**Generic SCons**

**Lessons Learned**

|  |  |
| --- | --- |
| Document Control: | |
| Project: | ETK: SCT\_Sconstools |
| Revision: | 0.5 |
| Last Change: | 06-Jun-2016 |
| Confidence Level: | 🞏 Public 🗷 Confidential |

CONFIDENTIAL AND PROPRIETARY PROPERTY OF ADAS SIBIU - ALL RIGHTS RESERVED

|  |  |
| --- | --- |
| Document State: | |
| State: | Draft |
| Author: | Lenin Palanisamy |
| Reviewed by: | Andre Fischer |
| Released by: | DD-MMM-YYYY |

Revision History

| Version | Date | Change Description | Author | Reviewed By |
| --- | --- | --- | --- | --- |
| 0.1 | 20-Jul-2015 | Initial template draft version | Lenin Palanisamy | Andre Fischer |
| 0.2 | 03-Aug-2015 | Updated section 2.1 | Tridip Bhagawati | Lenin Palanisamy |
| 0.3 | 05-Aug-2015 | Add items 1-4 in Section 2.1.1 | André Fischer | Lenin Palanisamy |
| 0.4 | 06-Aug-2015 | Add item 2 in Section 2.1 | Tridip Bhagawati | Lenin Palanisamy |
| 0.5 | 06-Jun-2016 | Added all LLs in sections 2.1.1, 2.2.1, 2.3.3 | Lenin Palanisamy | Lenin Palanisamy |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc452986337)

[1.1 Purpose 4](#_Toc452986338)

[1.2 Scope 4](#_Toc452986339)

[1.3 Reference documents 4](#_Toc452986340)

[1.4 Overview 5](#_Toc452986341)

[2 Lessons Learned 6](#_Toc452986342)

[2.1 Generic SCons Development and Maintenance 6](#_Toc452986343)

[2.1.1 Error and Exception Handling 7](#_Toc452986344)

[2.2 Generic SCons Project Management 9](#_Toc452986345)

[2.2.1 Estimation and Scheduling 9](#_Toc452986346)

[2.2.2 Metrics collection 10](#_Toc452986347)

[2.2.3 Risk Management 10](#_Toc452986348)

[2.2.4 Configuration Management 10](#_Toc452986349)

[2.3 Process improvements 10](#_Toc452986350)

[2.3.1 Release Process 10](#_Toc452986351)

[2.3.2 Testing Process 10](#_Toc452986352)

[2.3.3 Quality Process 10](#_Toc452986353)

# Introduction

## Purpose

Lessons Learned (LL) is knowledge or understanding which is gained by positive or negative experiences by a member of a project. It contains:

* success factors
* reasons for and impacts of plan deviations
* recommendations how to cooperate with stakeholders
* a list of "don't do"
* proposals for process, quality, management improvement actions
* accumulated data derived from the measurement data

The impact of this experience is significant of project operations and is applicable in processes, decisions or projects. Lessons learned should help to reduce costs, failures, mishaps or risks inside the SCT\_Sconstools tool in Engineering Tool Kit (ETK) project.

Evaluation of completed projects and record of the results and knowledge for further reference or use.

Lessons Learned are collected within the document for reuse by successive similar projects with the intention to improve quality and efficiency.

## Scope

This document applies for recording all lessons learned during the processes, development and management of SCT\_Sconstools tool in Engineering Tool Kit (ETK) project.

This document will follow the Continental Lessons Learned process.

## Reference documents

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Reference Document** | **Location** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5. |  |  |
|  |  |  |

## Overview

Lessons Learned is mandatory for SCT\_Sconstools tool in Engineering Tool Kit (ETK) project for continuous improvement on development and management activities.

Generic SCons development team will record the lesson that is a:

* Knowledge or understanding which is gained by positive or negative experience
* Applicable in a similar situation
* Helpful for other colleagues
* Proven solution or verified experience

Record experiences in the following template format in each key item in section 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title** | **Description** | **Solution** | **Author** |
| 1. | <Topic+Root Cause+Approach>  e.g., “Improve the quality by using review checklists” | What was the initial situation?  In which context did the root cause happen?  Where does the lesson apply? | Where are the proven correction actions? Which result was achieved? Which constraints do apply? |  |

# Lessons Learned

## Generic SCons Development and Maintenance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title** | **Description** | **Solution** | **Author** |
| 1. | “**Avoid confusion by jotting down in a document the changes done on applying GenericScons for new component**” | On applying GenericScons for component GS(Global Scheduler), I made changes in files to adapt GenericScons environment  I did not note down the changes I made in any document, and as the days passed I forgot the changes I made.  Lesson I learned – I met lot of confusion on which files did I changed and the reason why I changed. | To prepare a document in the beginning and note all the changes done.  In this way, developer will always remember the cause of changes and avoid confusions.  I suggest always preparing a document and note down the changes, which developer have to deliver to the component owner. | Tridip Bhagawati |
| 2. | “**To initiate Scons build node, add it to Scons Alias**” | I created a build node to do an action(read sdl file after this file is generated) in build time.  I want this action to work when I build sdl file using command line target “component\_sdl”  I was astonished as the action was not initiated. | To initiate a build node(can be an action in this case but it may be any other task to Scons) it has to be added to Scons Alias.  For eg: Alias(“”, build\_node) | Tridip Bhagawati |

### Error and Exception Handling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title** | **Description** | **Solution** | **Author** |
| 1. | **“Two environments with different actions were specified for the same target” –** A typical cause | When adding new targets to a GenericScons environment, this error may pop up.  **Problem:** Scons has gathered all the information on build targets integrated in build environment, and for at least one build target (e.g. an \*.obj-file) there are apparently two different ways of generating it according to the build scripts. | For any source file lists remove the “#” at the start, if existing.  **Explanation:** When defining a path, “#” indicates the location of SConstruct, so suppose you want to compile a certain source file for both release and debug. Then the resulting .obj’s must have different build locations. This is typically accomplished by “VariantDir”. The VariantDir-Command is effectless though, if “#” is prefixed, since then we well always build relative to SConstruct location. | André Fischer |
| 2. | **Multiple include directories in commandline** | It happens that the compilation command for source files contains directories twice, once in the project and once under 04\_Build, e.g.  -IH:\sandboxes\regular\SIB\_StereoImageBase\04\_Engineering\04\_Build\algo\sib\ti\_c674x\release\**01\_Source\_Code\algo\sib\common** -IH:\sandboxes\regular\SIB\_StereoImageBase\04\_Engineering\**01\_Source\_Code\algo\sib\common**  Such a behavior can result in the commandline becoming too long either for commandline or for the compiler!! | Include directories shall always be prefixed with “#”, i.e. given relative to SConstruct. If not, VariantDir duplicates this include as shown in the description. | André Fischer |
| 3. | **Command line too long for visual studio targets** | In scons\_adas\_extensions\msvc-addon.py the maximal length of command is defined, before the command is forwarded via temporary files to the vs compiler. In theory this should/could match the batch line limit of 8191. But setting the key “MAXLINELENGTH” to 8191 can result in errors, because of a bug in Scons 2.2.0 related to computation of the command line length. | In such cases where one tries to go for hard bounds such a bound needs to be positively verified. Even SCons docs may contain bugs!  This is why instead of 8191, at the moment (2015-08-05) the value 7800 is set.  **Info:** This bug has been fixed in more recent versions of SCons. | André Fischer |
| 4. | **Adding global compiler flags** | From time to time, integrators and/or project management require the use of certain compiler flags for certain targets by **ALL** components. Just adding those to the common\_config.scfg can lead to build failures in components. | Research on new flags in compiler documentation and/or on the net is indispensable, since any new globally set compiler options may interfere with already existing ones, global and/or local ones!  This may explain build failures for certain components on Jenkins server and needs to be communicated to the initiator before release. | André Fischer |
| 5. | **Provide the user friendly information for invalid configuration by adding exception handler** | When there is an invalid configuration value in shared/specific configuration item for a component, misleading message “old version of scons used” displayed. This is very difficult to locate and fix the configuration issues in GSCons | Implement Error and warning exception handler for each and every mandatory configuration item in GSCons and do thorough testing | Lenin Palanisamy |
|  |  |  |  |  |

## Generic SCons Project Management

### Estimation and Scheduling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title** | **Description** | **Solution** | **Author** |
| 1. | Improve the scheduling list of issues to be delivered by bringing the Release and testing processes for GSCons | There is no process in GSCons which will describe the work flow and release definition and testing process for quality deliverables. This was making difficult to plan the list of issues for a release. | Prepare the Release and Testing process for GSCons development and Maintenance activities. This will be used for all developers of GSCons. | Lenin Palanisamy |
|  |  |  |  |  |
|  |  |  |  |  |

### Metrics collection

### Risk Management

### Configuration Management

## Process improvements

### Release Process

### Testing Process

### Quality Process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title** | **Description** | **Solution** | **Author** |
| 1. | Improve the quality by defining the checklists for each issue and release | There is no checklist for GSCons to make sure that the all mandatory points are followed. This was making many reworks and repeated doubts. | Define the checklist templates for Issue and release which will be used for each issue and release. | Lenin Palanisamy |
| 2. | Improve the quality by maintaining the Lessons Learned document for continuous improvement | There was not LL document, in turn makes a lot of rework and no knowledge sharing | Prepare the LL document and encourage the team to update the document with their knowledge and positive and negative experiences. | Lenin Palanisamy |
| 3. | Improve the process of GSCons by reviewing quality updates in ADAS periodically for continuous improvements | Any project should adapt to common quality process following in the BU or organization. | Review the project specific quality process if there is a notification regarding quality updates in ADAS | Lenin Palanisamy |
|  |  |  |  |  |
|  |  |  |  |  |