# Mind-mapping: An effective technique to facilitate requirements engineering in agile software development

**Article** · December 2011

DOI: 10.1109/ICCITechn.2011.6164775



**2 authors:**

Imran Mahmud
Universiti Sains Malaysia



Vito Veneziano
University of Hertfordshire

# Mind-mapping: An Effective Technique to Facilitate Requirements Engineering in Agile Software Development

**Imran Mahmud, Vito Veneziano†**

Department of CSE, IUBAT—International University of Business Agriculture and Technology, Dhaka, Bangladesh

†School of CSE, University of Hertfordshire, Herts, United Kingdom

imran@iubat.edu, v.veneziano@herts.ac.uk

**Abstract**

**Merging agile with more traditional approaches in software development is a challenging task, especially when requirements are concerned: the main temptation is to let two opposite schools of thought become rigid in their own assumptions, without trying to recognize which advantages could come from either side.**

**Mind mapping seems to provide a suitable solution for both parties: those who develop within an agile method and those who advocate proper requirements engineering practice. In this paper, mind mapping has been discussed as a suitable technique to elicit and represent requirements within the SCRUM model: specifically, we have focused on whether and how mind maps could lead to the development of a suitable product backlog, which in SCRUM plays the role of an initial requirements specification document.**

**In order to experimentally assess how effectively practitioners could rely on a product backlog for their first development sprint, we have identified the adoption of mind maps as the independent variable and the quality of the backlog as the dependent variable, the latter being measured against the "function points" metric. Our hypothesis (i.e., mind maps are effective in increasing the quality of product backlogs) has been tested within an existing SCRUM project (the development of a digital library by an academic institution), and several promising data have been obtained and further discussed.**

**Key Index: Mind Map, Requirement Engineering, Agile, Scrum**

## I. INTRODUCTION

At the beginning of any software development project, one of the first burdens on project managers and developers' shoulders is to decide which approach and tools are to be adopted for the project, sometimes even before any significant requirement has been captured and/or analyzed.

Such decisions can be (and usually are) made by relying on literature or guidelines (see [1]): however, when they are not deterministically constrained by the problem domain or other "enforcing" factors (for example, in development of safety-critical systems, agile approaches are usually not recommended because of expected process features and outcomes such as accurate documentation, requirements evolution has to be traceable, etc.), still committing the project to a given process model (linear, incremental, agile: see, for instance, [2]) as well as to a given technology (or tool) comes with an incredible responsibility and it is often little more than a blind bet [3].

In this paper, we aim at evaluating how requirements-related activities (and particularly the requirements elicitation process) within an agile development project can be facilitated by using a specific cognitive and knowledge-management technique called "mind-mapping", together with its properly derived software tools, and how such technique could be reconciled with a sounder requirements management perspective and, at the same time, mitigate most of agile criticalities and unwanted side-effects.

## II. REQUIREMENTS ENGINEERING WITHIN AGILE AND, SPECIFICALLY, SCRUM

According to Baker [4], "a requirement is a description of the needs and desires for a system or application. A system requirement describes functions, features and constraints". Whilst agile philosophy has flourished upon the intrinsic volatile nature of requirements and customers perception of these, it is still an open question whether an agile development process can efficiently help capturing, analyzing and specifying as many and as worthy requirements as possible, by supporting the collection of a complete and consistent set of requirements, independently from (or on top of) any stakeholder individual perspective or wishes (where obviously most of agile advantages lay).

As reported from Cardozo et al. [5], agile methods are increasingly gaining popularity in the Software Industry. Many expectations seem leading to their adoption: agile methods supposedly enable software engineers to avoid

987-161284-908-9/11/$26.00 © 2011 IEEE

project problems such as low productivity, schedule delays, high costs, and lack of people motivation.

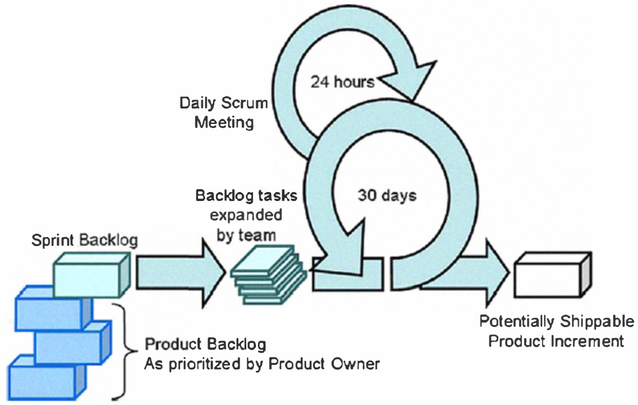


Fig 1. SCRUM model. Source: [6]

The product backlog is expected to be the master list of all functionalities desired in the product. When using SCRUM, it is not necessary to start a project with a lengthy, upfront effort to document all requirements: a SCRUM team and its product owner (PO) begin by writing down everything they can think of easily. This is almost always more than enough for a first sprint. The product backlog is then allowed to grow and change as more is learned about the product and its customers. Still, in order to fully exploit the potential of a product backlog, it is usually important to spend some more time in identifying all proper POs and gathering as much information as possible from them.

In order to capture (or elicit) requirements, several techniques, such as interview, use cases and scenarios (often further simplified and represented as "user stories" and/or clarified by "metaphors": see [7]), observation and social analysis, focus group, brainstorming and prototyping can be easily imported from RE and used within an agile development approach (for a list, see [8]; [9]). All of these techniques seem to rely on or, at least, not to breach the values as highlighted by the Agile Manifesto [10], namely

- Individuals and interactions over processes and tools
- Customer collaboration over contract negotiation
- Working software over comprehensive documentation
- Responding to change over following a plan

Whilst some attempts have been made to mitigate the supposedly well-known cons of an agile approach (e.g.,[11]) to software analysis and design, the temptation to adapt and export RE more traditional strategies and formal solutions to agile contexts come with a number of problems and issues that, in the long run, could prevent the agile approach to work in full [12], and leaves some problems untouched, like

- anticipating as many requirements as possible up-front, which would cost more money to change at a later stage
- leaving the programmers coding alone, without involving the customer in test-driven software development
- prototyping exercises that answer more than one question (one "spike") at once, keeping the customer away from the responsibility of "getting the requirements right"

The same adoption of traditional requirements management tools (RMTs) could invalidate or anyhow counter the expected positive effects of going agile, as these tools have been mostly developed around RE sets of activities embedded within a more traditional software development life cycle.

For all these reasons, it appears urgent to rethink requirements-related activities from a "native" agile view-point.

## III. MIND MAPPING AS AN AGILE ELICITATION TECHNIQUE

Out of several available techniques and derived tools, mind-mapping has been considered by both practitioners and academics as a useful manner to manage the requirements elicitation process within agile-based development projects [13][14][15][16], since the development of the initial product backlog..Mind mapping comes as the natural answer to a question well identified by Swan [17]: "Why use a linear tool for a nonlinear process?"

Swan writes: *"Requirements gathering like the consultant was doing on the cold call is, by its very nature, a nonlinear activity. People tend to offer up ideas and insights in no particular order. Trying to capture the sort of activity using traditional business applications can actually interfere with the process, as people get caught up trying to think about what should come first, then second, and third... and so on. Mind mapping, unlike other ways to capture information, it allows you to work the way people tend to think: nonlinearly."*

Mind-mapping encourages people to think of, organize and represent information within a radial hierarchy, by locating the most important concept at the center of a given diagram and relate it to other concepts (or details of the first concept, or both) "situated farther and farther away from the diagram center" [18]. Although mind mapping seems having ancient and glorious roots [19], the notational technique (the so-called "mind maps") was firstly developed by Tony Buzan in the Sixties [20]. At the very beginning, it was used for taking notes quickly and in a visually effective way. Nowadays mind-mapping has built up a reputation for being a cognitive tool exploiting the full potential of information processing by involving both human brain's hemispheres: the left one, specialized for performing tasks related to language, analytic thought, sequential activities; and the right one, specialized on tasks involving multidimensionality, creativity, emotions, rhythm, geo-spatial information processing. By using both sides of the brain, mind-mapping is supposedly expected to let "them work together and thus increases productivity and memory retention" [21]: cognitive psychology seems sharing and fostering such expectations (see, for instance, [22]; [23]).

Instructions on how to make a mind-map are publicly accessible over the internet: whilst it is the same Buzan who encourages people in creating their own "style" in developing mind-maps, the basic technique is to take a plain white sheet, to write the title of the subject on the middle of the page and to draw a circle around it. Lines from the circles can be drawn and subdivisions or subheadings are then added. These subheadings can be circled in turn and new lines, subheadings, etc. can be further derived.

As an example, we have outlined our paper in the following mind map (Figure 2). The technique seems ideal for "covering" the whole space of a given topic (or problem, or system) and to identify important correlations, both in terms of "extensiveness" (or completeness) and in terms of "accuracy" (or details).

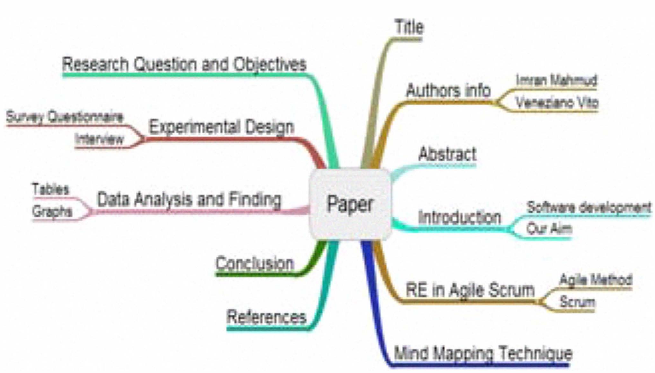


Fig 2. Mind map of the outline of this paper

## IV. RESEARCH QUESTIONS AND OBJECTIVES

How could mind-mapping assist software engineers to capture and process requirements outside traditional methods?

Several authors have already hinted at how and why such a technique could provide advantages in eliciting what customers need (rather than their perceived solutions: see [24]) or how to get there fast within an agile context (see [25]): others, like Hiranabe [15], highlight the practical and more agile-oriented immediate aspects of mind-mapping, as a technique useful for taking notes, developing meetings minutes and agendas, facilitating interaction between customers and developers.

In order to assess whether all assumptions and expectations rely on solid groundings, we have translated most of the aforementioned advantages into a set of objectives that could be experimentally evaluated. Our driving hypothesis was that mind-mapping, as a representational technique to be used across the whole requirements process used to develop an initial product backlog, intrinsically affects the ability:

- To identify a broader range of users and usages
- To identify more and more diverse requirements for and from different users

Whilst we were and still are fully aware that a longer and better structured research was needed to properly investigate the above objectives, as it usually happens in empirical research in software and requirements engineering, we took advantage of an existing project run by IUBAT (the development of a digital library) for deriving a case study around which to design a very simple two-conditions experiment: whether quality of requirements represented as a backlog product in SCRUM could be affected by the adoption of mind-mapping techniques during their elicitation and analysis process.

## V. EXPERIMENTAL PROTOCOL AND DESIGN

Four categories of stakeholders in our digital library were identified as participant in our experiment: three different types of users (librarians, faculty members and researchers, shortly referred as "teachers", and students) were identified as POs. Software developers were a fourth category of participants in our experiment: all participants were grouped by their purpose in dealing with requirements.

| Stakeholders | Purpose | N, sample size | Category Id |
|---|---|---|---|
| Librarians | Need to manage database of digital library and upload contents | 8 | A |
| Teachers | Faculty members and researchers might search various journals and articles, upload slide, handout or notes | 8 | B |
| Students | Students will search library to download e-book and other teaching materials. | 20 | C |
| Software engineers | Need to capture requirement for development as accurately as possible | 2 | D |

Each category of stakeholders (identified by a category id) had their requirements collected via a specific collection type.

The experimental design we adopted consisted of two conditions to be analyzed against two different groups of subjects within each category, one group acting as a control group (condition 1: no training in and no usage of mind-mapping) and the other one receiving some sort of intervention or treatment (condition 2: in our case, the exposure and training into using mind-mapping as a representational technique, and the explicit request of using it). A diagrammatic representation of the experimental design including a comparison between the two experimental conditions 1 and 2 is given below in Figure 3.

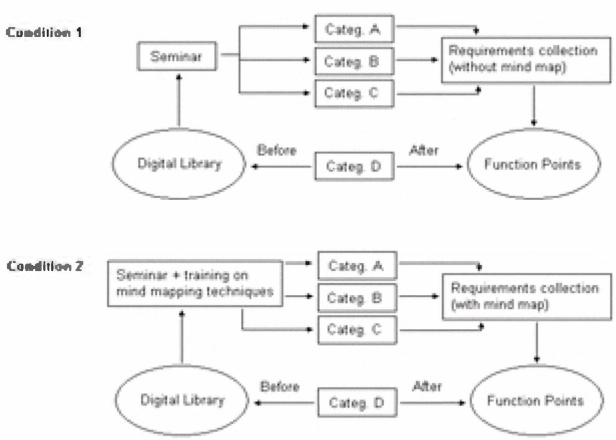


Fig 3. A comparison between the two experimental conditions.

As a basic protocol, our two experimental conditions (without and with mind-mapping) started with a seminar all participants had to attend, in which a brief introduction to the Digital Library was provided, together with an informal set of instructions on the task of collecting requirements.

However, whilst in condition 1 no instruction on how to capture and specify requirements was disclosed, for the group of subjects involved in condition 2, the seminar included an introduction to and some training into mind-mapping techniques. In both conditions, participants were asked to return, as an outcome, their necessary feedback on how the digital library content should be made accessible, including any relevant functional and non-functional requirement the digital library should meet.

The dependent variable has been assessed against a simple metric: the number of function points, as suggested by the IFPUG Functional Size Measurement Method (see [26]); for a broader reference, see http://www.ifpug.org/).

In condition 2, researchers used the seminar to introduce, train and discuss about mind-mapping techniques. Each member of the three groups belonging to the categories A, B and C was asked to hand-draw a mind map on a blank A4 sheet, and to submit it to software engineers alongside whatever the requirements collection type (for instance, teachers drew their mind maps during their interview). Once these were collected, software engineers counted the overall function points.

## VI. DATA ANALYSIS AND FINDINGS

Although the volume of collected data is far from allowing any analysis of strong statistical significance, we believe that at least some interesting trends can be highlighted, as represented by the tables below, starting from the analysis of the two bigger groups, made up of Students (category C).

| Participans Group C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Condition 1 | 3 | 4 | 3 | 3 | 6 | 5 | 3 | 5 | 3 | 4 |
| Condition 2 | 6 | 6 | 7 | 6 | 11 | 7 | 6 | 8 | 7 | 7 |

Table 1: Function points detected by the requirements specs of groups belonging to cat. C (Students)

We run an ANOVA (Single factor) on the function points outcome from this category of stakeholders and, since F > F*crit* (28.981 > 4.4138), we could reject the null hypotheses (mind-mapping having no effect on quality of requirements), as from the following Table 2.

| TABLE 2: SUMMARY ON DATA FROM TABLE 1 | | | | | | |
|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Average* | *Variance* | | |
| Condition 1 | 10 | 39 | 3.9 | 1.21 | | |
| Condition 2 | 10 | 71 | 7.1 | 2.32 | | |
| ANOVA | | | | | | |
| *Source of Variation* | *SS* | *df* | *MS* | *F* | *P-value* | *F crit* |
| Between Groups | 51.2 | 1 | 51.2 | 28.98 | 4.080E-05 | 4.41 |
| Within Groups | 31.8 | 18 | 1.76 | | | |
| Total | 83 | 19 | | | | |

Table 2: Summary on data from Table 1

We then analyzed results from the two groups of POs belonging to category B (teachers), as reported from Table 3.

| Participants Group B | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Condition 1 | 5 | 8 | 8 | 7 |
| Condition 2 | 9 | 11 | 13 | 12 |

Table 3: Function points detected by the requirements specs of groups belonging to cat. B (Teachers)

Again, as it emerges from the the ANOVA (Table 4) we performed on data collected from the groups B (Faculty members and researchers, i.e. "teachers") the difference between the two conditions seems significant.

| TABLE 4: SUMMARY ON DATA FROM TABLE 3 | | | | | | |
|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Average* | *Variance* | | |
| Condition 1 | 4 | 28 | 7 | 2 | | |
| Condition 2 | 4 | 45 | 11.25 | 2.9166 | | |
| ANOVA | | | | | | |
| *Source of Variation* | *SS* | *df* | *MS* | *F* | *P-value* | *F crit* |
| Between Groups | 36.125 | 1 | 36.125 | 14.694 | 0.0086 | 5.987 |
| Within Groups | 14.75 | 6 | 2.4583 | | | |
| Total | 50.875 | 7 | | | | |

Table 4: Summary on data from Table 3

In Table 5 and 6, we report data and analysis for members of the category A (librarians).

| Participants Group A | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Condition 1 | 17 | 14 | 12 | 14 |
| Condition 2 | 25 | 14 | 15 | 15 |

Table 5: Function points detected by the requirements specs of groups belonging to cat. A (Librarians)

As it can be easily noticed, differences between the two conditions are not significant between the two groups of librarians on whom we have tested out our hypothesis (F < Fcrit). However, whilst a larger sample would be needed for better evaluate these results, still they seem promising, especially when mind mapping techniques are used on stakeholders (like students or teachers) who are "less expert" POs than others (like librarians).

| TABLE 6: SUMMARY ON DATA FROM TABLE 5 | | | | | | |
|---|---|---|---|---|---|---|
| *Groups* | *Count* | *Sum* | *Average* | *Variance* | | |
| Condition 1 | 4 | 57 | 14.25 | 4.25 | | |
| Condition 2 | 4 | 69 | 17.25 | 26.9166 | | |
| ANOVA | | | | | | |
| *Source of Variation* | *SS* | *df* | *MS* | *F* | *P-value* | *F crit* |
| Between Groups | 18 | 1 | 18 | 1.1550 | 0.3237 | 5.9873 |
| Within Groups | 93.5 | 6 | 15.58 | | | |
| Total | 111.5 | 7 | | | | |

Table 6: Summary on data from Table 5

After getting a positive response from the three types of users (on top of the numbers, most subjects expressed their enthusiasm on mind-maps), we run a questionnaire on our group D (software engineers) after their first sprint backlog, in order to receive their feedback on how well the mind-mapping technique worked, on their side, to better receive and understand requirements included in the product backlog. Specifically, they were asked to evaluate mind-mapping (in a scale between 1, low, and 5, high) against the following five features of the overall product backlog:Results are given in the Table 7 below.

| | D1 | D2 | Mean value | % |
|---|---|---|---|---|
| General suitability | 4 | 5 | 4.5 | 90% |
| Usability | 4 | 4 | 4 | 80% |
| Analytical ability | 5 | 5 | 5 | 100% |
| Attributes finding | 5 | 5 | 5 | 100% |
| Satisfaction behind mindmap | 5 | 5 | 5 | 100% |

Table 7: how mind-mapping has been evaluated by stakeholders belonging to category D.

From this simple evaluation exercise, it is possible to highlight how mind-mapping seems very appropriate to analyze requirements (90%), helpful in capturing requirements (80%), and overall very supportive for making requirements as consistent and complete as possible.

## VII. CONCLUSION

Our hypothesis was that usage of a mind-mapping technique could lead to a significant improvement of the quality of the derived product backlog.The data we obtained show that, when mind mapping is adopted, the overall quality of the product backlog (measured against the "function points" metric) is significantly higher with "less expert" POs (library users), and at least not worst with more expert POs (librarians), than when the product backlog is the mere outcome of an informal requirements engineering approach.Whilst more experiments and larger amount of data would better ground the current investigation, it appears recommendable that mind-mapping and any derived technique should be considered as a valuable good practice to be pursued for setting up of a proper initial product backlog, whenever an agile development method like SCRUM is adopted.

## REFERENCES


[1] Cronholm S., 2008, "Using Agile methods? – expected effects", in G.A. Papadopoulos et al. (eds.), 2010, "Information Systems Development – Towards a Service provision Society", Proceedings of the 17th International Conference on Information Systems Development (ISD2008), Paphos, Cyprus, August 25-27, 2008, Springer

[2] Scacchi W., 2001, "Process Models in Software Engineering", in, J.J. Marciniak (ed.), *Encyclopedia of Software Engineering, 2nd Edition*, John Wiley and Sons, Inc, New York, December 2001

[3] Cohn M., 2009, "Best practices are dangerous when adopting agile", in his blog available at: http://blog.mountaingoatsoftware.com/ (http://blog.mountaingoatsoftware.com/best-practices-are-dangerous-when-adopting-agile)

[4] Natasha M. Baker (2008), Learn the Basics of Functional Requirements, Article published on http://www.brighthub.com/office/project-management/articles/11666.aspx

[5] Cardozo E.S. F., Araújo Neto J.B.F., Barza A., César França A.C., da Silva F.Q.B., 2010, "SCRUM and Productivity in Software Projects: A Systematic Literature Review", 14th International Conference on Evaluation and Assessment in Software Engineering (EASE), Keele University, UK, 12 - 13 April 2010

[6] Schwaber, K.(2004), *Agile Project Management with Scrum*, Microsoft Press, 2004

[7] Tomayko J.E., Herbsleb J., 2003, "How Useful Is the Metaphor Component of Agile Methods? A Preliminary Study", CMU-CS-03-152, School of Computer Science, Carnagie Mellon University (also indexed as CMU-ISRI-03-100)

[8] Paetsch F., Eberlein A., Maurer F., 2003, "Requirements Engineering and Agile Software Development", in WETICE '03 Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society Washington, DC, USA

[9] Sillitti A., Succi G., 2005, "Requirements Engineering for Agile Methods", in A. Aurum and C. Wohlin (eds.), *Engineering and Managing Software Requirements*, Springer

[10] Agile Manifesto, 2001. http://agilemanifesto.org/

[11] Oliveira Basto da Silva J.G.A, Rupino da Cunha P., 2006, Reconciling the irreconcilable? A software development approach that combines Agile with Formal, Proceedings of the 39th Hawaii International Conference on System Sciences – IEEE

[12] Tomayko J.E., 2005, "Engineering of Unstable Requirements Using Agile Methods", Carnagie Mellon University, USA, available at: http://cf.agilealliance.org/articles/system/article/file/1162/file.pdf

[13] Jaafar J., 2009, "Collaborative Mind Map Tool to Facilitate Requirement Engineering (RE)", MSc Thesis, University of Manchester: an homonymous paper (authors: Jaafar J., Atan M., Abdul Hamid N.N.A.) has been derived, accepted and to published in the Proceedings of the 3rd International Conference on Computing and Informatics, ICOCI 2011, Bandung, Indonesia, 8 – 9 June 2011

[14] Chenal D., 2008, "Mind Mapping Improves Software Requirements Quality, Communication and Traceability", *tech brief*, QAVantage

[15] Hiranabe K., 2008, "Exploring User Requirements through Mind mapping", http://www.change-vision.com/en/ExploringUserRequirementsThroughMindMapping_Letter.pdf

[16] Rathore A., 2007, "Requirements management, user stories, mind-maps, and story-trees", in his blog "epistemologic", available at: http://epistemologic.com/2007/04/08/requirements-management-user-stories-mind-maps-and-story-trees/

[17] Swan H., 2010, "Visually Mapping Requirements: An uncommon approach that just might make your job easier", RQNG, http://www.requirementsnetwork.com/node/2317

[18] Dykeman M., 2010, "Ten techniques for mind mapping from Tony Buzan", in his blog: http://thoughtwrestling.com/blog/mind-mapping-techniques-buzan/

[19] Tipper M., 2009, "Tony Buzan Did Not Invent Mind Mapping!", in his blog: http://www.michaelonmindmapping.com/blog/mind-map/tony-buzan-did-not-invent-mind-mapping/

[20] Buzan T., 2011, "Tony Buzan's Who's Who Curricula Vitae", ThinkBuzan.com, available at: http://www.thinkbuzan.com/downloads/tony_buzan_whos_who.pdf

[21] Larkin J.H., Simon H.A., 1987, "Why a Diagram is (Sometimes) Worth Ten Thousand Words", *Cognitive Science*, 11, 65-99

[22] Brinkmann A., 2005, "Knowledge Maps – Tools for Building Structure in Mathematics", *International Journal for Mathematics Teaching and Learning, CIMT*, University of Plymouth, available at: http://www.cimt.plymouth.ac.uk/journal/default.htm

[23] Zhang Y., Xiao S., Yang X., Ding L., 2010, "Mind Mapping Based Human Memory Management System", *International Conference on Computational Intelligence and Software Engineering* (CiSE 2010), vol., no., pp.1-4, 10-12 Dec. 2010

[24] Robertson S., Robertson J., 1999, "Mastering the Requirements Process", Addison-Wesley

[25] Larman C., 2003, "Agile and Iterative Development", Prentice Hall

[26] Haug M., Olsen E.W., Bergman L., 2001, "Software process improvement: metrics, measurement, and process modelling", Springer, p.99