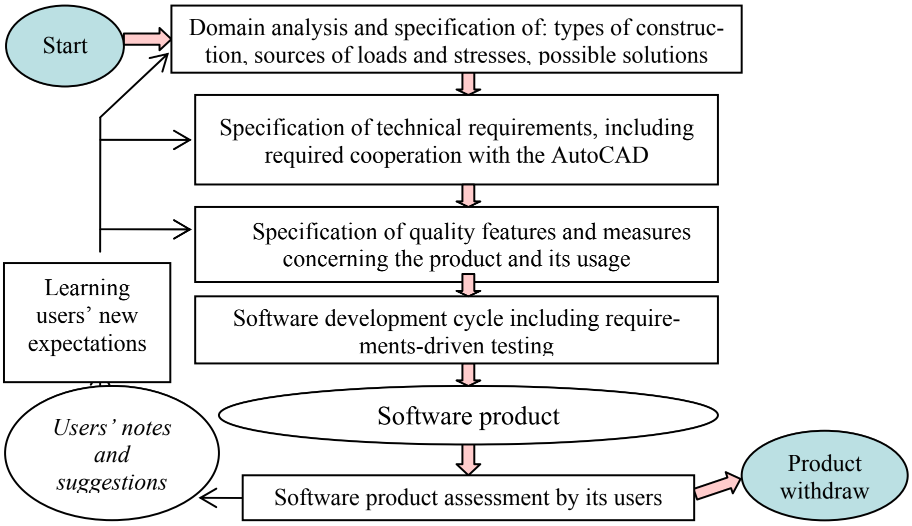

Start
Domain analysis and specification of: types of construction, sources of loads and stresses, possible solutions
Specification of technical requirements, including required cooperation with the AutoCAD
Specification of quality features and measures concerning the product and its usage
Learning users' new expectations
Software development cycle including requirements-driven testing
Users' notes and suggestions
Software product
Software product assessment by its users
Product withdraw

Fig. 1. Software life cycle of a product supporting engineering applications

4 Object of an Assessment − the BW System

The integrated software system BW (its name is an acronym from '*Tall Buildings*' expressed in Polish) has been developed to support the static and dynamic analysis of multi-storey buildings with large three-dimensional coupled shear wall structures. All analyses should be accurate and fast − the calculations are not possible without the computer support. The considered system provides calculations of internal forces, stresses and displacements due to the set of lateral and vertical loads and flexural moments. The static analysis is based on the developed variant of the continuous connection method using differential equations for the usually expected and the extreme stresses, the shear and normal ones [12]. The BW system enables also to analyze the flexural strength of a tall building − its dynamic analysis is performed using the hybrid continuous-discrete approach and the response spectrum technique to estimate the seismic response of a construction [14].

The BW system has evolved for many years. Its maintenance requires to respect principles of software engineering and to pay a special attention to software quality [3, 13]. Several programmers cooperate with the main authors of the technical solution. Its current version follows the Eurocode 8 standard, accepted by CEN (Comité Européen de Normalisation) in 2003. It analyses torsions of the coupled construction due to its displacements resulting from earthquakes or para-seismic effects. The BW system is intended to cooperate with the AutoCAD system to get dimensions of a construction from its technical drawings and also to transfer numerical values in the opposite direction to check visually the correctness of the given data. The BW system generates numerical and graphical results. It has been applied many times in practice to calculate ca 240 constructions, in the home country and abroad.

5 Quality Measures of Software for Civil Engineering Applications

The quality metrics have been specified during four years of work concerning quality of software products developed for civil engineering. Quality criteria and measures result from quality goals of the software project. Quality measures are, at first, the part of software requirements and then they become a subject of a product assessment by its users. The particular metrics have been worked out on a base of quality attributes given in the ISO 9126 [9] and as a result of cooperation with users. The developed decomposition of quality attributes is given in the Figure 2.

The scope of software ***functionality*** is related to the range of required types of constructions and their loads. The provided cooperation with the AutoCAD system and the data correction also belongs to the product functionality.

The main part of the software product assessment concerns its ***usability***. It refers to the software construction, user interface, ease of learn, and ease of use of the software product. Comprehensive software construction corresponds to subsequent steps of a design process and to designer's decisions made on a base of the obtained values. It is important feature from the user's point of view to provide an easy way to change the former set of input data and to perform calculations starting from the chosen point, not only from the very beginning. An interactive impact on calculations and a possibility to trace them seem to be valuable. User's interface should provide conformity with the terminology and notation used in the given technical discipline.

The usually long period of a program maintenance requires the ***portability*** of the product including an ease of installation on various computer platforms.

All quality measures along with user's subjective impressions constitute the main metric, namely the ***level of user's satisfaction*** from the considered product.

The following measures have been specified and then assessed:

Functionality

M1. Variety of building constructions analyzed by the product.

M2. Sufficient number of various load cases analyzed by the product.

M3. Mode of cooperation with the AutoCAD system, providing an effective way to check the correctness of the input data.

M4. Possibility for presenting subsequent steps of calculations.

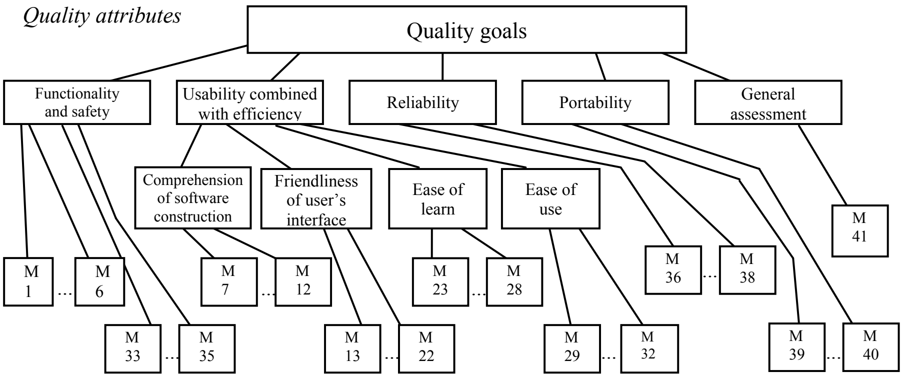


Fig. 2. The quality tree joint for software products applied in civil engineering

M5. Provided on-line correction of data describing the given construction.
M6. Correction of data concerning loads of the construction, including a withdraw of the previously given values.

Comprehension of software construction

M7. Comprehension of all provided software options and their use.
M8. Clarity of software construction and the provided ease of an access to software functions at any moment of use of a product.
M9. Ease of a program navigation using an icon tool bar.
M10. Readable and clear access to software functions by making use of menus.
M11. Provided way to prepare the input data.
M12. Availability of lists of choice for the input data.

Friendliness of screen elements provided by a user interface

M13. Comprehension of the content of lists of choice.
M14. Layout (arrangement) of objects presented on a screen.
M15. Size of character types and other screen elements, including graphics.
M16. Readability and comprehension of legends attached to drawings and charts.
M17. Provided kind and range of physical units.
M18. Conformity of labels, used next to elements of charts and drawings, with the technical notation applied in the given area.
M19. Comprehension of language expressions used in inscriptions and messages; ease of their interpretation (avoiding misinterpretation of them).
M20. Sufficient support with graphical forms.
M21. Proper use of colours applied to screen objects and a background.
M22. Ability to customize views of screen objects to the user's likings.

Ease of learn to use the software product

M23. The time required to learn to use the program.
M24. The initially required help of the program experienced user.
M25. Estimated frequency of mistakes made in one hour of a software use.
M26. Frequency of currently made mistakes compared with that at the beginning.
M27. Usefulness of the provided user's instruction.
M28. Usefulness of the built-in help option.

Ease of use of the software product

M29. Required help of an experienced user after the initial period of learn to use.
M30. Frequency of using the provided forms of help (in a given time period/unit).
M31. Feeling of a comfort of work at any moment during the use of the product.
M32. Sufficient number and clear content of the provided messages and alarms.

Safety of software and data

M33. Protection against an unauthorized user's access.
M34. Possibility to run a program for the incorrectly specified construction.
M35. Possibility to run a program for the incorrectly given load values.

Reliability

M36. Constant standby and ability to operate data (no unpredictable suspensions).
M37. Stability of program performance and a lack of unexpected side effects.
M38. Capability to maintain the specified level of performance under the stated conditions (the same parameters of performance after the multiple use of a product).

Portability

M39. Capability to run on the specified various platforms.
M40. Ease of program installation on any computer platform.

General assessment of the software product

M41. Level of users' satisfaction from the given software product.

6 Software Product Assessment Using a Questionnaire Survey

6.1 Structure and Content of Questionnaires

Each questionnaire includes an initial part which contains questions concerning users' experience with computing tools. Respondents have been asked about software products used at their every day work − a number and a kind of software systems of general purpose, the frequency of using each product, and separately distinguished tools suitably supporting professional applications in civil engineering.

The main part of a questionnaire has been developed to assess, first of all, the usability of the product. In its two first editions this part contained 26 questions, grouped into 3 sections concerning subsequently: user interface, ease of learn, and user's efficiency as a result of the provided ease of use [6]. In two last editions this part has been increased up to 41 questions, containing all measures quoted in the previous section. At the end, users are asked about their general assessment of a given product. An additional point in all four editions was a request for user's suggestions to improve the BW system.

The content of a questionnaire have been improved every year. The aim was to achieve process-able and comparable results. Each group of items has its name corresponding to the quality criterion to keep user's attention on a given subject. Expected forms of an answer should be clearly explained and suggested.

Two first editions showed that some questions were not equally interpreted by all respondents. In a consequence, various forms of answers were given. For example, the question "*A frequency of mistakes made at the very beginning of the software use compared to the current frequency of such mistakes*" was imprecise. The answers were like: a ratio (4:1, for example), number of times, percentage of cases during the use of software, and also in words like "*none*" or "*several*".

Table 1. The fragment of a questionnaire

M	Ease of learn to use the software product	
24	The initially required help of the program experienced user [1 – necessary, 2 – desired, 3 – partial, 4 – occasional, 5 – unnecessary]	
25	Estimated frequency of mistakes made in one hour of software use [1 more than 10, 2 – from 6 up to 10, 3 – from 3 up to 5, 4 – 1 or 2, 5 – zero]	
26	Frequency of currently made mistakes compared with that at the beginning [1 – no difference, 2 – a bit less, 3 – less than a half, 4 – rarely, 5 – by no means]	
27	Usefulness of the provided user's instruction [1 – bad, 2 – insufficient, 3 – sufficient, 4 – good, 5 – very good]	

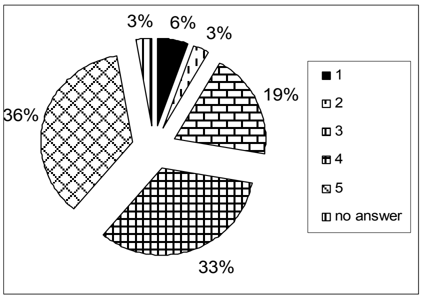
The need arose to specify precisely the suggested unit of each answer. The set of acceptable values of an answer should be attached to each question. The most comparable are numerals and thus just integers have been suggested as expected forms of answers. In next editions of the questionnaire, the possible forms of each answer along with their meanings were given in square brackets right behind each question as it is illustrated in the Table 1. The last column is reserved for answers.

6.2 Overview of the Obtained Results

The subsequent software quality assessments using a questionnaire survey took place four times − every year, since 2002. The 38 fulfilled copies were obtained in 2002, then 41 in 2003 [6], 36 in 2004, and 30 in 2005. It took 10 up to 15 minutes for each respondent to fulfil a questionnaire. An important question should be answered at the very beginning of the questionnaire survey − if problems in use the considered software product arise from a possible lack of users' experience at computing and software tools, or not. Answers given to the questions in the initial part have confirmed that civil engineers are skilled in computing – all of them use computers every day, all are experienced with popular tools of general purpose like the MS Word and MS Excel. All know and use the AutoCAD system.

All answers given by respondents during each edition of the questionnaire have been entered into the report sheet. Its each row is designed for one question and related answers. The first column points out the number of a question, and the next ones contain provided answers. The following data are calculated and maintained in last five columns: given minimum value, maximum value, sum of provided values, number of answers, and the average of provided answers. The fragment of the report sheet (limited to 15 respondents) obtained in the last edition is given in the Table 2.

In some cases of metrics, for example to assess the '*Feeling of comfort of work at any moment during the use of the product*' or the '*Sufficient number and clear content of the provided messages and alarms*', the average values do not make sense. So despite the comparable numerical results of the assessment the descriptive statistics has been done to present all obtained data on graphs. Two examples of graphs presenting statistical data are shown in the Figure 3 and the Figure 4.



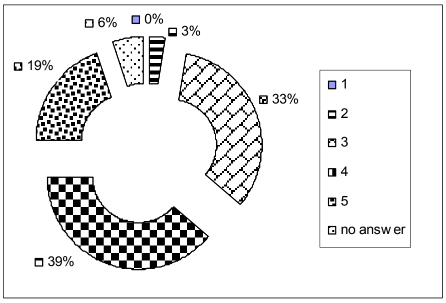


Fig. 3. Results of assessment of the M11 ('*Provided way to prepare the input data*')

Fig. 4. Results of assessment of the M5 ('*Provided on-line correction of data ...*')

Table 2. Fragment of the report sheet

User / *Question*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	*min*	*max*	*sum*	*n*	*average*
Functionality																				
M1	5	5	5	5	3	5	5	0	3	5	5	5	3	5	5	3	5	64	14	4.57
M2	5	5	5	3	5	5	5	5	3	5	3	5	5	5	5	3	5	69	15	4.60
M3	3	4	5	5	4	4	3	4	4	3	4	3	5	4	5	3	5	60	15	4.00
M4	4	4	5	4	4	4	5	0	4	4	4	4	4	4	4	4	5	58	14	4.14
M5	3	0	2	4	4	4	4	5	4	3	4	4	3	4	4	2	5	52	14	3.71
M6	3	0	2	4	4	4	4	5	4	3	4	4	4	4	3	2	5	52	14	3.71
Comprehension of software construction																				
M7	5	4	5	4	5	5	5	5	3	4	4	5	4	5	4	3	5	67	15	4.47
M8	4	4	4	5	5	5	5	5	3	5	4	5	4	4	5	3	5	67	15	4.47
M9	4	5	4	5	5	5	5	5	5	5	4	4	4	5	5	4	5	70	15	4.67
M10	4	4	4	5	5	5	5	5	4	4	4	3	4	5	5	3	5	66	15	4.40
M11	4	5	4	5	4	4	4	5	5	4	5	4	4	5	5	4	5	67	15	4.47
M12	4	5	4	5	5	5	5	5	5	4	5	3	4	5	4	3	5	68	15	4.53
Friendliness of screen elements provided by a user interface																				
M13	4	5	4	5	5	4	4	4	5	4	4	4	3	5	3	3	5	63	15	4.20
M14	4	4	5	5	5	5	5	5	4	4	4	5	5	4	4	4	5	68	15	4.53
M15	3	3	5	5	5	5	5	5	4	4	4	5	5	5	4	3	5	67	15	4.47
M16	4	4	4	3	4	5	5	5	3	4	5	4	4	5	4	3	5	63	15	4.20
M17	5	5	4	5	5	5	5	5	4	4	4	4	5	5	4	4	5	69	15	4.60
M18	5	4	5	4	4	4	4	5	3	4	4	4	4	5	4	3	5	63	15	4.20
M19	4	4	5	5	4	5	4	5	5	5	5	4	5	5	4	4	5	69	15	4.60
M20	5	3	5	5	4	5	5	5	5	4	5	4	5	5	5	3	5	70	15	4.67
M21	5	4	5	5	5	5	4	5	5	4	5	4	5	5	5	4	5	71	15	4.73
M22	4	3	3	3	4	3	4	0	2	3	4	2	4	5	5	2	5	49	14	3.50
Ease of learn to use the software product																				
M23	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	4	5	74	15	4.93
M24	3	5	4	2	2	3	4	4	2	2	2	4	1	4	4	1	5	46	15	3.07
M25	4	5	4	5	5	4	4	5	5	4	4	5	3	5	5	3	5	67	15	4.47
M26	4	5	4	4	5	4	5	0	1	4	4	5	3	5	5	1	5	58	14	4.14
M27	4	4	4	4	3	3	4	0	2	3	4	4	3	4	3	2	4	49	14	3.50
M28	3	4	3	3	3	3	4	0	3	3	4	4	3	4	3	3	4	47	14	3.36
Ease of use of the software product																				
M29	4	5	5	4	5	5	5	5	0	4	3	5	5	4	5	3	5	64	14	4.57
M30	4	4	3	4	4	4	4	5	3	3	3	4	3	4	5	3	5	57	15	3.80
M31	5	5	5	5	5	5	5	5	5	5	3	5	3	5	5	3	5	71	15	4.73
M32	4	5	4	5	5	5	5	0	3	4	4	5	4	5	3	3	5	61	14	4.36
Safety of software and data																				
M33	4	0	3	2	3	3	4	5	1	2	4	4	1	4	3	1	5	43	14	3.07
M34	3	3	4	4	3	4	4	0	3	4	3	4	3	4	3	3	4	49	14	3.50
M35	3	3	4	4	3	4	4	0	3	3	3	4	3	4	3	3	4	48	14	3.43
Reliability																				
M36	4	5	4	5	5	5	4	5	4	4	5	5	3	4	4	3	5	66	15	4.40
M37	4	5	4	5	4	5	3	5	4	4	5	5	3	4	5	3	5	65	15	4.33
M38	4	5	4	5	4	5	4	5	4	4	5	5	4	4	5	4	5	67	15	4.47
Portability																				
M39	4	0	4	5	5	3	4	0	3	4	4	4	4	5	4	3	5	53	13	4.08
M40	2	5	4	5	5	5	5	5	5	4	5	4	5	5	5	2	5	69	15	4.60
General assessment of the software product																				
M41	4	4	4	5	3	4	5	5	3	4	4	4	0	4	0	3	5	53	13	4.08

The general assessment of the product is relatively high (4.08). And 30 from among 40 particular measures have gained the average notes higher than 4.0. The facilities of data correction are the most difficult to improve them. It seems to be the general problem concerning any kind of software product − how to provide the satisfying level of verification and correction of data.

The highest average notes gained all measures concerning the comprehension of software construction and its reliability. The worst notes were given to the measures of safety of programs and their data − fortunately, these are not the key measures in the case of software products supporting applications in civil engineering.

In some metrics the maximally different results have been obtained. For example, in the case of M24 and M26 the range of values is from 1 to 5. There is a correlation between an experience in using the AutoCAD system and other software for engineering applications – users who assess their experience in using those products as the average require the help of an experienced user. The other explanation associates a user with his/her professional environment − respondents who cooperate with experienced users do not complain about the ease of learn to use the product.

7 Software Quality Improvement Resulting from the Users' Involvement in a Product Assessment

All notes and suggestions given by users have been carefully analyzed after each questionnaire survey. Fortunately, many of the polled users show their willingness to influence on a software product − each second respondent gave his/her particular suggestions, although the weight of them was not equal − from a suggestion of using boldface elements or letters on a specified chart, to identify a danger of loosing some data in the case of a too fast click in the described circumstances. Users' remarks and expectations regarding each new version of the BW product concern:

a) improvement of some specified software features and a system usage,
b) close cooperation with the AutoCAD system,
c) an increasing number of types of analyzed building constructions.

The chief of an assessment and the developers' leader (both are the authors of the presented paper) have worked out three ***enumerated lists of suggested changes***, separately for each kind of the listed above problems. Each item on a list has its several initially specified attributes: number, name, date of registration, weight (assigned using three different colors), description, justification (goal and related quality measures), addressee. The developed lists are transferred each time to programmers who then are obliged to fulfill several other attributes and explain what real modifications of the software product have been made. Thus the history of all changes and their justification is maintained − it also shows the favorite areas of programmers' work. The reality shows that programmers are eager to improve first of all the interface with the AutoCAD system. Thus the new or widely modified pre- and postprocessors cooperating with the AutoCAD have been subsequently developed. But the note given to the mode of cooperation with the AutoCAD (M3) remains the same, namely 4.00.

Authors of the BW system have taken into account many critical remarks concerning their product and have eliminated its some weak points pointed out in the questionnaire survey. So the directions of software product improvement follow exactly the users' notes, given opinions, and submitted suggestions.

Average values of most measures have grown since the previous assessment. For example, the variety of building constructions analyzed by the program BW (M1) has gained the average note 4.57 (previously 3.94). The comprehension of all provided software options and their use (M7) arose from 4.12 up to 4.47, the provided way to prepare the input data (M11) previously achieved an average 3.71 while 4.47 last time, and the availability of lists of choice for the input data (M12) got 3.94 and 4.53, respectively. The estimated frequency of mistakes made in one hour of software use (M25) has been currently assessed much better (4.47) than before (3.47). Also the average note given to the frequency of currently made mistakes compared with that at the beginning (M26) has grown from 3.82 up to 4.14. The feeling of comfort of work at any moment during the use of the product (M31) arose from 3.94 up to 4.73.

Usefulness of user's instruction (M27) and that of the built-in help option (M28) is still problematic and even a bit worse than previously: 3.50 and 3.36 instead of 3.88 and 3.56, respectively. So some help of the program experienced user (M29) is still required at the beginning (it got the average note 4.57 recently, and previously 3.82).

Again and again the data correction is assessed as a weakness of the product − the provided on-line correction of data describing the given construction (M5) and the similar measure related to the loads of the building (M6) are both currently assessed equally as 3.71 and are a bit worse than previously (3.76 and 3.88, respectively).

After four iterations of software development including periodical quality assessment of the product by its users, the average level of users' satisfaction from a product is slightly better (4.08) than that obtained the previous time (4.0).

8 Conclusions

Evaluation of software quality based on data coming from software users is the current problem in software engineering. The experience using a questionnaire survey to gain a continuous feedback from users on a software product has been described. Conclusions are related to: the form and the layout of the questionnaire, the obtained results, and the general idea of the applied approach. Some remarks concerning the content of the questionnaire are given in the section 6.1. After the careful analysis there are still some doubts if names of metrics and expressed possible answers are phrased well. Maybe the Likert scale will be used in the next edition of the questionnaire. In this method each item of the questionnaire is the statement and the respondents are asked to indicate their degree of agreement with the provided statements.

Users' expectations are growing every year. It is a real risk − if a software product remains unchanged then the general note given in its assessment may be each time worse than it was before. Notes are given to the particular measures of the software product in the changing environment − the level of average user's skills is growing. Also a range of users' expectations increases according to the hierarchy of needs described by Maslow [10] − users propose new requirements after their basic needs are satisfied. So the gained results of questionnaires may surprise when strictly compared − the new results may be worse than previous ones, although many improvements have been done in the meantime.

Despite all differences between the software products applied in civil engineering and any other software *the idea to incorporate users to the software process* looks promising. At least the ***periodical assessment of the software product*** by its available (possibly all) users is worth working out and decidedly recommended here. The assessment is based on the questionnaire survey. The content of a questionnaire and all forms of the expected answers should be carefully specified to get valuable results and to avoid misunderstandings. In other words, the questionnaire itself should be periodically improved, too. Users' answers and suggestions are a base of the specification of required changes. The history of all justified changes is maintained.

References

1. Agile Modeling(AM) Home Page, http://www.agilemodeling.com/, 2001−2005
2. Barbacci M. R., Ellison R., Lattanze A. J., Stafford J. A., Weinstock Ch. B., Wood W. G.: Quality Attribute Workshops (QAWs), 3rd edn, CMU/SEI-2003-TR-016, Carnegie Mellon Software Engineering Institute, Pittsburgh (USA), August 2003, http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03tr016.pdf
3. Begier B., Wdowicka E., Wdowicki J.: On the methodics providing software quality in civil engineering applications. In: Tasso C., Adey R. A., Pighin M. (eds.): Software Quality Engineering. Computational Mechanics Publications, Southampton-Boston (1997) 71−80
4. Begier B., Wdowicki J.: Quality criteria of software provided for the calculations of constructions in civil engineering (in Polish). In: Tadeusiewicz R., Ligęza A., Szymkat M. (eds.): The 3rd National Conference on Methods and Computer Systems in Research and Engineering MSK'01, Conference proceedings. Kraków (2001) 233−238
5. Begier B.: Software quality assessment by users. In: Huzar Z., Mazur Z. (eds.): Problems and methods of software engineering. Wydawnictwa Naukowo-Techniczne (2003) 417−431
6. Begier B., Wdowicki J.: Quality assessment of software applied in civil engineering by the users. In: 4th National Conference on Methods and Computer Systems in Research and Engineering MSK'03 (in Polish). Kraków (2003) 547−552
7. Begier B.: The UID Approach – the Balance between Hard and Soft Methodologies. In: Zielinski K., Szmuc T. (eds.): Software Engineering: Evolution and Emerging Technologies. IOS Press, Amsterdam (2005) 15−26
8. Cooke J.: Constructing Correct Software. The Basics. Springer Verlag, London (1998)
9. International Standard ISO/IEC 9126-1:2001, Software engineering – Product quality, Part 1: Quality model. ISO Copyright Office, Geneva (2001)
10. Maslow A.: Motivation and Personality, Harper and Row (1954)
11. Sikorski M.: Usability management in information technology projects (in Polish). Wydawnictwo Politechniki Gdanskiej, Gdansk (2000)
12. Wdowicki J., Wdowicka E.: System of Programs for Analysis of Three-Dimensional Shear Wall Structures. The Structural Design of Tall Buildings, Vol. 2 (1993) 295−305
13. Wdowicki J., Wdowicka E., Tomaszewski A.: Integrated System for Multi-storey Buildings – Use of Software Engineering Rules. In: European Conference on Computational Mechanics ECCM-2001, Cracow, Poland (2001) 1−20
14. Wdowicka E., Wdowicki J., Błaszczynski T.: Seismic analysis of the 'South Gate' tall building according to Eurocode 8. In: The Structural Design of Tall and Special Buildings, Vol. 14, John Wiley & Sons, Ltd, 1 (2005) 59−67