

Masters in Informatics Engineering  
Internship  
Final Report

# Rich Communicator for Android

Sandra Filipa Leitão Mendes

smendes@student.dei.uc.pt

WIT Software Supervisor:  
Eng. Nuno Costa

DEI Supervisor:  
Prof. Dr. Bruno Cabral



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA





**Departamento de Engenharia Informática**  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra

Pólo II, Pinhal de Marrocos, 3030-290 Coimbra  
☎ +351 239 790 000 | 📠 +351 239 701 266  
✉ [info@dei.uc.pt](mailto:info@dei.uc.pt)



**WIT Software, S.A.**

EN1/IC2, Km 185,6, Banhos Secos, Santa Clara,  
3040-032 Coimbra  
☎ +351 239 801 030 | 📠 +351 239 801 039  
✉ [info@wit-software.com](mailto:info@wit-software.com)

Sandra Filipa Leitão Mendes  
[smendes@student.dei.uc.pt](mailto:smendes@student.dei.uc.pt)  
Nº 2006131531

WIT Software supervisor  
Nuno Costa  
[nuno.costa@wit-software.com](mailto:nuno.costa@wit-software.com)

DEI supervisor:  
Bruno Cabral  
[bcabral@dei.uc.pt](mailto:bcabral@dei.uc.pt)



## ACKNOWLEDGEMENTS

To my parents, for whom I owe everything I have and am today, and for putting their children's education above all.

To my supervisors Nuno Costa and Bruno Cabral, for the help, the patience and the effort throughout this internship.

To WIT Software for the opportunity to work in an excellent environment and for the responsibility that was given to me during this internship.

To the team who accompanied me throughout these months, with all the necessary support and friendship.

Last but not least, to all my friends who have spent months hearing about the internship, work, Android and similar, but never stopped being there.

Thank you all for everything.



## ABSTRACT

In our days, communication and portability are key points in the people's lives. With the emergence of smartphones, it is expected that mobile devices do more than a simple call or send an SMS. It is necessary to develop applications to enrich the communication of their users, and take advantage of all available features.

To encourage the development of such applications, an initiative named Rich Communicator Suite was created and now supported by operators and other telecommunication companies. This effort defines a set of features such as voice and video calls, social presence, chat and chat group and file transfer, based on common standards, in order to develop applications compliant with each other.

Rich Communicator for Android aims to develop an application to Android devices that supports the features specified in Rich Communications Suite, over a 3G or wireless connection.

This project is part of a suite of products of WIT Software, which aims to close the gap between people devices, providing a full converged experience.

This internship lasts for one year and comes against a real demand in the market. There is also a real interest from companies and operators in this type of applications.

## KEYWORDS

"Android", "Communication", "Mobile", "Rich Communication Suite", "Smartphone", "Social Presence", "Softphone", "VoIP"





## TABLE OF CONTENTS

Acknowledgements .....	5
Abstract .....	7
Keywords .....	7
Index of Figures .....	iv
Index of Tables.....	iv
Glossary .....	vi
Acronyms .....	vii
1. Introduction .....	1
1.1. Motivation.....	1
1.2. Goals.....	2
1.2.1. Internship .....	2
1.2.2. Project .....	2
1.3. Document Structure.....	2
2. State of the Art .....	5
2.1. Overview .....	5
2.2. Competitors .....	6
2.3. Comparative Analysis.....	10
3. Requirements Elicitation.....	15
3.1. Rich Communicator Suite.....	15
3.1.1. RCS Release 1 <sup>[57]</sup> .....	15
3.1.2. RCS Release 2 <sup>[58]</sup> .....	16
3.1.3. RCS Release 3 <sup>[59]</sup> .....	16

3.1.4.	RCS-e Advanced Communications <sup>[60]</sup>	17
4.	Architectures and Technologies	19
4.1.	Background knowledge	19
4.1.1.	Android Applications Fundamentals	19
4.1.2.	Communicator Library	21
	PJSIP	21
	XDM Library	22
	FFmpeg	22
4.1.3.	Technologies	22
4.2.	Application Architecture	23
4.2.1.	WIT Communications Suite	24
4.2.2.	WIT Mobile Communicator	25
4.2.3.	WIT Mobile Communicator for Android	26
4.2.4.	Wrapper	28
5.	Project Management	29
5.1.	Software Development Methodology	29
5.1.1.	Scrum Basics	29
5.1.2.	How do Scrum works?	29
5.1.3.	Variant Used	30
5.1.4.	ROLES	31
5.1.5.	Product Backlog	31
5.1.6.	Sprint Backlog Example	35
5.2.	Planning	36
5.2.1.	Initial	36
5.2.2.	Revised	37
5.3.	Work breakdown	40

5.4.	Risk Management .....	40
5.5.	Software Test Plan .....	42
6.	Conclusions .....	43
6.1.	Future Work .....	43
6.2.	Lessons Learned .....	44
	References .....	45

## INDEX OF FIGURES

Figure 1 – Data representation of smartphone market share, in first quarter of 2011. ....	5
Figure 2 - Operating system share - recent devices acquirers (June - November, 2010).....	6
Figure 3 - Activity flow representation .....	20
Figure 4 – Communicator Library architecture and its external libraries. ....	21
Figure 5 - WIT Communications Suite architecture .....	24
Figure 6 - High-level architecture of WIT Mobile Communicator applications .....	25
Figure 7 – WIT Mobile Communicator for Android: detailed architecture .....	26
Figure 8 – Scrum methodology.....	30
Figure 9 – Sprint backlog example.....	35
Figure 10 – Assessment of risks priority. ....	40

## INDEX OF TABLES

Table 1 – Comparative analysis of competitors’ features. ....	13
Table 2 – Product backlog of WIT Mobile Communicator for Android. ....	34
Table 3 – Initial planning for the first semester.....	36
Table 4 – Initial planning for the second semester.....	37
Table 5 – Revised planning for the first semester. ....	38
Table 6 – Revised planning for the second semester. ....	39



## GLOSSARY

<b>Android</b>	An open-source operating system for mobile devices.
<b>Android NDK</b>	Technology that allows applications to use native code.
<b>Android SDK</b>	Software Development Kit that allows developing applications for Android devices.
<b>Feature Phone</b>	Term used to describe mobile phones with some features as mobile games and tones. They don't run a mobile operating system, but a proprietary firmware. If supports installation of third-party applications, it is only via a relatively limited platform. <sup>(1)</sup>
<b>Scrum</b>	An agile framework for manage projects.
<b>Smartphone</b>	Mobile phone that offers more advanced features than a feature phone. Runs a complete mobile operating system and they can run third-party applications, using an application programming interface. <sup>(2)</sup>
<b>Softphone</b>	Software program that allows its users to make calls over the Internet Protocol.
<b>Third-party application</b>	Programs written to work within operating systems, but are written by individuals or companies other than the provider of the operating system.

## ACRONYMS

<b>3GPP</b>	3 <sup>rd</sup> Generation Partnership Project
<b>API</b>	Application Programming Interface
<b>ARM</b>	Advanced RISC Machine
<b>CDMA</b>	Code Division Multiple Access
<b>GSMA</b>	Global System for Mobile communications Association
<b>HSS</b>	Home Subscriber Server
<b>HTTPS</b>	Hypertext Transfer Protocol Secure
<b>I-CSCF</b>	Interrogating – Call/Session Control Function
<b>IMS</b>	IP Multimedia Subsystem
<b>IP</b>	Internet Protocol
<b>MMS</b>	Multimedia Message Service
<b>NDK</b>	Native Development Kit
<b>OMA</b>	Open Mobile Alliance
<b>OS</b>	Operating System
<b>P-CSCF</b>	Proxy – Call/Session Control Function
<b>RCS</b>	Rich Communication Suite
<b>RISC</b>	Reduced Instruction Set Computer
<b>RTMP</b>	Real Time Messaging Protocol
<b>RTP</b>	Real-time Transport Protocol
<b>S-CSCF</b>	Serving – Call Session Control Function
<b>SBC</b>	Session Border Controller
<b>SDK</b>	Software Development Kit
<b>SIP</b>	Session Initiation Protocol
<b>SMS</b>	Short Message Service

<b>UI</b>	User Interface
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>URI</b>	Uniform Resource Identifier
<b>VoIP</b>	Voice over Internet Protocol
<b>WCS</b>	WIT Communications Suite
<b>WMC</b>	WIT Mobile Communicator
<b>XDM</b>	XML Document Management
<b>XML</b>	eXtensible Markup Language







## 1. INTRODUCTION

This document reflects the work done by Sandra Mendes, during the first semester of the one-year internship, within the scope of Master in Informatics Engineering of the Faculty of Sciences and Technology, University of Coimbra.

The internship, Rich Communicator for Android, is taking place at WIT Software, S.A.<sup>[3]</sup>, a company specialized in the development of software products for mobile devices and telecommunications companies and it is supervised by Bruno Cabral PhD, professor in the Department of Informatics Engineering and by Eng. Nuno Costa, project manager in WIT Software.

### 1.1. MOTIVATION

As technology evolves, more and more users want their devices to make available a wide range of features. One of the types of services that have been growing over time has the aim to enrich communications, like instant messaging services and applications that allow users to make voice and video calls over the Internet connection.

WIT Software has a set of products, WIT Communications Suite<sup>[4]</sup>, that allows users to have a centralized contact list, which is accessible from several devices and gives the ability to make voice and video calls over IP. Users are also able to send instant messages, to transfer files and share images or video during a voice call.

Currently, the following products compose the suite:

**WIT PC Communicator** – available for Windows<sup>[5]</sup>, Mac OS X<sup>[6]</sup> and Linux<sup>[7]</sup>, allows users to access above features by installing the application on computer.

**WIT Web Communicator** – through a browser user is able to access to his/her contacts and features without any kind of installation.

**WIT Mobile Communicator (WMC)** – currently available for iOS<sup>[8]</sup> devices, allowed the expansion of existing product to a new market segment, the mobile devices.

**WIT Toolbar Communicator** – is a plug-in for the browsers that provides services like SMS, MMS and calls and allows the user to mobilize the content of the Internet to the mobile phone.

**WIT Flash-to-SIP Gateway** – provides to any Flash-enabled browser all the communication services available in today's mobile and IP networks.

This project is part of this suite and it aims to expand WIT Mobile Communicator for Android<sup>[9]</sup> devices. This expansion is important for the company's strategy since users began to give high value to mobility and with the emergence of smartphones and tablets it is important that the software follows technology's evolution, and new applications that provide these features appear for a wide range of mobile devices.

As an evidence of these features importance not only for the users, but also for the operators, a project named Rich Communication Suite<sup>[10]</sup> (RCS) was created and currently supported by nearly an hundred operators, vendors and other telecommunication companies. This effort specifies a set of features based on common standards aimed to improve the users' communications. A detailed explanation about this specification can be seen in **Chapter 3 – Requirements Elicitation** of this document.

## 1.2. GOALS

The goals can be divided in two main sections: internship and project goals. The first ones correspond to the knowledge and experience acquired during the internship and the latter ones are related to the project itself and what is expected in its realization.

### 1.2.1. INTERNSHIP

The main goal in this section is to gain experience in an industrial environment, developing software to real customers and learn with project team and managers.

It also intended to improve the knowledge in software engineering using methodologies and software processes and to learn new technologies such as Android APIs, IMS-based systems and some protocols related with the project as SIP and MSRP.

### 1.2.2. PROJECT

WIT Mobile Communicator's goal is to develop an application where users can access to an address book from theirs Android equipments (version 2.1 or higher), where they can see if the contacts are online or not, which features are supported by each friend and start communications from there.

Thus, an overview of the main features to implement in this project is presented below:

- **Enhanced Address Book** – contact list with social presence and capabilities supported for each contact. User can make voice and video calls through WiFi or 3G connections.
- **Content Sharing** – ability to share image or video with a contact during an existing session.
- **File Transfer** – send images, videos or other files with or without an open session with the desired contact.
- **Enhanced Messaging** – existence of a unified composer for the several types of messages (SMS, MMS and IM) and ability to chat with one or more contacts at the same time.
- **Network Address Book** – the existence of remote contact list allows users to access to their contacts and features from several devices.

These features are specified in RCS Releases since the project aimed to be full-compliant with RCS Release 2. To a detailed analysis of RCS features and project goals for each semester of the internship, please refer to the sections **3 – Requirements Elicitation** and **5.2.1. – Planning** of this document, respectively.

## 1.3. DOCUMENT STRUCTURE

This report contains seven sections:

- **Chapter I – Introduction:** this section presents an overview of the internship, including the contextualization, the goals and the structure of whole document;
- **Chapter II – State of the Art:** in this chapter will be done a market study and will be analysed the competitors of the application to be developed;

- **Chapter III – Requirements Elicitation:** will be presented a list of features supported by the application and a brief description of each one, based on RCS specification;
- **Chapter IV – Architecture:** in this section will be explained the detailed architecture of the project and a brief explanation of the suite where WIT Mobile Communicator fits. Will be also presented the technologies to use to reach the proposed goals;
- **Chapter V – Project Management:** this chapter is intended to explain the used methodology, the development plan and the changes made to the plan during the development.
- **Chapter VI – Conclusions:** will be presented a final reflection about the project and the state of the project at the moment.

Apart from this document, the internship documentation is also comprised by a set of appendixes:

- Rich\_Communicator\_for\_Android\_-\_Appendix\_A\_-\_State\_of\_the\_Art;
- Rich\_Communicator\_for\_Android\_-\_Appendix\_B\_-\_Functional\_Specification;
- Rich\_Communicator\_for\_Android\_-\_Appendix\_C\_-\_High\_Level\_Architecture;
- Rich\_Communicator\_for\_Android\_-\_Appendix\_D\_-\_Software\_Quality;

Along this document and in the annexes, the project is referred to as WIT Mobile Communicator for Android. This is due to the fact this is the name of the product.



## 2. STATE OF THE ART

This chapter focuses in the analysis of the current market, and in the existing applications. It's important to conduct a comparative analysis between the various products already in the market, whose features are similar to those that we want to develop.

Firstly, will be presented a brief analysis of smartphone and mobile operating systems current market. This overview gives readers a glance about the thematic of this internship.

Then, will be listed all applications in the Android market and some found externally to it, that can be seen as competitor of WMC for Android. The application considered competitors are the ones that have all or part of the features of the application to develop.

Afterward, the applications will be analysed in more detail, finishing with a comparative analysis between the competitors and the WMC for Android.

This study allow us to know the current market and determine which are the major lacks, and in which topics WMC can improve to be a competitor at the same level or higher to those applications.

However, this section is a brief overview of the performed study. To see the whole analysis, please refer to the **Appendix A – State of the Art**.

### 2.1. OVERVIEW

The smartphone market share has been increasing over time. However, it still has a small part.

The figure below shows the distribution of smartphones market share on worldwide. These data were published by Gartner<sup>[11]</sup> and corresponds to first quarter of 2011.

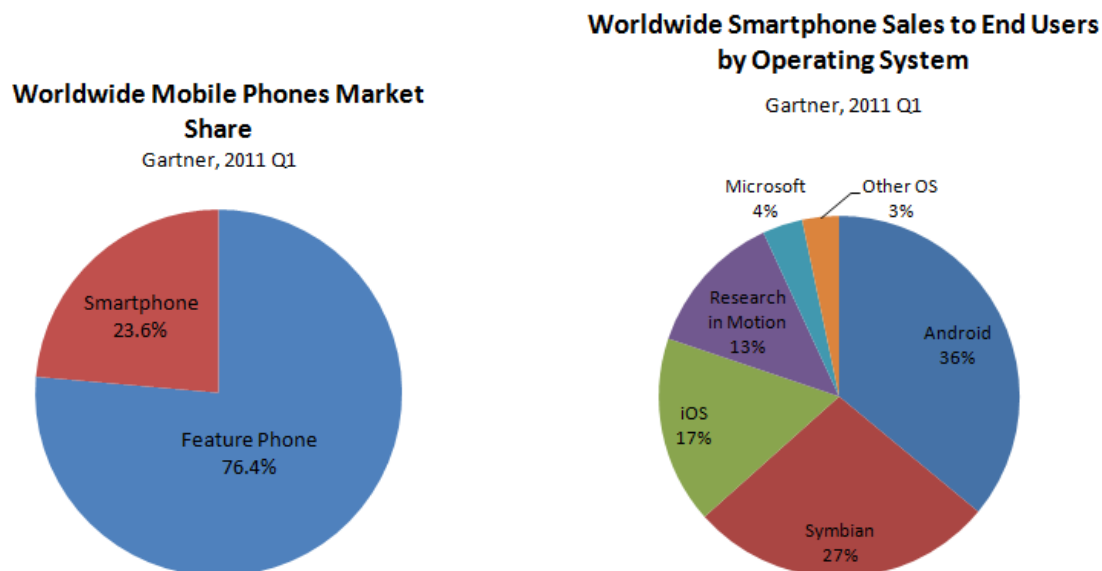
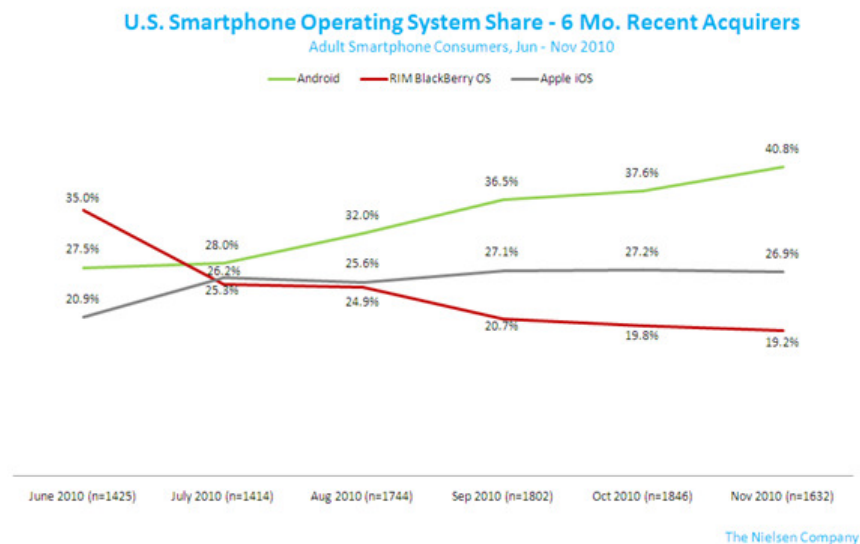


Figure 1 – Data representation of smartphone market share, in first quarter of 2011.

As can be seen in the charts presented above, smartphones represents a few part of the market, with 23.6% behind the feature phones. Considering the smartphone's sales in Q1 of 2011, Android OS is the leader with 36%, followed by Symbian<sup>[12]</sup> and iOS with 27% and 17% of the market sales, respectively.

However, as the smartphone market is constantly changing and the number of Android devices is growing fast (Figure 2, below), it is likely that this distribution change shortly.



**Figure 2 - Operating system share - recent devices acquirers (June - November, 2010)**

In Portugal two years were enough for Android platform achieve the second place in the smartphone market with 34% behind the 48% of Symbian platform<sup>[13]</sup>.

After this analysis we can conclude that Android platform is growing up and gaining market share, so the development of applications for Android it's a good bet for success.

## 2.2. COMPETITORS

Below, is presented a list of Android applications that have all or part of the features we want to develop, via 3G or WiFi connection, and a brief description of each one.

The list presented was retrieved from Android Market<sup>[14]</sup>, from the section of Communication applications. The user classification represents the assessment of application by its users on Android market, followed by the number of votes.

### **MSN Messenger: Mercury Donate**<sup>[15]</sup>

User classification: 4.5 (256)

An instant messaging client for MSN<sup>[16]</sup> and Yahoo<sup>[17]</sup> accounts. Features: File transfer, voice clips, group chat, support for multiple MSN accounts, multiple languages and swipe between contact lists and conversations.



**BeejiveIM** <sup>[18]</sup>

User classification: 3.8 (1,410)

Instant messaging client with support for Android, iPhone<sup>[19]</sup>, Blackberry<sup>[20]</sup>, iPad<sup>[21]</sup> and iPod Touch<sup>[22]</sup> platforms. Support for several accounts (AIM<sup>[23]</sup>, MSN, Yahoo, Facebook<sup>[24]</sup>, GTalk<sup>[25]</sup> and MySpace<sup>[26]</sup>). Features: chat, group chat, file transfer, switch between chats, chat history, address book management.

**IM+ Pro** <sup>[27]</sup>

User classification: 3.8 (993)

Supports Skype<sup>[28]</sup>, Facebook, AIM, MSN, Yahoo, ICQ<sup>[29]</sup>, Jabber<sup>[30]</sup>, GTalk and MySpace. Platforms supported: iPhone, iPad, iPod Touch, Android, BlackBerry, Windows Phone, Windows Mobile, Symbian, Palm OS<sup>[31]</sup> and Java Phones. Features: Pictures and audio file transfer, chat, group chat, chat history, geo-location and multi-language support.

**Bria Android** <sup>[32]</sup>

User classification: 3.3 (307)

A softphone based in SIP protocol, allowing user to make voice calls via 3G or WiFi network connection. Audio codecs supported: G.711, G.722, iLBC, GSM and G.729 (optional purchase). Features: voice calls, voicemail support, speakerphone, mute and hold, call history, multiple call support (swap between calls, merge calls, call transfer) and noise reduction.

**Talkdroid Messenger** <sup>[33]</sup>

User classification: 4.2 (30)

Support for MSN and Yahoo accounts, address book with contacts social presence, chat history, offline messages, swipe among chats.

**aSip** <sup>[34]</sup>

User classification: 2.8 (39)

Softphone SIP based, that supports Android and iPhone platforms. Features: voicemail notification, voice calls (G.711 and GSM audio codecs), speaker, mute and hold call, call history.

**VoIP Video SIP client** <sup>[35]</sup>

User classification: 2.4 (41)

Support for Windows Mobile, Symbian, iPhone, BlackBerry and Android. Audio codecs: G.729, G.722, G.711, Speex (16KHz), iLBC and GSM. Video codecs: H.263 and H.264. Features: voice and video calls, chat, contacts presence, call history, multiple calls support, conference call and echo cancellation.

**KakaoTalk** <sup>[36]</sup>

User classification: 4.5 (505,339)

Is an instant messaging client available for Android and iPhone platforms, in several languages. Features: chat, group chat and file transfer (pictures and video).

**WhatsApp Messenger** <sup>[37]</sup>

User classification: 4.6 (184,475)

Available for Android, iPhone, BlackBerry and Nokia platforms, this application allows users to make chat, group chat and file transfer (video, images and voice notes). User is also able to share location to his friends and has a chat history available.

**Skype**

User classification: 3.6 (170,733)

Skype is a softphone available for smartphones (Android, iPhone and Symbian) and computers (Windows, Mac OS and Linux). Features: voice and video calls, chat, group chat, conference calls, contact list with social presence.

**Yahoo! Messenger**

User classification: 3.9 (54,364)

Support Yahoo!, MSN and Facebook accounts. Features: chat, voice and video calls, file transfer (images and video).

**eBuddy Messenger** <sup>[38]</sup>

User classification: 4.3 (96,656)

Support for multiple accounts (MSN, Facebook, Yahoo, AIM, ICQ, GTalk, MySpace). Features: contact list with social presence, chat with friend from different services.

**Live Profile** <sup>[39]</sup>

User classification: 4.0 (20,779)

Instant messaging client for Android, iPhone and BlackBerry. Features: Contact list with social presence, chat, message receipts, image and video transfer.

**Trillian Messenger** <sup>[40]</sup>

User classification: 4.4 (3,230)

With support for Windows, Mac, iPhone, Android and Blackberry, allows users to chat and send pictures to their friends.

**Meebo IM** <sup>[41]</sup>

User classification: 3.8 (28,054)

This client available for iPhone, Android and BlackBerry platforms, supports AIM, MSN, Yahoo, MySpace, GTalk, Jabber and ICQ accounts. Allows to chat with contacts.

**fring** <sup>[42]</sup>

User classification: 3.7 (46,473)

Support for Android, iPhone and Nokia devices. Features available: contact list with social presence, voice and video calls, conference video calls, chat and history log.

**Kik Messenger** <sup>[43]</sup>

User classification: 4.2 (28,698)

A simple messenger client that allows users to chat, make group chat and transfer pictures.

**Nimbuzz** <sup>[44]</sup>

User classification: 3.9 (26,544)

Support for Windows, Mac and mobile devices. Features: chat, contacts list with social presence, voice calls, support for multiple accounts (Yahoo, MSN, AIM, GTalk, Facebook, ICQ, MySpace and SIP-based accounts).

**ooVoo Video Calls** <sup>[45]</sup>

User classification: 3.3 (15,765)

Support for Android, iPhone, PC, Mac and Web. Features: contact list with social presence, voice and video calls and conference video calls.

**MobileVOIP** <sup>[46]</sup>

User classification: 3.2 (2,192)

A simple application that supports SIP accounts, enabling users to make VoIP calls.

**CSipSimple** <sup>[47]</sup>

User classification: 4.4 (3,377)

CSipSimple is an open-source project that uses PJSIP<sup>[48]</sup> native libraries as its communications basic structure. The application currently supports voice calls, call log, integration with native contact list and dialer, chat one-to-one

and support for multiple SIP accounts. The codecs used in voice calls are iLBC, GSM, Speex, PCMA, PCMU and G.722.

### **Palringo Instant Messenger** <sup>[49]</sup>

User classification: 3.3 (7,606)

Support contacts from Facebook chat, MSN, AIM, Yahoo, GTalk, ICQ and Jabber. Features: chat, group chat, file transfer (images, video and voice notes), sharing location with friends.

### **Sipdroid** <sup>[50]</sup>

User classification: 4.0 (8,178)

This application is an emergent open-source project, currently with few features. Supports voice calls, video calls and access to the native contact list. The audio codecs used are G.722, PCMA, PCMU, Speex, GSM and BV16.

### **3CX Phone** <sup>[51]</sup>

User classification: 3.6 (1,718)

This is a freeware application developed by multinational 3CX. The user must have a SIP account as this company doesn't have own servers. The features available in this application are: voice calls, call history, support for multiple accounts and support for several platforms (Windows PC, Android and iPhone). The codecs used in voice calls are G.711, GSM and Speex.

### **Movicha Live Mobile Video Chat** <sup>[52]</sup>

User classification: 2.9 (774)

Support for Android, PC, Mac and it allows voice and video calls, contact list with presence and chat.

### **IMSDroid** <sup>[53]</sup>

User classification: Not applicable

IMSDroid is an open-source project developed to Android devices. It's a 3GPP IMS Client for Android, whose main goal is to show the available features in doubango<sup>[54]</sup> framework.

*"doubango is the world's most advanced open source 3GPP IMS/RCS framework for both embedded and desktop systems. The framework is written in ANSI-C to ease portability and has been carefully designed to efficiently work on embedded systems with limited memory and low computing power and to be extremely portable."* <sup>[54]</sup>

Current IMSdroid version implements partially the RCS Release 3 specifications and it's supposed to be full-compliant with this release in next versions.

The features in this version are:

- Voice calls;
- Video calls;
- Option to intercept outgoing calls from native dialer and native contact list: if option is enabled, the device doesn't allow user to choose which application to use;
- Content Sharing: images and audio files with or without an open session;
- Support for a XML Document Management Server: allows users to have a network address book;
- Doubango framework was carefully developed to work efficiently in devices with processing and memory limitations;
- Audio Codecs - G729AB, AMR-NB, iLBC, GSM, PCMA, PCMU, Speex-NB;
- Video Codecs - H.264, MP4V-ES, Theora, H.263, H.263-1998, H.261;

## Linphone<sup>[55]</sup>

User classification: 3.9 (892)

Linphone is a free SIP VoIP client with a very simplistic interface that uses native contact list as its buddy list and only has voice calls support. Presently supports voice calls, access to the native contact list and call log. The audio codecs used are Speex, GSM, PCMU and PCMA. iLBC availability depends on the ARM processor and Android OS version.

## 2.3. COMPARATIVE ANALYSIS

In this section will be presented a comparative analysis of the applications above mentioned, according to the following features and topics:

- **Voice calls** – ability to make VoIP calls;
- **Video calls** – possibility to make video calls through web traffic. The codecs used information can be found in the section 2.2. of this document, when available;
- **Content sharing** – allows users to exchange content during a call or without a session already open (the type of content is specified for each application that supports this feature);
- **Social presence** – includes a buddy list where user can see the availability and the profile of his contacts;
- **File Transfer** – ability to send/receive files (images, video, audio or others) to/from a friend, usually during a chat conversation;
- **Geolocation** – possibility to share location with friends;
- **Chat 1-to-1** – instant messaging service with another user;
- **Chat 1-to-many** – ability to chat with more than one user. In other terms, group chat where is possible to see which contacts are in the conversation;
- **Network Address Book** – synchronization of buddy list with a server to allow user to access to his contacts from any device where he connects. Applications support several accounts like MSN, Yahoo and similar, which can be consulted in the section 2.2. of this document to a detailed information of accounts supported for each product;
- **Platforms** – devices supported by the application or similar application from the same company, where user can access to his account and information;
- **User classification** – All applications in the Android market have an evaluation by its users, which can go from 1 to 5 stars. This indicator represents this classification. N/A grade (not applicable) means that the application is not available at Android market.

Below is presented a table with the competitors, where are presented the features supported by each one. WIT Mobile Communicator is also presented in the comparison table with the features that currently supports (time of this document creation).

Application	Voice calls	Video calls	Content Sharing	Social Presence	File Transfer	Geolocation	Chat 1-to-1	Chat 1-to-many	Network Address Book	Platforms	User Classification
MSN Messenger	✗	✗	✗	✓	✓	✗	✓	✓	✓	Android, Windows, Linux, Mac OS	4.5
BeejiveIM	✗	✗	✗	✓	✓	✗	✓	✓	✓	Android, iPhone, BlackBerry, iPad, iPod Touch	3.8
IM+ Pro	✗	✗	✗	✓	✓	✓	✓	✓	✓	Android, iPhone, iPad, iPod Touch, BlackBerry, Windows Phone, Windows Mobile, Symbian, Palm OS	3.8
Bria Android	✓	✗	✗	✗	✗	✗	✗	✗	✓	Android, iPad, iPhone, Windows, Linux, Mac OS	3.3
Talkdroid Messenger	✗	✗	✗	✓	✗	✗	✓	✓	✓	Android	4.2
aSip	✓	✗	✗	✗	✗	✗	✗	✗	✓	Android, iPhone	2.8
VoIP Video SIP client	✓	✓	✗	✓	✗	✗	✓	✗	✓	Android, Windows Mobile, Symbian, iPhone, BlackBerry	2.4
KakaoTalk	✗	✗	✗	✗	✓	✗	✓	✓	✓	Android, iPhone	4.5
WhatsApp Messenger	✗	✗	✗	✗	✓	✓	✓	✓	✓	Android, iPhone, BlackBerry, Nokia	4.6
Skype	✓	✓	✗	✓	✗	✗	✓	✓	✓	Android, iPhone, Symbian, Windows, Mac OS, Linux	3.6
Yahoo! Messenger	✓	✓	✗	✓	✓	✗	✓	✓	✓	Android, Windows, iPhone, Web, Mac OS	3.9

eBuddy Messenger	✗	✗	✗	✓	✗	✗	✓	✗	✓	Android, iPhone, Web	4.3
Live Profile	✗	✗	✗	✓	✓	✗	✓	✗	✓	Android, iPhone, BlackBerry	4.0
Trillian Messenger	✗	✗	✗	✓	✓	✗	✓	✗	✓	Android, iPhone, BlackBerry, Windows, Mac	4.4
Meebo IM	✗	✗	✗	✓	✗	✗	✓	✗	✓	Android, iPhone, BlackBerry, Web	3.8
fring	✓	✓	✗	✓	✗	✗	✓	✗	✓	Android, iPhone, Nokia	3.7
Kik Messenger	✗	✗	✗	✗	✓	✗	✓	✓	✓	Android, iPhone, iPad, iPod Touch	4.2
Nimbuzz	✓	✗	✗	✓	✓	✗	✓	✓	✓	Android, iPhone, Windows, Mac, Web	3.9
ooVoo Video Calls	✓	✓	✗	✓	✗	✗	✗	✗	✓	Android, iPhone, Windows, Mac, Web	3.3
MobileVoIP	✓	✗	✗	✗	✗	✗	✗	✗	✗	Android, iPhone, iPad, iPod Touch	3.2
CSipSimple	✓	✗	✗	✗	✗	✗	✗	✗	✗	Android	4.4
Palringo Instant Messenger	✗	✗	✗	✓	✓	✓	✓	✓	✓	Android, iOS, Symbian, Java, Windows Mobile, Windows, Mac OS	3.3
Sipdroid	✓	✓	✗	✗	✗	✗	✗	✗	✗	Android	4.0
3CX Phone	✓	✗	✗	✗	✗	✗	✗	✗	✗	Android, iPhone, Windows	3.6
Movicha Live Mobile Video Chat	✓	✓	✗	✓	✗	✗	✓	✗	✓	Android, iPhone, Symbian, Windows Mobile, Windows, Mac OS	2.9
IMSdroid	✓	✓	✓	✓	✗	✗	✓	✗	✓	Android	N/A

Linphone	✔	✘	✘	✘	✘	✘	✘	✘	✘	Android, iPhone, BlackBerry, WebOS, Windows, Linux, Mac OS	3.9
WIT Mobile Communicator	✔	✔	✘	✔	✔	✘	✔	✘	✔	Android, iPhone, Windows, Mac, Linux, Web	N/A

Table 1 – Comparative analysis of competitors’ features.

- ✔ - Application supports the feature
- ✘ - Application doesn’t support the feature

As we can see in the table there a wide range of communication applications. The most part focus in the instant messaging features, such as chat and group chat. However, there is already a large number of them that allows user to make VoIP calls and video calls.

One application to highlight is IMSdroid, not only due to the wide range of features it has, but mainly because the fact of one of its goals it’s to be full-compliant with RCS, like WIT Mobile Communicator for Android project.





### 3. REQUIREMENTS ELICITATION

This section will present the features of WIT Mobile Communicator for Android.

Since the application has the RCS specification as baseline, the requirements are already defined, so will be presented RCS Releases and its features specification.

For a detailed functional specification and to be able to see the flows between the screens, it's advised to read the full document, **Appendix B – Functional Specification**.

#### 3.1. RICH COMMUNICATOR SUITE

RCS is an initiative supported by GSMA and other companies related with telecommunications, that specifies a set of features based in common standards so that applications developed by different companies are compliant between them.

*“The GSMA RCS Project is a collaborative effort – now supported by nearly 100 operators, vendors and other telecommunication companies – to facilitate the introduction of commercial, IMS based rich communication services for UMTS and CDMA mobile as well as fixed networks.”<sup>[56]</sup>*

When this project was initialized, there were three specification releases. However, in April 2011, emerged a new RCS version, RCS-e. In the following sub-sections are presented the features of each release and a brief description of them.

##### 3.1.1. RCS RELEASE 1<sup>[57]</sup>

###### ENHANCED ADDRESS BOOK

This service feature defines the creation of an enhanced contact list, where you can see additional contact information beyond name and phone number. Some of the available information is:

- **Service capabilities:** list of services that are available for each contact at a given moment;
- **Portrait icon:** photo or image defined by the contact or by the user to represent a contact;
- **Social presence:** allows to see the current state of a contact (online, busy, away, offline);
- **Authorization management:** receiving and sending requests for sharing social presence;
- **Blacklist:** possibility of blocking contacts to stop receiving new invitations from these contacts;
- **Free text and favourite link:** allow users to publish a textual note and a favourite link, which is visible to authorized contacts.
- **Communication history per contact:** the user should see the entire communication history with a particular contact (messages, calls, content sharing).

###### CONTENT SHARING

This feature allows users to share content (images and video) during a session, usually a voice call, and can be initiated at any end point. The end of content sharing doesn't affect the voice call; however, the end of the voice call automatically leads to termination of content sharing.

## FILE TRANSFER

This feature allows the exchange of different content types (files) between two users, whether or not already exists an open session between them. This type of sharing is totally independent of other sessions that may exist between users (for example, voice calls).

## ENHANCED MESSAGING

RCS Messaging specification not only defines some changes to the user interface, but also adds some functionality for greater interaction between contacts.

The list below presents some key changes:

- **Unified Composer for SMS/MMS:** the message composer should be the same for different types of messages and the bearer technology is chosen by device during message composition according to the content of the message.
- **Threaded view conversations:** conversations and messages should be presented in threaded-mode, where you can see the whole conversation with a particular contact;
- **Chat service:** the application must provide a chat service with two types of chat available: one-to-one and one-to-many (group chat).

---

### 3.1.2. RCS RELEASE 2 <sup>[58]</sup>

The main purpose of RCS Release 2 is to provide the user the ability to access features from a wider range of devices.

#### BROADBAND ACCESS TO RCS FEATURES

Allows user to access to rich communication services from devices connected to the network via broadband access, for instance from a computer.

#### MULTI-DEVICE ENVIRONMENT

The goal of this feature is to provide support for multiple devices. Thus, it allows the user to have more than one device connected to the RCS services, choosing which he wants to use (for example, answer a call or reply a message).

#### NETWORK ADDRESS BOOK

The aim of Network Address Book is to provide users a centralized mechanism that allows them to manage contacts information. Thus, users may have contacts information updated regardless of the device they are using.

#### PROVISIONING AND CONFIGURATION OF RCS DEVICES/CLIENTS

The application should be automatically configured and transparently to the user, since it's not expected user to do manual settings to be able to enjoy the services.

---

### 3.1.3. RCS RELEASE 3 <sup>[59]</sup>

The main purpose of this release is to make some improvements in features defined in previous releases, so there aren't significant differences.

### BROADBAND ACCESS ENHANCEMENT

At this point are defined some guidelines corresponding to the use of broadband access connected-device as primary device. However, whenever there is a mobile device connected it will always be considered the primary device.

### CONTENT SHARING ENHANCEMENT

It's intended to provide the ability to sharing content with a contact without an open session. Thus, will be opened an uni-directional session for content exchange.

### SOCIAL PRESENCE INFORMATION ENHANCEMENTS

The improvements defined in release 3 to the social presence are based in the following points:

- **Personalized Invitation:** requests sent by user to his contacts may contain a personal message defined by him;
- **Geolocation:** users can share information about their location with their friends.

### MESSAGING ENHANCEMENT

In this section are defined some changes regarding the use of message service during the use of multiple devices:

- SMS and MMS are delivered to the primary device only;
- Messages sent from a broadband access client will not be part of the conversational view on mobile access device (and vice versa).

It also defines the ability to see the contacts presented in a group communication (Group Chat).

### NETWORK VALUE ADDED SERVICE (NVAS)

It's intended to provide user some services that take advantage from the network, like the ability of changing images when sent through content sharing; or text translation in real time during a chat conversation.

---

#### 3.1.4. RCS-E ADVANCED COMMUNICATIONS <sup>[60]</sup>

RCS-e complements the current RCS Release 2 specification. So, it includes all the features and specifications defined in that release with some optimizations. These optimizations and changes are presented below.

### REGISTRATION

The client registration in network can be made in two ways: manually and automatically. The first one only happens when it's the first time user is logging in. After that, the application will automatically register in the network.

## CAPABILITIES DISCOVERY

The fundamental mechanism that enables RCS-e is capability discovery. The RCS-e contacts in address book are identified and user can access to which features are supported by each contact. The capabilities update must be done at several moments:

- Every time an action is performed with the given contact;
- Periodically;
- When user change any detail or information about the contact;
- When capabilities change.

## CHAT

The basis of the chat feature is the same of the RCS Release 2, however there are some improvements to do in RCS-e:

- **IM store and forward** – the server is able to store sent messages and deliver them to the recipient when he becomes online;
- **Delivery and display notifications** – the users receive a notification when the message sent was received and when it was displayed in the screen of the recipient;
- **Local black list** – the application has a local blacklist for the messages not desired.

## 4. ARCHITECTURES AND TECHNOLOGIES

WIT Mobile Communicator for Android belongs to WIT Communications Suite, so there is already a basis architecture shared by all products of this suite.

In this section will be introduced the background knowledge needed to a better understanding of the application and its structure. After that, it will be presented the product architecture and its description.

To a better comprehension of the technologies and the global architecture of WIT Communications Suite, the **Appendix C – High-Level Architecture** should be consulted.

### 4.1. BACKGROUND KNOWLEDGE

Before discuss the application architecture, it's necessary to understand the fundamentals of Android platform and the technologies and libraries that will be used.

#### 4.1.1. ANDROID APPLICATIONS FUNDAMENTALS

Android applications have some essential components<sup>[61]</sup>:

##### ACTIVITIES

Each screen of the application represents an activity, so the flow between several screens implies the change of the active activity. The flow between activity works like a stack. The simplest way to understand their behavior is with a use case example.

##### Example: Communication's details

In the application to develop, the user wants to see the communications' history with a contact. Assuming that user is already logged, to do that he needs to pass through some steps:

- **1** – Open the contacts list;
- **2** – Select the desired contact, it goes to contact profile screen;
- **3** – Select the option corresponding to the communications history of the contact and the log screen will appear;
- **4** – Back to contact's profile screen;
- **5** – Back to the contacts list;

And the screens are:

- **A** – Contact list screen;
- **B** – Contact's profile screen;
- **C** – Communications history screen.

Now, let's see a figure representing each of these steps, with the activity's default behavior.

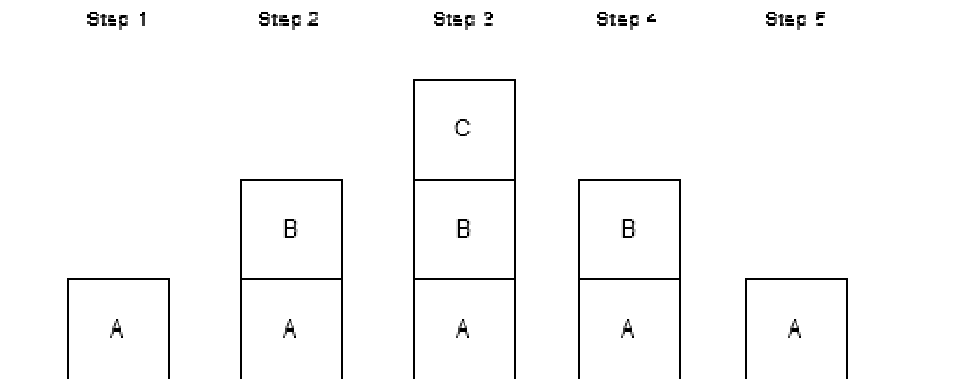


Figure 3 - Activity flow representation

As it's possible to see by clicking in device's back button, user will back to previous activity. It is possible to modify this behavior programmatically and obey the activity to close before change to another, according with activity's lifecycle <sup>[62]</sup>.

## SERVICES

A service doesn't have a visual interface. It runs in background and unlike the activities, operating system cannot free the corresponding memory.

## BROADCAST RECEIVERS

These components receive broadcast messages and react according to the type of the received message.

## CONTENT PROVIDERS

Allow application to store data that would be accessible from another application, i.e., it is like a database accessible from a URI.

## INTENTS

Intents <sup>[63]</sup> are asynchronous messages used to communicate with the activities, services and broadcast receivers. These components name the actions being requested and may carry additional information as extra arguments. Intents have a great importance in Android applications, since they define the flow of the entire application.

## INTENT FILTERS

Intent filters <sup>[63]</sup> are associated to others components (activities, services and broadcast receivers) and are used to inform the Android application which components are able to handle a given intent. The intents may be filtered by action, category and data.

### 4.1.2. COMMUNICATOR LIBRARY

Communicator library is shared by the products that belong to WIT Communicator Suite. It is composed by a set of structures named managers, which each of them is responsible to handle a particular set of events. For example, login manager is responsible to handle the events of register and unregister of the client on the server; the messaging manager is responsible to send and receive the messages and the management of the chat rooms and sessions.

A set of external libraries are also part of Communicator, some of which are developed by external companies and other developed internally at WIT Software. These libraries are developed in C and C++ programming languages.

The figure below represents the architecture of Communicator Library.

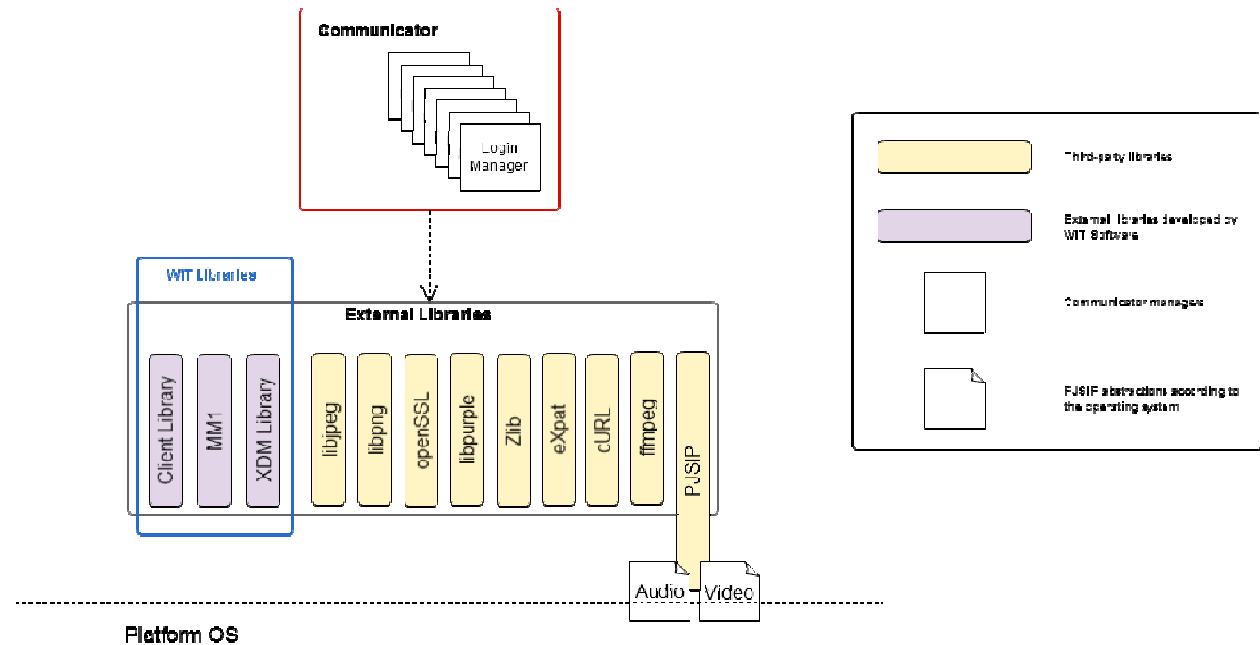


Figure 4 – Communicator Library architecture and its external libraries.

Some of these libraries will not be used directly from application, however it is necessary include them since they are needed to compile Communicator Library.

Will be described the most relevant libraries to WIT Mobile Communicator and the function of each.

#### PJSIP

PJSIP is a SIP stack that supports a set of features defined by SIP standards. This library provides too an user agent library (PJSUA-LIB) which has a set of Application Programming Interfaces (API) that allows external applications to easily access to the available features.

The available APIs are:

- **PJSUA-API Basic API** – provides access to the basic operations in PJSUA like creation, initialization, and configurations (for example, logging);

- **PJSUA-API Signaling Transport**—API for managing SIP transports like creation and configurations, and supports multiple transport instances;
- **PJSUA-API Accounts Management** – API for managing configurations relative to user accounts. The API supports several accounts and each account has information about user like credentials and presence data;
- **PJSUA-API Calls Management** – management of calls and information related to them;
- **PJSUA-API Buddy, Presence and Instant Messaging** – interface to manage all features related to presence and IM.
- **PJSUA-API Media Manipulation** – management of media, from device sounds to audio in calls.

In this library will be necessary to implement the specific abstractions to the Android platform relative to audio and video, since it differs from platform to platform.

## XDM LIBRARY

This library is used to manage the XML documents, according to OMA XDM<sup>[64]</sup> standards, related to contacts information retrieved from the server. This is a library developed internally by WIT Software.

## FFMPEG

Ffmpeg<sup>[65]</sup> is used to compress the frames captured on video calls with the desired codec. It is also used to decompress received frames before present them to the user during a video call too.

---

### 4.1.3. TECHNOLOGIES

In this section will be presented the technologies used in the application development. The most part of these technologies were already defined due to Communicator Library. Since this project uses that common library, it is recommended to use the same technologies in some cases, for instance, in streaming protocols and in the access to the network address book.

## ANDROID SDK

To develop applications for Android devices it is necessary use the Android SDK that provides a large number of APIs and functions that facilitate the programming for these devices. The final application is presented in a file with the extension “.apk”, which is directly installable in the device.

Android already has several versions, currently in the 3.1. Initially, it was intended that the application was developed for devices 1.6 and higher. However, due to some limitations in APIs of the versions under 2.0 was decided to develop the application to devices higher than that.

Android SDK also includes an emulator that allows programmers to test their applications, and a debugging tool called Dalvik Debug Monitor Server (DDMS) that facilitates the debugging of an application with a detailed logging service.



## ANDROID NDK <sup>[66]</sup>

As mentioned above it will be necessary use common libraries developed in C and C++, and Android programming language is Java based, so it will be necessary to use a tool that allows the use of native code in Android applications.

To solve this issue there is a technology called Android Native Development Kit (Android NDK) that allows programmers to implement part of the Android application using native code. The benefits of this are mainly the ability to reuse existing code and the possibility to optimize the performance of the application.

This is a key point in Android since the devices are limited in terms of processing and memory, and Java programming language is a bit heavy in comparison with other programming languages. If some critical functions with a high use of memory and processor are developed in Java, this may reduce the application performance, so it is a better choice to implement those functions in C or C++ and make use of Android NDK to access them.

## SESSION INITIATION PROTOCOL (SIP)

Is a signalling protocol used for establishing and controlling multimedia communication sessions, such as voice and video calls over IP.

*“(SIP) ... an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.”* <sup>[67]</sup>

## REAL-TIME TRANSPORT PROTOCOL (RTP) <sup>[68]</sup>

A standard protocol for delivering audio and video over IP networks. RTP is used in communication and entertainment systems that involve streaming media, such as telephony and video conference applications. It is used with another protocol that monitors transmission statistics and quality of service (QoS) and helps in synchronization of multiple streams – **RTP Control Protocol (RTCP)**.

In WIT Mobile Communicator this protocol is used to send/receive audio and video packets after a session has been established through the SIP protocol.

## 4.2. APPLICATION ARCHITECTURE

First, will be described briefly the global architecture of WIT Communications Suite to a better contextualization of the product. It is again highlighted the importance of consulting **Appendix D – High-Level Architecture** to a detailed description of these architectures and the technologies mentioned above.

#### 4.2.1. WIT COMMUNICATIONS SUITE

The products belonging to the WIT Communications Suite are developed with the intent to share common services and servers. Thus, these products are linked by core layer, and the general structure is shown in the figure below.

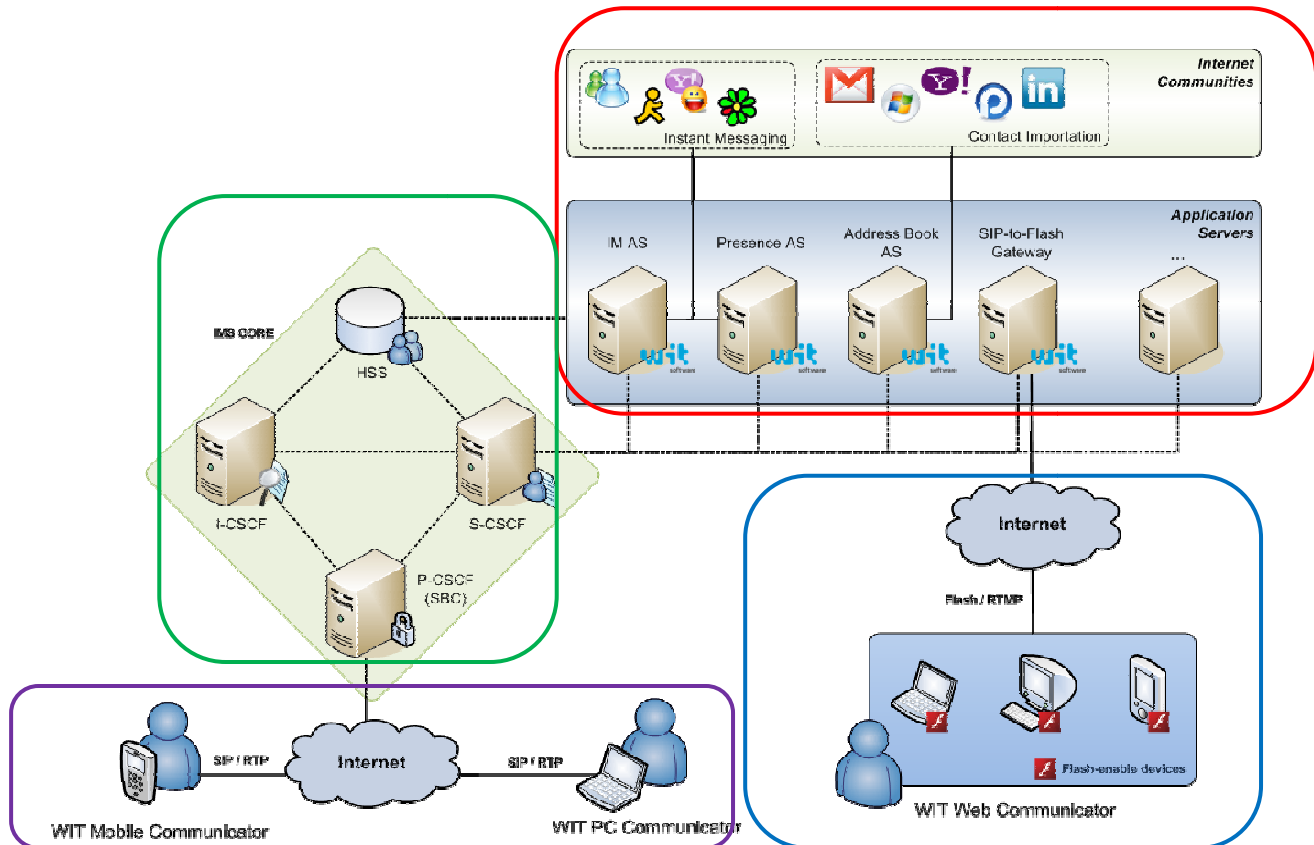


Figure 5 - WIT Communications Suite architecture

The figure above can be divided in four main sections:

- IP Multimedia Subsystem<sup>[69]</sup> Core (green)
- Application Servers (red)
- Clients using SIP/RTP (purple)
- Clients using other protocols (blue)

The clients using SIP/RTP as protocol to connect to the system, connects directly with P-CSCF, that will redirect the requests to the corresponding nodes to be handled. The clients using other protocol connects to a gateway which makes the protocol translation and then routes requests to IMS core.

WIT Web Communicator is an example of client using Flash. The internship scope is inserted in the Clients using SIP/RTP, specifically in the WIT Mobile Communicator clients.

Thus, the remaining sections will not be addressed since they are not the focus area of the project.

#### 4.2.2. WIT MOBILE COMMUNICATOR

WIT Mobile Communicator is a set of products belonging to WCS developed to several mobile devices. The application to be developed is the expansion of WMC to Android devices. The figure below represents the high-level architecture of these products.

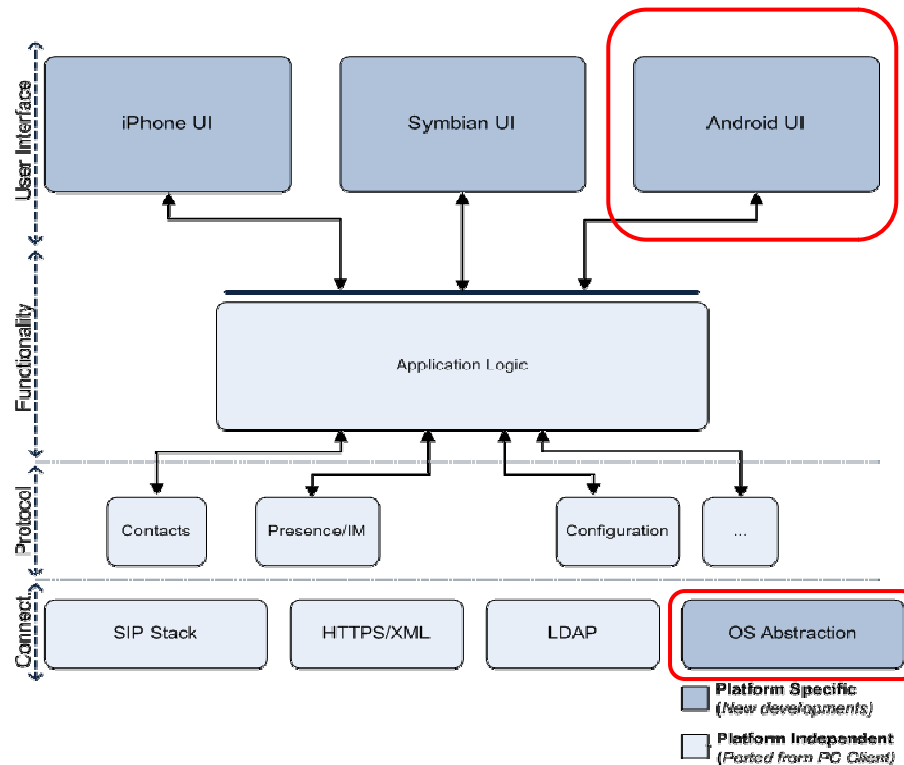


Figure 6 - High-level architecture of WIT Mobile Communicator applications

In the application development will be used a common library (Communicator Library) where are defined the logic functionalities. This library already exists and defines all low-level operations related with SIP negotiations, communication with server, contacts management, among others.

This library must be generic enough to be integrated with the various applications, independently of the operating system or the device.

Thus, as we can see in the presented figure, for each device is necessary to develop the user interface and the OS abstraction. So, these will be the parts to be implemented in the WIT Mobile Communicator for Android (red above).

For a better architecture perception, then will be presented a detailed description of the application to develop.

### 4.2.3. WIT MOBILE COMMUNICATOR FOR ANDROID

The application can be divided in two main parts: native code and graphic user interface (GUI).

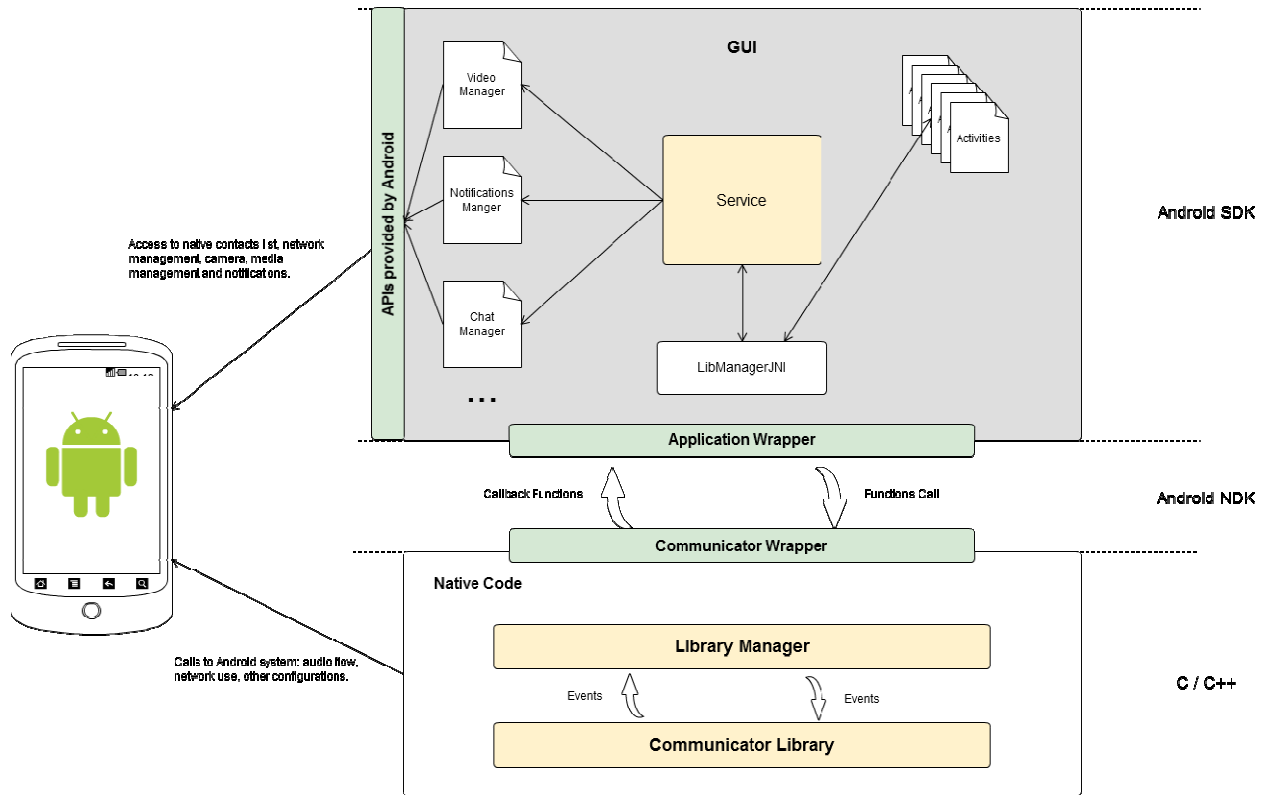


Figure 7 – WIT Mobile Communicator for Android: detailed architecture

### APPLICATION LIBRARIES

This part of application consists of a set of libraries in native code (C and C++): Communicator Library and Library Manager. Communicator Library architecture is specified in section 4.1.2. and since it works by events the Library Manager has the aim to convert the events received from Communicator to functions callbacks in wrapper; and the function calls to events to be handled by Communicator Library.

At this level are actions that take direct access to Android hardware, like audio capture and reception, network and other configurations.

### GRAPHIC USER INTERFACE

In this side of the application will be implemented the changes between screens and the whole service behind that. This is the Android logic layer.

The components of this part of the application are:

- **Service** – it will be initialized when application starts and is responsible to make the core operations, including the management of the application fundamentals.
- **LibManagerJNI** – is the interface where the Library Manager will connect to send information to the upper layer and where the application will call the functions in the native code layer.
- **Activities** – is the set of screens visible to the user. These include the changes between the screens and the management related to them.
- **APIs provided by Android** – Android provides a set of APIs that intends to facilitate the use and the access to device-related configurations, and allowing a high integration between the application and the device. E.g., contacts API allows to access to the native contacts database, providing information about contacts, their number phone and other information. Other APIs allow camera management, screen and networks management, among others.
- **Notification Manager and other managers** – Accessing directly to the Android APIs route the requests from application to device and vice-versa.

## Service and Activities Flow

The application service is initialized when application starts to be ready to handle any operation needed by the activities. The service performs the communication with the native functions.

Each activity has a broadcast receiver that allows activities and service to communicate between them through intents. The most part of broadcast receivers are defined with the activity creation. Nevertheless, it is necessary defined some of them in a global way in the start of the application, to get a higher integration with the device. For instance, the application will be able to intercept the outgoing calls performed from the native contact list. To do that it is necessary to filter the outgoing call intent generated by the device, allowing user to choose which application he wants to use to make the call. If the broadcast receiver is registered inside the activity, it will only be enabled when the activity is active, so in this case it would not be able to handle the intent.

## Handle events reception

Not only GUI is able to call functions in native libraries. The opposite it is possible too. Native libraries need to send events to the GUI by calling callback functions, for instance, in a new incoming call notification or when call status is updated.

To receive these notifications, the GUI has an event receiver (LibManagerJNI) that is always listening for new events and then is responsible to forward them to the respective handlers. This is done in parallel with other application operations.

## Notifications

Android has an own notification system that allows applications present notifications in the home screen of the device. This notification system has a status bar in the top of the screen, and is able to warn users that have new notifications. Rich Communicator for Android will make use of this native notification system in some situations. For instance, when the user receives a new request for sharing social presence information, one notification will be presented in this status bar, warning the user with device vibration. From this notification list, user will be able to access directly to the application by taping the notification message. Other example is an icon that there are always present in the top of the screen, when user is logged in the application, informing him of that.

## APIs provided by Android

Android SDK has a set of APIs that allows application to make use of some native features, in an easier way. WIT Mobile Communicator will use these API to get a higher integration with the device. One of the APIs used is the Contacts API that allows to access to the contacts information from the device's address book.

The application will be integrated with the contacts list in two ways: the ability to make calls directly from the native contact list; and native contacts will be presented in the application's contact list too.

---

### 4.2.4. WRAPPER

To connect the both layers it is necessary to have a wrapper, which makes the conversion of native functions in Java-perceptible functions. The wrapper works like an interface in each layer, and Android NDK will make the connection between that two layers.

In the libraries wrapper, there will be a conversion from the Java data types to C data types, while the GUI wrapper will make the opposite conversion. It is also necessary represent all the C structs used in the connection as Java objects.

For instance, assuming that exist a struct user in C that is comprised by name, phone and address, and it is necessary to send it to the GUI. For the GUI layer recognize the data type, there must be a user class in Java that is also comprised by name, phone and address. Thus, in the wrapper will be defined that the user struct in C corresponds to the user class in Java.

## 5. PROJECT MANAGEMENT

Project management has a high importance in a software development project. It allows to planning the tasks to do and the deadlines to meet, to take more control and better guidance on the project. In this section will be presented the methodology used in the project, the assigned tasks and the deadlines for each of them. It will be also done a comparison between the initial and the final planning of the tasks, the differences between them and why those discrepancies occurred.

Also in this section, it will be presented the risks inherent to the project and described the method of testing the software quality used during the development of WIT Mobile Communicator for Android.

### 5.1. SOFTWARE DEVELOPMENT METHODOLOGY

The methodology used in this project is based in Scrum<sup>[70]</sup>. Scrum is an agile framework that allows planning a project with the ability to change requirements or adapt new features easily.

Sometimes, companies use a rigid methodology, like “*The Waterfall*”, in software development projects. In this type of planning is necessary define the tasks and estimate its duration that will be done in the next several months. Humans are not perfect estimating time, so it is likely that this “guess” fails. Furthermore, if during the development new ideas come, there is no space or time to change or add new features. The plan is already defined and that is the way it has to go.

Scrum is a different approach for planning projects, where team can make adaptations and the priority of requirements can change.

---

#### 5.1.1. SCRUM BASICS

In Scrum there are three primary roles:

**Product Owner** – is responsible to define all the features and inputs of the project and to create a prioritized list with the desired features.

**ScrumMaster** – the main goal of ScrumMaster is to protect the team from outside interference, ensuring the success of the team. He will guide the team in Scrum practices.

**Team** – it is typically composed of 5 to 10 elements and may include developers, analysts, testers and designers with a high autonomy; however the number and the type of the elements may change from team to team. The team is responsible for developing the product, however they may also provides ideas to the Product Owner.

---

#### 5.1.2. HOW DO SCRUM WORKS?

At the beginning of the project, the project owner defines the set of features that application should have, and create a list with them – **Product Backlog**. It is dynamic, since items can be added or deleted at any time during the project. It is assigned a priority to each item of product backlog, where the features with high-priority are completed first.

The development is structured in cycles of work called **Sprints** with a fixed duration (in this case 2 weeks). At the beginning of each sprint, team selects the priority items from product backlog, to complete during the sprint.

Daily, the team has a brief discussion about the work done until that moment, and update a chart corresponding to that sprint (**Sprint Backlog**).

At the end of each sprint a meeting is scheduled to review the work done during that sprint, and at the beginning of the next sprint, new tasks are assigned to the team elements.

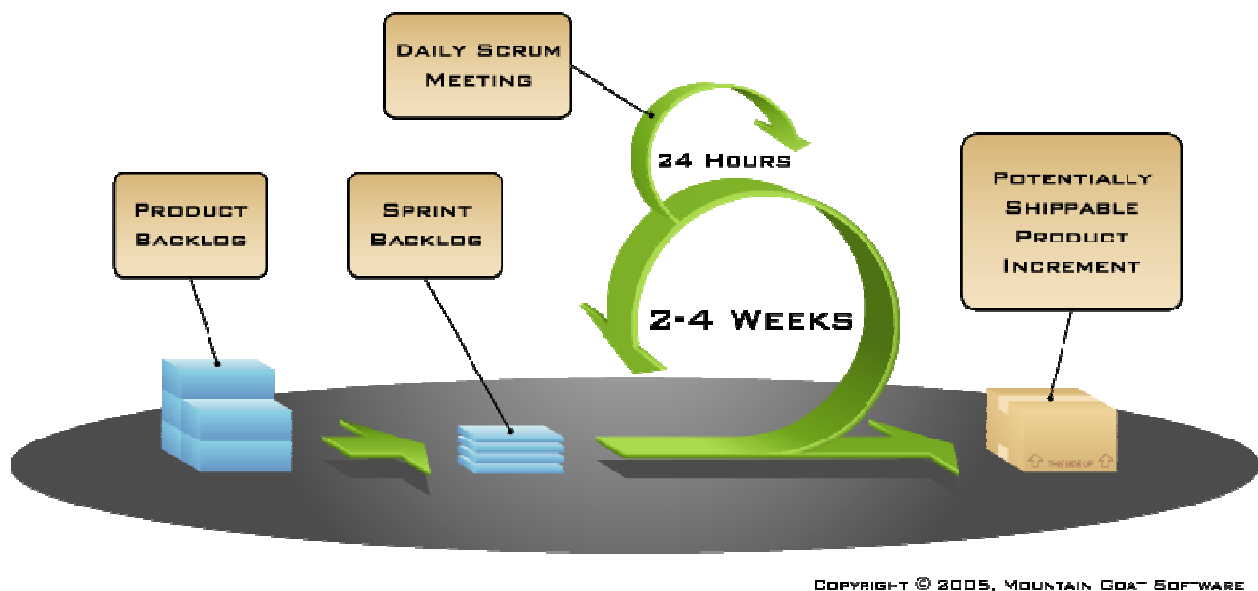


Figure 8 – Scrum methodology

### 5.1.3. VARIANT USED

It is important to note some detailed in the variant that differ from the Scrum above mentioned:

**Product Owner and ScrumMaster** - although not desirable, in this project the role of product owner and the ScrumMaster are played by the same person – the supervisor of the project.

**Product Backlog** – the idea of being product owner to develop the product backlog, it is not entirely applicable, since the features to develop are already defined in the RCS specification and the product backlog was performed by the team. However, the items priority and the new features needed are still defined by product owner according to the needs of potential clients.

**Sprint Review** – the meeting scheduled at the end of the sprint is not only used to review the work done until that moment, but is also used to define the tasks to the next sprint.

**Start the Next Sprint** – every new sprint, the tasks and its estimated time are discussed in a team meeting. In this meeting are defined the tasks to do during the next sprint.



---

#### 5.1.4. ROLES

Following are presented the roles and the corresponding owners.

**Product Owner** – Pedro Pereira/Nuno Costa

**Scrum Master** – Nuno Costa

##### **Team**

- João Raimundo
- Rui Teixeira
- Sandra Mendes

---

#### 5.1.5. PRODUCT BACKLOG

The table 2 represents the product backlog, where are defined the tasks done during the internship, a brief description of each one, the estimated time to develop the feature, the priority initially assigned to the tasks and the sprint number where the task was finished.

All the tasks presented below correspond to the ones done by the author.

The tasks below the dashed line correspond to future work, not done during the project.

Category	Task	Description	Estimate	Owner	Priority	Sprint
Documentation	<b>State of the Art</b>	Study of the market	120	Sandra Mendes	Medium	-
Documentation	<b>Functional Specification</b>	Analysis of application's features	32	Sandra Mendes	High	-
Documentation	<b>Risk Plan</b>	Analysis of the risks that project is subject	32	Sandra Mendes	Medium	-
Documentation	<b>RCS features</b>	Analysis of RCS features	24	Sandra Mendes	High	1
Documentation	<b>High-Level Architecture</b>	Create architecture document	24	Sandra Mendes	Medium	2
Documentation	<b>Product Backlog</b>	Define the product backlog of the project	16	Sandra Mendes	High	2
Technologies	<b>Code Experience</b>	Fundamentals of Android, JNI and Android NDK	40	Sandra Mendes	High	2
Libraries	<b>Compile PJSIP</b>	Compile PJSIP with Android NDK	24	Sandra Mendes	High	3
Application	<b>Android Application Basics</b>	Basic structure of the application	24	Sandra Mendes	High	3
Application	<b>Application Service</b>	Development of core service	24	Sandra Mendes	High	3
Application	<b>Login and Logout</b>	Implements authentication	8	Sandra Mendes	High	3
Voice Calls	<b>Voice Call Layouts</b>	Development of voice calls screens	8	Sandra Mendes	High	4
Voice Calls	<b>Service and Layout Integration</b>	Communication between service and layouts	16	Sandra Mendes	High	4
Voice Calls	<b>Outgoing Voice Call</b>		8	Sandra Mendes	High	4
Voice Calls	<b>Incoming Calls</b>		8	Sandra Mendes	High	4
Voice Calls	<b>Enhanced GUI</b>	Call status, counter and other details	8	Sandra Mendes	Medium	4
Voice Calls	<b>Account Settings</b>	Network profile and settings	5	Sandra Mendes	Medium	4
Voice Calls	<b>Status Bar Notification</b>	Login notification in Android home screen	8	Sandra Mendes	Medium	4
Voice Calls	<b>Calls from contact list</b>	Ability to make call from native contact list	24	Sandra Mendes	High	4
Testing	<b>Voice Calls Tests</b>	Tests and fixes in voice calls	16	Sandra Mendes	Medium	5
Wrapper	<b>PJSIP wrapper</b>	Video calls wrapper	60	Sandra Mendes	High	5
Video Calls	<b>Capture Video</b>	Capture video from camera device	20	Sandra Mendes	High	6
Video Calls	<b>Video Call Layouts</b>	Development of video call screens	8	Sandra Mendes	High	6
Video Calls	<b>Receive and Handle Events</b>	Handle events received from native library	16	Sandra Mendes	High	6
Video Calls	<b>Video Preview</b>	Present video captured in the device's camera	16	Sandra Mendes	High	6
Video Calls	<b>Frames codification</b>		16	Sandra Mendes	High	6

Video Calls	<b>Codecs and Audio issues</b>		16	Sandra Mendes	Medium	7
Video Calls	<b>Frames reception</b>		24	Sandra Mendes	High	7
Testing	<b>Video Calls Tests</b>	Tests in video calls	24	Sandra Mendes	Medium	7
Video Calls	<b>Samsung issues</b>	Added task: video issues on Samsung devices	8	Sandra Mendes	Medium	7
Video Calls	<b>Camera Rotation</b>	Issue: camera rotated 90º	40	Sandra Mendes	Medium	8
Video Calls	<b>Contact list issue</b>	Added task	16	Sandra Mendes	High	8
Video Calls	<b>Technicolor issues</b>	Added task: issues on tablets Technicolor	16	Sandra Mendes	Medium	8
Video Calls	<b>Crash after end calls</b>	Added task	40	Sandra Mendes	High	9
Video Calls	<b>General fixes</b>	Added task: General issues fix in video calls	24	Sandra Mendes	High	9
Testing	<b>Android - iPhone tests</b>	Added task	8	Sandra Mendes	High	9
Video Calls	<b>Video Preview</b>	Support under 2.2	24	Sandra Mendes	High	9
Video Calls	<b>Rotate frame display</b>		40	Sandra Mendes	High	10
Documentation	<b>Intermediate Report</b>	Intermediate report documentation	160	Sandra Mendes	High	12
Video Calls	<b>Video Quality Tests</b>	Testing and improvement of video quality	40	Sandra Mendes	High	13
Testing	<b>Social Presence Tests</b>		8	Sandra Mendes	Medium	15
Testing	<b>Contacts Management Tests</b>		8	Sandra Mendes	Medium	18
Settings	<b>Settings Screen</b>	Menu and configurations	16	Sandra Mendes	Medium	14
Chat 1-to-1	<b>Chat one-to-one screens</b>		32	Sandra Mendes	High	14
Chat 1-to-1	<b>Sending Messages</b>		24	Sandra Mendes	High	14
Chat 1-to-1	<b>Messages Reception</b>		24	Sandra Mendes	High	15
Chat 1-to-1	<b>Is Typing feature</b>	Shows when contact is typing something	16	Sandra Mendes	Medium	15
Chat 1-to-1	<b>Notifications</b>		16	Sandra Mendes	Low	15
Chat 1-to-1	<b>Smileys</b>		16	Sandra Mendes	Medium	16
Chat 1-to-1	<b>Chat Tests</b>		16	Sandra Mendes	High	18
Chat 1-to-1	<b>Chat wrapper</b>	Wrappers and functions between Android and CommLib	40	Sandra Mendes	Medium	20
File Transfer	<b>Send file from chat</b>	Choose the file and prepare to send it	24	Sandra Mendes	High	17
File Transfer	<b>Preview file to send</b>		24	Sandra Mendes	Medium	18
File Transfer	<b>Receive file</b>	Receive a file and preview it	32	Sandra Mendes	Medium	17
File Transfer	<b>File Transfer tests</b>	Test and fixes	24	Sandra Mendes	High	18
Chat 1-to-1	<b>Persistent chats</b>	Save current chats between sessions	16	Sandra Mendes	High	16
RCS-e: Planning	<b>Planning</b>	Requirements analysis and planning	24	Sandra Mendes	Medium	19

		tasks				
RCS-e: Migrate project	<b>RCS-e project creation</b>		24	Sandra Mendes	High	19
RCS-e: Screens	<b>Application Layouts</b>		40	Sandra Mendes	High	20
RCS-e: Structure	<b>Application Structure</b>		16	Sandra Mendes	High	19
RCS-e: Authentication	<b>Authentication</b>	Manual and automatic authentication	24	Sandra Mendes	High	20
RCS-e: Social Networking	<b>Publish Facebook</b>	Publish a post on Facebook	32	Sandra Mendes	Medium	21
RCS-e: Dialer	<b>Dialer feature</b>		24	Sandra Mendes	Medium	21
RCS-e: User profile	<b>User profile information</b>		24	Sandra Mendes	Medium	21
Demo	<b>Demos Preparation</b>		120	Sandra Mendes	High	-
Testing	<b>Test application</b>	Prepare final tests and testing	100	Sandra Mendes	High	22/23
Testing	<b>Stabilization</b>	Application stabilization and small changes and fixes	60	Sandra Mendes	High	24
Documentation	<b>Final Report</b>	Internship final documentation	40	Sandra Mendes	High	24
Chat 1-to-many	<b>Chat group screens</b>		20		Medium	
Chat 1-to-many	<b>Chat group feature</b>		50		Medium	
Content Sharing	<b>Image Sharing</b>	Share an image during a voice call	40		Medium	
Content Sharing	<b>Video Sharing</b>	Share a video during a voice call	40		Medium	
Messaging	<b>Unified composer</b>		50		Medium	
			2149			

Table 2 – Product backlog of WIT Mobile Communicator for Android.

### 5.1.6. SPRINT BACKLOG EXAMPLE

		Start Date		17-Jan-11		17-Jan-11		18-Jan-11		19-Jan-11		20-Jan-11		21-Jan-11		22-Jan-11	
		Insert New Line		Insert New Line At End		Day 1		Day 2		Day 3		Day 4		Day 5		Day 6	
Work item ID	Sprint Work item Description	Driver	Status	Pri	Initial	Spent	Left	Spent	Left	Spent	Left	Spent	Left	Spent	Left	Spent	Left
1	High-level architecture document	sandra.mendes		1.0	8	0	8	0.7	7.3	0	7.3	7.9	0	0	0	0	0
2	Functional specification document	sandra.mendes		1.0	16	3.8	12	7.6	4.6	4.6	0	0.6	0	0	0	0	0
3	Product Backlog Changes	sandra.mendes		1.0	3	0	3	0	3	0	3	0	3	0	3	0	3
4	Intermediate report elaboration	sandra.mendes		1.0	28	0	28	0	28	0	28	0	28	9.8	18	2	16
5	WIT Internship Presentations	sandra.mendes		1.0	8	0	8	0	8	0	8	0	8	0	8	0	8
6	Demo Vodafone preparation	sandra.mendes		1.0	5	0	5	0	5	4.7	0.3	1.5	0	0	0	0	0
7	Video quality analysis	sandra.mendes		1.0	16	5.8	10	0	10	0	10	0	10	0	10	0	10
Totals						9.6	74.4	8.3	66.1	9.3	56.8	10	49.2	9.8	39.4	2	37.4

Figure 9 – Sprint backlog example.

This is part of a sprint backlog. It is created by removing tasks from product backlog, then estimated time is calculated and the task is added to the sprint backlog. This cycle repeats until the sum of task's estimated time is the time of sprint (in this case 80 hours). Every day the time spent in the tasks is filled, and it is automatically subtracted from the total estimated. The tasks should be simple and relatively short, in order to perform a better time estimation, and to be possible to see the progress of the tasks along the sprint. Each sprint lasts for 15 days and a new sprint backlog should be created for each sprint.

## 5.2. PLANNING

During the project development there were some changes to the plan due to several reasons, for example, changes on requirements and tasks that were more complex than expected, increasing the task duration.

Thus, in this section it will be presented the initial and the revised planning, the discrepancies between them and the reasons to those differences.

### 5.2.1. INITIAL

The initial goal, as presented on the internship proposal, was to develop an Android application that was an expansion of WIT Communications Suite and was full-compliant with RCS Release 3.

Thus, the following tasks were planned for the 1<sup>st</sup> and 2<sup>nd</sup> semesters.

#### First Semester

This semester was focused in research, study of the market and competitors as well as defining and understanding more clearly the requirements needed and how to implement them. It was also intended to develop the features corresponding to the RCS Releases 1 and 2.

The table with the detailed tasks is presented below.

<b>Initial documentation of the project</b>	Detailed analysis of RCS features State of the Art and competitors' analysis Product Backlog elaboration
<b>Prototyping some features</b>	-
<b>RCS Release 1 features development</b>	Enhanced address book Content Sharing File Transfer Enhanced Messaging
<b>RCS Release 2 features development</b>	Multi-device environment Network Address Book
<b>Internship intermediate documentation</b>	Intermediate Report Intermediate Presentation

Table 3 – Initial planning for the first semester.

## Second Semester

The final part of the internship aimed to develop RCS Release 3 features, to prepare the solution to demos to clients and to prepare the final documentation of the internship.

<b>RCS Release 3 features development</b>	Broadband Access Enhancement Multi-device Enhancement Content Sharing Enhancement Social Presence Information Enhancement Messaging Enhancement Geolocation Enhancement Network Value Added Service Provisioning and Configuration of RCS devices/clients
<b>Preparing solution to demos</b>	-
<b>Internship final documentation</b>	Final Report Final Presentation

Table 4 – Initial planning for the second semester.

### 5.2.2. REVISED

There were several changes to the planning, so below is presented the actual schedule of tasks followed by a brief explanation of each. This revised planning only takes in account the tasks assigned to the author.

## First Semester

<b>Initial documentation of the project</b>	Detailed analysis of RCS features State of the Art and competitors' analysis Product Backlog elaboration System Architecture document Functional Specification
<b>Technologies Experience</b>	Android SDK Fundamentals Java Native Code Android NDK
<b>Communicator Library</b>	Communicator Library structure overview Compile external libraries with Android NDK
<b>RCS Release 1 features development</b>	Native address book Voice calls Video calls

<b>RCS Release 2 features development</b>	Multi-device environment
<b>Internship intermediate documentation</b>	Intermediate Report Intermediate Presentation

Table 5 – Revised planning for the first semester.

In the first semester the internship started by analyzing RCS features, followed by a market study and competitors' analysis. It were elaborated some documents relevant to the project development such as product backlog and architecture document.

It was necessary to study the technologies that would be used, as well as Communicator Library and its external libraries.

The features developed in this first semester were authentication, a contact list based on native contacts and voice and video calls with these contacts. These features are compliant with the other clients belonging with WIT Communications Suite.

The discrepancies between initial and revised planning, namely the small number of features developed, is due to the fact that some difficulties have been encountered over the developments. A brief description of these challenges is presented below.

**Inexperience on development platform and technologies** – it was necessary to invest some time in learning the used technologies. It was the first time that this platform was used, so it was needed to do some code experiences and some tutorials mainly about Android projects, Android NDK and Java Native Code.

**Integration with Communicator Library** – the lack of knowledge about this library was a key point in the delay of this project. To use this library and its external sources it was necessary to compile all libraries within Android project. To do that, the external libraries needed to be compiled separately, using flags and alternative code to the code that was not supported by Android platform.

**Use of three programming languages** – the use of Java, C and C++, and the need to use a wrapper to connect them, implied that it was necessary to invest considerable time in learning and developing the function needed to convert the code through Android NDK.

**Variety of Android devices** – the wide range of these devices and the differences between the various manufacturers, raised some challenges in the voice and video calls development. The use of camera and audio hardware differs from device to device and the API available to access to the camera resources (such as frames captures and their handling) is somewhat limited.

**Dynamic environment** – since the beginning of the project the interest of a client influenced the course of developments, since the features and the progress was made taking into account its feedback.



## Second Semester

<b>RCS Release 1 features development</b>	Video calls improvement Chat one-to-one File Transfer
<b>RCS-e features development</b>	Manual and automatic authentication Native address book Chat one-to-one
<b>Other features</b>	RCS Release 2: Dialer RCS-e: Dialer RCS-e: Share on Facebook feature RCS-e: User profile
<b>Preparing solution to demos</b>	
<b>Internship final documentation</b>	Final Report Final Presentation

Table 6 – Revised planning for the second semester.

In the second semester of the internship were made some tests and improvements to video calls quality, developed the chat one-to-one and part of file transfer feature.

The planning changed substantially since a new strategy was adopted: create a new version of the application that was compliant with RCS-e. Despite the similarities between the two versions, it was necessary change the base of whole project followed by the development of some features: authentication as described in section 3.1.4., chat one-to-one.

In addition to these features were developed others not defined in RCS releases such as a dialer integrated in the application, and a feature that allows users to share in their Facebook account that are using a WIT Communications Suite product.

Finally, it was necessary to test the application and make the changes/fixes needed to demonstrate it to clients. Although this task is only defined in the second semester, there were several demos and delivers to a potentially client during the development.

Overall, planning has changed greatly, not only due to the challenges and changes during the development but also due to ambitious goals. The platform and the technologies used were new to the team, and it was necessary to integrate the new client with all existing base. This time of learning was not taken into account when the initial planning was defined.

The need to meet the demand of potential customers also influenced the development throughout the internship.

### 5.3. WORK BREAKDOWN

Since the team is composed by three elements it's necessary to clarify the tasks performed during this internship. The part developed in the scope of this internship is described in the section 4.2.3..

However, in an initial stage of the internship, the project had only one collaborator (the author). Thus, it was necessary work in the libraries layer, in particular with PJSIP library, and with the wrapper that connects this layer to the GUI.

Also in the chat feature, it was necessary to develop the wrapper and the functions in Library Manager to send the events to Communicator Library.

Thus, the tasks performed by the author are presented in the Revised Planning section. The social presence and contacts management tasks were performed by another team member.

To a list with the tasks performed, please refer to the Product Backlog, in the section 5.1.5.

### 5.4. RISK MANAGEMENT

Risk management is a very important task in a project's development because it allows monitoring risks and minimizing their impact, if they occur.

WIT Mobile Communicator for Android is a project that is subject to the occurrence of some risks, since new technologies will be used, will be integrated several programming languages and the developer team is comprised of new collaborators without experience on company and in the platform and its technologies.

However identifying risks is not an easy task since they are constantly changing according to internal and external factors at any given time. Thus, this section intends to analyze project risks and create a contingency plan for the ones that have highest priority or which are considered relevant.

Each identified risk shall be qualified according to its occurrence likelihood and impact if it occurs. Therefore, priority was given by the following figure<sup>[71]</sup>.

		Impact			
		Negligible	Marginal	Critical	Catastrophic
Likelihood	Certain	High	High	Extreme	Extreme
	Likely	Moderate	High	High	Extreme
	Possible	Low	Moderate	High	Extreme
	Unlikely	Low	Low	Moderate	Extreme
	Rare	Low	Low	Moderate	High

Figure 10 – Assessment of risks priority.

## INEXPERIENCE OF THE TEAM ON DEVELOPMENT PLATFORM (ANDROID) AND TECHNOLOGIES USED

**Priority:** Extreme

Team elements have a lack of knowledge about Android platform and could have some difficulties in using new technologies so there may be a delay in these tasks execution.

**Mitigation plan:**

- Train doing some tutorials and seeing some examples;
- Analyze the time taken doing each task and adjust the estimated time to the next ones;
- Assign tasks according to skills of each employee;
- Conduct a study based on the technologies to be used before starting the tasks, and if it is necessary, ask to someone who has worked with these technologies.

## AMBITIOUS SCHEDULES

**Priority:** High

If the proposed deadlines are too ambitious (sometimes defined by the customer), it may results in a lack of time to complete the goals and the need to work over budget or fail the deadlines.

**Mitigation plan:**

- Agree realistic deadlines with customers;
- Divide tasks into smaller tasks;
- Reconsidering the estimated time given to the execution of each task;
- Ensure that tasks are being assigned according to the knowledge of each team member;
- If necessary, try to negotiate deadlines.

## MARKET ACCEPTANCE

**Priority:** High

The number of applications that provide some or all the features of WMC for Android is constantly increasing. Thus, the releasing time is a key point to show the product to the customers.

**Mitigation plan:**

- Schedule demos to show prototypes to potential clients;
- Get feedback from demos to adapt the product to the market.

## DYNAMIC ENVIRONMENT

### Priority: High

This project grew in a very dynamic environment. The platform and technologies used are constantly evolving, so the requirements and the project planning can change according to this evolution and to the feedback gathered in demos with the several clients.

### Mitigation plan:

- An agile methodology helps in these changes;
- Being aware of market evolution and changing the requirements priority or event the requirements themselves accordingly.

## 5.5. SOFTWARE TEST PLAN

The tests done to the application can be divided in two parts: tests performed during or after the features development by the developers, and the tests done by the software quality team.

The first ones represent the tests that are done when developing a feature to test if it is already finished or if it is going in the right direction. After an application is given as completed the application will be sent to the software quality department with a test specification, where the members of that team will realized exhaustive tests to the application.

The tests performed by the team are not only the ones specified in the document but also ad-hoc tests, in case of something are missing.

In the final tests, the application passed with 100%, corresponding to the implemented features.

The complete test specification of the project can be consulted in the **Annex D – Software Quality**.

## 6. CONCLUSIONS

The WIT Mobile Communicator for Android is intended to bridge the gap existing in the communication applications. Thus, the main goal of the project was to develop an application for Android devices, that offers to users a variety of features allowing the enrichment of the way to communicating in real-time, having a high integration with the device itself.

Considering the proposed goals, both at a personal and at the project levels, it can be said that the objectives were achieved. As in all projects some challenges were experienced during the development, such as technical limitations of the devices or the learning of new technologies. However, it is overcoming these challenges that we grow professionally and personally, and at the final we can say that we are ready to enter in the professional environment.

It is necessary to emphasize the importance of this internship and all the knowledge acquired during these months. Several new technologies were used and the experience of working in a company, with a team was very grateful.

During the first semester, the focus was on research and in the implementation of a prototype with voice and video calls for demonstration to a potential client. The market and competitors' analysis were performed after a better comprehension of the features to implement. New knowledge about the platform and the technologies was acquired and the prototype was developed.

In the second semester the planning changes substantially, with the emergence of a new version of RCS and the need develop a version that was compliant with it. The chat one-to-one and file transfer features were implemented and various demonstrations were prepared to several potential clients.

Thus, both solutions have already taken interests in some companies, having already been demonstrated during the developments.

The RCS Release 2 version has shown its importance in the market, being acquired by a customer.

### 6.1. FUTURE WORK

The changes on the requirements, the demos and the lack of knowledge in the platform and technologies, were the key points to the discrepancies between the initial and the final planning. Thus, some of the features were postponed.

As future work, it is expected to continue the developments of RCS and RCS-e features that was postponed, making the application full-compliant with the current RCS releases and following new features.

In addition to the new features to implement, it is necessary provide more options in the existing features, such as a high variety of audio and video codecs, and support the largest possible number of Android devices.

Specifically, the next steps in both versions is to implement image and video sharing during an open session and provide the ability to chat with more than one contact at the same chat room, *ie*, group chat.

## 6.2. LESSONS LEARNED

During this internship it was possible to understand that in the industrial environment, not always it's easy to meet the deadlines imposed by the potential customers. Since customers are the key point in business, it is important to acquire their interest as soon as possible. This way, it is important to invest some time in prototypes to be in the front line showing new products.

A good documentation in a project influences its development. It is important to keep it updated and ready to do changes if it is needed. It is also important to clarify the requirements before starting implementations, saving time in possible mistakes and changes.

## REFERENCES

- [1] *Feature Phone*, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
[http://en.wikipedia.org/wiki/Feature\\_phone](http://en.wikipedia.org/wiki/Feature_phone)
- [2] *Smartphone*, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
<http://en.wikipedia.org/wiki/Smartphone>
- [3] WIT Software, S.A. [Online] – Cited on July 11, 2011. Available from:  
<http://www.wit-software.com/>
- [4] WIT Communications Suite [Online] – Cited on July 11, 2011. Available from:  
<http://www.wit-software.com/products/communications-suite/>
- [5] Microsoft Windows [Online] – Cited on July 11, 2011. Available from:  
<http://windows.microsoft.com/pt-PT/windows/home>
- [6] Mac OS X [Online] – Cited on July 11, 2011. Available from:  
<http://www.apple.com/macosx/>
- [7] *Linux*, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
<http://en.wikipedia.org/wiki/Linux>
- [8] *iOS*, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
[http://en.wikipedia.org/wiki/iOS\\_%28Apple%29](http://en.wikipedia.org/wiki/iOS_%28Apple%29)
- [9] Android [Online] – Cited on July 11, 2011. Available from:  
<http://www.android.com/>
- [10] GSMA Rich Communication Suite [Online] – Cited on July 11, 2011. Available from:  
[http://www.gsmworld.com/our-work/mobile\\_lifestyle/rcs/gsma\\_rcs\\_project.htm](http://www.gsmworld.com/our-work/mobile_lifestyle/rcs/gsma_rcs_project.htm)
- [11] Gartner (2011, May 19). *Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year* [Online] – Cited on July 11, 2011. Available from:  
<http://www.gartner.com/it/page.jsp?id=1689814>
- [12] Symbian, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
<http://en.wikipedia.org/wiki/Symbian>
- [13] IDC Portugal (2010, December 15). *Terminais Android Impulsionam Mercado de Telemóveis em Portugal no Terceiro Trimestre* [Online] – Cited on July 11, 2011. Available from:  
[http://www.idc.pt/press/pr\\_2010-12-15.jsp](http://www.idc.pt/press/pr_2010-12-15.jsp)
- [14] Android Market [Online] – Cited on July 11, 2011. Available from:  
<https://market.android.com/>
- [15] MSN Messenger: Mercury Donate, Android Market [Online] – Cited on July 11, 2011. Available from:  
<https://market.android.com/details?id=im.mercury.android.donate>
- [16] MSN Messenger, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
[http://en.wikipedia.org/wiki/MSN\\_Messenger](http://en.wikipedia.org/wiki/MSN_Messenger)
- [17] Yahoo! Messenger [Online] – Cited on July 11, 2011. Available from:

<http://messenger.yahoo.com/>

[18] BeejiveIM [Online] – Cited on July 11, 2011. Available from:  
<http://www.beejive.com/>

[19] iPhone, Apple [Online] – Cited on July 11, 2011. Available from:  
<http://www.apple.com/iphone/>

[20] BlackBerry [Online] – Cited on July 11, 2011. Available from:  
<http://us.blackberry.com/>

[21] iPad, Apple [Online] – Cited on July 11, 2011. Available from:  
<http://www.apple.com/ipad/>

[22] iPod Touch, Apple [Online] – Cited on July 11, 2011. Available from:  
<http://www.apple.com/ipod/>

[23] AIM [Online] – Cited on July 11, 2011. Available from:  
<http://www.aim.com/>

[24] Facebook [Online] – Cited on July 11, 2011. Available from:  
[www.facebook.com](http://www.facebook.com)

[25] GTalk [Online] – Cited on July 11, 2011. Available from:  
<http://www.google.com/talk/>

[26] MySpace [Online] – Cited on July 11, 2011. Available from:  
<http://www.myspace.com/>

[27] IM+, Shape Services [Online] – Cited on July 11, 2011. Available from:  
<http://www.shapeservices.com/en/index.php>

[28] Skype [Online] – Cited on July 11, 2011. Available from:  
<http://www.skype.com/intl/pt/home/>

[29] ICQ [Online] – Cited on July 11, 2011. Available from:  
<http://www.icq.com/en>

[30] Jabber [Online] – Cited on July 11, 2011. Available from:  
<http://www.jabber.org/>

[31] Palm OS, Wikipedia [Online] – Cited on July 11, 2011. Available from:  
[http://en.wikipedia.org/wiki/Palm\\_OS](http://en.wikipedia.org/wiki/Palm_OS)

[32] Bria Android, CounterPath [Online] – Cited on July 11, 2011. Available from:  
<http://www.counterpath.com/bria-android-edition.html>

[33] Talkdroid Messenger, Android Market [Online] – Cited on July 11, 2011. Available from:  
[https://market.android.com/details?id=com.meeble.talkdroidpro&feature=search\\_result](https://market.android.com/details?id=com.meeble.talkdroidpro&feature=search_result)

[34] aSip, VNET-CORP [Online] – Cited on July 11, 2011. Available from:  
<http://www.vnet-corp.com/aSip.htm>

[35] VoIP Video SIP client, VoipSwitch [Online] – Cited on July 11, 2011. Available from:



<http://voipswitch.com/en/products/softphones/mobile-softphones/softphone-for-android/>

[36] KakaoTalk [Online] – Cited on July 11, 2011. Available from:  
<http://www.kakao.com/talk/en>

[37] WhatsApp Messenger [Online] – Cited on July 11, 2011. Available from:  
<http://www.whatsapp.com/>

[38] eBuddy Messenger [Online] – Cited on July 11, 2011. Available from:  
<http://www.ebuddy.com/android/>

[39] LiveProfile [Online] – Cited on July 11, 2011. Available from:  
<http://www.liveprofile.com/>

[40] Trillian Messenger [Online] – Cited on July 11, 2011. Available from:  
<http://www.trillian.im/android/>

[41] Meebo IM [Online] – Cited on July 11, 2011. Available from:  
<http://www.meebo.com/android/>

[42] fring [Online] – Cited on July 11, 2011. Available from:  
<http://www.fring.com/>

[43] Kik Messenger [Online] – Cited on July 11, 2011. Available from:  
<http://kik.com/>

[44] Nimbuzz [Online] – Cited on July 11, 2011. Available from:  
<http://www.nimbuzz.com/en>

[45] ooVoo Video Calls [Online] – Cited on July 11, 2011. Available from:  
<http://www.oovoo.com/home.aspx>

[46] MobileVOIP, Android Market [Online] – Cited on July 11, 2011. Available from:  
[https://market.android.com/details?id=finarea.MobileVoip&feature=search\\_result](https://market.android.com/details?id=finarea.MobileVoip&feature=search_result)

[47] CSipSimple [Online] – Cited on July 11, 2011. Available from:  
<http://code.google.com/p/csipsimple/>

[48] PJSIP – Open Source SIP Stack [Online] – Cited on July 11, 2011. Available from:  
<http://www.pjsip.org/>

[49] Palringo Instant Messenger [Online] – Cited on July 11, 2011. Available from:  
<http://www.palringo.com/en/gb/>

[50] Sipdroid [Online] – Cited on July 11, 2011. Available from:  
<http://sipdroid.org/>

[51] 3CX Phone [Online] – Cited on July 11, 2011. Available from:  
<http://www.3cx.com/VOIP/voip-phone.html>

[52] Movicha – Mobile Video Chat [Online] – Cited on July 11, 2011. Available from:  
<http://www.movicha.com/>

[53] IMSdroid [Online] – Cited on July 11, 2011. Available from:

<http://code.google.com/p/imsdroid/>

[54] doubango – open source 3GPP IMS/LTE framework for embedded systems [Online] – Cited on July 11, 2011. Available from:

<http://doubango.org/>

[55] Linphone [Online] – Cited on July 11, 2011. Available from:

<http://www.linphone.org/>

[56] GSMA RCS Project, GSM World [Online] – Cited on July 11, 2011. Available from:

[http://www.gsmworld.com/our-work/mobile\\_lifestyle/rcs/gsma\\_rcs\\_project.htm](http://www.gsmworld.com/our-work/mobile_lifestyle/rcs/gsma_rcs_project.htm)

[57] Rich Communicator Suite Release 1, Functional Description 1.2 [Online] – Cited on July 11, 2011. Available from:

<http://www.gsmworld.com/documents/FunctionalDescriptionv1.2.pdf>

[58] Rich Communicator Suite Release 2, Functional Description 1.1 [Online] – Cited on July 11, 2011. Available from:

[http://www.gsmworld.com/documents/FunctionalDescription\\_v1.1%281%29.pdf](http://www.gsmworld.com/documents/FunctionalDescription_v1.1%281%29.pdf)

[59] Rich Communicator Suite Release 3, Functional Description 1.0 [Online] – Cited on July 11, 2011. Available from:

[http://www.gsmworld.com/documents/FunctionalDescription\\_v1.0%281%29.pdf](http://www.gsmworld.com/documents/FunctionalDescription_v1.0%281%29.pdf)

[60] RCS-e specification Version 1.1 – Final [Online] – Cited on July 11, 2011. Available from:

[http://www.gsmworld.com/our-work/mobile\\_lifestyle/rcs/rcse\\_specification.htm](http://www.gsmworld.com/our-work/mobile_lifestyle/rcs/rcse_specification.htm)

[61] *Application Fundamentals*, Android Developers [Online] – Cited on July 11, 2011. Available from:

<http://developer.android.com/guide/topics/fundamentals.html>

[62] *Activity Lifecycle*, Android Developers [Online] – Cited on July 11, 2011. Available from:

<http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle>

[63] *Intents and Intent Filters*, Android Developers [Online] – Cited on July 11, 2011. Available from:

<http://developer.android.com/guide/topics/intents/intents-filters.html>

[64] *OMA XML Document Management*, Open Mobile Alliance [Online] – Cited on July 11, 2011. Available from:

[http://www.openmobilealliance.org/Technical/release\\_program/xdm\\_v2\\_0.aspx](http://www.openmobilealliance.org/Technical/release_program/xdm_v2_0.aspx)

[65] FFmpeg [Online] – Cited on July 11, 2011. Available from:

<http://www.ffmpeg.org/>

[66] Android Native Development Kit, Android Developers [Online] – Cited on July 11, 2011. Available from:

<http://developer.android.com/sdk/ndk/overview.html>

[67] *RFC3261: Session Initiation Protocol*, Internet Engineering Task Force [Online] – Cited on July 11, 2011. Available from:

<http://www.ietf.org/rfc/rfc3261.txt>

[68] *Real-time Transport Protocol*, Wikipedia [Online] – Cited on July 11, 2011. Available from:

[http://en.wikipedia.org/wiki/Real-time\\_Transport\\_Protocol](http://en.wikipedia.org/wiki/Real-time_Transport_Protocol)

[69] G. Camarillo and M. A. García-Martín, *The 3G IP Multimedia Subsystem (IMS)*, 3<sup>rd</sup> ed. West Sussex, United Kingdom, John Wiley & Sons Ltd, 2008.

[70] P. Deemer and G. Benefield, *THE SCRUM PRIMER – An Introduction to Agile Project Management with Scrum*, Version 1.04, 2007.

[71] James D. (October 19, 2009), *IT Risk Management*, Realsearch – Software Engineering at NCSU [Online] – Cited on July 11, 2011. Available from:

<http://agile.csc.ncsu.edu/SEMaterials/RiskManagementJamesDemitriou.pdf>