

---

# Code Review Process

## Table of Contents

1. Agreement of goals and requirements .....	1
2. Product functionality investigation .....	1
3. Automatic tools implementation .....	1
4. Code Review .....	2
5. Code Fixing Verification .....	2
6. Code Review Iterations .....	2

## 1. Agreement of goals and requirements

At first, we need to know, what is the goal of the code review:

- Do you need to prepare the product to the certification?
- Do you need to verify the product quality?
- Is the external code review a part of you quality assurance process?

In the first case we need to investigate the certification requirements.

In other cases we need to familiarize with your enterprise coding standard and code review checklists (if exist). Moreover, if the design documents review is also desirable we need to obtain design document template(s).

If you have no coding standards and/or templates, but wish to invent practice of their using, we help you to create standards and templates suitable for your company. In this case the review itself is postponed to let your engineers update documents and source code to comply the standard.

If you have no code review checklist, we create it basing on your requirements. The checklist may be re-used for future external and local code reviews.

Finally we need to negotiate format of code and document review reports. We can use your template or provide ours: OKTL-000022. We can also submit description of defects to your bug tracker.

## 2. Product functionality investigation

Before the review we need to understand what the source code is intended for. We need to understand product requirements, algorithms, read design documents (even if do not review them). Review of code related to areas which we are experienced in (networking, telecommunication, Linux) requires less time for this stage, but we can review code from other areas as well.

## 3. Automatic tools implementation

Obviously some coding standard violations may be found automatically. We can implement tools for automatic "pre-review". One of existing free tools is gcc, which produce a lot of very useful warnings.

We can implement such tools and provide provide them with source code and documentation - you can change and enhance them without our help.

One of use cases of these tools - applying them to all files committed to repository. We can help your system administrator to configure CVS or Subversion to achieve such functionality.

## 4. Code Review

Before the review we need to fix the versions of documents and source code to be reviewed. It's recommend to tag them in your repository.

If amount of code/documents is too big, the review may be splitted to several stages:

- tag first part;
- review first part;
- tag next part;
- review next part, etc.

to avoid reviewing of too old versions.

Finally we create one or several review reports.

## 5. Code Fixing Verification

As an optional service we can perform the verification after documents/code fixing by your engineers.

Result of this procedure - updated review reports where notes about defects are marked as "checked" and "partially fixed" and "not fixed".

## 6. Code Review Iterations

Sometimes it's useful to perform several iterations of the code review. For example, first iteration may check code standard conformance while the second one may verify design conformance, look for bugs and optimization opportunities, etc.