(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 March 2013 (21.03.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/037688 A2

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/EP2012/067329

(22) International Filing Date:
5 September 2012 (05.09.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1209/KOL/201 1 14 September 201 1 (14.09.201 1) IN

(71) Applicant *(for all designated States except US):* **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 Munchen (DE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only):* **GUPTA, Shrinath** [IN/IN]; No. 768 17th F Main road, 3rd cross, 6th Block, Koramangala, Bangalore 560 095 (IN). **SAMARTHYAM GOVINDACHETTY, Ganesh** [IN/IN]; #203, Snehanjali Residency, CV Raman Nagar, Bangalore 560 093 (IN).

(74) Common Representative: **SIEMENS AKTIENGESELLSCHAFT;** Postfach 22 16 34, 80506 Munchen (DE).

(81) Designated States *(unless otherwise indicated, for every kind cf national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind cf regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with declaration under Article 17(2)(a); without abstract; title not checked by the International Searching Authority*

(54) **Title:** A SYSTEM FOR PRIORITIZING VIOLATIONS IN A PIECE OF CODE

(57) **Abstract:**

Description

A system for prioritizing violations in a piece of code

The invention relates to prioritizing violations in a piece of code, more particularly the invention relates to prioritizing violations in a piece of code by using specification requirements used for developing a software design on whose basis the piece of code is developed.

For developing a piece of code, a software designer has to develop a software design to be followed by a software developer, so that the software developer can use the software design as reference to develop the piece of code. The software designer uses specification requirements provided by a project manager or a customer for developing the software design. The specification requirements generally represent needs and a wish list of the customer or the project manager for developing the piece of code to be used for a particular project or part of a product.

During development cycle or life cycle of the piece of code, the piece of code is being generated by multiple software developers and hence prone to violations in the piece of code with respect to the software design and specification requirements .

One possible way to prevent the violations is to identify the violations using static analysis tools by analyzing the piece of code without executing the piece of code. However, the static analysis tools generate an overwhelming number of violations which may not be relevant for the time being to be fixed. These static analysis tools don't indicate time critical violations which risks performance of the system. Being non-indicated, the time critical violations may not be attended by the software developer.

It is an object of the invention to prioritize violations occurring in a piece of code.

The object of the invention is achieved by a system of claim 1 and a method of claim 8.

The idea of the invention is to prioritize violations in a piece of code which can be defined as non-compliant to a software design. The prioritization is functioned by receiving a set of violations identified in the piece of code and a set of priority factors based on specification requirements by a prioritizer and further prioritizing the violations to provide a priority number to each violation by same prioritizer on a basis of the priority factors.

According to an embodiment of the invention, a system for prioritizing the violations in a piece of code includes the prioritizer and a code analyzer adapted to analyze said piece of code to generate the set of violations. This helps to provide an integrated system for identifying the violations in real time and prioritizing the violations using the prioritizer.

According to an exemplary embodiment, the code analyzer analyzes said piece of code either statically or dynamically. This provides an option for the system to analyze the piece of code with an option for either to analyze the code statically if still the piece of code is under development and analysis is required to be done for a developed portion of code or dynamically when the piece of code is developed and a software tester is required to prioritize violations by executing the piece of code.

According to a further embodiment, the system includes a factoring device for collecting the specification requirements and for processing the specification requirements to provide the priority factor to the prioritizer. This provides for an integrated environment to prioritize violations while generating the priority factors by processing the specification requirements.

According to one embodiment, the prioritizer is adapted to prioritize the violations by providing a rank to each of the violations on a basis of a function of the priority factors relating to the violation. Ranking of the violations provides with another parameter for the software tester to take in consideration while deciding severity of the violations to be removed from the piece of code.

According to another embodiment, the system includes a user interface adapted to receive a weighted value to be assigned to each of the priority factors, wherein the prioritizer is adapted to provide ranks to the violation on the basis of the function of the weighted priority factors. This provides for intervention of the software developer in the prioritizing process to prioritize the violations according to his requirements or the project's requirements.

According to yet another embodiment, the prioritizer is adapted to prioritize the violations by providing a classification of the violations on a basis of non-compliance to an aspect of the software design. Such a system is equipped for prioritizing the violations with another option of prioritizing which is related to different aspects of software design, so that the software tester can choose the violations based on classification, if he is looking for a category of violations to be removed.

The present invention is further described hereinafter with reference to an embodiment shown in the accompanying drawing. The illustrated embodiment is intended to illustrate, but not to limit the invention. Herein,

FIG 1 is a schematic diagram of a system for prioritizing violations in a piece of code.

Various embodiments are described with reference to the drawing, wherein like reference numerals are used to refer to single elements throughout. In the following description, for purpose of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more embodiments. It may be evident that such embodiments may be practiced without these specific details.

For developing the piece of code, a software developer requires a reference from a specification requirement to be covered during the development and maintenance phase. To generate an understandable reference, a software designer develops a model of the specification requirements with details which are understandable by the code. Such a model is called a software design. However, during development and maintenance of the piece of code, the software developer tends to make violations of the software design which makes the piece of code non-compliant to the software design.

FIG 1 illustrates a schematic representation of a system 1 for prioritizing a violation 2 of a piece of code 3 including a prioritizer 4, a factoring device 8 and a code analyzer 7 cooperating together to prioritize violations 2 of the piece of code 3 to provide a priority number 6, a rank 9 and a category 13 of violations for each of the violations 2.

The priority number 6 is generally an integer value, ranking 9 of the violation 2 is based on a severity scale like I, II, etc. and the category 13 of the violation 2 denotes which attribute of the software design the violation 2 is representing like technical attributes, industry coding standard attributes, etc.

In an alternate embodiment, the prioritizer 4, the factoring device 8 and the code analyzer 7 cooperate together to prioritize violations 2 of a piece of code 3 to provide the priority number 6 and not the rank 9 and the category 13 of violations for each of the violations 2. Alternatively, the prioritizer 4, the factoring device 8 and the code analyzer 7 cooperate together to prioritize violations 2 of a piece of code 3 to provide the priority number 6 and only one of the rank 9 or the category 13 of violations for each of the violations 2.

In an alternate embodiment, the priority number 6 can be used by the prioritizer 4 to generate the rank 9 for the violation 2. For example (any ranking method can be used here), first 2%, 2%-5%, 5% to 25%, and 25% to 100% as severity values (ranks) I, II, III, IV and V (respectively).

The code analyzer 7 receives the piece of code 3 and analyzes said piece of code 3 to generate the set of violations 2. The set of violations 2 is obtained by the code analyzer 7 by statically analyzing the piece of code 3. Alternatively, the code analyzer 7 can obtain the violations 2 for the piece of code 3 by executing the piece of code 3 dynamically. Yet alternatively, the code analyzer 7 can be any general purpose processing unit like the central processing unit of a personal computer which can be adapted to receive the piece of code 3 and analyze the piece of code 3 to generate violations 3 to a software design requirement in the piece of code 3.

In an alternate embodiment, the system 1 does not comprise the code analyzer 7. Rather, the system 1 directly receives the preprocessed violations 2 from a user via a system interface allowing user to provide the violations 2.and can use the violations 2 for prioritizing the violations 2.

Specification requirements 12 based on which the piece of code 3 is developed can be both functional as well as non-functional. An example of a non-functional specification requirement 12 can be quality attributes of the piece of code 3. An example of functional requirements 12 can be technological attributes like memory leaks while execution of the piece of code in a real world application.

The specification requirements 12 can be either discretely provided by a user through an input device or can be fetched by the code analyzer 7 by querying a database stored in a memory device to the code analyzer 7. The memory device can be external like an USB device or internal like a hard drive. Alternatively, the specification requirements 12 can be consolidated in a configuration file for a particular application or project or product and can be either downloaded by the code analyzer 7 from a server or can be made available through the memory device.

The prioritizer 4 receives a set of violations 2 identified in the piece of code 3 and a set of priority factors 5 based on specification requirements 12 and prioritizing the violations 2 to provide a priority number 6 to each violations 2 by using the priority factors 5. The prioritizer 4 prioritizes the violations 2 by mapping the priority factors 5 onto the violations 2 in the set. In an alternate embodiment, the prioritizer 4 can prioritize violations 2 by using any other mathematical algorithm.

In an alternate embodiment, the prioritizer 4 can be a general purpose control processing unit of a personal computer or any other processing unit like microprocessor, etc. adapted to receive the violations 2 and priority factors 5 to provide the priority number 6 for the violations 2 on a basis of the priority factors 5.

The factoring device 8 collects the specification requirements 12. It is adapted to process the specification requirements 12 to provide the priority factors 5 to the prioritizer 4. Alternatively, the factoring device 8 can be a general purpose control processing unit of a personal computer or any other processing unit like microprocessor, etc. adapted to receive the specification requirements 12 to priority factors 5 to the prioritizer 4. Yet alternatively, the system 1 does not comprise the factorizing device 8, rather the prioritizer 4 is adapted to receive the priority factors 5 pre-calculated from a memory device.

The priority factors 5 are derived from the specification requirements 12. For example, with respect to non-functional requirements 12 like quality attribute, the priority of quality attributes and sub-attributes will be processed by the factorizing device 8 to determine the priority of the quality attributes to be taken in consideration while being prioritized by the prioritizer 4. The prioritizer 4 maps the quality attributes as the priority factors 5 onto the violations 2 to bring out the priority number 6 for the violations 2.

The priority factors 5 can also be derived from the functional specification requirements 12, like the technological attributes with respect to code related issues, runtime issues, etc. Each of these technological attributes can be given priorities with respect to a project or product and hence the priority factors 5 are generated by the factorizing tool 8 for these technological attributes.

There may be other non-functional requirements 12, functional requirements 12 or combinations thereof, like the domain of an application on which the piece of code 3 will be executed, e.g. healthcare, industry, automation, etc., the type of the application (example embedded system where resource utilization is critical), attributes of components to which the piece of code maps to according to the software design (example reusability, stability, criticality of the components), types of part of code (example hazardous, non-hazardous in healthcare domain), temporal attributes (example "maintainability" issues are not important during testing phase of the piece of the code, but quite important while product launch/project implementation), etc.

Each of the priority factors 5 can further be divided into sub-factors and each of the priority factors 5 can be consolidated by a function (g) of the sub-factors for a particular priority factor (fi) 5, where "i" denotes an integer and "fi" denotes the "ith" factor. The subfactor can be represented as fsl, fs2, fs3, ……, fsn. The priority factor (fi) 5 can be calculated by equation [1]

Fi = g (fsl, fs2, fs3, ……., fsn) [1]

The prioritizer 4 prioritizes the violations 2 by providing a rank 9 to each of the violations 2 on a basis of a function of the priority factors 5 relating to the violation 2. The ranking of violation is determined on severity scale like Severity I, II, III, etc. The priority factors 5 can be represented as fl, f2, f3,...., fn. The rank (Rn) 9 for each of the priority factor can be calculated by a function (f) of the priority factors 5 as represented in the equation [2].

Rn = f (fl, f2, f3,...., fn) [2]

The prioritizer 4 receives a weighted value 11 to be assigned to each of the priority factors 5 from a user via a user interface 10, such that the prioritizer 4 is adapted to provide ranks 9 to the violation 2 on the basis of the function of the weighted priority factors 5. In an alternate embodiment, the prioritizer 4 takes an average of the weighted factors 5 to determine the rank (Rn) 9. Yet alternatively, the prioritizer scales the priority factors 5 on a scale like [0, 10]. For example if the reliability is of very high importance to a particular project or product, then the priority factor 5 for reliability can get a value of 10. If portability is not at all important for a project or product, then the priority factor 5 for portability can get a value of 0. The prioritizer 4 multiplies the value of all the priority factors 5 for a particular violation 2 and can compute the value of Rn for that violation 2.

Patent Claims

1. A system (1) for prioritizing violations (2) in a piece of code (3) comprising:
- a prioritizer (4) adapted to receive a set of violations (2) identified in the piece of code (3) and a set of priority factors (5) based on specification requirements (12) and adapted to prioritize the violations (2) to provide a priority number (6) to each violation (2) by using the priority factors (5),
wherein the violations (2) are defined as a non-compliance to a software design used for developing the piece of code (3) and the software design is modeled on the basis of the specification requirements (12).

2. The system (1) according to claim 1 comprising a code analyzer (7) adapted to receive the piece of code (3) and analyze said piece of code (3) to generate the set of violations (2).

3. The system (1) according to claim 2, wherein the code analyzer (7) analyzes said piece of code (2) either statically or dynamically.

4. The system (1) according to any of the claims 1 to 3 comprising a factoring device (8) adapted to collect the specification requirements (12) and adapted to process the specification requirements (12) to provide the priority factors (5) to the prioritizer (4).

5. The system (1) according to any of the claims from 1 to 4, wherein the prioritizer (4) is adapted to prioritize the violations (2) by providing a rank (9) to each of the violations (2) on a basis of a function of the priority factors (5) relating to the violation (2).

6. The system (1) according to the claim 5 comprising a user interface (10) adapted to provide a weighted value (11) to be assigned to each of the priority factors (5) ,
wherein the prioritizer (4) is adapted to provide ranks (9) to the violation (2) on the basis of the function of the weighted priority factors.

7. The system (1) according to any of the claims 1 to 6, wherein the prioritizer (4) is adapted to prioritize the violations (2) by classifying the violation (2) to a category (13) of the violations (2) on a basis of non-compliance to an aspect of the software design.

8. A method for prioritizing violations (2) in a piece of code (3) comprising:
- receiving a set of violations (2) identified in the piece of code (3) and a set of priority factors (5) based on specification requirements (12) by a prioritizer (4),
- prioritizing the violations (2) to provide a priority number (6) to each violation (2) by using the priority factors (5) ,
wherein the violations (2) are defined as a non-compliance to a software design used for developing the piece of code (3) and the software design is modeled on the basis of specification requirements (12).

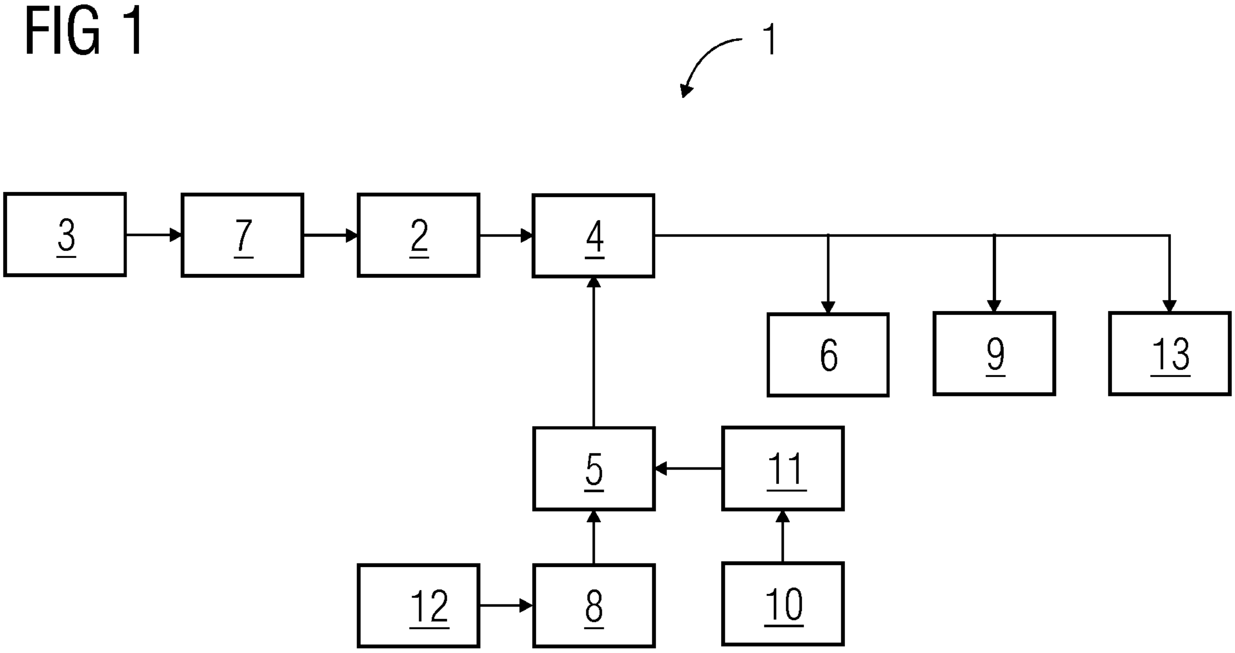
9. The method according to the claim 8 comprising:
- analyze said piece of code to generate the set of violations (2) by a code analyzer (7) .

10. The method according to any of the claims 8 or 9 comprising :
- collecting the specification requirements (12) by a factoring device (8), and
- processing the specification requirements (12) to provide the priority factors (5) to the prioritizer (4) .

11. The method according to any of the claims 8 to 10, wherein prioritizing the violations (2) by providing a rank (9) to each of the violations (2) on a basis of a function of the priority factors (5) relating to the violation (2) .

12. The method according to any of the claims 8 to 11, wherein the violations (2) are prioritized by classifying the violation to a category (13) of the violations (2) on a basis of non-compliance to an aspect of the software design.

FIG 1



PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT

(PCT Article 17(2) (a), Rules 13fer.1 (c) and Rule 39)

Applicant's or agent's file reference 201 0P24329WO	**IMPORTANT DECLARATION**	Date of mailing *(day/month/year)* 25 October 201 2 (25-1 0-201 2)
International application No. PCT/EP20 12/067329	International filing da\.e*(day/month/year)* 5 September 201 2 (05-09-201 2)	(Earliest) Priority da\.e*(day/month/year)* 14 September 201 1 (14-09-201 1)
International Patent Classification (IPC) or both national classification and IPC G06F9/44		
Applicant SIEMENS AKTIENGESELLSCHAFT		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below

1. ☒ The subject matter of the international application relates to:
   - a. □ scientific theories
   - b. □ mathematical theories
   - c. □ plant varieties
   - d. □ animal varieties
   - e. □ essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes
   - f. □ schemes, rules or methods of doing business
   - g. ☒ schemes, rules or methods of performing purely mental acts
   - h. □ schemes, rules or methods of playing games
   - i. □ methods for treatment of the human body by surgery or therapy
   - j. □ methods for treatment of the animal body by surgery or therapy
   - k. □ diagnostic methods practised on the human or animal body
   - l. □ mere presentations of information
   - m. □ computer programs for which this International Searching Authority is not equipped to search prior art
2. □ The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:
   □ the description □ the claims □ the drawings
3. □ A meaningful search could not be carried out without the sequence listing; the applicant did not, within the prescribed time limit:
   - □ furnish a sequence listing on paper complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
   - □ furnish a sequence listing in electronic form complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
   - □ pay the required late furnishing fee for the furnishing of a sequence listing in response to an invitation under Rule 13ter.1 (a) or (b).
4. Further comments:

Name and mailing address of the International Searching Authority European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31 -70) 340-2040 Fax: (+31 -70) 340-301 6	Authorized officer VAN RIJSSEL, Debbie Tel: +31 (0)70 340-471 9

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 203

No International Search Report is issued, because the current international application exclusively includes subject-matter which the International Searching Authority is not required to deal with (PCT International Search and Preliminary Examination Guidelines 9.01).
The claims 1-7 (system) and 8-12 (method) do not specify any concrete technical means for executing the steps mentioned in the claims, e.g.
receive a set of violations and a set of priority factors ;
prioritize the violations by using the priority factors ;
receive and analyze a piece of code;
collect and process specification requirements;
prioritize the violations by providing a rank to each of the violations;
prioritize the violations by classifying the violations to a category.
The subject-matter of the claims is therefore considered to be of a purely abstract or intellectual character, which does not go beyond performing purely mental acts (PCT International Search and Preliminary Examination Guidelines 9.07).
In the current international application, the claimed invention, when viewed as a whole, is of abstract character, and thereby does not provide a practical application having a useful , concrete and tangible result (PCT International Search and Preliminary Examination Guidelines A9.07 [l]).
The claimed system and method are items essentially of an abstract or intellectual character. Moreover, since the claimed subject matter does not specify any apparatus or technical process for carrying out at least part of the scheme, the scheme does not have to be searched and examined as a whole (PCT International Search and Preliminary Examination Guidelines A9.07 [2]).

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guideline C-VI, 8.2), should the problems which led to the Article 17(2) declaration be overcome.