

Automation for the Change Management Process

Dan Adrian Marior*

**Department of Automation, Electronics and Mechatronics, University of Craiova, 107 Decebal Street, RO-200440 Craiova, Romania (e-mail: marior.adi@gmail.com)*

Abstract: The present paper presents the author's approach on studying one of the existing plant/production processes in the Ford factories across the world, in order to automate it through applied research. This process has been put in place to deal with exceptional situations such as changes in the usual day to day life of the factory. The designers of the Manage the Change process (MTC) took into consideration modifications that directly impact on the production process and those which do not. The process turned out to be very complicated to automate (in part due to the human element) and it is not the only one of its kind.

Keywords Manage the Change process, workflow,

1. INTRODUCTION

Most if not all of the companies across the globe have automated systems to help them manage the business, electronic systems which are usually a mixture of hardware equipments (sensors, robots, assembly lines, computing equipment, tools, etc.) and software programs, in various forms, such as ERP and/or CRM software, collaboration and communication software, security software, and others. Some of these systems constitute decision support systems, the useful category of software which provides data for management to provide better and better solutions and actions to the increasingly harsh conditions the business world has to face to continue being on the market (for example CRM software, HelpDesk software, Business Intelligence Software or modules in various software conglomerates such as Sharepoint) (De et al., 1985).

The DSS (Decision Support Systems) category thus represents software that captures, records, forwards and possibly processes data according to various business rules, in order to complete the puzzle of data used to manage the company. Usually DSS systems do not take decisions by themselves (no matter how simple) unless directed or programmed to. There are 3 categories of DSS, passive (do not offer suggestions or solutions), active (offer suggestions or solutions) and collaborative (allow the decision maker to make his own choices or modifications, overruling the ones calculated by the DSS – there is a repetitive process of refining the solutions). The difference between DSS and systems that can actually manage entire plants (Bullinger et al., 1986), with minimal or no human interaction (Chakraborty et al., 2005), is very important and has to be underlined.

The second section of the paper reveals a few details about the change process in the Ford plants which is the target of the case study. The automation potential of the Manage the Change flow in Ford plants was found to be sufficient for an implementation. The next section

describes the process in detail and the fourth section describes the approaches the authors chose to the aforementioned process only to conclude with the future stages of the corresponding application lifecycle.

2. MANAGE THE CHANGE PROCESSES

The Ford plant in Craiova itself has been going through a change process since the taking over from the prior owner of the factory assets.

Change processes sometimes happen at the level of whole factories (for example Dacia Pitești S.A. started up a new factory in Morocco to take over a part of the production, the retooling and training processes of the employees are going through since Ford acquired the assets in order to be able to produce new models).

These changes have been profoundly thought out by companies that make it their business to sustain and consult other companies while going through such changes. LaMarsh is such an example and its model has become one of the best choices in the field. Their model can be seen in figure 1; a preliminary analysis of the figure reveals the inner logic of the process, which resembles designing an adaptive system. The LaMarsh process consists of a few preliminary stages (identifying, preparing and planning the change, continues with the implementation of the current change and goes on continuously with the monitoring for any change in the process and adapts to it).

The Ford plant here in Craiova adheres to the same model and it has been translated to the process that can be seen in figure 2 (appendix B). The process is called Manage the Change and every change implementation is overseen by an adhoc interdepartmental group called Manage the Change Committee (MTCC).

The MTCC deals with most of the practical aspects of the change implementation; there are nevertheless other permanent departments such as SREA (Supplier Request

for Engineering Approval) and WERS Support (Worldwide Engineering Release System) which will be discussed in the following section.

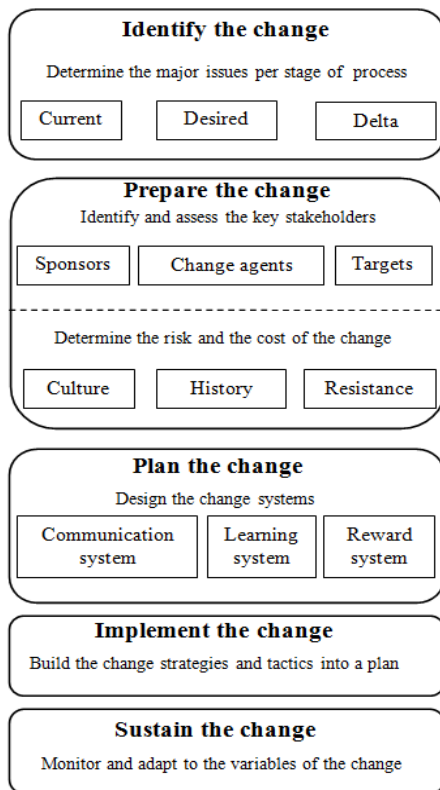


Fig. 1. Change management – LaMarsh model

The MTCC has a chosen leader (either elected or appointed by the top management).

In the course of the research the authors have found there are two versions of the process, the ones illustrated in figures 2 and 3 (appendix) respectively. Actually, in a sense, the two depict different aspects of the same process, so they could be considered different views on the same subject. There are differences in the execution of the flow and also in the manner of implementation which will be discussed also in the next section.

3. PROCESS DESCRIPTION

All the description of the process is in accordance with the corresponding internal documents that had to be studied for the first part of the authors' research. The current process for managing the changes has been designed with the first step being all the necessary approvals that have to be collected. In order to move on to the second step the system has to wait for the approvals to be in. The approving is assigned to the SREA and the WERS Support departments.

These two departments are comprised of engineers, project managers and technology specialists and eventually other persons whose opinions are valuable in the context of any change. WERS is also the name of a software application accessible only to the Ford

employees from the internal portal of the company, which deals with the preliminary aspects of implementing a change and is the medium through which the departments of Product Engineering release parts and assemblies/subassemblies. The first sequential version of the workflow is more about the documents and the logging of the change implementation process, as can be seen in figure 2. The second figure is quite self explanatory and the focus will not be on it very much, but there are a few aspects worth mentioning.

Firstly, this is the abstraction of the steps to be taken in case anyone notices a change is in order, never mind its importance. The person (or persons) should fill in an electronic form through the dedicated internal web portal and from here it is posted or sent to the MTCC. Should all the relevant stakeholders agree upon the change, it goes on to the next step, in which the MTCC ensures that all the prerequisites for the implementation of the change have been met. Otherwise the form returns (after the proper logging) to the submitter(s) with the recommendation that all necessary approvals should be obtained. Secondly, in case not all the prerequisites are met, the fact is logged and the form returns to the submitter with the announcement that it will not be implemented. Ultimately, in the most fortunate case, the change is implemented and is followed only by the corresponding general announcements. A much more detailed view is shown in figure 3 (Appendix A).

The first stage of the process is represented by the synergy between the full SREA approval and the publishing of the change notice in the WERS system (WERS acknowledgement). After this step the MTC Leader has a discussion with the MTCC in which the committee rules over the feasibility of applying the change. This can be regarded as an ad hoc feasibility study, as there are certain changes that may do more bad than good (have a low benefit/cost ratio, or after a more thorough looking into the matter prove to be quite useless in reality and would only be regarded as money ill spent). Conversely, if the filter applied in step 2 of the process has helped deeming the change necessary, the full MTCC approval is needed to continue. If the change does not have any impact on the production processes it is implemented accordingly (for example changing the uniforms of the workers).

The next step forces the MTC Leader to load the change into the MTC database and to oversee the completion of every item on the MTC checklist, as needed to begin the implementation. Afterwards a revision is in place, by the MTCC, in order to agree on proceeding with the change. In case of a failure to agree on proceeding, the originating document returns to the author, otherwise the MTC checklist is one again processed and if all items are covered and there are no open issues, the MTC Leader proceeds with the change. Otherwise, the incomplete issues are discussed between the MTCC and the Leader and the next steps are determined and implemented. The process logic flow then returns to the step where the Leader checks for item completion (from the list).

Returning to the best case scenario where all the items on the checklist have been completed and there are no open issues, the MTC Leader next ensuring the notification form is completed by all the customers and sent to all affected plants. The change is next logged in the database. The next phase is the feedback collection phase, which is standard enough for all kinds of processes. Actually, in this case, the designers of the process specified a period of 30 days to allow for the feedback collection.

4. WORKFLOW IMPLEMENTATION

The authors approached the automation of the workflow in two stages. The first one was to transform the original process to be accommodated with the help of a sequential workflow, although it was much better suited for state machine workflows. This first attempt would likely allow studying how the workflow behaves as a whole (what errors or problems would appear) and the integration with the other modules or software it would eventually have to communicate with; in other words, the sequential workflow would allow, in the authors' vision, for a more appropriate testing phase for the project. Ultimately, after the preliminary conclusions would have been drawn, the real implementation with the help of state machines could replace the sequential version.

Any state machine workflow can be transformed into a sequential one, but the transformation does not always render a perfect replica. There are issues regarding the ability of the state machines to return with the execution to a previous point which can only partially be represented with "while" code sequences. In order to implement both versions the authors used a Sharepoint 2010 based development environment. As the virtual machine used by the author in other papers – for example in (Marior et al., 2011) – was very slow, due to performance issues, the better approach was to simulate a virtual machine with the help of a physical machine and special backup software. However, this approach was an extreme one. The performance issues were due to virtualizing Sharepoint Server 2008; Microsoft recommends using a separate machine dedicated to running their database server (tec, 2012), but depending on the chosen infrastructure performance lag can be acceptable, especially in development environments (cri, 2012).

The Sharepoint infrastructure is comprised of a dedicated portal to host the workflow, and this in turn contains various lists and libraries to help implement it. The lists `Changes_type`, `SREAandWERS-approvals`, `Issues`, `Changes_end`, `Feedback`, `Pending approval`, and `No_Impact` are customized announcements lists used to post certain pieces of information regarding the flow of activities (for example the successful stop of the workflow execution) or to make certain decisions. The lists `Tracking_approvals`, `Tracking_issues` and `Feedback_tracking` are just what their name implies, tracking lists which contain indicators of various parameters. For example, in the list `Tracking_approvals` there is an indicator counting the number of approvals of

the document (practically the number of persons who have approved it) so this ensures the workflow does not start until the indicator has not reached the desired value (for example 5).

A short analysis revealed that the tool of choice for this kind of workflow is Sharepoint Designer 2010, and there was no reason to use Visual Studio 2010. Sharepoint Designer 2010 has suffered many improvements since the 2007 version, in order to be usable in approaching even more complicate workflows or scenarios, which were almost impossible to reach before without the use of Visual Studio. It can build workflows that are based on content type, lists, site level and also it can support building reusable workflows (can be imported in Visio or Visual Studio). In the present case study a site level (site wise) workflow was chosen. The process is modelled in figure 4. It is a three step workflow; the second steps represents a one day delay or wait time to allow for the collection of the feedback items, as per the description of the process in section II, except the authors chose the value of one day instead of 30. The authors would also like to remark that unfortunately the code behind the graphical programming representation in figure 4 is not possible to obtain.

As long as the value of the indicator has not reached for example 5 (the MTCC has 5 members), the workflow does not start. The indicator counts the number of items in the `SREAandWERS – approvals` list. If the condition is met and there are no open issues (no items in the `Issues` list) the flow continues by checking if the changes will impact the production process or not and sends an appropriate email to the MTC Leader (the content is programmatically adapted to either situation). At the end of this second step an item is created in the `Changes_end` list, representing the general announcement that the implementation has been successful. In step three the workflow engine checks the indicator in the `Feedback_tracking` list (which represents the number of feedback items inserted in the `Feedback` list) and sends an appropriate email which informs the Leader about the presence of items. This way, if no items have arrived in the `Feedback` list, there is no need to waste time looking for them; this step concludes the workflow.

The other way to implement the MTC workflow is to develop a Visual Studio 2010 application. This IDE is perfect for designing and creating applications for the Sharepoint environments in general, not only workflows. A short feature comparison between Sharepoint Designer and Visual Studio reveals the ability of the latter to provide functionality for the real MTC workflow, based on state machines. The application is in its initial development phase, and it models the workflow with state machines, transitions, activities and events. In figure 5 there is the document outline of the Visual Studio project.

The first two states are represented in tree view, with their events that trigger various actions; the transition from state one to state two can also be seen. The first state represents the first step of the workflow in figure 3, the collection of the approvals and state two represents the

second step, in which the MTC filter is applied. Of course, the event that triggers the transition from state one to state two is the full approval collection from the SREA department and the WERS application.

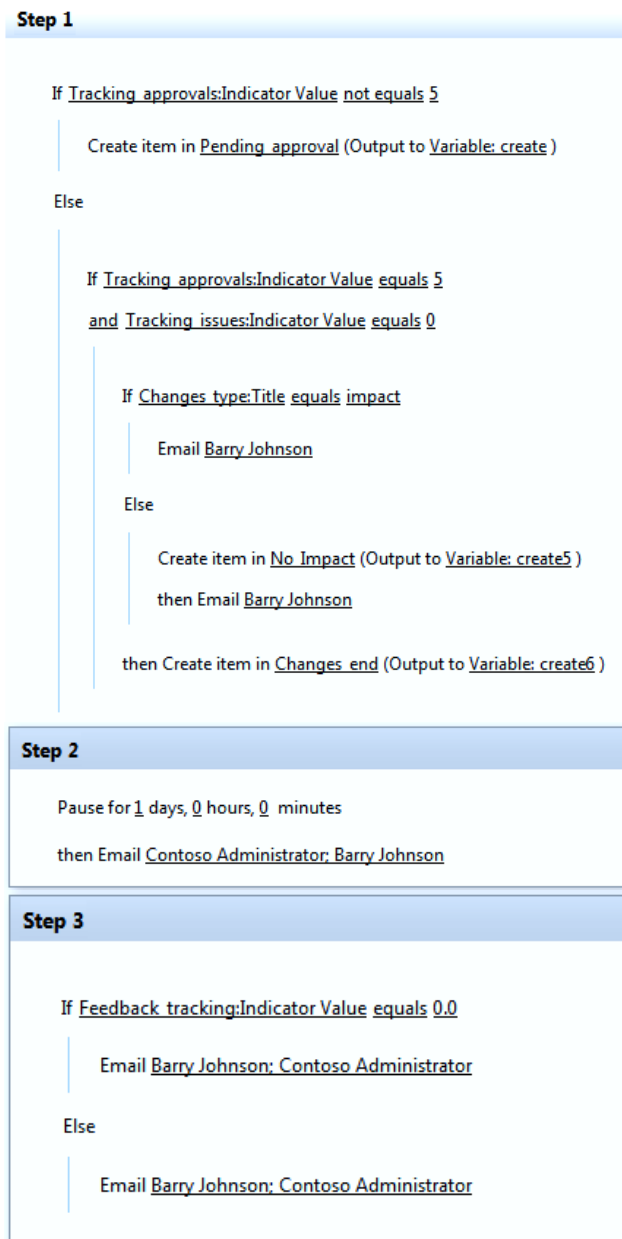


Fig. 4. MTC sequential workflow

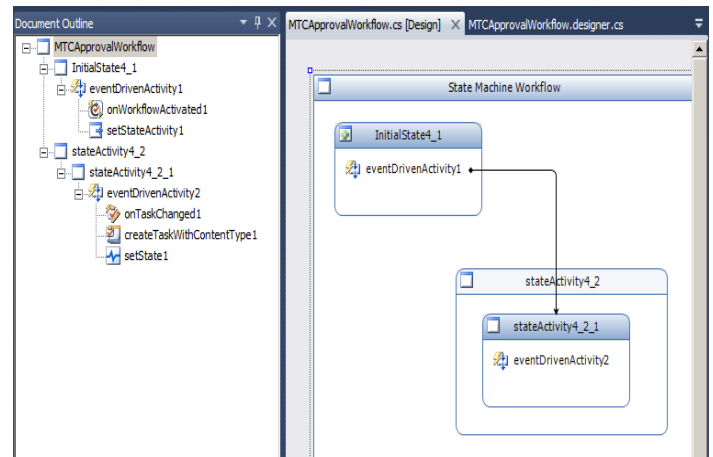


Fig. 5. MTC Visual Studio Project View

5. CONCLUSIONS

The first approach to the process, with the help of Sharepoint Designer graphical programming is complete, and needs no further modifying. However this is not the case with the Visual Studio application, which still needs to go through the development and testing phases, followed by the deployment in production in case it meets the expectations. The initial goals of the research were only partially achieved but for now it appears they will be reached. The premises for automation were unclear at the beginning as the processes which include document passing of some kind can be at least partially automated, but it was not obvious if modifications would be necessary in order to achieve the same for the present process. For the first method of implementation there were necessary a series of modifications, indeed, but the authors hope that no more will appear in the course of development of the second application.

ACKNOWLEDGMENT

This work was supported (in part) by the grant no. POSDRU/6/1.5/S/14b.

Appendix A. FIRST APPENDIX

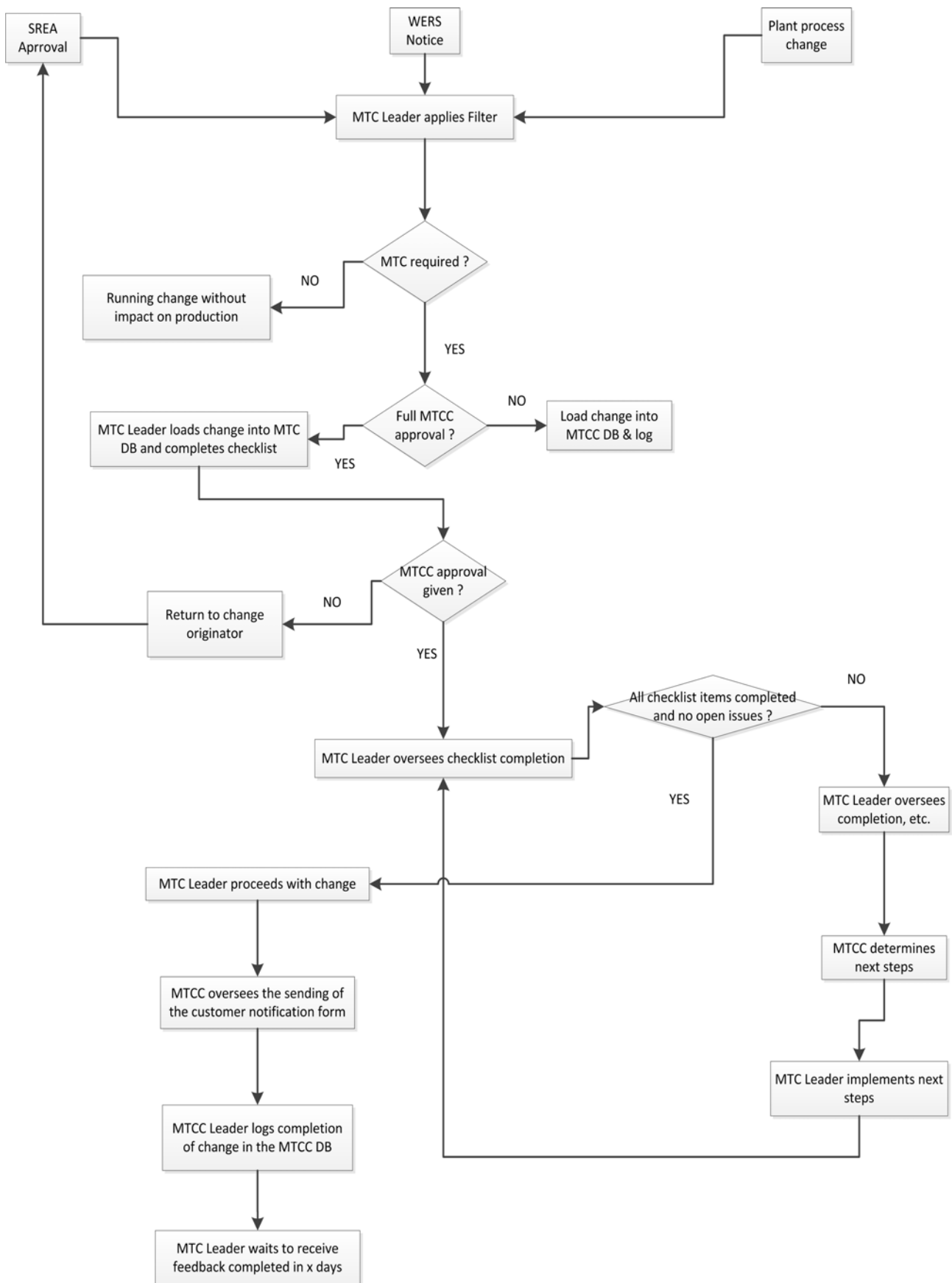


Fig. 3. Normal view of the MTC process

Appendix B. SECOND APPENDIX

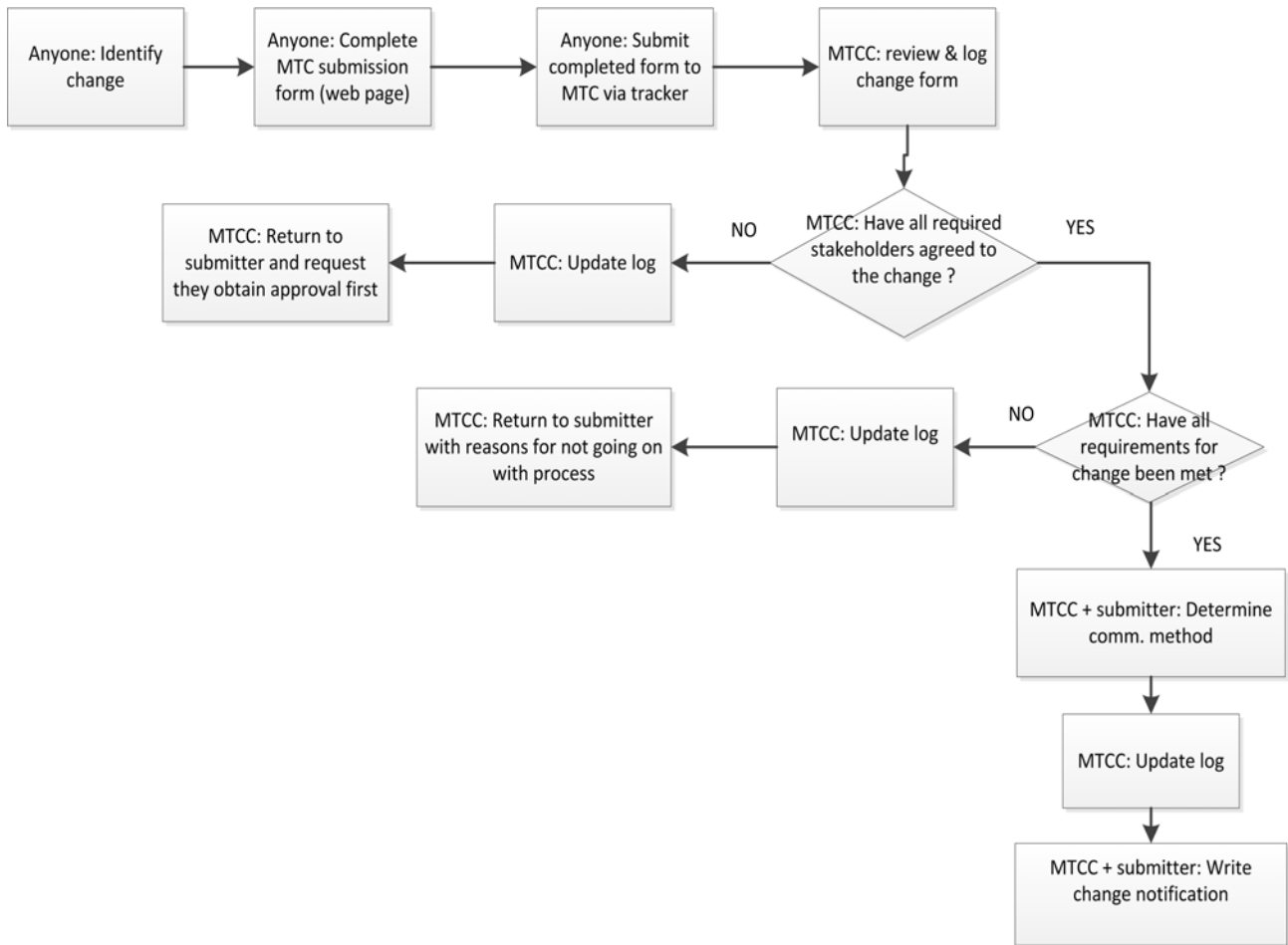


Fig. 2. Short perspective on the MTC process

REFERENCES

- De S., Nof S., Whinston A. B. (1985), "Decision Support in Computer – Integrated Manufacturing", *Decision Support Systems*, Vol. 1, No. 1, pp. 37-56.
- Bullinger H.J., Warnecke H.J., Lentz H.P. (1986), "Toward the factory of the future", *International Journal of Production Research*, Vol. 24 No. 4, 1986, pp. 697-741.
- Chakraborty S., Tah D. (2005), "Real time statistical process advisor for effective quality control", *Decision Support Systems*, Volume 42, Issue 2, November 2006, pp 700-711.
- Marior D. A., Cîrciumaru C. (2011), "Management workflow for consultants in a company", *Proceedings of the 15th International Conference on System Theory, Control and Computing*, October 2011, Sinaia, Romania
<http://technet.microsoft.com/en-us/library/ff898403.aspx>, 12.02.2012 at 15:46
<http://www.criticalpathtraining.com/Members/Pages/default.aspx>, 13.02.2012 at 18:00