Design of Controlled and Customizable Platform for Frontend Developer's Repetitive Tasks

Saurav Gupta[1]

[1]*Department of Computer Science Engineering, Lovely Professional University, India*

---***---

Abstract – *In the world of the internet, the development of software has become very crucial. Also, different modern lifecycle, models, principles evolved with the rapid growth of the IT market. Every day on average around 252,000 new websites are created, which is the major reason for the IT boom. Everyone wants there's business on the internet. A whole advertisement-based revenue model is based upon the websites. You might have come across the product ad which you just browsed on your favorite e-commerce sites. It might look very helpful in some cases, but mostly it looks very disturbing. The nature of the ad depends on the context of the usability. In the developer's community, we only care about the subject line for which we opened up any particular site, not the ad that we unnecessarily tend to see. A developer is an IT expert who works on multiple things parallel based on the type of task which is assigned to him. If he/she came across some of the bugs on the need to look for the solution of the newly assigned task then. There is a high chance that the developer must visit various sites as per hit and trial basis on some of the reliable sources, such as Stackoverflow, Github, and official docs, etc. In this paper, We are going to propose the design and solution for all the repetitive sites that a frontend developer might need while making frontend development that can have full customizable and controlled abilities such that one can just focus on the task without the distraction of the ad by combining all to one place. It will be the one-stop-shop solution for the frontend developers.*

In addition to the custom widgets, we aim to combine learning sources such that if one wants to excel in the concepts and stay updated then it has that capability. We follow the whole user behavior using the 3rd party libraries such that we can track the most used widgets and non-usable widgets based on the continuous feedback cycle we can re-work this ecosystem with iterations of the widgets. People tend to forget the name of tools, which they used in past. But now it won't happen if you like some websites want to integrate and save for later used here you can do this as well.

Key Words: Customize and control platform, frontend tools, Software development. Software Engineering

1. INTRODUCTION

Customize and control platform, frontend tools, Software development 1. INTRODUCTION In this section, we have discussed the need for having such a tool where we can organize the sub-app more diligently. The tool is made for people who want control with customization over tools, not being controlled by the advertisements tools. This perfectly suits front-end developers along with the Front-end developer Roadmap.

1.1 Problems with too many tools

If you're used to shifting from one app to the other, you may not realize how it's dragging you down. generally, you remembered multiple usernames and passwords, for one. Automatic login has made that easier, but you'll need to remember the sensitive information eventually. The more tools you have, the extra you will need to note all that credential data somewhere. This can make development slow, which can affect the development quality. One can forget the last visited site where he found the solution. No manual history.

1.2 Benefits of Consolidating tools

It's easier to navigate for the solution in a single web app than to traverse and find into multiple sites. You can use, arrange as per your need personalize as per your need in a single platform. On top of that, you have control of the sub-apps. What you need is a tool to do everything in one place. At a glance, you can see if any of your apps have new activities. Consolidating your apps/widget would not only save time, but you'd also be less likely to miss something important because of its arrangement.

You don't need to remember too many usernames and passwords. Here only one username and password will solve the problem. No switching issues. With time you may tend to forget the name of the site you logged in to a few times back. But everything is present in a single platform. The platform will take care of your activities.

2. PROBLEM STATEMENT

Today world has done so much progress in terms of collaboration and integration. Nowadays developer needs a tool where integrations of various frequent, newly developed, popular tools should be there and one can pick and add it to its queue. Sharing and collaboration of personal settings should be possible. Such that one can share its dashboard with anyone for better collaboration. On top of that out of tools library, the developer can pick and arrange it to its dashboard with tracking capabilities.

These frontend tools are also incorporated with the frontend roadmap, where updated docs related to the learning community will be shared.

3. RELATED WORK

Frontend tools are created for Frontend developers. This design can help developers to save time. This framework is the end to end application. The tools which we are planning to add here are very much popular in the frontend dev community. Right from starting of the development to implementation Frontend developers may need the help of some tools on daily basis. The popular frontend tools which we are planning to add here are as follows.

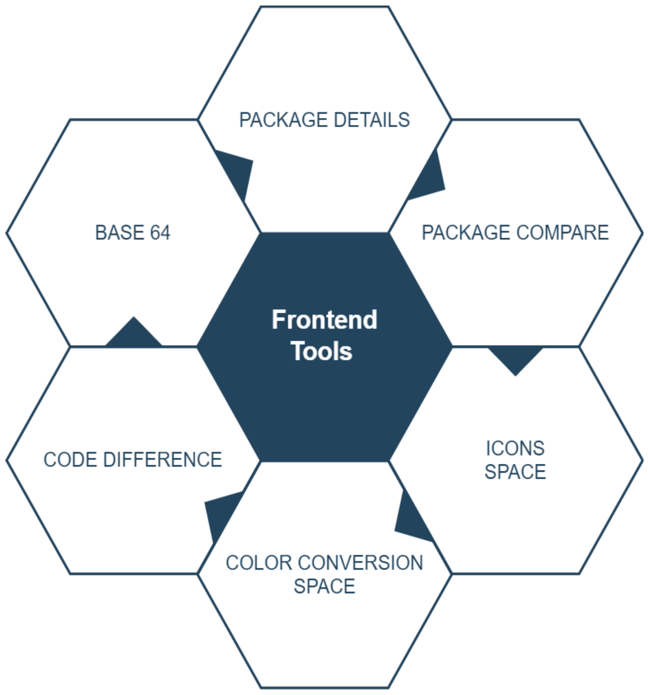


Figure 1. Single tool solution

3.1 Package Details

This tool is the most popular among the Javascript community. In June 2019, a total of around 1.3 million packages are registered. The npm registry is serving around 125 billion requests with 6 petabytes per month. So adding these tools will serve the purpose of finding the cost of adding an npm package to the bundle

3.2 Package Compare

This tool is very much helpful if you want to decide over similar behavioral packages under the npm registry database. Compare package download counts over time will help you the finding the most popular package over the selected time frame.

3.3 Icons Space

Icons tools will help the developers to select and find the right icons on the fly. We should add the free, high-quality, open-source icon library with at least 700+ icons. Include them any way you like SVGs or CSS.

3.4 Color Conversion Space

Color selection can be scary but if we add the Select Color and convert it into Other Color spaces such as Hex, RGB, HSL, etc then it will be going to very useful.

3.5 Code Diff

Code Diff is a tool to compare text differences between two text files. Enter the contents and check it on the fly.

3.6 Base64 Converter

Decode and Encode Base64 data or text with this online base64 decoder/encoder

3.7 CSS Font Preview

CSS font preview tool allows the developer to preview the font or change the settings of the font properties.

3.8 SVG Converter

Optimize the SVG and can be converted into JSX, TSX, React Native, CSS, and Base64 format. This will solve the formatting issues.

3.9 JS/TS Converter

In this tool just Type your JS/TS code here and see it with correct formatted form or in minified form. These tools can help us in the optimization of code as well.

3.10 CSS Converter

Type your CSS code here and see it with the correct formatted form or in minified form with optimized form.

3.11 Time Converter

Since timezone can not constant for all the globe, it depends on the locations. Tools responsible for the unification of different timezones can solve the very common issue.

If the developer’s full attention is on the tools without any ad structure then the production level surely increases. So this suggested design will not have any advertisement.

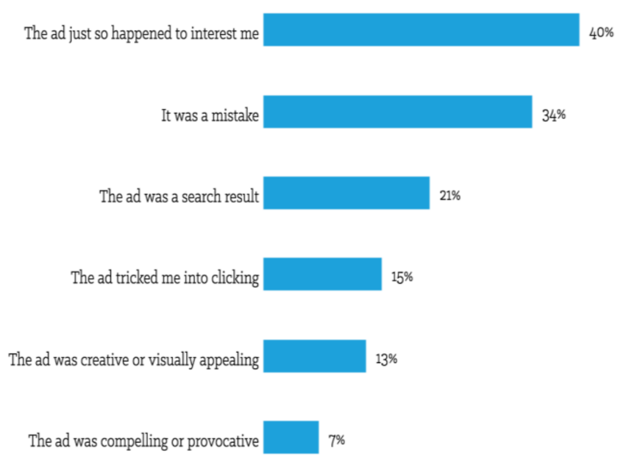


Figure 2. What caused a user to click on an advertisement

4. METHODOLOGY

In this section, we discussed the Software development life cycle model selection. Although in most of the cases of design of software will fall under the waterfall model. But since our projects are more towards the Adaptive nature than predictive. We left with other options such as V-Model, Sashimi, etc. Here a well-defined approach is followed. Requirement, Design, Implementation, Testing, Deployment, and maintenance are the core steps that need to be followed one after another in the waterfall model.

As you can see from figure 3. The next step is dependent on the previous step. To implement our design under waterfall we need to behave more predictive, which in really can't.

Since this is just the starting of a new project which is not so established so can't comment on its future status. Because everything needs to go through from feedback look in we may iterate through various points later on.

So in that case waterfall doesn't fit in our case. We need to look for something where we can model fast, should be adaptive, and re-iterate will not be a pain.

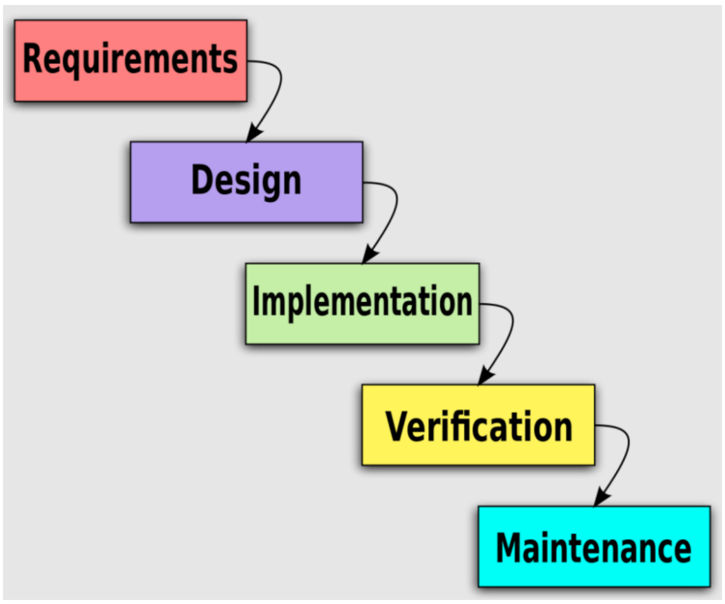


Figure 3. Waterfall Lifecycle model

If you look at the Waterfall method, it logically makes sense. You define what you want to build, and then you design how you're going to build it, then you build it, then you validate it, and then you deploy, and users use it. So what were some of the challenges that the software industry was facing that led to the emergence?

- In the verification phase where you put all the components together and try to find out whether the system is working as expected

- the software is developed and deployed, the clients were using the system and they were saying this is not what they were expecting, or this is not what they need, this doesn't work for them

This led to another Software development mindset known as Agile frameworks which include subtypes such as Scrum, Kanban, and Lean Startup.

To implement this we followed the Lean startup model because of its incremental, adaptive, and iterative nature. Here we can Build, Measure through feedbacks, learn what exactly the user is looking for and make parallel changes in our design. It is very flexible. Here we can learn faster and build the right product with that we can speed the market. Because if we deliver at the correct time then it makes more sense then to deliver aftermarket need changes. Lean startup models us useful in case of doubtful market cases and where we have a lot of high probability risks.

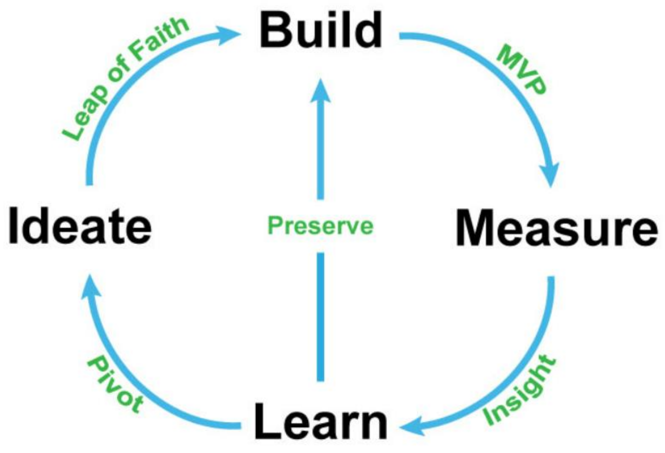


Figure 4. Lean Startup model

5. TOOLS AND TECHNOLOGY

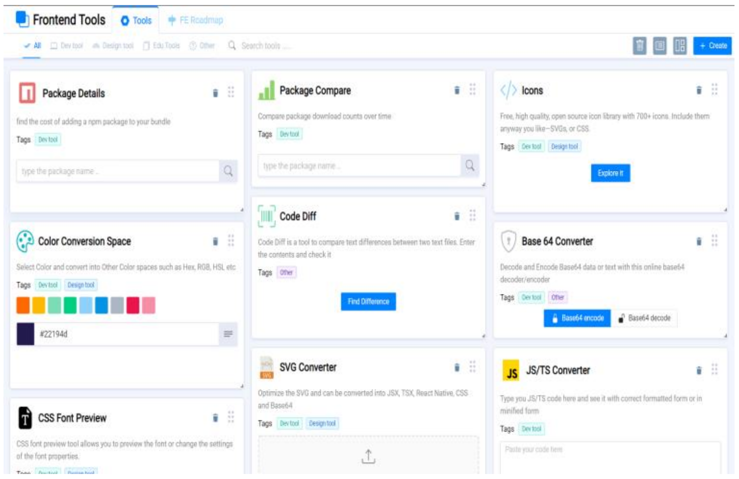
Visual Studio Code by Microsoft was used to develop the prototype of its implementation. We divided the implementation into the backend and frontend codebases.

Backend is written in NodeJS environment using javascript scripting language where we used MongoDB NoSQL database with addition to various open APIs wrappers. The frontend is written using the most famous javascript framework React with hooks. Frontend included HTML(Hypertext Markup Language), CSS(Cascading style sheets), and Javascript. As we used React because of its lightweight and component followed structures where routing using ReactRouter, store using Redux are the key functions that we can add through external npm imports to the projects.

We used the most matured CSS version knows as SASS(syntactically awesome style sheets), which is a stable, and powerful professional-grade CSS extension in the world.

5.1 Developer Dashboard

This is the personalized view of the dashboard where we have the option to changes the views out of the following type a. Tabular View, b. Mailbox View, and c. Masonry View. Since it is fully controlled so one can drag and drop and arrange as per his choice. There will not be any ads.



Package Details
find the cost of adding a npm package to your bundle
Tags Dev tool

Package Compare
Compare package download counts over time
Tags Dev tool

Icons
Free, high quality, open source icon library with 700+ icons. Include them anyway you like—SVGs, or CSS.
Tags Dev tool Design tool

Color Conversion Space
Select Color and convert into Other Color spaces such as Hex, RGB, HSL etc
Tags Dev tool Design tool
#22194d

Code Diff
Code Diff is a tool to compare text differences between two text files. Enter the contents and check it
Tags Other

Base 64 Converter
Decode and Encode Base64 data or text with this online base64 decoder/encoder
Tags Dev tool Other

SVG Converter
Optimize the SVG and can be converted into JSX, TSX, React Native, CSS and Base64
Tags Dev tool Design tool

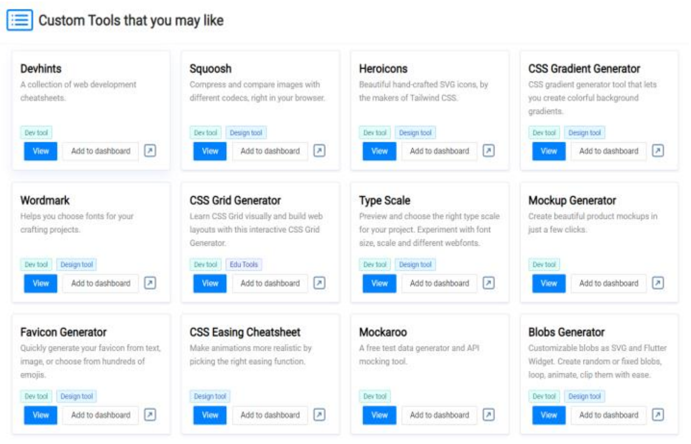
JS/TS Converter
Type you JS/TS code here and see it with correct formatted form or in minified form
Tags Dev tool

CSS Font Preview
CSS font preview tool allows you to preview the font or change the settings of the font properties.

Figure 5. View of personal dashboard

5.2 Custom Tools Library

This section contains the list of tools library which developer can pick and add it to the dashboard. This is like a control board for the dashboard of your choice.



Custom Tools that you may like

Devhints
A collection of web development cheatsheets.
Dev tool

Squoosh
Compress and compare images with different codecs, right in your browser.
Dev tool Design tool

Heroicons
Beautiful hand-crafted SVG icons, by the makers of Tailwind CSS.
Dev tool Design tool

CSS Gradient Generator
CSS gradient generator tool that lets you create colorful background gradients.
Dev tool Design tool

Wordmark
Helps you choose fonts for your crafting projects.
Dev tool Design tool

CSS Grid Generator
Learn CSS Grid visually and build web layouts with this interactive CSS Grid Generator.
Dev tool Edu Tools

Type Scale
Preview and choose the right type scale for your project. Experiment with font size, scale and different webfonts.
Dev tool Design tool

Mockup Generator
Create beautiful product mockups in just a few clicks.
Dev tool

Favicon Generator
Quickly generate your favicon from text, image, or choose from hundreds of emojis.
Dev tool Design tool

CSS Easing Cheatsheet
Make animations more realistic by picking the right easing function.
Design tool

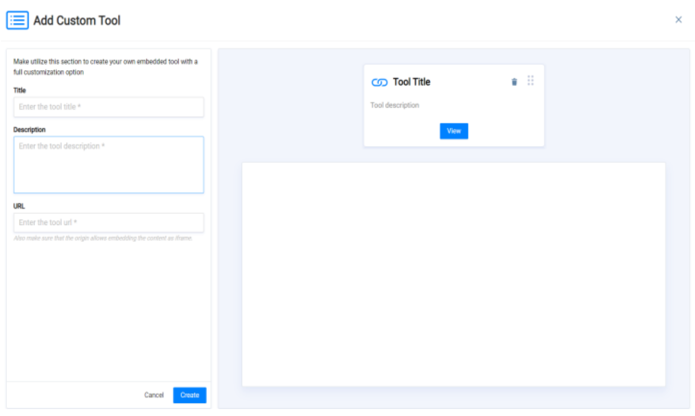
Mockaroo
A free test data generator and API mocking tool.
Dev tool

Blobs Generator
Customizable blobs as SVG and Flutter Widget. Create random or fixed blobs, loop, animate, clip them with ease.
Dev tool Design tool

Figure 6. View of Tools library

5.3 Create Personal Tool

If some tool is not present in the library still can be added to the dashboard from any source. Just enter the URL, title and it is done. We can suggest similar libraries to other developers if they are interested in similar things. We take feedback and iterate the design again. Here Lean Startup model comes in handy.



Make utilize this section to create your own embedded tool with a full customization option

Title: Enter the tool title *

Description: Enter the tool description *

URL: Enter the tool url *

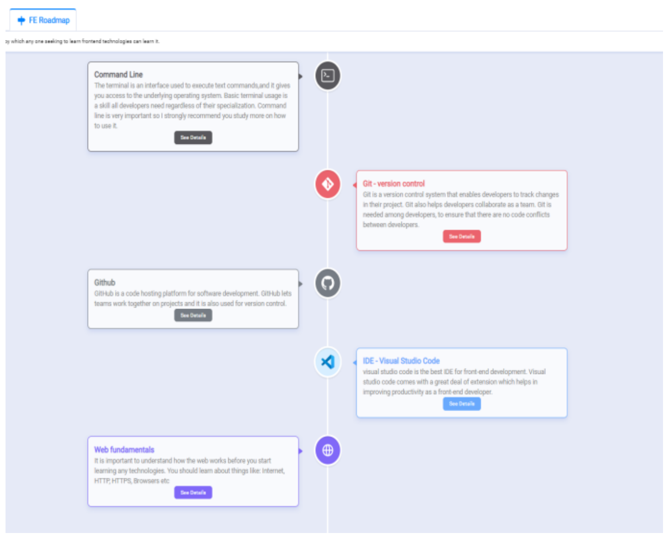
Also make sure that the origin allows embedding the content as iframe.

Tool description

Figure 7. Form to create a tool

5.4 Frontend Roadmap

Learning is the key to stay updated in the developer's community so we add this component as well where a developer can select the area of interest and learn the concepts related to his field through FE Roadmap.



y which any one seeking to learn frontend technologies can learn it.

Command Line
The terminal is an interface used to execute text commands,and it gives you access to the underlying operating system. Basic terminal usage is a skill all developers need regardless of their specialization. Command line is very important so I strongly recommend you study more on how to use it.

Git - version control
Git is a version control system that enables developers to track changes in their project. Git also helps developers collaborate as a team. Git is needed among developers, to ensure that there are no code conflicts between developers.

Github
GitHub is a code hosting platform for software development. GitHub lets teams work together on projects and it is also used for version control.

IDE - Visual Studio Code
visual studio code is the best IDE for front-end development. Visual studio code comes with a great deal of extension which helps in improving productivity as a front-end developer.

Web fundamentals
It is important to understand how the web works before you start learning any technologies. You should learn about things like: Internet, HTTP, HTTPS, Browsers etc

Figure 8. View of Frontend learning Roadmap

6. RESULTS

Total 100 Frontend developers were asked to use this tool during their normal working hours instead of traveling through other tools. I have taken feedback in form of the multi-select options. then, I found these interesting insights in figure 9.

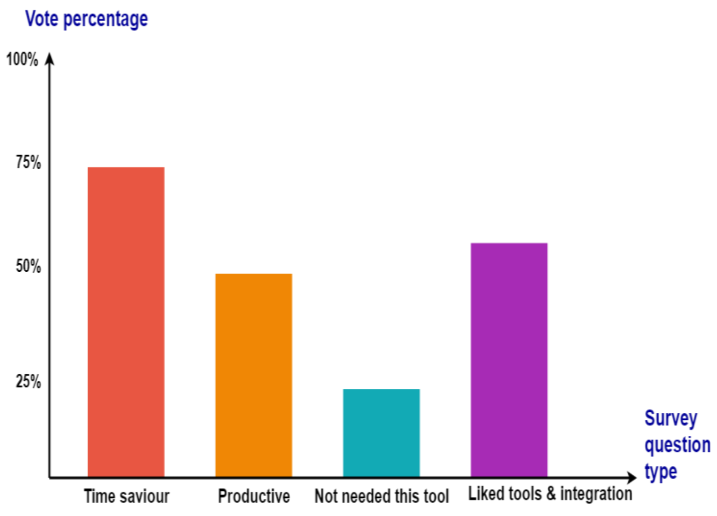


Figure 9. Analysis of survey with frontend developers

Hence our result showed that our developed frontend tool system is good as the one-stop-shop for Frontend developers as 75% of them think of this tool as a Time savior, and 50% found that it increased the productive there's production level, and 60% likes the tools and its way of integration.

7. CONCLUSION

In this rapid world of software, we need to focus on collaboration and integration things, Here our solution fits well in terms of perfect time saviors and does increase productivity, by combining all the tools into one. With a single Sign-in page a developer can see everything without the need of seeing any unwanted advertisement. Just do want you to want to do without any distractions. But yes we have to be very sure while choosing this kind of design. Since this is very flexible, we need to update our design with the change of nature of the market, that is why we choose the Lean startup model over others. This design perfectly suits Frontend developers. With the tracking system of each module, we can better serve the developers with the most frequently used module to the topmost level.

8. FUTURE WORK

In these tools we have focused on the tools integration part, in the future, we can also think more collaboration part. We can add more such tools in the library base on the next iteration of the feedback from developers. We can also how can we add the designers to this ecosystem such that Designer and frontend developers can collaborate here without any hassle without different systems. We can add discussion modules as well such that sharing of an idea can be made more easily. We can also add a Front-end interview roadmap. Such that a common interview prep ecosystem can also be integrated and used among all.

REFERENCES

[1] A. R. Gilal, J. Jaafar, M. Omar, S. Basri, and I. Din, Balancing the Personality of Programmer: Software Development Team Composition,|| Malaysian J. Comput. Sci., vol. 29, no. 2, pp. 145–155, 2016.

[2] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer, Genprog: A generic method for automatic software repair,|| Ieee Trans. Softw. Eng., vol. 38, no. 1, pp. 54–72, 2012.

[3] Mary Shaw. Writing Good Software Engineering Research Papers, International Conference on Software Engineering, IEEE Computer Society, pp. 726-736, 2003

[4] Marvin V. Zelkowitz and Delores Wallace. Experimental validation in software engineering. Information and Software Technology, Vol 39, no 11, 1997, pp. 735-744.

[5] Samuel Redwine, et al. DoD Related Software Technology Requirements, Practices, and Prospects for the Future. IDA Paper P-1788, June 1984.

[6] Zhi Wang, Bing Li, Yutao Ma, An Analysis of Research in Software Engineering: Assessment and Trends. arxiv.org, vol. 1407, no. 4903

[7] Wong W E, Tse T H, Glass R L, et al. An assessment of systems and software engineering scholars and institutions (2002–2006). Journal of Systems and Software, 2009, 82(8): 1370-1373.

[8] Wohlin C. An analysis of the most cited articles in software engineering journals-2000. Information and Software Technology, 2007, 49(1): 2-11.

[9] Hoonlor A, Szymanski B K, Zaki M J. Trends in computer science research. Communications of the ACM, 2013, 56(10): 74-83.

[10] Glass R L. An assessment of systems and software engineering scholars and institutions. Journal of Systems and Software, 1998, 43(1): 59-64.

[11] Chuang S-W, Luor T, Lu H-P. Assessment of institutions, scholars, and contributions on agile software development. Journal of Systems and Software, 2014, 93(7): 84-101.

[12] Wong W E, Tse T H, Glass R L, et al. An assessment of systems and software engineering scholars and institutions. Journal of Systems and Software, 2011, 84(1): 162-168

[13] Man Yong. The Design and Implementation of Online Examing System. Hunan University, 2012.

[14] Chuang S-W, Luor T, Lu H-P. Assessment of institutions, scholars, and contributions on agile software development. Journal of Systems and Software, 2014, 93(7): 84-101.