# ESROCOS COMMUNICATION AND OUTREACH MANUAL

**ESROCOS_D6.1**

| **Due date of deliverable:** | 19-01-2017 |
| **Start date of project:** | 01-11-2016 |
| **Duration:** | 27 months |
| **Topic:** | COMPET-4-2016 Building Block a) Space Robot Control Operating System |
| **Work package:** | WP 6100 |
| **Lead partner for this deliverable:** | GMV |

“This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement No 730080.”

<table>
<thead>
<tr>
<th>Dissemination Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
</tr>
<tr>
<td>CO-1</td>
</tr>
<tr>
<td>CO-2</td>
</tr>
</tbody>
</table>

Prepared by: ESROCOS team  
Approved by: DFKI  
Authorized by: Malte Wirkus

| Code: | ESROCOS_D6.1 |
| Version: | 1.0 |
| Date: | 17/01/2017 |
| Internal code: | GMV 20173/17 V1/17 |
DOCUMENT STATUS SHEET

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Pages</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>21/11/2016</td>
<td>50</td>
<td>First Version, draft skeleton</td>
</tr>
<tr>
<td>0.2</td>
<td>06/01/2016</td>
<td>50</td>
<td>Added more content: Public websites and software development sections.</td>
</tr>
<tr>
<td>0.3</td>
<td>09/01/2016</td>
<td>50</td>
<td>Added content to section on CI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Spell checking</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added content to &quot;Scope&quot; and &quot;Content&quot; sections</td>
</tr>
<tr>
<td>0.4</td>
<td>13/01/2017</td>
<td>21</td>
<td>Added Acknowledgement Text for publications</td>
</tr>
<tr>
<td>1.0</td>
<td>17/01/2017</td>
<td>21</td>
<td>Final document V1.0, delivered for SRR</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

1. INTRODUCTION .................................................................................................................. 5
   1.1. PURPOSE ..................................................................................................................... 5
   1.2. SCOPE ......................................................................................................................... 5
   1.3. CONTENTS .................................................................................................................. 5

2. REFERENCE AND APPLICABLE DOCUMENTS ............................................................... 6
   2.1. APPLICABLE DOCUMENTS ....................................................................................... 6
   2.2. REFERENCE DOCUMENTS ......................................................................................... 6

3. TERMS DEFINITIONS AND ABBREVIATED TERMS ..................................................... 7
   3.1. DEFINITIONS ............................................................................................................. 7
   3.2. ACRONYMS .............................................................................................................. 7

4. COMMUNICATION AND OUTREACH MANUAL ......................................................... 8
   4.1. GENERAL COMMUNICATION AND OUTREACH ..................................................... 8
       4.1.1. Internal communication ....................................................................................... 8
           4.1.1.1. Project Website Internal Area ................................................................. 8
           4.1.1.2. Mailing lists ............................................................................................... 10
           4.1.1.3. Individual team contacts ......................................................................... 10
           4.1.1.4. Document and Deliverable development ............................................... 11
           4.1.1.5. Discussion ................................................................................................. 12
       4.1.2. External communication .................................................................................... 12
           4.1.2.1. Press releases ......................................................................................... 12
           4.1.2.2. Scientific Publications ............................................................................. 12
           4.1.2.3. Face-To-Face Outreach ......................................................................... 13
           4.1.2.4. Public Websites ...................................................................................... 13
   4.2. SOFTWARE DEVELOPMENT ................................................................................... 15
       4.2.1. Version control system ..................................................................................... 16
       4.2.1.2. Issue tracking .............................................................................................. 17
       4.2.1.3. Source Code distribution and Build System ............................................. 17
       4.2.1.4. Continuous Integration ............................................................................. 19

5. SUMMARY OF MATERIALS ............................................................................................... 20
LIST OF TABLES AND FIGURES

Table 2-1: Applicable documents ........................................................................................................ 6
Table 2-2: Reference documents ........................................................................................................ 6
Table 3-1: Definitions .......................................................................................................................... 7
Table 3-2: Acronyms .......................................................................................................................... 7
Table 4-1: Possible platforms for publishing ...................................................................................... 12
Table 4-2: Possibilities for Face-To-Face Outreach ............................................................................. 13
Table 5-1: Comparison among RCOS. ................................................................................................. 20

Figure 4-1: Website Log-In for internal area ......................................................................................... 8
Figure 4-2: Accessing file repository and forum from website.............................................................. 8
Figure 4-3: File repository of internal area of the website .................................................................. 9
Figure 4-4: Forum of the internal area of the website ....................................................................... 10
Figure 4-5: Threads in the forum have a life-time ............................................................................... 12
Figure 4-6: Public Website of ESROCOS ......................................................................................... 14
Figure 4-7: ESROCOS project page from DFKI ............................................................................... 15
Figure 4-8: Package oriented software development ......................................................................... 15
Figure 4-9: Package oriented software development - Implementation workflow ............................. 16
Figure 4-10: "Issues" tool of GitHub .................................................................................................. 17
Figure 4-11: Autoproj project specification ......................................................................................... 18
1. INTRODUCTION

1.1. PURPOSE
This document will summarize the communication and outreach strategy followed within ESROCOS. It contains all print materials and references to implementations of online materials produced within the project. The document's purpose is, that project partners have a guideline how project internal and external communication should be performed.

1.2. SCOPE
The document is an early deliverable in WP6. It contains the resources for internal communication and outreach and describes the workflow for these activities. The document contains all the information that has been identified until the date of the deliverable. While internal communication is to large degree already established, workflows might change and outreach activities will become more and more important during the course of the project. Thus, the document might be subject to updates during the project so that the document could be used as an internal reference.

1.3. CONTENTS
This document is structured as follows:
- Section 1. Introduction presents the purpose, scope and structure of the document
- Section 2. Reference and Applicable Documents lists other documents that complement or are needed to understand this document.
- Section 3. Terms Definitions and Abbreviated Terms defines terms and acronyms used in the document
- Section 4. Communication and Outreach Manual Contains the main content of the communication and outreach manual:
  - General communication and outreach subdivided into sections internal and external communication
  - Instructions for the software development workflow
- Section 5. Summary of Materials gives a tabular overview about the material references in this document.
2. REFERENCE AND APPLICABLE DOCUMENTS

2.1. APPLICABLE DOCUMENTS

The following is the set of documents that are applicable:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Date</th>
</tr>
</thead>
</table>

Table 2-1: Applicable documents

2.2. REFERENCE DOCUMENTS

The following is the set of documents referenced:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RD.1]</td>
<td>GIT Website : <a href="https://git-scm.com/">https://git-scm.com/</a></td>
</tr>
<tr>
<td>[RD.3]</td>
<td>ROS Website : <a href="http://www.ros.org/">http://www.ros.org/</a></td>
</tr>
<tr>
<td>[RD.5]</td>
<td>GitHub Website : <a href="https://github.com/">https://github.com/</a></td>
</tr>
<tr>
<td>[RD.6]</td>
<td>GitLab Website : <a href="https://about.gitlab.com/">https://about.gitlab.com/</a></td>
</tr>
</tbody>
</table>

Table 2-2: Reference documents
3. TERMS DEFINITIONS AND ABBREVIATED TERMS

3.1. DEFINITIONS

Concepts and terms used in this document and needing a definition are included in the following table:

<table>
<thead>
<tr>
<th>Concept / Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2. ACRONYMS

Acronyms used in this document and needing a definition are included in the following table:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESROCOS</td>
<td>European Space Robotics Control and Operating System</td>
</tr>
<tr>
<td>ERF</td>
<td>European Robotics Forum</td>
</tr>
<tr>
<td>RCOS</td>
<td>Robot Control and Operating System</td>
</tr>
<tr>
<td>ROCK</td>
<td>Robot Construction Kit</td>
</tr>
<tr>
<td>ROS</td>
<td>Robot Operating System</td>
</tr>
<tr>
<td>CI</td>
<td>Continuous Integration</td>
</tr>
<tr>
<td>CVS</td>
<td>Concurrent Versions System</td>
</tr>
</tbody>
</table>
4. COMMUNICATION AND OUTREACH MANUAL

4.1. GENERAL COMMUNICATION AND OUTREACH

In this section we describe the means for communication within the project team. That includes information on how to get in contact with the whole team, or individuals. Also a guideline for collaborative working on documents will be given.

For external communication, we will present our main public representations and as also how to get in touch with space and robotics community, e.g. via publication, workshops or media.

4.1.1. INTERNAL COMMUNICATION

4.1.1.1. PROJECT WEBSITE INTERNAL AREA

We have set up a website that provides an internal area. The website is accessible with the URL http://www.h2020-esrocos.eu/ (also http://h2020-esrocos.gmv.com/ can be used). Figure 4-1 shows how to log in to and how to access the private area. The internal area provides a repository to upload files and a forum for discussion. Figure 4-2 shows how to access the Forum and the Files repository.

![Figure 4-1: Website Log-In for internal area](image1)

![Figure 4-2: Accessing file repository and forum from website](image2)

The files repository (Figure 4-3) is used for larger binary files. In particular Deliverable documents, developed as Microsoft Word Documents or Excel Sheets are distributed using the file repository of the website.

Each project partner has the possibility to upload/download files and to create folders.

Following we list what the files repository could be used for, and in which folder the corresponding files should be stored (if there is a specified location):

- Development files for deliverables: Stored in separated folder for each deliverable, e.g.: "D1.1 Technology Review".
• Document Template files: Stored in the sub-folder “Templates”.
• Re-usable files for documentation, e.g. commonly used image files: Stored in sub-folder “Misc Data”
• Documents of general interest to share with partners: Stored in sub-folder “Misc Data”
• Meeting protocols: Stored in “Meetings”

![Figure 4-3: File repository of internal area of the website](image)

The forum is used for in-depth discussion and finding consensus on particular topics. The advantage over e-mail is that focus is kept on one topic within a discussion thread. The forum should be used for instance for the following issues:

• Discussions about technical topics, which are not about a enhancements of a particular software module (for software feature request or implementation issues, see Section 4.2.1.2)
• Discussions about formulation/content of deliverables
• Discussions about outreach activities

Figure 4-4 shows a screenshot of the forum.
4.1.1.2. MAILING LISTS

Mailing lists are a tool for distributing information to a defined group of people. If a mail is sent to a mailing list, it will be forwarded to each e-mail address, which is registered within the mailing list.

For communicating with all project partners the mailing list esrocospartners@gmv.com was created. All contact persons for each project partner should be registered within this mailing list.

To get in touch with the project reviewers from, the mailing list esrocosreviewers@gmv.com was created.

Both mailing lists are maintained by GMV. In order to get an additional e-mail address added to one of the lists, get in contact with Jorge Ocón Alonso (jocon@gmv.com).

4.1.1.3. INDIVIDUAL TEAM CONTACTS

To get in contact with individual partners, we provide a list with contact information in the internal section of the ESROCOS website. The file can be found in the files section of internal area of the ESROCOS website under “Misc Data/individual_contact_information.xlsx”.

Project partners are invited to extend this list by additional contacts or correcting and adding information keep the document up to date.
4.1.1.4. DOCUMENT AND DELIVERABLE DEVELOPMENT

In ESROCOS, we agreed on using Microsoft Word as primary tool for document writing. To cope with the well-known drawbacks of using Word for collaborative work, we suggest the following workflow:

- When starting a document, always use the latest template from the “Templates” subfolder of the files repository of the website.
- Each deliverable should have its own folder containing the following content:
  - Main Document Master: Should be named after the pattern "ESROCOS_#{DELIVERABLE_ID}_#{DELIVERABLE_NAME}_V{VERSION}-MASTER.docx" and represents the most current version of the document.
  - Contributions: This is the place where collaborators upload their contributions. The contributions should be named following the pattern "ESROCOS_#{DELIVERABLE_ID}_#{DELIVERABLE_NAME}_V{VERSION}-#{COMPANY}.docx" (e.g. "ESROCOS_D1.1_TechnologyReview_V0.3-KUL.docx").
  - Version History: Each time a new MASTER was generated by merging contributions in the main document, the old MASTER is supposed to be moved into the "Version History" subfolder.
  - Submissions: Each time a document was ‘officially’ submitted, a copy goes to the "Submissions" sub-folder. The file should be named according to the following pattern: "#{YYYYMMDD}_ESROCOS_#{DELIVERABLE_ID}_#{DELIVERABLE_NAME}_V{VERSION}-#{COMPANY}.docx" (e.g. "20170108_ESROCOS_D1.1_TechnologyReview_V1.docx"), where #{YYYYMMDD} corresponds with the date the document was submitted (not finalized).

- The main responsible person for a deliverable is in charge of creating the above mentioned folder structure and compile an initial MASTER document. The document templates make use of the AutoText feature of Word. Please refer to the Document Misc Data/Modification of AutoText in Word Documents.pdf for information on how to modify the AutoText.

- The main responsible person for a deliverable is supposed to ask partners for contributions to sections of the document.

- Each partner always downloads the latest MASTER document before writing.

- During writing, each partner must have the modification tracking feature of Word enabled.

- Each partner uploads his contributions to the “Contributions” subfolder following the naming guideline mentioned above.

- The main responsible of a deliverable is in charge of collecting all contributions from time to time and merge them into a new MASTER document and increasing the version number. The main responsible has to update the document status sheet section in the document with meaningful comments about the changed since the previous document MASTER.

- The main responsible is in charge of uploading the new MASTER and moving the old MASTER to the version history subfolder. For some documents it is good practice to announce a date contributions will be merged. It might also be a good practice for some documents to send a short notification when there is a new MASTER available for a deliverable.

- The main responsible is in charge of finalizing the document. And submitting it. The main responsible is in charge of uploading a copy of the submitted document to the “Submissions” subfolder as mentioned above.
Document Submission to PSA

When reporting to PSA, documents must be uploaded to “JIRA”. Current, only GMV has an account at JIRA under https://spaceroboticssrc.atlassian.net/

For a deliverable responsible submitting to PSA involves the following steps:

- Finalize the document as described in the workflow above
- Send a notification mail to Jorge and Miguel (jocon@gmv.com, mmunoz@gmv.com) that the document is ready for submission. Add a direct link to the document location (within the “Submission” subfolder) to the mail.

4.1.1.5. DISCUSSION

The internal section of the website provides a forum that should be used for discussions on distinct topics. Threads within the forum can be flagged as “Open” or “Closed” indicating whether the issue discussed is still open for discussion or has finished.

![Figure 4-5: Threads in the forum have a life-time](image)

Each partner in ESROCOS has the ability to Comment on open threads or creates new threads. Changing the state of a thread is only possible for the thread creator.

4.1.2. EXTERNAL COMMUNICATION

4.1.2.1. PRESS RELEASES

Press releases should be conducted on major milestones of the project. Each project partner can carry out press releases to attract public interest within their respective countries.

4.1.2.2. SCIENTIFIC PUBLICATIONS

Every project partner is invited to write publications on their own or with other partners from within the ESROCOS consortium or also external partners. Papers published within ESROCOS must include the following Acknowledgement:

“This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement No 730080.”

The following table summarizes a few conferences or journals from the area of robotics or computer science that might be possibilities to publish results from ESROCOS.

<table>
<thead>
<tr>
<th>Conference/Journal</th>
<th>Full name</th>
<th>Comment/Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-SAIRAS</td>
<td>International Symposium on Artificial Intelligence, Robotics and Automation in Space</td>
<td>Must be space related</td>
</tr>
<tr>
<td>RSS</td>
<td>Robotics: Science and System Conference</td>
<td>Excellent quality international conference</td>
</tr>
<tr>
<td>ICRA</td>
<td>IEEE International Conference on Robotics and Automation</td>
<td>High-excellent quality international conference</td>
</tr>
<tr>
<td>IROS</td>
<td>International Conference on Intelligent Robots and Systems</td>
<td>Probably biggest robotics conference. Focused on robotic systems. Around 900 papers get accepted here.</td>
</tr>
</tbody>
</table>
4.1.2.3. FACE-TO-FACE OUTREACH

The following table summarizes some vents that could be good opportunities for face-to-face outreach:

<table>
<thead>
<tr>
<th>Event</th>
<th>Information</th>
<th>What we could do there</th>
</tr>
</thead>
<tbody>
<tr>
<td>European Robotics Forum (ERF)</td>
<td><a href="http://www.macs.hw.ac.uk/erf2017/">http://www.macs.hw.ac.uk/erf2017/</a> Located at March 22-24, 2017 in Edinburgh, Scotland, UK</td>
<td>Organization of a Session</td>
</tr>
<tr>
<td>CeBit</td>
<td>20.-24.3. Hannover, Germany <a href="http://www.cebit.de/home">http://www.cebit.de/home</a></td>
<td>Promote ESROCOS as a visitor. Promote ESROCOS with a project flyer on DFKI booth.</td>
</tr>
</tbody>
</table>

4.1.2.4. PUBLIC WEBSITES

The primary source of information for public is our project website. Currently it is reachable under the URL http://www.h2020-esrocos.eu/ or http://h2020-esrocos.gmv.com. The URL http://www.h2020-esrocos.eu/ has been registered. Figure 4-6 shows a screenshot of the front page of the website.

The public website contains updated information about the project and its progress. This includes:

- General project objectives
- Information about the consortium
- Project schedule include milestones
- Information about deliverables
- Information about publications
- A news section, where for example important events are mentioned
A private area (cf. Section 4.1.1.1) Any project partner can propose changes to the website such as the posting of news or addition of a publication. For updating content of the Website, currently GMV is in charge. If one of the partners wants an update to the website, Jorge Ocón Alonso (jocon@gmv.com) should be contacted.

Beside the main web online representation of the project, according to the partners individual company rules there might be presentations of the project on the company websites as well. For example DFKI has a small project introduction page for each project they are working on. For ESROCOS this project page is located at http://robotik.dfki-bremen.de/en/research/projects/esrocos.html. The page contains a general information text about the project alongside with contact information. Figure 4-7 shows a screenshot of the project information page of DFKI.

Figure 4-6: Public Website of ESROCOS
The content of the DFKI project page is supposed to be changed only if major technical results have been achieved (maybe 1-2 times during the course of the project). Changes are maintained by DFKI, but any partner can propose changes such as reformulations. To request a change, Malte Wirkus (malte.wirkus@dfki.de) should be contacted.

4.2. SOFTWARE DEVELOPMENT

One major characteristic of the software developed in ESROCOS is, that it has its foundation relying on other software products such as TASTE, BIP or other algorithmic libraries. This is a common situation for robotic frameworks, usually being a definition of a programming methodology, a set of development tools, domain-specific software libraries and an extendable collection of software packages. Existing robotic software development frameworks have established best practices on how to deal with this situation, which results in a package oriented software development workflow.

As shown in Figure 4-8 one of the responsibilities of the framework’s development tools is to take care of establishing a common workspace from the separated packages. In Frameworks like ROS [RD.3] or ROCK [RD.2], the following types of packages can be distinguished: There are such packages, which are considered as core packages. Core packages are such packages that establish...
the infrastructure of the framework. In this category fall for example the build system, the component interface related libraries or code template generators, middleware and glue-code generators. Additional tools like logging tools, visualization tools or domain-specific software libraries/components can be considered as both core and contribution packages. Especially ROS showed the potential that lies in providing a mechanism for users to contribute to a growing contribution packages set. These are packages developed using the framework, that where released to public by the original developers (the users). The framework therefore must provide a workflow to publish the packages. The third category, the user packages are those packages the user develops but is not made public. The framework development tools should support the incorporation of such coming from previously unknown sources.

Beside the ability to scale with a growing number of functionalities that are implemented in separate packages, the package-oriented development workflow also provides means for maintaining this large number of packages by allowing maintenance on a per-package basis. Software is split into small chunks, where each of which should represent a single functionality or purpose within the framework. The individual packages can be each on its own be easier understood than a whole complex system could be, resulting in a larger potential group of contributors/maintainers. This increase software maintenance and improves feature tracking and testing.

As shown in Figure 4-9, the maintenance of the framework itself is rather a testing of integrated packages and providing feedback to the individual package development communities on an issues- or requirements-basis.

In ESROCOS, we aim at also establishing a package-oriented development workflow for the RCOS. The following subsections introduce tools and give usage suggestions for them.

4.2.1.1. VERSION CONTROL SYSTEM

Each software or documentation package developed for the RCOS should be stored within a version control system. We suggest the use of GIT [RD.1]. There are several GIT hosting services available to public, providing a GIT server together witch storage space and a web interface providing access the GIT repository. Of the GIT hosting, GitHub [RD.5] is probably the most popular. As an alternative, there are GIT hosting environments available that can be run on an own server. At DFKI, for example a GitLab [RD.6] server is running. A very useful feature of the hosting
environments is that beside source code hosting, they provide tools for issue tracking and discussion board for merge requests, a wiki for documentation and possibility for integration of continuous integration methods.

At this point we have not yet decided on a code hosting strategy. It has to be clarified if for all partners it is okay to have their code available to public, which would be the case if we choose GitHub.

GitHub allows the creation of so called Organizations, which are used as grouping criteria for projects. Each organization can have a number of repositories and have a group of people being assigned to. Referring to Figure 4-8, we would create an organization srocos-core ([https://github.com/srocos-core/](https://github.com/srocos-core/)), as the place, where all RCOS-specific packages will go to, which are not yet maintained at another source location.

People assignment to an organization is made by linking their GitLab profile to the organization. While no separated user account will be required for the organization, anybody who is going to contribute will require a GitHub profile.

For using GIT we suggest looking at the official Git tutorial [RD.7] and using the GIT Cheat Sheet [RD.4] in the beginning, while still memorizing the commands.

4.2.1.2. ISSUE TRACKING

Software feature request, bug reports and other issues related to software that might be reported or require discussion should all be handled within the issues tool of the GIT hosting environment. Figure 4-10 shows a screenshot of a discussion thread from a package of the ROCK software project.

![Figure 4-10: "Issues" tool of GitHub](https://example.com)

The Issue system is open to anyone. Any logged-in user can create an issue. It is in the responsibility of the package maintainers to answer an issue contemporary.

4.2.1.3. SOURCE CODE DISTRIBUTION AND BUILD SYSTEM

In Figure 4-8 there are “development tools” mentioned, that establish a workspace for the developers. The main development tool used in ESROCOS for this purpose is the tool Autoproj [RD.8].

Autoproj allows easily installing and maintaining software that is under source code form (usually from a version control system). It has been designed to support a package-oriented development process, where each package can have its own version control repository (think “distributed version
control”). It also provides an easy integration of the local operating system (Debian, Ubuntu, Fedora, maybe MacOSX at some point).

Unlike the ROS build system, it is not bound to one build system, one VCS and one integration framework. The philosophy behind autoproj is:

- Supports both CMake and autotools, and can be adapted to other tools
- Supports different VCS: cvs, svn, git, plain tarballs.
- Software packages are plain packages, meaning that they can be built and installed /outside/ an autoproj tree, and are not tied at all to the autoproj build system.
- Leverage the actual OS package management system. Right now, only Debian-like systems (like Ubuntu) are supported, simply because it is the only one I have access to.
- Handle code generation properly

![Figure 4-11: Autoproj project specification](image)

The idea in an autoproj installation is that people share definitions for a set of packages (cf. Figure 4-11 top) that can depend on each other. Then, anyone can cherry-pick in these definitions to build its own installation (cf. Figure 4-11 bottom. In practice, one builds a complete configuration per-project).

Therefore the installation is subdivided into the following entities:

- Packages,
- Package sets,
- Buildconf.

To setup a project, the project maintainer sets up a buildconf for his particular project. The buildconf includes package sets and selects the so called layout, a selection of individual packages.

The package sets are maintained by a group of content providers, e.g. a group of framework developers, or a research institute developing their own software packages. A package set itself is just a grouping for individual packages. The packages themselves must include the following information:

- how to get the package’s source code
- how to build the package
- on what the package depends. This can be either another package built by autoproj, or an operating system package.

Also an identifier and classification is given to the package by its name (e.g. drivers/camera_usb_driver).
To setup a development system, one would now simply download the autoproj_bootstrap program and instruct it to bootstrap the installation specified in the buildconf:

```
wget http://rock-robotics.org/stable/autoproj_bootstrap
ruby autoproj_bootstrap git https://url_to_package_set.git
```

The autoproj_bootstrap program installs console programs to use autoproj:

- amake: build whole project or current package
- amake package_name: build package of given name
- aup: update (e.g. git pull) repositories of whole project or current package
- aup package_name: update (e.g. git pull) repository of given package
- autoproj rebuild: rebuild the whole project
- autoproj doc: generate documentation for each package

To get started after bootstrapping usually one would update your installation and build if afterwards:

```
aup
amake
```

It is good practise for project maintainers, to provide a shell script file that executed the above mentioned commands.

4.2.1.4. CONTINUOUS INTEGRATION

The package integration utility was described in Section 4.2.1.3, but still other aspects of continuous integration are foreseen in ESROCOS: There should be infrastructure for automated documentation generation, automated builds with build reports, automated tests and code coverage statistics available. The continuous integration infrastructure is not yet set up and is strongly dependent on the chosen hosting strategy, since some GIT hosting services as GitHub have support for integrating CI utilities via plug-ins to the web-services.
5. **SUMMARY OF MATERIALS**

<table>
<thead>
<tr>
<th>Document</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESROCOS Website</td>
<td><a href="http://h2020-esrocos.gmv.com">http://h2020-esrocos.gmv.com</a> or <a href="http://www.h2020-esrocos.eu/">http://www.h2020-esrocos.eu/</a></td>
</tr>
<tr>
<td>Internal discussion forum (internal area of website)</td>
<td><a href="http://h2020-esrocos.gmv.com/forum/">http://h2020-esrocos.gmv.com/forum/</a> or <a href="http://www.h2020-esrocos.eu/forum/">http://www.h2020-esrocos.eu/forum/</a></td>
</tr>
<tr>
<td>Document Template for Deliverables</td>
<td>Internal area of website: Templates/Text Document-Template.docx</td>
</tr>
<tr>
<td>Modification of AutoText in Word Documents</td>
<td>Internal area of website: Templates/Modification of AutoText in Word Documents.pdf</td>
</tr>
<tr>
<td>Package Set Definition</td>
<td>Not yet available</td>
</tr>
<tr>
<td>Source code host</td>
<td>Not yet available</td>
</tr>
<tr>
<td>Main buildconf</td>
<td>Not yet available</td>
</tr>
<tr>
<td>Project information page from DFKI</td>
<td><a href="http://robotik.dfki-bremen.de/en/research/projects/esrocos.html">http://robotik.dfki-bremen.de/en/research/projects/esrocos.html</a></td>
</tr>
</tbody>
</table>