

Internal distribution code:

- (A) [-] Publication in OJ
- (B) [-] To Chairmen and Members
- (C) [-] To Chairmen
- (D) [X] No distribution

**Datasheet for the decision
of 14 May 2024**

Case Number: T 0156/21 - 3.5.01

Application Number: 18214173.9

Publication Number: 3671587

IPC: G06Q10/06

Language of the proceedings: EN

Title of invention:

A METHOD FOR CHANGING A SOFTWARE CODE EXECUTED BY A PRODUCTION SYSTEM

Applicant:

SAP SE

Headword:

Changing a software code executed by a production system/SAP

Relevant legal provisions:

RPBA 2020 Art. 11
EPC Art. 56, 111(1)

Keyword:

Remittal - (yes) - special reasons for remittal

Decisions cited:

T 0641/00, T 0787/18, T 0072/20, T 0098/21



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0156/21 - 3.5.01

D E C I S I O N
of Technical Board of Appeal 3.5.01
of 14 May 2024

Appellant: SAP SE
(Applicant) Dietmar-Hopp-Allee 16
69190 Walldorf (DE)

Representative: DeltaPatents B.V.
Fellenoord 370
5611 ZL Eindhoven (NL)

Decision under appeal: **Decision of the Examining Division of the
European Patent Office posted on 9 November 2020
refusing European patent application No.
18214173.9 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman W. Chandler
Members: N. Glaser
L. Basterreix

Summary of Facts and Submissions

- I. This appeal is against the decision of the examining division to refuse the European patent application No. 18214173.9 pursuant to Article 97(2) EPC on the ground of lack of inventive step (Article 56 EPC).
- II. The examining division considered that claim 1 was an obvious implementation of a software change management workflow on a conventional distributed information system, such as the one disclosed in D1 (US2016/378434) or D2 (US9 965 377), see points 13 and 14 of the decision.
- III. The appellant requested that the decision under appeal be set aside and a patent be granted on the basis of the main request, on which the decision was based or, alternatively, on the basis of a first or a second auxiliary request filed with the statement of grounds of appeal.

The appellant further requested that the case be remitted to the examining division for further prosecution.
- IV. In a communication accompanying the summons to oral proceedings, the Board set out its preliminary opinion that the invention did not appear to involve an inventive step (Article 56 EPC). The Board also tended to consider the auxiliary requests to be inadmissible under Article 12(2), (4) and (6) RPBA.
- V. In a reply, the appellant presented arguments in favour of inventive step.

VI. At the oral proceedings, the Board discussed inventive step of claim 1 of the main request. Following the discussion, the appellant's main request was that the case be remitted to the examining division for further prosecution.

VII. After due consideration of the appellant's arguments the Chairman announced the decision.

VIII. Independent claim 1 of the main request reads as follows:

"1. A method for changing a software code executed by a PS ("production system"), the method being implemented by a computer system comprising a CI ("continuous integration server"), a CMCS ("change management computer system"), a SCMS ("source code management computer system"), a plurality of DCS ("developer computer system"), one of the DCS being assigned to one developer, a TMS ("transport management system"), and a TS ("test system") being coupled by a network, the CMCS comprising a workflow engine for enforcing and documenting a workflow for processing change documents, wherein a change document comprises a specification for specifying a desired functionality of the software code, the change document having an assigned change document identifier and a change document status, the status being one of a first status ("development approved"), a second status ("implemented"), a third status ("successfully tested") and a fourth status ("productive"), and the change document being assigned to a developer, the TS and the PS having a respective deployment interface for deploying of an artifact, the deployment interface being controlled by the CMCS,

the method comprising:

- outputting of the assignment of change document identifier to the developer by CMCS,
- pulling a copy of the master code from the SCMS onto the DCS of the developer,
- entering changes to the copy of master code into DCS to provide staging code that has the desired functionality,
- sending request for staging code push from the DCS to SCMS for storing staging code on SCMS, the request comprising the change document identifier,
- executing the request by the SCMS and embedding the change document identifier into the staging code,
- sending a request for voter build of the staging code from the SCMS to CI,
- executing the request for voter build by the CI by obtaining the staging code from the SCMS, extracting the document identifier from the staging code, wherein the CI uses the CMCS to check whether (i) the extracted change document identifier matches the change document identifier of the CMCS, (ii) the assigned status being the first status, and (iii) the assignment is to the developer, and only if all conditions (i) to (iii) are fulfilled generating an artifact for the staging code, testing the artifact,
- if the test is successful,
 - o merging the staging code with the master code and storing the merged master code on the SCMS to enable access to the changed master code resulting from the merging by other developers,
 - o sending a request for CI build of the master code from the SCMS to CI,
 - o executing the request for CI build by the CI by obtaining the master code from the SCMS, extracting the document identifier from the master code, wherein the CI uses the CMCS to check whether (i) the extracted change document identifier matches

the change document identifier of the CMCS, (ii) the assigned status being the first status, and (iii) the assignment is to the developer, and only if all conditions (i) to (iii) are fulfilled generating an artifact for the master code, testing the artifact,

o sending a request from CI to CMCS for the creation of a container, the request comprising the change document identifier, wherein the CMCS uses the TMS for the creation of the container, the TMS returning a container identifier to CMCS

o sending the artifact from CI to CMCS, wherein the CMCS inserts the artifact into the container provided by TMS

o sending a request from the CI to the CMCS for transitioning the status from the first to the second status, the request comprising the change document identifier, wherein the CMCS triggers the TMS to deploy the container on the TS whereby the CMCS controls the deployment interface of the TS to allow deployment of the artifact inside the container,

- testing the artifact on the TS by a tester and if successful,

o sending a request from the TS to the CMCS for transitioning the status from the second to the third status, the request comprising the change document identifier,

- confirming the change document by a change manager, o triggering the processing in the CMCS for transitioning the status from the third to the fourth status, wherein the CMCS triggers the TMS to deploy the container on the PS, whereby the CMCS controls the deployment interface of the PS to allow deployment of the artifact inside the container."

Reasons for the Decision

1. Background of the invention
 - 1.1 The invention relates to a combination of continuous integration (CI) of software development and change management computer systems (CMCS).
 - 1.2 CI is popular because it fosters creativity and quick development cycles, but does not cover requirements in the area of compliance, such as auditable documentation for regulatory purposes. CMCS are known to enforce a standardized workflow for changing, testing and deploying of changed software code including the assignment of tasks to developers, testers and change managers. It also requires the documentation of all relevant activities in an auditable way that ensures fulfilment of regulatory requirements, see page 2 of the application.
 - 1.3 The invention proposes a method for changing software code which combines the concept of CI with the safety and quality control of CMCS see page 4, 2nd paragraph, of the application as originally filed. The CMCS controls the integration of new code into the source code management computer system (SCMS) as well as the deployment of artifacts (machine code) emerging from the CI/SCMS domain in accordance with a predefined workflow which is implemented by the CMCS.
 - 1.4 Figure 1 of the application illustrates a CMCS 190 coupled to a computer network 150 for communication with various developer computer systems (DCS) 140, a source code management computer system (SCMS) 180, a continuous integration server (CI) 160, a transport

management computer system (TMS) 170, a test computer system (TS) 120 and a production computer system (PS) 130. The CMCS stores change documents 199 which specify a desired functionality of the software code to be executed by the PS, a document ID, a Status, such as *1-development approved, 2-implemented, 3-successfully tested and 4-productive* and a Person ID.

A developer gets assigned a change document to implement its functionality by setting the Person ID to that developer's ID. S/he logs on to a DCS, authenticates him/herself, requests and downloads a copy of the master code from the SCMS. S/he enters changes into the master code to implement the functionality defined in the change document. The changed copy of the master code is called "staging code". It embeds the change document ID and is stored in the SCMS and the CI is informed (request for voter build).

The CI uses the CMCS to perform two series of checks (217 and 231) which must be met before the method for changing code is allowed to continue, that is, before the CI is allowed to generate an artifact for the staging code (step 219) and later on for generating an artifact for the new version of the master code (step 232a). The checks involve three conditions; firstly whether the change document on which the staging code was developed has the correct document identification, secondly whether the status of the document equals "development approved" to prevent the implementation of functionalities without explicit approval, and finally whether the staging code was from the correct developer. Each generated artifact is tested which is driven by the CI or CMCS.

2. Article 56 EPC

2.1 The examining division cited D1 and D2 in its inventive step argumentation as examples of a conventional distributed information system, see point 14 of the impugned decision, whereas it considered the method steps of claim 1 to define a *change management workflow for software development*. It then argued that a person skilled in the art would implement the workflow on such conventional systems in an obvious manner and concluded a lack of inventive step.

2.2 Although the Board could agree with many of the individual points in the reasoning of the examining division, it is not convinced of its overall structure.

The analysis of the technical features of claim 1 at point 13 of the impugned decision cited D1 for most features and referred at the same time to D2, and only to D2 for the transport management system (TMS), which creates a container for use by the CMCS which inserts the generated artifact into it. In other words the examining division combined D1 and D2. Such a combination was explained by the statement on page 3 of the impugned decision that **"the technical features are all well-known in the art as exemplified by documents D1 and D2"**. Later on in point 14, the technical problem was then defined as how to implement a workflow "in a conventional distributed information system such as the one disclosed in D1 or D2" (Board's emphasis).

2.3 This is not a correct application of the problem-solution approach or the COMVIK approach (T 641/00 - *Two identities/COMVIK*, OJ EPO 2003, 352), which requires starting from a single piece of prior art. When starting from exemplifying "conventional" prior

art, each cited, single document must disclose all the features.

- 2.4 In the Board's view such an approach also leads to a rather generic or abstract analysis of the invention. In this case, it appears that the invention should be analysed more specifically as either a further development of the revision control system of D1, by the same applicant, or a modification to the known continuous integration technique. Starting from either would then entail a precise analysis of the differences and the effect of the differences in the context leading to a statement of the problem solved.
- 2.5 Starting from D1 would require an explanation of the role of the differentiating features, to identify their technical effect (if any) and also to explain how the different entities of the computer system, that is, the CMCS, SCMS, DCS, CI, TS and TMS, implement each one or more particular method steps. Such a distributed implementation of continuous integration needs to be properly assessed and argued, in particular in regard of the different types of checks and tests performed. This probably requires prior art about CI, such as the one referred to in the application (see below).
- 2.6 The Board therefore essentially agrees with the appellant that the examining division went too far by considering all method steps of claim 1 to define a non-technical change management workflow for software development without providing enough detailed reasoning on the individual steps.
- 2.7 Starting from continuous integration on the other hand, or even discussing it while starting from D1, would probably mean taking into account the appellant's

argument that:

The present invention addresses a problem which known systems of continuous integration (CI) face with a large number of small pieces of software being delivered by developers to update the master code. This usually leads to a bottle neck of the SCMS, such as the revision control system (RCS) of D1. The present invention solves this problem by defining different stages of checking and testing and further on by distributing functionality among the alleged known entities of a combined SCMS and CMCS computer system. All features of claim 1 are inventively combined so as to provide increased security when developers change the software code.

- 2.8 However continuous integration does not appear to have been considered in any detail and the Board has doubts whether the search even covered it, because the examining division considered these features to be notorious or non-technical. Secondly, the field of search indicated in the search report is G06Q10 (business methods).
3. Article 111(1) EPC gives the Board the discretion to either exercise any power within the competence of the department whose decision is appealed, i.e. in this case the examining division, or remit the case to that department for further prosecution. According to Article 11 RPBA, the Board shall not remit a case unless special reasons present themselves for doing so.
4. The Board considers that the present situation constitutes a **special reason** according to Article 11 RPBA justifying a remittal of the case to the department of first instance, because it is not in a position to

examine inventive step of claim 1 according to the main request without a further search being carried out. This is consistent with previous decisions T 787/18, reasons, point 6, T 98/21, reasons, point 10, T 72/20, reasons, point 5.

5. The Board therefore exercises its discretion under Article 111(1) EPC and Article 11 RPBA and remits the case to the department of first instance on the basis of the main request for further examination and an additional search for the aspects mentioned above.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The case is remitted to the examining division for further prosecution.

The Registrar:

The Chairman:



T. Buschek

W. Chandler

Decision electronically authenticated