



## **White paper**

**Improving embedded software quality  
using source code reviews**

### COPYRIGHT NOTICE

© Copyright 2016 Atollic AB. All rights reserved. No part of this document may be reproduced or distributed without the prior written consent of Atollic AB.

### TRADEMARK

**Atollic**, **Atollic TrueSTUDIO** and the Atollic logotype are trademarks or registered trademarks owned by Atollic. ECLIPSE™ is a registered trademark of the Eclipse foundation.

### DISCLAIMER

The information in this document is subject to change without notice and does not represent a commitment of Atollic AB. The information contained in this document is assumed to be accurate, but Atollic assumes no responsibility for any errors or omissions. In no event shall Atollic AB, its employees, its contractors, or the authors of this document be liable for any type of damage, losses, costs, charges, claims, demands, claim for lost profits, fees, or expenses of any nature or kind.

### DOCUMENT IDENTIFICATION

ASW-WPCR                      July 2011

### REVISION

First version                      July 2011  
Second version                      August 2016

#### **Atollic AB**

Science Park  
Gjuterigatan 7  
SE- 553 18 Jönköping  
Sweden

+46 (0) 36 19 60 50

**E-mail:** [sales@atollic.com](mailto:sales@atollic.com)

**Web:** [www.atollic.com](http://www.atollic.com)

#### **Atollic Inc**

241 Boston Post Road West (1st Floor),  
Marlborough,  
Massachusetts 01752  
USA

+1 (973) 784 0047 (Voice)  
+1 (877) 218 9117 (Toll Free)  
+1 (973) 794 0075 (Fax)

**E-mail:** [sales.usa@atollic.com](mailto:sales.usa@atollic.com)

**Web:** [www.atollic.com](http://www.atollic.com)

# Contents

Abstract .....	1
Introduction.....	2
Finding errors earlier is cheaper.....	3
Source code reviews.....	4
Preparing for a source code review .....	4
Performing the source code review.....	7
Summary.....	9

# Tables

**No table of figures entries found.**

## ABSTRACT

A major problem in the embedded systems industry of today is inferior software quality, which is much more costly and difficult to address in embedded systems that has been deployed in the field, compared to PC or mobile applications that typically can download new releases automatically from the internet.

Several approaches are available to improve software quality. One of the cheapest and most effective is to perform source code reviews, where developers study each other's source code and point out potential problems already at the development stage.

By deploying methods and tools outlined in this white paper, you and your team can delivery higher quality software with less effort.

## INTRODUCTION

Most software developers are aware of the fact that almost all software products are shipping with at least some bugs. If this is not the case in your product, you are probably shipping a product with a very small amount of software, or you are most likely enforcing various methods for software quality management, including code reviews, static source code analysis, test quality measurement using advanced code coverage analysis, test automation, etc.

As embedded systems tend to contain a lot more software now compared to only a couple of years ago, the problem of shipping high-quality software is increasing. Many strategies exist to improve embedded systems software quality, including:

- Source code reviews (peer review)
- Complying to formal coding standards that promote “best-practice” coding
- Keeping comment level and code complexity at sound levels using code metrics
- Manual testing
- Automated unit testing
- Measuring test quality using advanced code coverage analysis
- Etc.

This white paper will discuss methods and tools for source code review (peer review), which is one of the most cost-effective ways of improving software quality. Performing source code reviews enable embedded software teams to deliver higher-quality software with a very limited effort.

## FINDING ERRORS EARLIER IS CHEAPER

Finding the cause of errors using a debugger is often necessary, but to fix a bug, it must first be detected. It is far cheaper to find the bug before the test phase is started, not to mention before the product is delivered to customers. Development teams should thus strive to find and correct bugs as early as possible in the development cycle.



Figure 1 - It is cheaper to correct software problems early in the development process

It is clearly a good strategy for development teams to spend more efforts on software quality earlier in the development process, as any problem that is found and fixed during the development phase is cheaper than finding and fixing it during the testing or maintenance phase.

A very good way of finding potential problems early in the development process is to perform source code reviews, where colleagues study each other's source code and point out potential problems before the product goes into the test phase.

## SOURCE CODE REVIEWS

A source code review (peer review) is an activity where colleagues study each other's source code, and try to find potential problems, which can then be fixed at an early stage in the development process.

In its simplest form, a team member could study the source code files from a colleague, write down any potential problems that have been detected on a piece of paper, and give it back to the developer for assessment and possibly correction (some review comments may turn out to be invalid, or may be valid but cannot not be fixed at this time for other reasons).

It is common to let several developers take part in a code review session. In such a case, developers typically analyze the source code files that have been appointed to them in their own time. When all developers have reviewed the source code files that were appointed to them, a code review meeting can be held, where all developers discuss all review comments that has been made. The review meeting might decide that some problems found during the review must be corrected.

A source code review is thus typically performed in three stages:

- The Individual (review) phase where the reviewers study the source code and make review comments.
- The team (review meeting) phase where reviewers discuss what to do with review comments, and possible assign review comments for fixing to specific team members.
- The rework (problem fixing) phase where select team members resolve the problems that has been appointed to them.

Up-until-now, it has not been common to have any tool support for source code reviews in embedded systems development tools. **Atollic TrueSTUDIO Pro** is one of the first embedded systems development tools to integrate features for source code reviews and to run code review meetings. This enables embedded developers to perform this useful technique with a minimum of effort.

---

## PREPARING FOR A SOURCE CODE REVIEW

**Atollic TrueSTUDIO Pro** allows a team member (such as the team leader) to create a source code review session at any time. When a review session is created, a name and description of the review have to be defined, as there might be several reviews during the project's lifecycle.

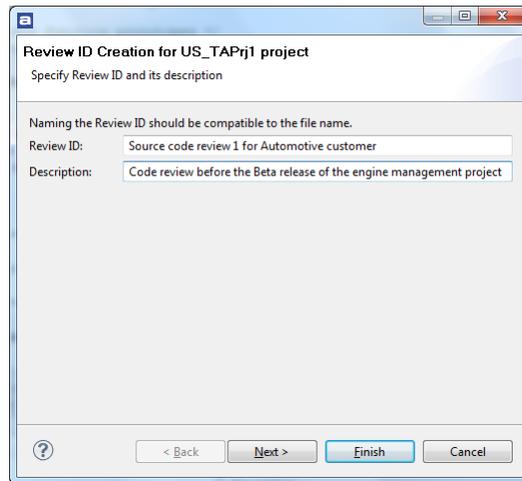


Figure 2 - Creating a source code review session

The files in the **Atollic TrueSTUDIO** project that will undergo review must then be selected:

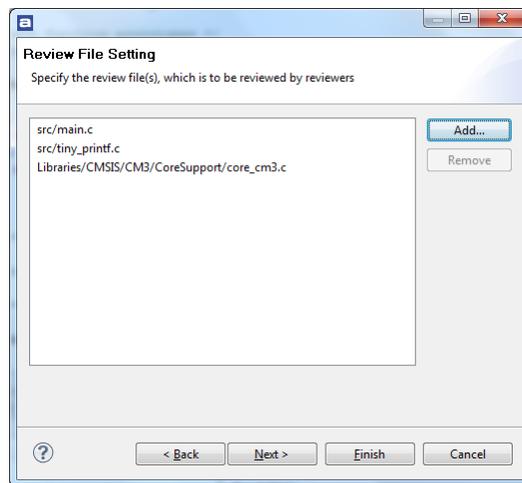


Figure 3 - Selecting files to undergo the source code inspection

The name of the reviewers who will take part in the review is selected too:

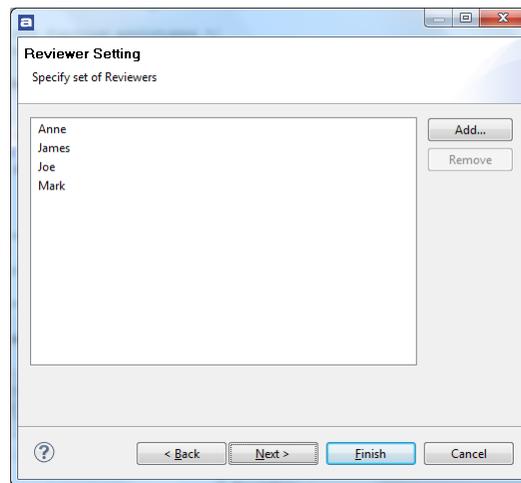


Figure 4 - Selecting reviewers

Finally, the enumerated selection values that the reviewers shall be able to make in review comment drop-down lists must be defined. By default the selection values below are pre-defined.

Review comment classification types:

- Coding standards violation
- Program logic problem
- Optimization problem
- Usability problem
- Code clarity problem
- Code is missing
- Code is irrelevant
- Suggestion
- Other

Review comment severity:

- Critical
- Major
- Normal
- Minor
- Trivial

Review comment resolution decision:

- Valid needs fixing
- Valid fix later
- Valid duplicate
- Valid but won't fix
- Invalid won't fix
- Unsure validity

Review comment status:

- Open
- Resolved
- Closed
- Reopened

All these values can however be changed according to company or project needs. For example, your project might want to add a review comment classification type that is labeled "Portability problem".

Once the source code review session has been created in the **Atollic TrueSTUDIO Pro** IDE, other developers in the same project can connect to the review session and start to review the source code files right from inside the IDE.

---

## PERFORMING THE SOURCE CODE REVIEW

Once the source code review session is created, team members can start to add code review comments. Adding a review comment is easily done by right-clicking in the left margin of any code line in the editor.

The reviewer can then add a title and description to the review comment, as well as classification (such as logic error, portability problem, etc.) and severity (such as critical, major, minor etc.).

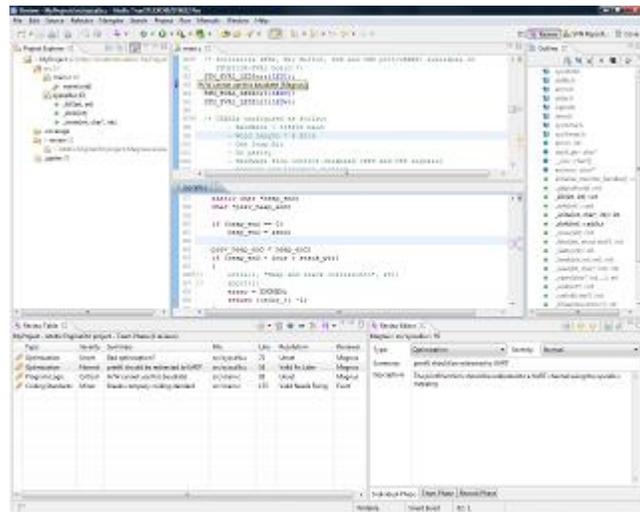


Figure 5 – Atollic TrueSTUDIO Pro integrates features for code reviews and meetings

In the second (review meeting) phase of the review, all reviewers sit down and discuss all the different review comments together in a code review meeting (typically using an LCD projector) in a meeting room. Some review comments might be decided to be invalid, other are deemed valid and needs to be fixed, etc.

The review meeting typically discuss if the review comment is valid, if it needs to be fixed, etc. For review comments that needs fixing, the review meeting can optionally assign detected problems to different team members for fixing, and he or she will then get a work list with all the review comments that was assigned to him or her.

Using the simple methodology of source code reviews, software quality can be improved. Reviewers can study the code in their own time, and all team members taking part in the code review meeting improve their coding skills by learning from the colleagues and their review comments.

It is however of paramount importance that team members do not take review comments personally, but instead separate personal ego from potential problems in the code. Everyone makes mistakes while developing complex software, which most professional developers are well aware of. By discussing the review comments openly, everyone gets a chance to improve their coding skills and the software quality is improved.

Now that functions for code reviews are getting integrated into professional C/C++ environments, there is no longer any excuse not to use this methodology, at least for the most critical parts of the application.

Source code reviews can be done only one or many times during the same project as appropriate. They can also address all software in the project, or only a small part of the project dependent on needs and priority. Remember that it is better to review only the most important part of your software, rather than not at all!

## SUMMARY

Software complexity and code size increase every year, and so does the problem of releasing high-quality software. Performing source code reviews is one of the cheapest and most effective ways of improving software quality.

By exposing source code to a second eye, many problems can be found at an early stage during development, when it is much quicker and cheaper to correct the problem.

New embedded systems development tools, like **Atollic TrueSTUDIO Pro**, integrate features for source review seamlessly into the C/C++ development environment. This for the first time empowers development teams who want increase their software quality using source code reviews with proper tool support.

More information about Atollic and **Atollic TrueSTUDIO** is available here:

[www.atollic.com](http://www.atollic.com)

[www.atollic.com/truestudio](http://www.atollic.com/truestudio)