Artículo de investigación

Creativity-oriented software development

Разработка программного обеспечения ориентированная на творчество

Desarrollo de software orientado a la creatividad

Recibido: 10 de julio del 2019 Aceptado: 24 de agosto del 2019

Written by:
Alexei I. Razumowsky[281]
ORCID: 0000-0003-1769-0178
SPIN: 6169-9374

Abstract

This Article describes principles and a practice to support individual creative activity in software development. It shows how significant the visual means of development are, by which any insufficient information is supplemented, and an impact on the person's short term memory is reduced. The framework conditions of a dominant idea of a creative human factor are determined, namely, personal experience, comfort and availability heuristic. For the purpose of localization in a single visual data context, a creative and contextual form is proposed that serves for various possible information elements to be placed therein and further recorded, and this gives rise to a creative insight of another solution. An example of how such software system is implemented, and a sample for an algorithm developed by it are presented. A description is given for an Eureka software model that supports a developer's creative process activity. The scientific novelty of this Article is methodological principles and a practice to support an individual creativity comfort in software development. In circumstances where data losses are neutralized, and thus the total volume of appropriate information is increased, the reduction of an impact on the developer's short term memory is achieved, which leads steadfastly to acceleration of solutions and to improvement of their quality.

Keywords: Creative process, creative-contextual form, individuality of the solution, insight, availability heuristics.

Аннотация

В статье описаны принципы и практика поддержки индивидуальной творческой активности при разработке программного обеспечения (ПО). Показана важность визуальных средств разработки, посредством которых организуется восполнение недостаточной информации и снижается нагрузка на кратковременную память человека. Определены базовые условия доминанты творческого человеческого фактора: личный опыт, комфорт и эвристика доступности. Для позиционирования в едином визуальном контексте данных предложена креативно-контекстная форма, которая служит размещению на ней и последующей фиксации всевозможных информационных элементов, что провоцирует творческий инсайт очередного решения. Приведен пример реализации такой программной системы, а также образец алгоритма, разработанного ее посредством. Описана модель программного комплекса «Эврика», реализующего обслуживание активности творческого процесса разработчика. Научной новизной статьи является методологические принципы и практика поддержки комфорта индивидуального творчества в разработке ПО. В условиях нейтрализации информационных потерь, а значит, увеличения общего объема необходимой информации, достигается снижение нагрузки на кратковременную память разработчика, что неуклонно ведет к ускорению принимаемых решений и повышению их качества.

Ключевые слова: творческий процесс, креативно-контекстная форма, индивидуальность решения, инсайт, эвристика доступности.

[281] Candidate of Technical Sciences, Associate Professor, Senior Researcher V.A. Trapeznikov Institute of Control Sciences RAS, Moscow State University of Technology "STANKIN". Moscow, Russia

Resumen

Este artículo describe los principios y una práctica para apoyar la actividad creativa individual en el desarrollo de software. Muestra cuán significativos son los medios visuales de desarrollo, mediante los cuales se complementa cualquier información insuficiente, y se reduce el impacto en la memoria a corto plazo de la persona. Se determinan las condiciones marco de una idea dominante de un factor humano creativo, a saber, experiencia personal, comodidad y disponibilidad heurística. Para fines de localización en un solo contexto de datos visuales, se propone una forma creativa y contextual que sirve para que varios elementos de información posibles se coloquen allí y se registren más, y esto da lugar a una visión creativa de otra solución. Se presenta un ejemplo de cómo se implementa dicho sistema de software y una muestra de un algoritmo desarrollado por él. Se proporciona una descripción de un modelo de software Eureka que admite la actividad de proceso creativo de un desarrollador. La novedad científica de este artículo son los principios metodológicos y una práctica para respaldar la comodidad de la creatividad individual en el desarrollo de software. En circunstancias en las que se neutralizan las pérdidas de datos y, por lo tanto, se aumenta el volumen total de información apropiada, se logra la reducción de un impacto en la memoria a corto plazo del desarrollador, lo que conduce constantemente a la aceleración de las soluciones y a la mejora de su calidad.

Palabras clave: Proceso creativo, forma contextual creativa, individualidad de la solución, perspicacia, heurística de disponibilidad.

Introduction

We will understand a creative software development as an informal and individually detached process of a stepwise accumulation, transformation and structuring of algorithmic solutions. We should highlight an individual search for and adoption of solutions in the proposed conception. This is necessary because current practices to develop software by templates, instructions or various valuation metrics cater to data consolidation and abstraction. Furthermore, a developer limits himself to conditions where he must follow some prescribed operating steps. That is to say, a software engineer is held hostage to the development technology, and his creative activity is suppressed significantly.

Anyway, ways to enhance effectiveness of software development including analysis, designing and monitoring its productivity, are now explored in technological innovations. A creating role of a developer himself, his qualities, experience, knowledge and responsibility remain over the horizon.

A search for effective technological solutions is often aimed at forming and manipulating numerical and abstract information, which will not provide a solution to a problem of full assimilation by a person of data used. This study makes an effort to fill such gap and defines as its objective the exclusion of multiple data losses at different stages and levels of software solution development. It seeks to arrange for information to actually get into a person's consciousness, for accumulation of various redundant data with imaging tools used, and for preservation of individually significant values that correspond to a creative insight of a solution, and for further reproducing of them again, when required, to continue the development.

The issues of full accumulation by a person of information used in software development are caused by an insufficient support for human creativity, and, with regard to existing software technologies, can be defined as follows:

- Insufficient information in designing and algorithmization;
- Project data unification and depersonification;
- Leak of individually significant information.

Such issues have two main causes:

- Software process rationalisation;
- Unification of software development tools.

Seeking a rational approach in software development and usage is reasonable and natural. However, there is another point of view, "we will never find a process that could give us an opportunity to design applications in a strictly rational way" (Parnas and Clements, 1986). But it was further noted that, "if we bear in mind an ideal process, it becomes easier to measure a project success". In other words, a developer's creativity, as the authors believe, should be in accordance with specific rules of a software process, which will allow for its controllability. Consequently, the main issue of software development is an issue of harmonious interaction of creativity and reasonableness in a search for, development and adoption of solutions. It also needs to be understood that narrowing a freedom of a developer's creativity leads to a poor professional growth, a lack of initiative or negligence (Stroustrup, 2013). Furthermore, an opinion is expressed in the work (Ackoff and Emery, 1972) that, "restrictions imposed by common logic and abstract mathematics on solution procedures make it almost impossible to study truly creative solutions and consign a science of solutions to a combination of perfunctory and, for this reason, boring and unvaried examples of solution adopting". In software development, an opportunity is needed to make all appearing ideas having graphic representation explicit. Any ideas created, associations and estimations need be made explicit because of the person's limited short term memory identified by Miller's Magical Number 7 +-2 (Miller, 1956).

Literature review and analysis

Today the principle area to improve the effectiveness of software development is to increase understanding of a source code, dark data, various data charts and design patterns. Visualization of various information is closely linked to understanding. Software visualization and information visualization emerged as far back as in the 1980-90s, but a study of the processes of interdependence and replaceability of human-computer interaction system elements still remains a blind-spot. We will be interested in the aspect of quality and features of the process in which a person makes a decision on algorithm development or correction. Among the most popular works in information visualization, the works (Petre, 1995; Lanza, 2003; Lanza, Ducasse, Gall and Pinzger, 2005; Thomas and Cook, 2005; Ware, 2000; Tufte, 1997, 2001) stand out. Software visualization is aimed at supporting understanding of a structure of software systems and algorithms, their analysis and error research with respect to content correction (Petre, 1995; Lanza, 2003). It is stated in the work (Ware, 2000) that the visualization allows for understanding vast amounts of data while difficulties in understanding often occur at a level of interpretation of excessively simplified information (Petre, 1995). In order to perceive information better, the work (Lanza, 2003) offers a number of metrics providing software solution developers with evaluatable information relevant to the code, namely, a hierarchy structure, a size and coherence of classes and methods, and attributes used. Visualization tool CodeCrawler offered in the work (Lanza, Ducasse, Gall and Pinzger, 2005) uses 2D spaces to visualize an object-oriented software. In the work (Thomas and Cook, 2005), information visualization was focused on creating approaches to provide abstract information in intuitive ways. The works (Tufte, 1997, 2001) argue that visualization errors may be minimized by means of changing a density with which any useful information is given.

In addition to direct matching up the understanding of information and its visualization, there are also the works related to analyzing and monitoring individual elements of software environments. In addition to enhancing the effectiveness of a software code analysis, some tools are offered to adjust project architectural solutions (Aghajani, Bavota, Linares-Vasquez, Lanza, 2019; Caracciolo, Lungu, Truffer, Levitin, Nierstrasz, 2016; Bacchelli, Cleve, Mocci, Lanza, 2017; Würfel, Lutz, Diehl, 2016; Baltes, Moseler, Beck, Diehl, 2015). Sufficiently significant efforts pursued by researchers are geared to a search for means to monitor a task integrity and to delete errors of the program performance (Würfel, Lutz, Diehl, 2016; Baltes, Moseler, Beck, Diehl, 2015).

Researchers engaged in visualization issues and adjustment of information, completely overlook a factor which, in our opinion, is the only factor that affects the success of a software process. The main accumulating force of a person that can overcome ambiguity of a subject, a proposed idea, a relative point of view and a sensitive solution is its creative work. The researches on the phenomenon of creativity are around 100 years old, during which it is managed to identify its stages, characteristics, and to connect it to behavioral motives and drives. However, due to the irrationality of creativity, it is still quite difficult to gauge the actual origin of creativity.
Multiple works in psychology, training and analysis are nowadays dedicated to studying creativity and factors contributing to its better manifestation and realization (Lopes, 2017; Yuan, 2016; Himaieva and Nykyforova, 2015; Byrge and Tang, 2015 Chang, Chien, Yu. et al., 2016). The work (Yuan, 2016) shows the paths to develop students' independent behavior, proactivity and curiosity through fostering their creativity. The work (Himaieva and Nykyforova, 2015) refers to verbal and nonverbal creativity in training. The work (Byrge and Tang, 2015)

discusses how to understand training materials better. The work (Chang, Chien, Yu et al, 2016) studies the enhancing of a problem analysis and abilities to creatively generate and plan a solution to such problem. An individual issue related to creativity is a stress phenomenon and a motivational phenomenon (Yeh, Lai, Lin et al., 2015). An important collateral issue of software development is a risk that the work (Tyagi, Hanoch, Hall et al., 2017) is devoted to within the context of creativity.

It should be taken into account that many tasks are addressed creatively with a common-sense view and not with operating procedures introduced. In this context, the works based on common sense (Aloimonos, Fermueller, 2015; Ji, Yu, Zhao et al., 2018; Di Caro, Ruggeri, Cupi et al., 2015) should be emphasized. The work (Aloimonos and Fermueller, 2015) covers a new eyesight model where the eyesight is a part of a reasoning system. The work (Ji, Yu, Zhao et al., 2018) which defines and solves an issue of extracting common sense at a concept level is particularly interesting. The work (Di Caro, Ruggeri, Cupi et al., 2015) discusses an issue of determining semantic similarity based on the data obtained under the rule of reason, which might be useful for ranking any text-based information.

Methodology

Let us define an individual creative process as a creative human factor (CHF). A general order for an individual action of the CHF is as follows: the CHF subjectifies a local task and develops a solution creatively. There are two classes of information that stipulate the solution development, actual information and imaginary information or perceptual imagination (Anderson, 2010). The first one is characterized by all visible elements of a software process, namely, language and structural representation of an algorithm, comments, debug data and interface elements of design environment. Imagery information is the one that is hidden from a designer's view and hands but paves the way forward, by associations and intuition through multiple insights, for further development of a software project. In terms of creativity, imagery information has an absolute advantage over actual information. However, it is imagery information that has the most to lose as it is impossible to be further reproduced to the sufficient extent, but only the entourage of a creative solution process could be partly recollected.

We assume that imagery information can be partly recorded by being rendered graphically in order to be further used over and over. Solution images created will actualize comprehensiveness and coherence of miscellaneous data shown visually and graphically, and will make it possible to find new solutions and also to correct the existing ones. The recording and reproducing of such images at any time will allow for recovery of the entire development chain so as to review again and look closely at the steps already taken. It is important to note that the very possibility for a solution, an insight, to occur exists only if a key to it already lies in unconscious experience (Ponomarev, 1976). Therefore, visualization of perceptual information is a kind of a bridge between creativity and personal experience in resolving a specific problem. It is only personally-significant information that is recorded and reproduced but not data that can be summarized and further expressed in an abstract way. Such personally-significant information is compliant with preparing and recording insight data, and is expressed as elements of a new experience.

Therefore, the experience gained is a determining element of a solution, and gain in personal experience in the general solution process will consequently become a factor for intensification of a CHF dominant idea. Furthermore, attention should also be paid to conditions for solution maturation, which corresponds to the incubation stage in the description of the creation stages given by English professor G. Wallis in 1926 (Wallas, 1926). Such fact can be taken into account by creating an analog of the "blackboard" method offered by A. Newell in 1962 and described, for example, in the book by G. Booch (Booch, 1993).

The blackboard method includes three elements: a blackboard, a set of knowledge sources and a control unit that controls such sources. Moreover, an active element, the control unit, plays the following role: each knowledge source is connected to the control unit and sends its ideas to it, as well as the control unit can activate the knowledge sources. However, we can expand the role of the control unit and let it perform voluntary operations with the knowledge sources: to add, delete, displace, actualize, organize, save and reproduce. As such, the control unit becomes an active controlling element which acts outside unspecified predefined rules and is only limited to a blackboard size and multiple knowledge sources. The software implementation of such control unit apparently presents not only difficulties in creating an extensive base of descriptions for rules and conditions of the automated solution synthesis. Therefore, a reasonable solution to this issue is a substitution of the control unit for the CHF or an individual creative process. The blackboard should now be called otherwise, namely, a creatively contextual form (CCF), underlying the fundamental role that the creative process plays in solution development (Fig. 1).

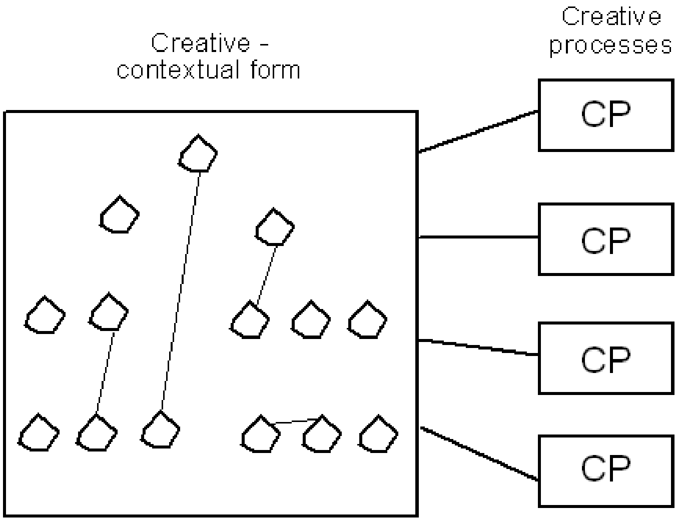

Fig. 1. Informational correspondence model of creative-contextual form and creative human factor

A contextual side of the CCF means that a person develops and adopts a solution based on his personal experience, comfort and information accessibility. Therefore, the contextual configuration of elements in the CCF is based on so-called availability heuristic which is defined as a possibility or assessment of a choice due to easiness of recollection or association (Kahneman, Slovic and Tversky, 1982), and is a determinative component in which a person searches for and adopts a solution, and gains knowledge.

A step sequence to develop yet another algorithmic solution begins with its incubation based on information elements given in the CCF. After a new solution is found and produced, it is recorded in an algorithm at a project or software level and then registered and positioned in the CCF. The elements and their details that are the most essential for a developer act as data that fill the CCF. These may be software code details, comments, debug details, presentation charts or program output drawings. Their value is not only to save in a database in general but also to correlate with each other in a certain way which is significant for a developer. Being multiplexed correctly in a redundant context provides a developer with a comfort to use the availability heuristic and, as a result, the incubation of another solution. A solution can be immediately saved in the CCF database. Solutions positioned in the CCF and recorded will provide not only for better incubation of another solution through the availability heuristic but also for reproduction of the earlier recorded solutions with additional information entourage.

It is especially important when repeated returns to previous development iterations are required, and it plays an expansion role of the person's short term memory, which contributes directly to a better overview and usage of the CCF data for another incubation and a future creative insight.

Software "Eureka"

For the purpose of a proper and flexible human-computer interaction in algorithm and software design and programming, a special software that supports and maintains the activity of a developer's creative process was developed. A task was set to find and provide for such a human-computer interaction regime where a developer's creative process in solution development and production would take an essential priority over a computer.

ISSN 2322- 6307

The software called Eureka contains, in its core, a visual block that operates the CCF.

A task of a creative search for an algorithm is defined by the following conditions achieved simultaneously:

- To preserve redundancy with which a designed solution is described;
- To preserve an individual development of an algorithmic idea that is based on the personal pertinence of an information element in particular conditions in which it is applied;
- To preserve a particularity of a solved issue without the need for generalization and premature classification;
- To support an original informal presentation of information elements;
- To provide for the visibility of local outcomes of project development, debugging and functioning, as well as a possibility to correlate them with other information elements of a project or algorithm.

Each of the above conditions is directly related to the expression of personal preferences and indicates no barriers in developing an individually comfortable solution, which means a free activity of the CHF. The main objective in solution development is the achievement of such a state of available data correlated and located in the CCF in which an insight corresponding to a birth of the sought idea will occur.

There are three main areas assumed to be in processing the CHF data located in the CCF (Fig. 2):

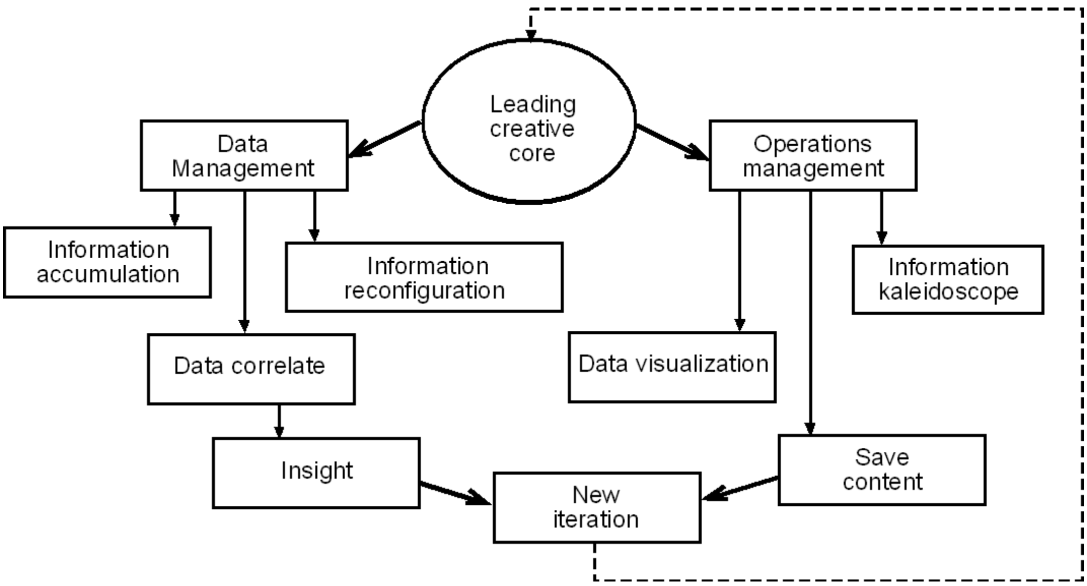

Fig. 2. Functional-information model of the Eureka software

- To form a vision of a solution including any operations with data, which is aimed at improving and better understanding a local problem and providing for the incubation of a final idea;
- To save a picture appropriate to an insight moment;
- To represent saved pictures of an insight.

The main distinction of the Eureka software is information and visual support provided to achieve another insight that is incubated by monitoring the CCF data correlated to each other. This model is too focused on the natural development iteration.

Let us consider heuristic development of an algorithm to build a 2D shape equidistant (2DSE) as an example of the Eureka software usage.

Such algorithm is distinguished by the need for storing and using all possible full information about the structure of the original shape and details of a consistent algorithm change, including data that are outcomes of the local project iterations, and various intermediate calculations. The essential knowledge integrity of the algorithm may only be insured by creating an information and functional environment which would have a definitive possibility to accumulate, present and save a whole range of information.

The essence of the algorithm to calculate the 2DSE is to gradually find significant angle bisector lines between the lines containing shape segments, and then to cross the obtained bisector network with lines equidistant to each shape segment. The points obtained in crossing will consistently form target equidistant shapes (Fig 3).

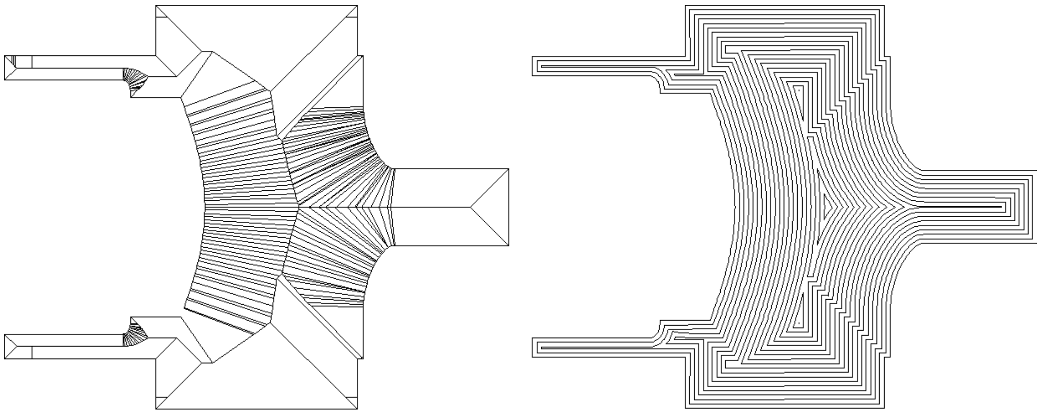

Fig. 3. The results of the algorithm for constructing 2DSE

Let us make qualitative assessments of multiple information elements required to develop another local solution. In order to store data of each edge of the original shape, a structure is used that consists of 9 fields, and each of such fields is directly used not only for calculations but also for simultaneous monitoring intermediate outcomes in both text and graphical formats.

Let us compare two cases of complexity, at the beginning and in the middle of the algorithm operation. A situation at the beginning of the algorithm operation is significantly characterized by initial data only, and each of the bisector arrays contains just one point, an initial point, therefore the monitoring of data is not very difficult. The second case differs from the first one with thickened bisector arrays and more complicated understanding and interpretation of current outcomes, without which it is impossible to develop and make a decision on further algorithm development. The number of information elements required for monitoring increases approximately from 1.5 to 7 times for the shapes with at least 500 points. Let us estimate the approximation of such elements to create conditions of incubation of another solution. For the first case, the number of necessary information elements is about 10 to 20.

For the second case, 15 to 140 of such elements are required accordingly. An increase by 1.5 – 7 times of the elements required for monitoring occurs due to the fact that the number of bisectors points the coordinates of which must be taken into account rises significantly from zero at the beginning of the algorithm operation. In the simplest case, we have at least 7 elements at the beginning of the algorithm operation and at least 11 elements in the middle of its operation as two more coordinates and a segment crossing index are added. But averagely, the elements which are significant for monitoring increase by 4-5 points at once, by a pair of coordinates, from each side of the shape segment, which means a sharp rise in the monitoring and solution development elements, approximately by 5 times.

Such estimates are obviously made subject to what a person think as to how significant an information element is in the context of others. In monitoring, searching and producing a solution in the CCF visualization environment based on the availability heuristic, a developer acts as an expert, and a choice made by him is supported by

ISSN 2322- 6307

a personal heuristic vividness. This is the value of a creative approach, as opposed to a solution process synthesis where there is a rigid system of selection rules and measures. On the contrary, when there is a creativity freedom, solutions are developed and assessed by a developer based on personal experience, knowledge, abilities and personal experience, knowledge, abilities and responsibility applied without any technologic or axiomatic obstacles.

Fig. 4 describes the need for information about any problematic situation which should be solved to be combined in a single visualization context. This picture illustrates one of the CCF obtained in the Eureka system in designing the 2DSE algorithm in a situation when an unexpected error occurs in step 25 of the iteration in the algorithm operation.

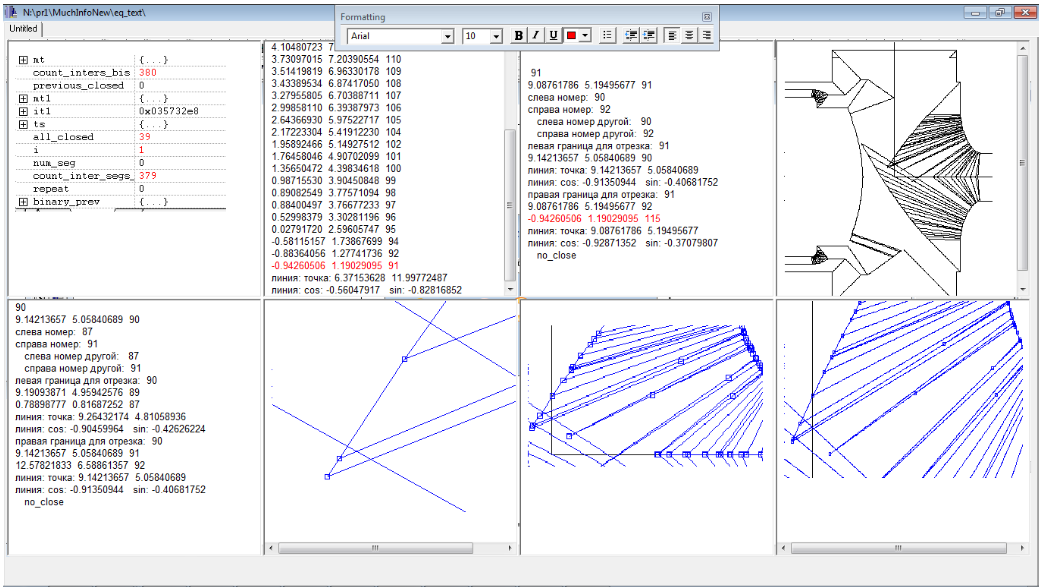

Fig. 4. The CCF of the next iteration of the 2DSE algorithm

CCF includes 8 fields of slides for selected information elements, each of which contains important information for the creative development of solutions to correct the error of the algorithm. These 4 slides show the situation from a graphical point of view at different scales. 3 more slides include a text-only debug and log information corresponding to a current data associated with the intersection of the bisectors. The last field of the slide shows individually important debugging data cut out at the time of the error as part of the MS Visual Studio screen. The CCF includes 8 slide fields for the chosen information elements each of which contains important information for the creative development of solutions to correct the error in the algorithm operation. The data of four slides display the situation graphically in various scale. 3 more slides contain text-based, debug and log file information relevant to current data and related to the bisecting lines crossed. The last slide field displays individually significant debug data clipped off at the time when the error occurred, as a sub-screen of MS Visual Studio.

The CCF arrayed data obtained using the Eureka software, i.e. recorded insights of solutions, have a truly infinite value. The described practice of displaying individual creative human traits can substantially enhance software quality and reliability.

Discussion

The order of analysis by users of the complex "Eureka" visually presented on the CCF information suggests the presence of completeness. Observation, recollection and associations excite the creative insight, based on its nature accompanied by an emotional outburst, which signals the achievement of such completeness. At this point, the path to the next solution becomes clear, and its context - holistic. Immediately, the KKF is committed and its data is placed in storage. If necessary, you can quickly return the developer to any previously recorded situation by expanding the previously recorded data. Thus, there is a significant reduction in information losses, which will provide an easier and more reliable correction or expansion of the project. Obtained by using the complex "Evrika" dataset CCF, i.e. – written insight, solutions, truly priceless. The described practice of manifestation of individual creative qualities of a person is able to improve the quality and reliability of SW. In addition, the computer-stored information of the CCF data equips the developer with the confidence that any action taken by him will not lead to a fatal deadlock, and its result can be easily supplemented, expanded or changed.

The procedure by which the Eureka software users analyze information visually presented in the CCF requires a whole picture. Observation, recollection and associations initiate a creative insight accompanied, based on its nature, by emotional outburst which indicates such wholeness. At this moment, a path to the next solution becomes clear, and the context of such path becomes holistic. A comprehensive set of information elements in the CCF which are collected by the time of an insight, can be immediately recorded, and then such data are put in storage. Where appropriate, a developer can be quickly returned to any project development situation recorded earlier, by reopening the recorded data in the CCF, as though memories of the past are coming back. Such memories return by the extraction of data from the storage but not due to a person’s memory and consciousness. The outcome of such extraction can always be kept close at hand other than what the short term memory can keep. The storage capacity can be quite large; therefore there are no visible limits for an individual informational constraint.

A personal experience and comfort in the awareness of a development path regulate human activities. A developer decides which information, and whether such information should be placed in the CCF or not. How the data placed to be correlated with other CCF information is also determined individually. How CCF information to be detailed or extended is also decided by a person. Only a developer cannot miss a moment when a clear view of a solution occurs in a developer’s mind. The next moment, he decides based on the availability heuristic whether to record, supplement or transform this picture of a solution. Such practice makes a developer sure that any action taken by him will not bring to a deadlock.

Conclusion

Note that, at the time when project information is significantly increased, there is a notable reduction in information loss. This contributes to easiness and reliability of the software process control.

Let us outline the main objectives achieved by a creativity-oriented software development methodology:

– Direct monitoring of the informational expansion or modification of a project enables significant information to be correlated in the general context and accumulated;
– Detailed restoring of informational content of any development stage previously recorded reproduces memories about the picture of such development stage;
– Simultaneous monitoring of several algorithm versions ensures the adoption of such versions as those more acceptable from a creative point of view;
– Review of and search for solutions in a specific individual data context generate a heuristically responsible solution;
– A possibility to accumulate any information volume will avoid the risk of a project development data loss.

The procedure offered in this Article for accumulation, positioning and visualization of information elements provides an opportunity to place and record any graphically represented information to the full specific and individual extent. In such event, each CCF will reflect experience and a mindset of an individually expressed solution.

The scientific novelty of this work is developing a software development methodology and practice by providing comfort of the human creativity due to the fact that an impact on the short term memory is reduced. A personal experience to combine perceptual, unbiased, local or any other information in order to give rise to a creative insight of a solution, as well as its accumulation and further opening contribute to a systematic and consistent character of a software process.

Unfortunately, in the modern world of software development, there is not even a hint of matching up solutions and creative activity of a person who produces them. Consequently, a person's general responsibility for his action and the final product quality reduces inevitably.

We also note that interdisciplinary research operations aimed at a search for psychological basis to further improve the software process quality are needed essentially. First of all, we should study roles, a place and an order of abstraction in the software analysis and development. It is important to note the order in which solutions are creatively found and adopted in terms of tactical achievement of a desired goal.

References

Ackoff R.L. and Emery F.E. (1972). On Purposeful Systems. Chicago, Aldine-Atherton, 288.

Aghajani E., Bavota G., Linares-Vasquez M., Lanza M. (2019). Automated Documentation of Android Apps. IEEE Transactions on Software Engineering, accepted for publication. IEEE CS Press.

Aloimonos Y. and Fermueller C. (2015). The Cognitive Dialogue: A new model for vision implementing common sense reasoning. IMAGE AND VISION COMPUTING, 34, 42-44.

Anderson J.R. (2010). Cognitive Psychology and Its Implications. New York: Worth Publisher.

Bacchelli A., Cleve A., Mocci A., Lanza M. (2017). Mining Structured Data in Natural Language Artifacts with Island Parsing. Journal of Science of Computer Programming (SCP), Elsevier.

Baltes S., Moseler O., Beck F., Diehl S. (2015). Navigate, Understand, Communicate: How Developers Locate Performance Bugs. ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM).

Booch G. (1993). Object-Oriented Analysis and Design with Applications (2nd Edition).

Byrge C., Tang, C. (2015). Embodied creativity training: Effects on creative self-efficacy and creative production. THINKING SKILLS AND CREATIVITY. 16, 51-61.

Caracciolo A., Lungu M., Truffer O., Levitin K., Nierstrasz O. (2016). Evaluating an Architecture Conformance Monitoring Solution. IWESEP@SANER, 41-44.

Chang Yu-S., Chien Yu-H., Yu K.-C. et al. (2016). Effect of TRIZ on the creativity of engineering students. THINKING SKILLS AND CREATIVITY, 19, 112-122.

Di Caro L., Ruggeri A., Cupi L. et al. (2015). Common-Sense Knowledge for Natural Language Understanding: Experiments in Unsupervised and Supervised Settings. 14th International Conference of the Italian-Association-for-Artificial-Intelligence: ADVANCES IN ARTIFICIAL INTELLIGENCE, Book Series: Lecture Notes in Artificial Intelligence, 9336, 233-245.

Himaieva Y., Nykyforova V. (2015). Creativity and personality motivation and their interconnection as basis of graphic designers and marketing specialists' creative activity. SCIENCE AND EDUCATION, 1, 13-24.

Ji Y., Yu Y., Zhao H. et al. (2018). Verb Based Conceptual Common Sense Extraction. INTERNATIONAL CONFERENCE ON INFORMATION SCIENCE AND SYSTEM, 256-260.

Kahneman D., Slovic P. and Tversky A. (1982). Judgment Under Uncertainty: Heuristics and Biases. New York: Cambridge University Press.

Lanza M. (2003). Program Visualization Support for Highly Iterative Development Environments. Proceedings of VISSOFT, 62-67.

Lanza M., Ducasse S., Gall H., Pinzger M. (2005). CodeCrawler. An information visualization tool for program comprehension. ICSE 05: 27th International Conference on Software Engineering, 672-673.

Lopes A.G. (2017). Creative Activity Mediated by Technology. Artifacts, Technology and Ideas that Could Shape Our Lives. AHFE International Conference on Human Factors and System Interactions, ADVANCES IN HUMAN FACTORS AND SYSTEMS INTERACTION. 592, 371-382.

Miller G. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, 63 (2), 86.

Parnas D. and Clements P. (1986). A Rational Design Process: How and Why to Fake It. IEEE Transactions on Software Engineering, Vol. SE-12(2).

Petre M. (1995). Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming, Comm. ACM. 38 (6), 33-44.

Ponomarev Y.A. (1976). Psychology of creativity. Moscow: Science, 303.

Stroustrup B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley Publishing Company.

Thomas J.J. and Cook K.A. (Ed.) (2005). Illuminating the Path: The R&D Agenda for Visual Analytics Archived. 2008-09-29 at the Wayback Machine. National Visualization and Analytics Center, 30.

Tufte E.R. (2001). The Visual Display of Quantitative Information. Graphics Press, 2nd edition.

Tufte E.R. (1997). Visual Explanations. Graphics Press.

Tyagi V., Hanoch Y., Hall S.D. et al. (2017). The Risky Side of Creativity: Domain Specific Risk Taking in Creative Individuals. FRONTIERS IN PSYCHOLOGY, 8. Article Number: 145.

Wallas G. (1926). The Art of Thought. New York: Harcourt, Brace and Company.

Ware C. (2000). Information Visualization. Morgan Kaufmann.

Würfel D., Lutz R., Diehl S. (2016). Grounded requirements engineering: An approach to use case driven requirements engineering. Journal of Systems and Software.

Yeh Yu-C., Lai G.-J., Lin C. Fu et al. (2015). How stress influences creativity in game-based situations: Analysis of stress hormones, negative emotions, and working memory. COMPUTERS & EDUCATION, 81, 143-153.

Yuan Y. (2016). Creative activity as a method of organizing future music teachers' self-preparation for teaching practice. SCIENCE AND EDUCATION, 4, 21-24.