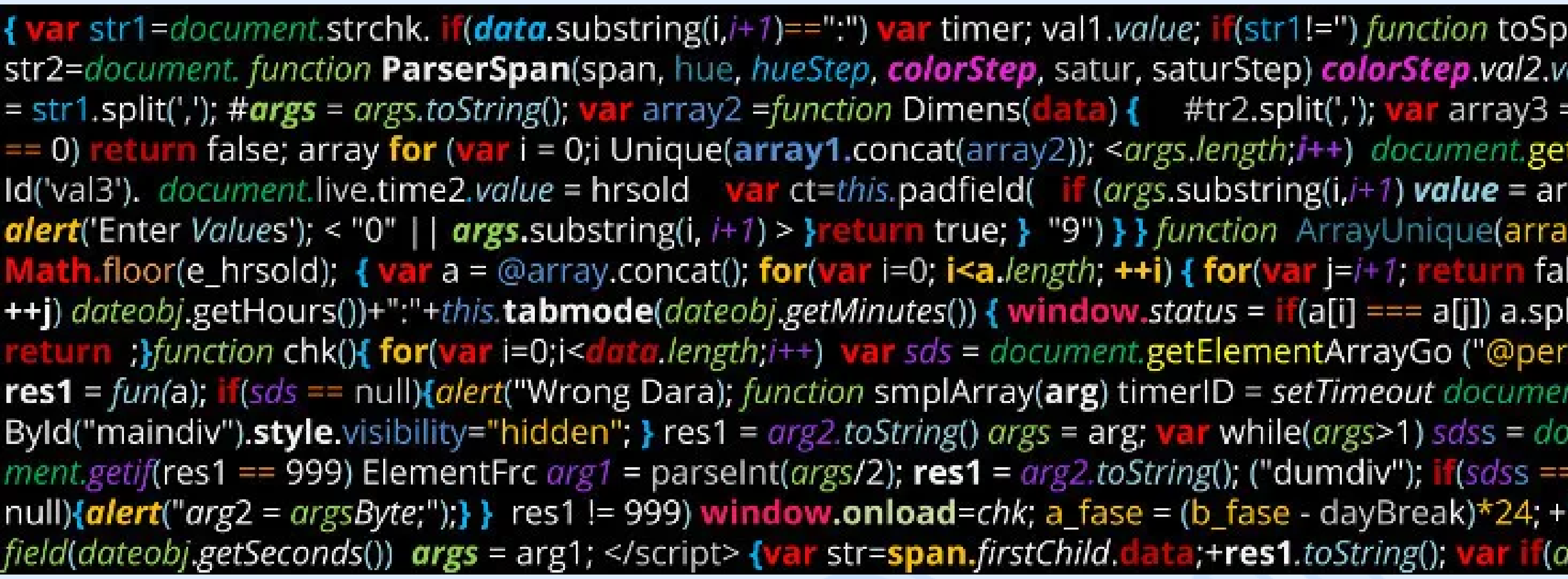

```
{ var str1=document.strchk. if(data.substring(i,i+1)==":") var timer; val1.value; if(str1!=") function toSp
str2=document. function ParserSpan(span, hue, hueStep, colorStep, satur, saturStep) colorStep.val2.v
= str1.split(','); #args = args.toString(); var array2 =function Dimens(data) {    #tr2.split(','); var array3 =
== 0) return false; array for (var i = 0;i Unique(array1.concat(array2)); <args.length;i++)  document.ge
Id('val3').  document.live.time2.value = hrsold    var ct=this.padfield(   if (args.substring(i,i+1) value = ar
alert('Enter Values'); < "0" || args.substring(i, i+1) > }return true; }  "9") } } function  ArrayUnique(arra
Math.floor(e_hrsold);  { var a = @array.concat(); for(var i=0; i<a.length; ++i) { for(var j=i+1; return fa
++j) dateobj.getHours())+":"+this.tabmode(dateobj.getMinutes()) { window.status = if(a[i] === a[j]) a.sp
return  ;}function chk(){ for(var i=0;i<data.length;i++)  var sds = document.getElementArrayGo ("@per
res1 = fun(a); if(sds == null){alert("Wrong Dara); function smplArray(arg) timerID = setTimeout documen
ById("maindiv").style.visibility="hidden"; } res1 = arg2.toString() args = arg; var while(args>1) sdss = do
ment.getif(res1 == 999) ElementFrc arg1 = parseInt(args/2); res1 = arg2.toString(); ("dumdiv"); if(sdss ==
null){alert("arg2 = argsByte;");} }  res1 != 999) window.onload=chk; a_fase = (b_fase - dayBreak)*24; +
field(dateobj.getSeconds())  args = arg1; </script> {var str=span.firstChild.data;+res1.toString(); var if(a
```

Despite many different coding conventions, there is no uniform standard that affords a clear and unambiguous evaluation of code quality.

When evaluating code, you may notice regional differences in standards. For example, coding standards often differ from country to country, with US standards closely reflecting that in Russia and differing from coding syntax used in Europe.

Major problems might appear at the bespoke development stage, using Framework, as well as when programming custom elements like forms, templates, and interaction specifications.

As you can see, there are various approaches to code-quality evaluation, and which method you choose depends on each specific case.

Code Review and Quality Evaluation Process

There are certain checkers that help developers validate the code; for example:

- http://phpcodechecker.com/
- Use Markup Validation Service to validate HTML
- Use CSS Validation Service to validate CSS
- Web Developer extensions (Firefox, Chrome, and Opera) offer some handy features including quick links to validators

Developers also use various services for automated code review with popular lint utilities:

- C++ / http://www.cplusplus.com/doc/tutorial/
- Java / checkstyle
- Scala / scalastyle

- JavaScript / jslint
- Coffeescript / coffeelint
- Perl / Perl:Critic
- CSS / CSS LINT
- PHP / PHPLint
- Python / pep8

This review is done as early as the coding stage and is followed by many tests for bugs and compliance with the functional requirements specified in the SRS (software requirement specification).

However, if you are not a professional developer, these checkers will not help you evaluate the quality of the written code.

Are there any tools that could be useful for code evaluation?

Unfortunately, there are no such tools, as there is no unified coding standard. The standards of different countries, and even of different development teams from the same city, can deviate from general standards. It is not critical and does not affect software performance, but can slow down the maintenance of your program if the code delivered by third-party developers deviates from the coding syntax to which your company is accustomed.

For example, you are accustomed to a particular style of recording variables, you separate nesting levels by spacing in the lines of code, or your team has a pool of most commonly used Frameworks and integration methods, which they would like to keep on using.

Another factor that points to the developer’s proficiency is the selection of modern frameworks and libraries they use. Such frameworks and libraries as Symphony & Laravel for PHP, Angular.js & React.js for JavaScript, AFNetworking, JSONModel and MagicalRecord for iOS guarantee you a great system performance, high functionality and a broad spectrum of opportunities to use APIs supplied by the frameworks. Using such high-profile and time-proven services, you can speed up the development process and reduce costs while improving the app's quality.

If you intend to maintain software in-house, it is easier to decide on all nuances, including the coding standards, preferable frameworks, programming languages and libraries of your choice, at the stage of development.

Our programming expertise and a large development ensure that our software meets all your demands and syntactic requirements.

The primary measure of software quality is its flawless performance. To help you competently evaluate the quality of the code, we are ready to show you the examples of the projects developed by our team.

Software Quality Evaluation

ISO standards (ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models) determine software quality using the following factors:

- **Functionality**
  - Operability
  - Compliance with standards
  - Functional compatibility
  - Security
  - Precision
- **Reliability**
  - Completeness
  - Recoverability
  - Fault tolerance
- **Usability**
  - Ease of training
  - Intuitive
  - User-friendliness
- **Efficiency**
  - Efficient use of time
  - Effective usage of resources
- **Maintenance**
  - Stability
  - Analysability
  - Controllability
  - Changeability
- **Portability**
  - Ease of installation
  - Substitutability
  - Compatibility

When evaluating the app's quality, pay attention to the following factors:

- Compliance with the SRS
- No bugs
- Stable performance under peak loads
- Invulnerability
- Fast performance

There are two major approaches that can help you evaluate the quality of the software, developed for you.

Ensure that these factors are met:

- You can check the software with running tests (whether in-house tests or with the help of third-party testers). If the app works as intended, its performance is stable, and there are no critical errors, you can be sure that its quality is excellent.
- The second approach involves a close examination of the code. It is only necessary when a company faces the need to support and maintain the app developed by third-party developers in-house. We are ready to provide you with the source code of the app, which can be easily evaluated by your in-house professional developers.

Last but not least, keep in mind that the most important quality estimation of any software application is its degree of compliance with the demands, fixed in the software specification and its stable, bug-free performance.