JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Realtime Text Editor with Multiple Users

Abhishek Verma, Arjun Kanojia, Dr. Nidhi Saxena
Department of Computer Science & Engineering,
Shri Ramswaroop Memorial College of Engineering and Management,
Tiwari Ganj, Faizabad Road, Lucknow (UP).

Abstract- The Multiuser Live Text Editor is a platform where users can interact with each other and write code together. It has support for C++, Java, and Python. This application is based on the concept of changing functionality, which is the basis of the shared editor. Pair programming is an agile software development method originating from Extreme Programming (XP), where two developers work together on a single computer. The two work together to design, code, and test user stories. Best of all, these two have different systems, and everyone spends the same amount of time on the keyboard. One use of integration is to refer to the keyboard's programmer as the driver and the other as the navigator. The navigator focuses on the general direction of work. Collaboration between developers can take place in person or remotely. Pair programming is a collaborative method that involves a lot of communication. The idea is for drivers and travelers to communicate, share routes, and solve problems that would be difficult for a developer to investigate.

1. INTRODUCTION

2code can be thought of as two people and one machine. To be c lear, it's multiple people and a machine that allows multiple peo ple to come together and work together. In this application, two people have a keyboard and a mouse. One programmer should w rite the code and another programmer should look at the code. It checks whether the code is suitable for the requirement.

It also notes, analyzes and identifies errors in the number written and what to do next. The coder's personal responsibility is to wr ite the code properly and he doesn't care what he writes because other programmers have proven it. Roles can change at any time ; the driver becomes the supervisor and vice versa. Both make a couple and work well, saving time and making code easy to deb ugBest of all, they both have equal capabilities, and each uses the same clock on the keyboard. One implementation of pair programming refers to the programmer on the keyboard as the controller and the navigator as the navigator. Navigator focuses on general guidance on programming. Collaboration between developers can take place face-to-face or remotely. Pair programming is a collaborative process that requires a lot of communication. The idea is for the driver and navigator to communicate with each other, discuss routes, and solve problems that were difficult for the developer to diagnose alone. However, this agile software development process is not for everyone. To develop effective teamwork in closed and collaborative computing, learning requires skills that not all programmers have. It requires both programmers to have the soft skills needed to work together and the hard skills needed to write and test code. Some businesses may use this application, while others may choose not to. The process starts with the developer finding a job. They suggest small goals like counting, measuring, or just writing in hours. Any advice or correction discussion will be made after each location so as not to disturb the driver's journey.

2.PROBLEM STATEMENT

Pair programming involves two developers working on a single workstation. One of the developers acts as the driver, the other a s the navigator.

Drivers use workstations to record numbers, travelers view num bers in real time. Each time, two developers switch roles, so ever yone has the opportunity to take the direction of the project and t urn the solution into actual code.

In general, the task of the traveler is to determine the direction o f the policy and to suggest how it should be implemented. A pair programming project can consist of two developers with similar skills and experience, or it might bring together a senior developer who mentors a junior developer.while another navigation programmer guides the drive r as he writes code.Bitter programming may sound confusing, bu t the truth is it's a great way to bring development teams together to create better products.

3.LITERATURE REVIEW

Marina Pimenova. (2001) According to Marina Pimenova Seven out of the ten PP studies regarded paired skill level as one of the determinant factors of PP's effective-ness the two categories of skill level used were actual and perceived skill. The actual skill level was determined based on programming experience, academic background, and students' academic performance. Perceived skill level was measured subjectively according to the skill of a student's partner relative to their own perceived skill (i.e. "better", "about the same", or "weaker"). The consensus from these studies is that PP works best when the pair has a similar skill level. However, two correlation studies show contradictory findings on the association between students' skill level and PP's effectiveness. Muller and Patberg report there is no correlation between the two variables and Made ski refutes this finding.

Yu Kong. (2007) According to Yu Kong the two studies that investigated the effect of Felder-Silverman learning style reported that learning style did not significantly affect pair compatibility or the perception of students towards pairing. In terms of work ethic, Williams et al. report that pairing students of similar work ethic enhances pair compatibility, and Layman reports that students' perception to-wards pairing is not affected by their work ethic. Williams et al. also investigated students' time management ability and found it has no effect on pair compatibility. In 2004, Muller and Patberg coined the term "feel-good" which refers to how comfortable pairs feel during the PP session. They report that the feel-good factor is correlated with a pair's performance. Made ski [S68] had similar findings where a positive correlation between the feel-good factor and pair performance (quality of software) was found.

Deepak Kumar. (2019) According to Deepak Kumar PP's effectiveness was measured using various factors, organized in four categories: technical productivity, pro-gram/design quality, academic performance, and satisfaction. Technical productivity, measured by 31 (44%) of the 70 studies was the most common method used to assess PP's effectiveness, followed by program/design quality (30 studies, 43%). A subset of 16 studies (23%) evaluated PP's effectiveness based on students' academic performance in final exams, mid-terms, assignments, projects, and course grades. Besides the objective measurements, PP's effectiveness was evaluated subjectively in 22 studies (31%) using students' perceived satisfaction experiencing PP sessions.

Shruti Kothari. (2020) According to Shruti Kothari Pair Programming (PP) - all production code is written by two people at one screen/keyboard/mouse. Pair programming is a collaborative approach that makes working in pairs rather than working in individual for code development One programmer writes a software artifact (e.g. program code or UML diagrams) and other programmer continuously assures quality of the software artifact by watching, asking questions, looking for some alternative approaches, helps to avoid defects etc. The two programmers switch their roles after some time: creator, is also called Driver becomes quality assurer, is also called the Navigator, and vice versa

Dybå & Dingsøyr. (2008) In 2008, Dybå & Dingsøyr carried out a SLR of Agile Software Development empirical studies to find the empirical evidence for benefits, limitations, and strengths of agile methods. They found low strength of evidence supporting PP in agile methods. However, in 2007, Dybala et al. conducted a SLR focusing on effectiveness of PP. The study investigated the empirical evidence and supported the claims that PP is more advantageous than solo programming. The PP 's aspects investigated were related to effectiveness focusing ―duration‖ (time spent to produce the system), ―effort‖ (person hours spent), and ―quality of the final product‖. The SLR as an intermediate analysis was extended by Hanna et al as a full-scale analysis in 2009, which summarized pair programming experiments published up to and until 2006. In addition, the studies published up to August 2007 were also taken into account.

4. METHODOLOGY

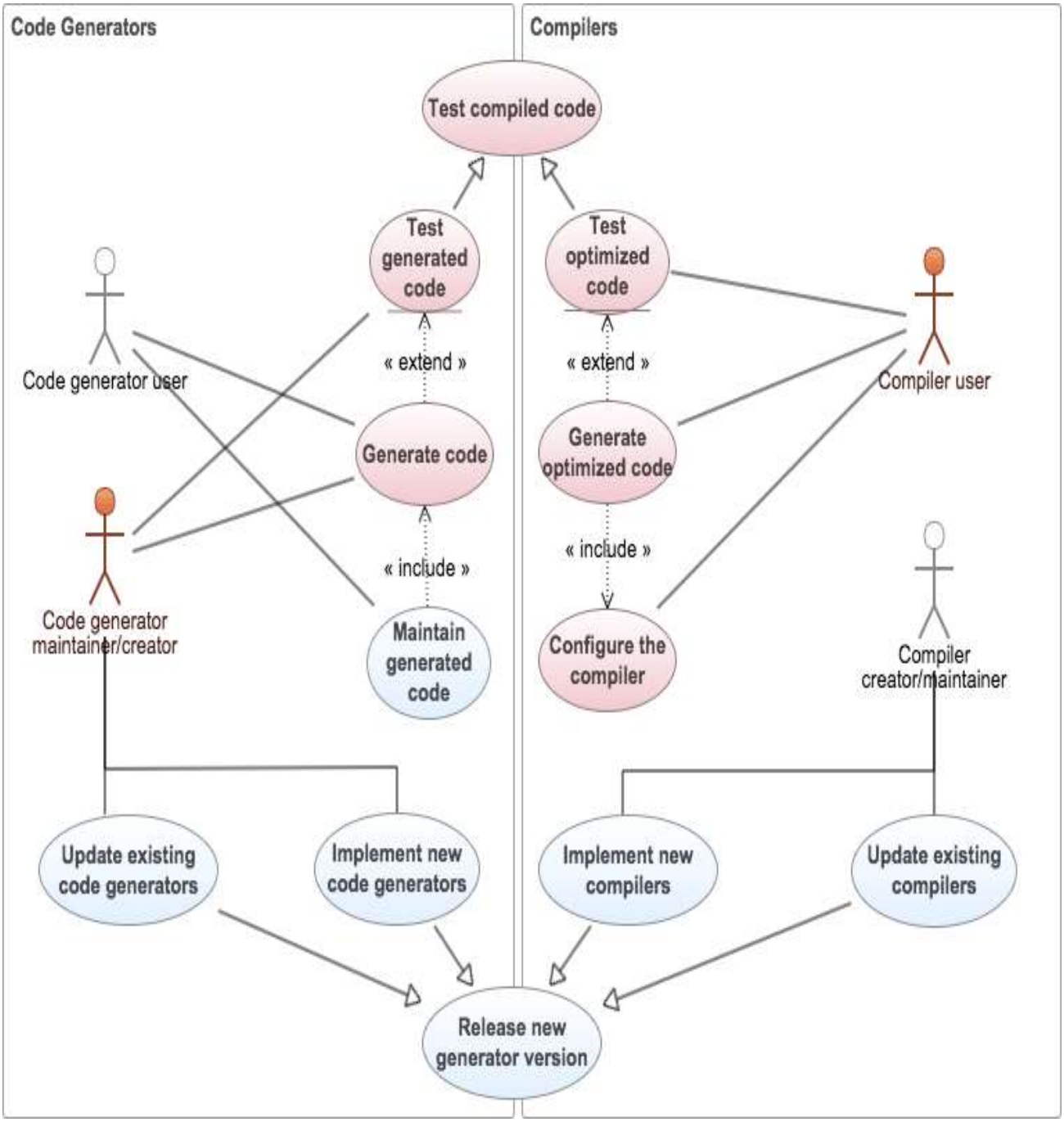
The proposed methodology in order to do Realtime coding with multiple users as follows –

- A GUI which shows the editor with options of different programming languages, and option of video call and invite too.
- After opening the site, the user will see the editor, and can start coding with multiple users, with the help of invite button.
- The user will share the room code with other users to join the same room.
- Other user will join the room by invite link, and can start coding in the same room.
- Users can also do the video call for better understanding and code sharing.
- In This way The Multiple users can collaborate and code efficiently.
- Users can also have multiple themes for text editor.
- It Can Be used for Teaching purpose, interviews, small-level

The main objective of our project is:

- Real Time Editing: Edit code in a real time editor. Input and output are also real time.
- Run code: Run code against a custom test case in currently supported languages.

• Private Channel: Only the users who have access to the unique URL can edit in the editor.



• Video Calling: Users can interact via video calling.

• Rich text presence: Highlight the cursor position of another user.

5.MODULE DESCRIPTION

1. ***VIDEO CAPTURE MODULE:*** Video capture module includes the detection of real time video frame captured using the user's web camera (live feed).

2. *CODE EDITOR MODULE*: Code editor Module will help in editing code. We are using ACE (Ajax.org Cloud9 Editor). Ace is a standalone code editor written in JavaScript. Our goal is to create a browser-based editor that matches and extends the features, usability and performance of existing native editors.

3. *REAL TIME COMMUNICATION MODULE*: For Realtime communication we are using socket.IO. Socket.IO allows bi-directional communication between client and server. Bidirectional communications are enabled when a client has Socket.IO in the browser, and a server has also integrated the Socket.IO package. While data can be sent in a number of forms, JSON is the simplest.

4. *COMPILER MODULE*: For Compiler we have use Compile Run library. It has different languages it provides different compiler facilities that are needed for coding.

***5. AUDIO MODULE*:** Audio capture module includes the detection of real time audio captured using the user's Microphone.

6.IMPLEMENTATION OF REALTIME COMMUNICATION

Socket.io - Client-Server Communication:

Socket.IO is a library that enables low-latency, bi-directional, and event-based
communication between client and server. It is built on top of WebSocket
protocol and provides additional guarantees such as reverting to HTTP long-polling or
automatic reconnection. It provides connection over TCP while Socket.io does
a library for abstracting WebSocket connections.

Peer.js – Communication Client – Client

Peer.JS wraps the browser's WebRTC implementation to provide a complete,
configurable and easy-to-use API for peer-to-peer connections. Equipped
ID only, a partner can create a P2P data or media stream connection to a
remote peer Why do I use peer JS? With PeerJS, we don't have to worry about that
STUNs, candidates for ICE or creating a server. We can even avoid implementation
WebSockets too. PeerJs provides a complete, configurable peer-to-peer
an API and a server named PeerServer for easy connection establishment

between PeerJS clients.

Video calls using Web RTC

WebRTC (Web Real-Time Communication) is a free and open-source project providing web browsers
and mobile applications with real-time communication (RTC) through an application programming interface
(API). It enables audio and video communication to work on websites by enabling direct peer-to-peer communication, eliminating the need to install plugins or download native applications.
WebRTC allows browsers to stream files directly between each other, reducing or eliminating them entirely
the need to host files on the server side. Web Torrent uses WebRTC transport to enable peer-to-peer file sharing
sharing using the Bit Torrent protocol in the browser. Some websites use it to allow users to send files

7.CONCLUSION

Two heads are higher than one. If the driving force encounters a hitch with the code, there might be two
heads higher than one. If the driving force encounters a hitch with the code, there might be two of them who'll clear up the hassle. greater efficient. commonplace thinking is that it slows down the mission of completion time due to the fact you are efficaciously placing two programmers to develop a single program, as a substitute for getting them
to work independently on distinct packages. however, research has proven that programmers running on equal software are only 15% slower than when those programmers work independently, in place of the presupposed 50% slowdown. Fewer coding mistakes. due to the fact, there is any other programmer searching over your work, it results in higher code. In fact, an earlier take look suggests that it affects 15% fewer bugs than code written with the aid of solo programmers. Plus, it lets the motive force stay focused on the code being written even as the opposite attends to outside subjects or interruptions.
An effective way to proportion expertise. Code Fellows talks approximately how it can assist programmers' research from their peers in this blog post. it'd allow programmers to get on the spot face-to-face instruction, that's a lot higher than online tutorials and quicker than searching out resources on the net. Plus, you can research matters higher from your associate, particularly in regions that can be surprising to you. builders also can pick out up exceptional practices and better techniques from greater superior programmers. it is able to additionally facilitate mentoring relationships among programmers. Develops your team of workers' interpersonal abilities. participating in a single venture enables your crew to respect the value of verbal exchange and teamwork.

8.REFRENCES

[i] Salomon, G., Distributed Cognitions: Psychological and educational considerations. Learning in doing: Social, cognitive, and computational perspectives, ed. R. Pea and J.S. Brown. 1993, Cambridge: Cambridge University Press.

[ii] Constantine, L.L., Constantine on Peopleware. Yourdon Press Computing Series, ed. E. Yourdon. 1995, Englewood Cliffs, NJ: Yourdon Press.

[iii] Beck, K., Extreme Programming Explained: Embrace Change. 2000, Reading, Massachusetts: Addison Wesley.

[iv] Williams, L., et al., Strengthening the Case for Pair Programming, in IEEE Software. submitted to IEEE Software. Online at http://www.cs.utah.edu/~lwilliam/Papers/ieeeSoftware. PDF

[v] Williams, L.A. and R.R. Kessler. The Collaborative Software Process. in International Conference on Software Engineering 2000. submitted for consideration. Limerick, Ireland. Online at http://www.cs.utah.edu/~lwilliam/Papers/ICSE.pdf

[vi] Nose, J.T., The Case for Collaborative Programming, in Communications of the ACM. 1998. p. 105-108. 7. Humphrey, W.S., A Discipline for Software Engineering. SEI Series in Software Engineering, ed. P. Freeman, Musa, John. 1995: Addison Wesley Longman, Inc. 8. Humphrey, W.S., Introduction to the Personal Software Process. 199

[vii] Addison-Wesley Flor, N.V. and E.L. Hutchins. Analysing Distributed Cognition in Software Teams: A Case Study of Team Programming During Perfective Software Maintenance. in Empirical Studies of Programmers: Fourth Workshop. 1991: Able Publishing Corporation.

[viii] Fagan, M.E., Advances in software inspections to reduce errors in program development. IBM Systems Journal, 1976. 15: p. 182-211.

[ix] Johnson, P.M., Reengineering Inspection: The Future of Formal Technical Review, in Communications of the ACM. 1998. p. 49-52.

[x] Lave, J. and E. Wenger, Situated Learning: Legitimate peripheral participation. 1991, New York, NY: Cambridge University Press.

[xi] Weinberg, G.M., The Psychology of Computer Programming Silver Anniversary Edition. 1998, New York: Dorset House Publishing. DeMarco, T. and T. Lister, Peopleware. 1977, New York: Dorset House Publishers.

[xii] Cockburn, A., Crystal "Clear": A human-powered software development methodology for small teams, Addison-Wesley, 2001, in preparation. Online at http://members.aol.com/humansandt/crystal/clear. 16. Highsmith, J., Adaptive Software Development, Dorset House, 1999.

[xiii] Cockburn, A., Characterizing People as Non-Linear, First-Order Components in Software Development, in International Conference on Software Engineering 2000. submitted for consideration. Limerick, Ireland. Online as Humans and Technology Technical Report, TR 99.05, http://members.aol.com/humansandt/ papers/ nonlinear/nonlinear.htm.