How to Enhance the Creativity of Software Developers: A Systematic Literature Review

Reshma Hegde
Microsoft Dynamics AX
Microsoft Corporation
Fargo, USA
rehedge@microsoft.com

Gursimran Walia
Computer Science Department
North Dakota State University
Fargo, USA
gursimran.walia@ndsu.edu

Abstract—Success during software development depends on the creativity of software engineers. Knowledge plays a very important role in enhancing the creativity of software developers. Knowledge is available in different forms like repository knowledge (experiences of past projects) and community knowledge (gained through communication among software engineers). To help software engineers be more creative and successful, a systematic literature review was undertaken to find how knowledge influences creativity. The systematic literature review reports the various knowledge sources and how these can be accessed by developers to be more creative, and the methods used to access the knowledge sources.

I. INTRODUCTION

Knowledge plays a significant role in the creativity of software engineers and creativity of software engineers defines the success of projects. Creativity refers to “the ability to produce new and original ideas and things” [1]. Graham Wallas described the stages of the creative process as, the preparation stage, where data is collected to solve a problem, the incubation stage, where the brain works unconsciously on the problem with existing knowledge, the illumination stage, where ideas start emerging and links are established between known facts, and the verification stage, where ideas are worked on so that it can be communicated to others [8]. The creative process is described as rapid idea generation phase followed by periods of incubation and reflection before these ideas are refined or new ideas are created [16].

Software development is conceptually a complex, knowledge intensive and cognitive activity. Effective software development relies on the knowledge collaboration and on the creativity of software developers [9]. Creative thinking requires the ability to integrate internal (stored in head) and external knowledge (stored across different artifacts and developers) for performing a task. Creative thinkers search for new ideas by manipulating existing knowledge to see different problems, opportunities and solutions [4]. On that, previous researchers have conducted studies to analyze the role of various forms of knowledge (prior knowledge, analogies etc.) on the creativity during software development. This includes case studies [6, 13], experiments [12, 7], surveys and interviews [10] with students and professionals from various domains (not just limited to SE field). Their research found that knowledge plays a positive role and can help professionals to be more creative in their work. Knowledge in different forms helps software engineers to come up with as many ideas as possible to solve a problem. The results also show that prior knowledge helps engineers get new ideas/inspiration for the current problem [5].

Further, knowledge in the form of examples helps engineers improve their understanding of the problem, and to add innovation to their solution or to already exist ones [10]. Similarly, knowledge in the form of analogies helps engineers see the problem from different viewpoints and generate more ideas to solve their problem. The results from an empirical study showed that designers came up with more ideas with the help of analogies (as compared to without analogies) [7]. Similarly, sharing of knowledge among peer engineers makes the engineers see a problem in a different perspective which triggers creative ideas. In an experiment, software engineering and social workers were provided an existing object and tools to share their ideas and connect with other peers. The results showed that subjects who shared their knowledge generated more ideas and developed an innovative product [17].

While the above results provide some evidence that using “knowledge” can improve creativity, a subset of previous research [5,7,10,17] narrowed down the focus on different forms of knowledge (e.g., prior knowledge or analogies) that had an impact on the creativity of software engineers. It was found that information was scattered among different papers and each provided information on only one form of knowledge, how that knowledge type can be accessed to enhance creativity, and what problems are faced [5]. Hence, it was imperative to collect this scattered information and organize them in a single study to be able to better understand the effect of knowledge (in different forms) on creativity of software engineers. To realize this, a systematic literature review was conducted.

A systematic literature review is a formalized, repeatable process in which researchers systematically search a body of literature to document the state of knowledge on a particular subject. This approach is more commonly used in medical field to document high-level conclusions drawn from a series of detailed studies [3]. To be effective, a systematic review was driven by an overall goal: Identify all the sources and forms of knowledge that helps improve creativity of software engineers, the ways that the knowledge is accessed, and their limitations.

II. MOTIVATION

Our software companies (like Microsoft) want employees to be creative and come up with new innovations. Creativity among software engineers has become very important for the long-term success of the company. In order to understand what triggers or enhances creativity among software engineers, an ad hoc literature review identified that among several factors (i.e., intellectual abilities, knowledge, thinking, personality,
motivation and environment) knowledge has the most impact on the creativity among software engineers [13].

Software engineering is knowledge intensive and collaborative task. Its success depends on the collaborative knowledge and skills of the software engineers involved in the process [9]. Software engineering knowledge is scattered across different resources (e.g., artifacts, code, documents, peers, lessons learned etc.) and continually flows through the entire software development ecosystem. From this perspective, knowledge plays an important role in triggering the creativity of software engineers. The main motivation is to help software engineers understand the importance of knowledge and properly utilize the knowledge to have a creative edge. This work aims to help software engineers understand the benefits of different forms of knowledge (e.g., prior knowledge, examples and analogies) and include them in their day to day software development activities, and to help motivate the software developers to store and share their experiences with their peers and generate creative ideas to solve their problems.

III. SYSTEMATIC LITERATURE REVIEW

This section describes the process used for performing a systematic literature review of, role played by knowledge in enhancing the creativity of software professionals. This section provides a high-level description of the review protocol. More details of the review protocol can be referred in [19].

The systematic review is based on guidelines established by Kitchenham et al. [2, 3]. The purpose of performing a systematic literature review is similar to that of performing any scientific experiment. Procedures are established, followed, and reported on so that other researchers are capable of replicating the work. Following a systematic review process also provides a high degree of control over the type and quality of reference works that will be included in the review and helps to provide support for the conclusions of the literature review. In accordance with the guidelines established by Kitchenham [2-3], the review protocol specified the research questions to be addressed, established a list of databases, conference proceedings, journals, etc. from which primary sources were selected, and established criteria for including sources, extracting data from the included studies, and evaluating their quality. The subsequent steps closely mirror those of any other experiment in that the protocol is executed; the results are analyzed to address the research questions, and are discussed.

Research Questions: A high-level research question (What is the role played by knowledge in creativity of professionals in software development organization?) was decomposed into three more specific RQ’s and related sub-questions. A list of these RQ’s is available in Table I. RQ#1 gathers and analyzes the evidence from the literature to investigate the extent to which the knowledge has an effect on creativity. RQ#2 gathers and analyzes different ways in which knowledge is accessed and utilized to be creative. It also gathers all the shortcomings of the knowledge access methods. RQ#3, a meta-question gathers additional evidence along with the information from RQ’s 1 and 2 to suggest improvements of the knowledge access methods to support creativity in software development.

Source Selection and Search: Initially, an ad hoc review was performed in order to assist with the development of search strings and to provide a list of potential conference proceedings and journals to be manually searched. Selection criteria were developed to establish a list of initial databases to be searched and conference proceedings or journals to be searched manually. References from primary sources were also included if they were relevant. The source list which was searched included these Databases: ACM Digital Library, IEEE Explorer, APA PsycINFO, ScienceDirect.

Study Execution: After executing search strings on all databases and manually searching through selected sources, 6696 papers were found, that were initially screen for inclusion based on their title and abstract. After this, 43 papers remained and were read in their entirety to evaluate their quality and applicability. After reading all of these papers, 16 remained.

Study Inclusion and Exclusion Criteria: The papers chosen for inclusion in the literature review are all based on empirical research in order to provide greater confidence that the results are not anecdotal. A set of inclusion/exclusion rules were created so that only the most relevant studies were considered for the systematic review (as shown in Table II).

Quality Assessment: Quality assessment was performed to assess the study design, bias, validity, and generalizability of results for the final list of papers. Each paper was evaluated using a quality assessment checklist in accordance with the guidelines published by Kitchenham et al. The results of the quality assessment were good and can be found in [19].

Data Extraction and Synthesis: Once all of the papers had undergone quality assessment, data extraction was performed on all papers. A data extraction form (see Table III) was developed to ensure consistent extraction across all papers. This data was synthesized to answer the RQ’s (in Table I).

Validity Threats: Most of the sources found during the review focused on the relationship between knowledge and creativity in the requirements and design stages of software

<table>
<thead>
<tr>
<th>TABLE I. RESEARCH QUESTIONS AND MOTIVATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research Question # and Description</td>
</tr>
<tr>
<td>RQ1. Is there any evidence that knowledge influences creativity in Software Engineering?</td>
</tr>
<tr>
<td>RQ2. Is there any evidence that the Knowledge access methods identified in literature enhances the creativity of software professionals?</td>
</tr>
<tr>
<td>RQ3. What are the ways to improve the current knowledge access methods so that it can support the creative thinking process of software professionals?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. INCLUSION AND EXCLUSION CRITERIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inclusion Criteria</td>
</tr>
<tr>
<td>Publications that are related to Knowledge and creativity in software engineering.</td>
</tr>
<tr>
<td>Publications that reported the results on creativity in Software Engineering.</td>
</tr>
<tr>
<td>Publications that contained empirical results.</td>
</tr>
<tr>
<td>Publications that are related to creativity and knowledge which were non software engineering, but could be applied to software engineering.</td>
</tr>
</tbody>
</table>
engineering. There was no literature that focused on the creativity during the testing stage of software development. We anticipate that this relationship (Knowledge vs. Creativity) would hold true for other stages of software development, due to the nature of testing as a knowledge intensive activity as well. However, there is no empirical evidence to underscore this claim. Also, the sources which reported the effect of knowledge on the creativity during the design included designers that were not limited to SE background. However, the results were consistent across SE and non-SE studies.

IV. REPORTING THE REVIEW

This section reports the results organized by each RQ using the information extracted and synthesized from all the papers.

A. Question 1: Does knowledge influences creativity?

A review of the literature indicates that knowledge impacts creativity during software construction. Knowledge (available in various forms) helps software engineers come up with a larger number of solutions to their problems (some of which may be original). While most of the empirical studies focused on the relationship between design knowledge and the creativity, the relationship and the resulting effects can be applied to any development stage since each stage involves knowledge transfer of some sort. The review also uncovered studies from the Psychology and Human Cognition whose findings and results were generalized to the SE domain.

The review identified many forms of knowledge that can influence the creativity of software professionals. These include: a) Prior knowledge— useful information (ideas that could be extracted from design documents, magazines, sketches, notes, lessons learnt etc.) from previous projects. The previous projects could be one’s own projects or other’s projects; b) Analogies— are the entities which look similar to the entity under construction. This could belong from the same domain or other domains; c) Dynamic knowledge in tools – pertains to the knowledge that is embedded into the tool and is available to the engineers at any time or when user demands. Details of these knowledge forms and the empirical results on their effect on an individual developers’ creativity follows:

Cognitive Psychology has often cited the use of the prior knowledge as an important ingredient to create novel solutions. Humans are able to use the prior information (e.g., examples of other’s work) to learn produce new ideas and solutions. Similar to the psychology field, prior knowledge helps software engineers to generate more (and new) ideas or be creative. Specifically, software engineers during the early phase of design heavily depend on the prior knowledge (form their own and their peer’s prior work) in order to produce new ideas. Prior knowledge represents conceptual ideas, lessons and representations captured or collected when solving a problem Bailey et al., reported the results from interview with 14 design engineers and an online survey with 28 additional participants that showed that the reuse of prior knowledge is highly valued and an important step during early phases of design [5]. In early phases, people are still looking for directions rather than actual solutions. They provide inspiration to new and creative ideas. Prior knowledge can be well defined solutions to recurring problems, case based solutions, design decisions and reasons behind those decisions, design history for a particular item. The results showed that designers tended to access previous work or project to get inspiration. Results showed that designers stored artifacts like notes and sketches from their current project as they were sure that they will need them in their future projects; however designers mostly prefer to analyze other designer’s work more than their own past work to gain new ideas and inspiring solutions [5].

Examples (of prior knowledge) are helpful in enhancing the creativity of software engineers. Examples are any kind of material, product, prototype or digital artifact that contributes directly or indirectly to the design. Examples can come from the designer’s own work or external sources like other designer’s work, web, blogs, magazines etc. Herring et al. conducted an interview with designers to understand the use of examples in creative design. The results showed that examples helped designers in idea generation phase. Examples also helped the designers to understand the problem better and helped them compare similar products to see the novelty in their idea. During the evaluation phase, examples were used as references to see how their design has evolved from where they started [10]. Therefore, prior knowledge plays an important role in providing a starting point and inspiration for new ideas.

Analogies are another form of knowledge which helped designers to come up with creative and innovative solutions by analyzing the design solution from a different domain. Analogies help compare two similar entities, find similar feature and integrate them. This can lead to creative solutions. As pointed out by Johan Hoorn, people come up with creative solutions when they combine similar entities together, compare them and look closely to their intersecting features and integrate them. He explains creative process as entities being associated, selected, integrated and adapted until optimal solution is found [11].

Nathalie Bonnard reported the results from a series of experiments to show that creativity during the design phase can be improved with the help of analogies [7]. In one of the experiments, subjects were asked to design a chair for cyber cafe and were provided with sources to make analogies. Two analogies provided belonged to the same domain and were nomadic stool and rocking chair. The other two were from different domain and were canoe and logotype. The results showed that the group that used analogies was able to draw features from many more analogies than the unguided group.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main RQ’s</td>
<td>The research question(s) that the paper answers.</td>
</tr>
<tr>
<td>Motivating study</td>
<td>The previous study from which the study was inspired if any.</td>
</tr>
<tr>
<td>Type of study</td>
<td>The type of study (e.g., survey, experiment etc).</td>
</tr>
<tr>
<td>Findings</td>
<td>The main findings of the paper.</td>
</tr>
<tr>
<td>Validity threats</td>
<td>Threats in the findings or study techniques.</td>
</tr>
<tr>
<td>Why paper is interesting?</td>
<td>Highlights of the paper/most interesting finding or technique.</td>
</tr>
<tr>
<td>Knowledge management</td>
<td>The knowledge management or access technique discussed in the paper.</td>
</tr>
<tr>
<td>Improvement</td>
<td>Improvements for the current study if applicable.</td>
</tr>
<tr>
<td>Empirical evidence</td>
<td>The empirical evidence for the question(s).</td>
</tr>
<tr>
<td>Relation with creativity</td>
<td>Any information that indicates that knowledge improves or influences creativity.</td>
</tr>
</tbody>
</table>
who did not have sources to make analogies. The guided group came up with significantly more number of sources and more ideas to generate [7]. In another experiment, professional designers and students were asked to design a renewed torch. They were asked to find a new gesture to use the torch and move away from the traditional switch. Very minimal information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].

The engineer’s knowledge base consists mainly of scientific principles, formulae and rules. The scientific principles are linked to the real world, where creative products will be used. This linking or applying the principles to the real world is done by the engineer. If we want to enhance the creativity and creative output of an engineer, we should be able to build the capability of applying the scientific principles to the real problems. The information technology (IT) tools can gather information about the problem was given to the participants, but additional information or analogies were provided upon request. The results showed that professional designers asked more questions to understand the problem better and visualize the problem from various perspectives. Professional designers applied these various viewpoints into their problem solving at an early stage. Overall, these results showed that analogies help designers to extract more details about the problem in hand and in coming up with more ideas to solve the given problem [7].
concluded by performing qualitative evaluation and through assessment of creativity observed during all the project phases on ten completed research and development projects in Turkish’s scientific and technological council. The results from the case study showed that there is a strong correlation between knowledge sharing and the creativity in all project types. It is important for an organization to promote knowledge sharing between their employees to allow them be more creative [13]. Similarly, Edmonds et. al., reports the results from a retrospective study on bicycle design which showed that for a designer to be creative, he should have access to diverse and dynamic knowledge. Based on the results from the study, it was concluded that **dynamic knowledge** should be made available to the designer depending on the problem.

A system like KSS (Knowledge support system) has been proposed by several authors that can provide the knowledge or information to the designers during the design process. The KSS includes the design history in the form of predecessor designs, design rules and knowledge editors and provides access to all the information and expertise. This system was proposed after concluding that embedding knowledge into the tools can provide knowledge dynamically to the user [6].

Similar to the above study, Bales et. al. conducted an observational study to see how designers organize tangible information in their work space and how they utilize this information in their work. The results from the study showed that the **prior knowledge** and the **domain knowledge** are keys to creative and successful design. They also observed the designers individual and collaborative work spaces and showed that the tangible information is moved between these workspaces as design evolves and refines. Based on these results, the authors concluded that, for any tool to support creativity, it has to provide multiple workspaces and ability to move the information between these work spaces to share their ideas with others and get feedback to refine their solutions [15].

On similar lines Joshua et. al., conducted a study to see how software tools can help the designers in being creative. The tool used in the study provided personal and collaborative workspaces for sharing ideas and work. People were asked to work on individual ideas, and then move it to the collaboration space and work collaboratively on the idea. The results of the study showed that the tool helped design multiple solutions for the problem and also work on them in parallel. The tool also helped the teams share their ideas with others, get feedback and work in collaboration which led to creative solutions [16].

**Limitations**: Based on these results, there is ample evidence to show that the knowledge access and sharing methods help improve the creativity of software engineers, especially during the software design. However, there were few limitations with the knowledge access method that inhibits the creative power of software engineers. For example, when using the repositories, searching for prior knowledge is cumbersome due to plethora of information stored in them. It is challenging to finding the right information efficiently. Software engineers also expect to have a visual search to help them locate the right information easily. Knowledge users also felt that the information was stored in an inconsistent manner and wanted some kind of consistency so that they can easily make their way to the right information they are looking for [5]. Similarly, during the knowledge collaboration, the most difficult problem is to find the right people to collaborate with. It is reported that people are not always aware of whom to contact to get help when they are in problem or get feedback from. At times when people find the right people to collaborate with, others may be busy with their work and would not completely participate in collaboration or they might deny participating [9]. Therefore, it is important for an organization to create work environment for the engineers to be able to share their ideas and knowledge and collaborate in order to support creative software development.

C. **Question 3: Improving the creative thinking process**

Multiple studies suggest that the knowledge access methods need to be improved for supporting creativity [5, 4, 9, and 18]. Multiple studies reported knowledge stored in repositories as one of the main sources of knowledge used by software engineers to solve problems. While prior knowledge gained from repositories is very useful, it is not being used effectively as searching is very difficult and in few occasions important information from the data is missing from these repositories. The search mechanisms used in existing repositories are not very powerful thereby making searching for the right things extremely difficult. This is the reason why software engineers seldom use prior knowledge stored in repositories.

One way to improve this could be to make search more powerful with introducing visual search. In this kind of search, a person can visually see the data through which he is searching, the user can modify the search based on their needs, and the user can see how the data is organized so that he/she can search the data in an effective way. Once the search provides the right information in an efficient manner, software engineers will start using the repositories to enhance their creative thought process. Regarding the incompleteness of the information stored in the repositories, it is important to collect the story behind the knowledge being stored. For example when a design was chosen over alternate designs, storing the reason as to why it was chosen over others is a good approach. This would help the person looking at this knowledge better understand the original thought behind the decision and help relate it to his or her current problem [5].

Based on the results reported in IV.A, **analogies** are an important form of knowledge which triggered creativity among software engineers. Maiden et. al., [4] conducted a workshop to show that though using analogies during the requirements engineering generated more creative ideas, it had inherent limitations. The workshop targeted collecting requirements for a future air traffic management system for managing departures from major European airports. Twenty one professionals from the departure management and scheduling departments in the UK and France attended the workshop. One of the main lessons learned from this workshop was the realization that in order to support the use of analogies during requirement stage, people need to be trained on how to use analogies and to be able to extract useful information from analogies to transfer knowledge from one domain to generate new ideas and use it in the targeted domain. Therefore, usage of analogies should be supported by providing relevant trainings so that people know how to extract knowledge from analogies and generate creative
ideas. While this workshop was focused on the requirements phase, the results are applicable at other phases.

Based on the results reported in IV.B, sharing knowledge or collaboration is one of the common ways to promote creativity. Andrawina [18], reports the results from a survey that organizational factors like top management’s support to knowledge sharing and technology factors like computer network and electronic bulletin boards, discussion groups to collaborate and share knowledge influence positively and encourage people to share knowledge and apply them to be innovative. The survey recommended that organizations should have management who appreciates and rewards knowledge sharing behaviors among their employees and provide help required for knowledge collaboration. This would help organization and its people to be innovative and creative [18].

Additionally, the results reported that the two major knowledge access methods were repository and collaboration. The data in the repository does not contain dynamic contextual data whereas the collaboration gives access to dynamic and contextual knowledge. The best approach would be to integrate both types of access methods by providing access to both the repository knowledge and the contextual knowledge. Software tools that can combine both these approaches would act as single point of contact for a software developer to search for information when he/she wants feedback or help with his or her problem. Such tools would increase the knowledge access and in turn help software engineers be more creative [9].

V. DISCUSSION OF FINDINGS AND FUTURE WORK

The knowledge types reported in literature that impacts the creativity of software engineers were broadly classified as: prior knowledge, analogies, and dynamic knowledge. Prior knowledge provided useful information about previous projects and helped software engineers in the initial stages of idea generation. They provided a starting point or motivation to start with ideas to solve the problem. Analogies helped software engineers link two objects from same or different domain and create new ideas. Tools provided knowledge when required by the software engineer and help software engineers concentrate on the original problem. Knowledge is accessed using different ways. The most common knowledge access techniques found during the review was searching for prior knowledge in repositories, internet, wikis, books and magazines. Another technique was collaboration, where software engineers shared theirs ideas and thoughts with others. They build on each other’s ideas and provide constructive feedback. They covered each other’s knowledge gap as a group.

Knowledge is very useful for software engineers and it makes them more creative and but not all software engineers were utilizing the available knowledge. The reasons for this were that searching for the right information in large repositories was cumbersome and when the information was found relevant information but would be missing in the repositories. Searching huge database and finding the right information was difficult as the search techniques were not very powerful. Gaining new knowledge via collaboration was difficult too as finding the right experts to share the ideas and get inputs was difficult and when right people were found they were busy in their work and had minimal or no time to collaborate. To improve searching and make it easier the knowledge should be stored in a consistent way, searching techniques should be improved and powerful visual techniques should be added. We should try to combine repository knowledge with collaboration tools so that software engineers can find all the knowledge in one location. Software engineers should be trained on analogy making and utilizing analogies to extract useful information and come up with new ideas. Management of software industries should support and enhance knowledge sharing behavior among employees.

REFERENCES