

1-1-2013

Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges

Shakira Banu Kaleel
kshakira@ryerson.ca

Ssowjanya Harishankar
sharisha@ryerson.ca

Follow this and additional works at: http://digitalcommons.ryerson.ca/compsci_techrpts



Part of the [Computer Engineering Commons](#)

Recommended Citation

Kaleel, Shakira Banu and Harishankar, Ssowjanya, "Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges" (2013). *Computer Science Technical Reports*. Paper 4.
http://digitalcommons.ryerson.ca/compsci_techrpts/4

This Technical Report is brought to you for free and open access by the Computer Science at Digital Commons @ Ryerson. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenges

Shakira Banu Kaleel
Ryerson University,
Toronto, Canada
kshakira@ryerson.ca

Ssowjanya HariShankar
Ryerson University,
Toronto, Canada,
sharisha@ryerson.ca

Abstract— Highly volatile requirements of mobile applications require adaptive software development methods. Several attempts to address challenges in mobile software engineering have found agile methodology to be appropriate for mobile application development. This project report provides a detailed analysis on various challenges involved in mobile software development which are addressed using Agile-SCRUM methodologies. An efficient mobile software development concept derived from Agile-Scrum methodology is designed in this project. A light-weight Android application for secure and incremental backup has been developed using the proposed methodology. An in-depth illustration of the practical experience in developing the application has been discussed. Unlike other prominent languages like Java, the use of Python for Android platform has emerged recently. Hence developing the secure-backup application in Python was a challenge, which has been dealt in this report. We believe our proposed methodology has a potential to help developers deliver improved quality of mobile applications in short time.

Keywords: agile, scrum, mobile software engineering, mobile application, android, python, sl4a

I. INTRODUCTION

A. Mobile Software Engineering

Massive diffusion of mobile phones, especially smart phones and mobile broadband services has lead to enormous and rapid development of mobile software applications. The number of customers for 3G, 4G and 5G networks has been on the rise since 2008. It was foreseen that there will be four times increase in customers count leading to a dramatic increase in cellular bandwidth by 2012 [12]. It was also predicted in [13] that the world revenue would grow from about 1 billion in 2008 to approximately 8 billion in 2013, where the mobile social computing would stand third place, after music and gaming.

Today we have a number of programming environments available for developers to create and release mobile applications which include Xcode for Apple devices, Android Development Tools (ADT) plugin for Eclipse environment offered by Android, Microsoft's Visual Studio environment offered for Windows phone and similar environment for Blackberry, Symbian and such others. Out of a study

conducted in [1], it was observed that small or medium sized mobile applications were developed by one or two developers who were responsible for the point of conception until the official release of applications. The development process or tracking of development activities are not given adequate importance by developers, but only some best practices have been adhered. Also, since most of the applications developed were for recreational purposes and were not business centric, the software engineering process involved has not been formal. The recent growth in the number of mobile applications demands the mobile industry to deliver high product quality. This can be achieved by incorporating standard software development methodologies in mobile software development.

B. Agile and Scrum practices for Mobile devices

Agility means being able to move quickly and easily. Using agile methodologies in software development process makes the process flexible enough to adapt quickly and easily to the changing trends and technologies. In mobile software engineering, methodology is of great importance, since software applications are ever changing and evolving based on immediate user requirements. For teams focused on customer satisfaction via development of a well-designed application by means of a production process, agile methodology is adopted. The SCRUM methodology is an iterative-incremental agile development framework. Projects are developed through series of iterations called "sprints". Each sprint is ideally a bunch of low level SDLC processes where a small sub feature is driven from idea conception, planning till testing phase, which takes around two to four weeks to complete. Short SCRUM meetings are held on a daily basis to discuss status updates of the team's sprint tasks. Post sprint, retrospective meeting is conducted to document the shortcomings and improvements. It has been found that agile-scrum process is best suited for project development scenarios with highly emerging or rapidly changing requirements as in the case of mobile software development.

C. Android Operating System

After the launch of iphone Appstore in 2008, various other mobile engineering companies began their development and launch of mobile applications. Google acquired Android Inc. back in 2005 with the intention to extend their successful business to include the mobile market. A significant

distinction between Android and iPhone platforms is that Android is not tied to any particular hardware and can be used on many mobile devices unlike iPhone. Other Android OS market partners are HTC, LG, Samsung, Motorola and NTT Docomo. In this project we have chosen Android platform for development of a mobile software application. Android architecture consists of four layers which are: a) Linux Kernel, b) Dalvik virtual machine and Android runtime environment, c) application framework and d) the applications layer.

D. Scope

One of the scopes of this project is to perform a detailed analysis of agile and scrum practices that suit mobile (Android) software engineering in the best possible ways. We begin by studying the a) available literature of agile software development methodologies, b) list a number of challenges faced in mobile software engineering and c) focus on the best practices recommended for addressing the challenges and developing mobile software applications. Research work in [1], [3], [4], [6], [7] and [8] have proved that Agile and SCRUM methodologies are one of the best practices to be adhered for mobile software development process. We then intend to construct a software development methodology best suited for mobile application development. The other scope of this project is to apply some of the discussed practices in developing a software application for Android platform, document our experience and challenges faced in the process. We aim to construct a backup application for Android OS and incorporate Rsync over Secure Shell (SSH) to make it secure. We have also included the concept of “reuse” and “sub-version control” in our proposed methodology as well as in the android application.

II. LITERATURE REVIEW

A. Challenges in mobile software development

In this section, we highlight various challenges faced in mobile application engineering and mention those that we have attempted to address in our project. Unlike the process-intensive traditional software applications like those that run on desktops, servers or laptops, the software engineering process involved in developing mobile applications pose additional requirements and issues. Some common issues include: integration of software with hardware, limited storage and traditional issues like reliability and performance [1]. Other problems arise due to wireless feature that needs support for bandwidth variability and intermittent connections. Due to mobility of mobile phones, there are issues with location-dependent information retrieval and address migration.

Development of mobile computing applications is difficult especially because they need to be adaptive, as applications need to be aware of and adapt to changing resources like battery and bandwidth. Security in mobile platforms is a major issue since it is an open platform and malware applications could be easily installed without proper detection services. Interaction between applications is an area with potential problems and web applications that connect to internet pose different set of challenges. According to [2], User Interface design needs significant consideration as the

screen size and resolution differs for different platforms like iPad & iPhone and various other Android platforms. Smart phones have accelerometer that reacts to device movement, touch screen that responds to gestures, virtual keyboard, multiple cameras, Global Positioning System and such other features. These features place requirements like sensor handling, compatibility of native and hybrid application, user interface management and testing complexity. A number of platforms available for operating system like Android may require different set of features to function in the application. Every application that runs on a mobile device depletes the device's battery life to some extent, which is a major concern.

As new software applications are just a tap away from the end users, customer satisfaction has always been the prime objective of mobile software engineering. The rapid growth of task-specific mobile devices with limited lifetime exerts pressure on the development and maintenance of adaptive mobile applications. The critical factor that limits commercial success of mobile devices is the extremely narrow time available for release of new software applications. A slow-paced software engineering process could device a sturdy application, but would result in releasing out-dated applications to the market. Hence the mobile software development process is time constrained. In this project, we aim to construct a reliable light-weight Android application using test-driven and iterative-incremental development approach.

B. Best practices recommended for Mobile Software Engineering from literature

The functioning of mobile software engineering is directly based on the budding demands from end users. The mobile software industry is more focused on the product launch than on the process adhered in product development. Mobile industries can sustain only if they reflect opinion of end users and rapidly launch the software products on demand. Market consciousness plays a vital role in mobile software engineering. It is vital to understand and predict the market and also incorporate relevant practices from the start to end of the mobile development lifecycle. Following are the two main methodologies that have been identified as the best practices for mobile software engineering. We have incorporated some highlighting features of these methods to construct our mobile software development methodology and develop our Android application.

1) Agile Methodology:

It is evident that large scale software development projects are moving away from process-intensive approach towards agile and Scrum approaches [1]. Agile methods are driven towards establishing cost efficiency and customer satisfaction. The mobile software application has user interface modules which react immediately to user interactions, and since this is an environment with rapidly changing user demands, the development of such applications requires an agile approach. Among the available software development methods, agile methods are the most appropriate and potential solution for mobile application development according to [4], [5] and [6]. To prove this, the author in [6] has performed a comparative analysis of ideal agile characteristics with traits observed in mobile software development and showed how mobile

engineering requires agile characteristics to be incorporated. In a review of literature conducted in [11], the key principles of agile manifesto has been summarized according to [10] as “individual and interactions over process, working software over extensive documentation, customer collaboration over contract negotiation and responding to change over following a monotonous plan”. These show the basic difference in importance given to certain activities by agile methodologies when compared to traditional software development methodologies. There are multiple agile processes with many distinctive features such as: Adaptive Software Development (ADP), SCRUM, Extreme Programming (XP), Crystal and Feature-driven development (FDD) [17]. Best suited features from all these processes are being adopted by various teams to tune to their flavour of agility.

Agile software development is considered to be a powerful and proficient approach to accommodate rapidly changing demands of the market and to cope up with modern software development trends. According to [7], “Agile software development methodologies claim to speed up the delivery of software solutions to a client’s rapidly changing requirements”. Agile methods are believed to enhance certain features of mobile application development process such as flexibility, productivity, sustainability, learning, responsiveness, leanness, lightness, simplicity, technical quality and active user collaboration in software development [4], [7]. This methodology uses iterative-incremental model for development process to achieve quick delivery of software and to improve risk-management capabilities. Some of the highlighting agile characteristics that well align with requirements of mobile application development include:

a) *Test-driven development*: Testing goes hand-in-hand with development process.

b) *Continuous customer involvement*: Mobile industry focuses on demands from the market or end-users. Customer involvement during all phases of SDLC is considered highly important to avoid unpleasant or unprecedented variations in requirements. This helps to adapt accordingly.

c) *Prioritization of requirements*: According to iteration, level of abstraction and current scope, the requirements may vary and has to be prioritized. Once they are prioritized, the design and implementation processes are focused on satisfying the highly significant requirements first and then towards attending to less significant requirements.

d) *Effective Communication*: The primary means of communication is face-to-face people interaction. Agile process believes in building projects around stimulated individuals in the team.

e) *Enhanced quality assurance*: The iterative-incremental model and process-wide review-learning sessions ensure enhanced quality assurance.

f) *Skilled developers*: Developers are agile, knowledgeable, collocated and collaborative.

g) *Process-wide reviews and learning sessions*: Knowledge obtained from product development is in turn

incorporated in the development process through reviews and learning sessions to ensure product analysis.

h) *Adaptive process*: The process is adaptive since learning from the previous iterations or releases are efficiently incorporated into the process for betterment. Continuous attention to technical fineness and good quality design boosts agility.

To achieve minimum time to launch the application, it is necessary to have regular and early releases of operational software. This concept of regular release builds of the software is an important feature of Agile methodology. In this project, we have applied Agile methodology in software development phases to build a light-weight secure backup Android application. Here, we have covered the following features of agile methodology: test-driven development, quality assurance, prioritization of requirements, periodical reviews, effective communication, skilled developers and adaptive processing.

2) *Scrum Methodology*:

Scrum is a well-known agile methodology which is characterized by iterative-incremental sprints. Sprints are short and time bound in nature. According to [9], Scrum is an approach of project delivery which includes self-organizing process into iterations which consists of planning, black-box engineering process and review or post-mortem phase. Scrum is an empirical management technique to control and manage software development process. It mainly focuses on customer satisfaction, product reliability and product quality by means of self-organizing iterations. The dynamics of the team and process are revealed and built during project execution.

When compared to the traditional white box or waterfall model, Scrum integrates the following phases into a sprint: use-case development, design, implementation, integration and testing. So the Scrum-agile process, in other words, the Black Box process consists of the following phases in general: product requirement, project planning, Sprint and reviews & release [9]. Planning is essential in sprints so as to reduce the risk of roadblocks during development phase. Post-mortem is another critical part that helps strengthen the knowledge-base of best practices. SCRUM differs from waterfall and spiral models by providing: a) responsiveness to environment throughout the process, b) unlimited team flexibility and creativity during iterations, c) knowledge transfer in the form of team work during project and d) a very high probability of success [18].

Three phases involved in Scrum are: a) Pregame: This includes planning and architecture design of the software based on items in the product backlog, b) Game: System is developed in sprints, where functionality is enhanced to produce updates and c) Postgame: marks the closure of the development process, including final documentation, pre-release stage testing and release [18].

In a mobile application development project using SCRUM practices, distributed teams work autonomously and communicate through the daily SCRUM meetings. Daily Scrum is 15 minutes long, conducted to a) report and update

new impediments, scrutinize resources and skills required to tackle those impediments, b) monitor and share statistics of automated tests, builds and defects, c) present knowledge session of new features to team and d) conduct post-mortem & reviews to impart lessons-learned in the process to team and update knowledge-base. In accordance to the development pace of the application, SCRUM meetings may be conducted on a daily basis or after each shift and an elaborate meeting could be held on a weekly basis between various teams to ensure effective management of planned activities. It was found in [8] that a number of teams that worked on mobile application development using SCRUM framework were successfully able to deliver working software on time. Considering issues they faced with time constraints and with just-in time learning of mobile application development process, the teams managed to accomplish project deliverables on time using SCRUM framework.

In our project, we have followed SCRUM practices by conducting status update meetings where we discussed about tasks related to implementation and testing phases. Reviews and changes to the plan, architecture, tools & technologies, source code and quality assurance scenarios were made during these meetings. As the number of team members involved in this project is two, the meetings were held through one of the modes: face-to-face meet up, Skype or phone calls. Generally the team size of a project following SCRUM model is around seven [22]. We have adopted certain SCRUM practices to help us develop the secure-backup Android application for a smaller team size of 2 members. In our case, the team members share roles of Scrum Master, developer, software tester, Quality Analyst and such others. Apart from the SCRUM and Agile practices, we have used Bit Bucket for hosting our project code and related documents as part of version control system. Version controlling helps in ensuring the integrity of the artifacts being developed in the development lifecycle. Web hosting helps us in sharing documents internally within team, keeping track of team member's activities and following up on deliverables. It also assists us in providing feedback on the tasks accomplished and helps in maintaining up-to-date project related data.

III. DESIGNING SOFTWARE ENGINEERING METHODOLOGY FOR MOBILE APPLICATION DEVELOPMENT

Our extensive study involving various research papers on agile-scrum methodologies and challenges involved in mobile software development process helped us derive a development methodology. Traditional software development methodologies should be redesigned to accommodate adaptive nature of mobile applications. In our approach, certain vital practices that we found essential for mobile software development lifecycle have been included into the traditional Agile and SCRUM methodologies. Fig. 1 shows the proposed Agile-Scrum methodology for mobile application development with some additional features.

This methodology consists of the different phases namely: Phase 1: Requirements Analysis, Phase 2: Design and Development, Phase 3: Test and QA, Phase 4: Product

acceptance phase and Phase 5: Release to Market. When a customer demands for a mobile software product, the concept is fed into Phase 1. In this phase, we have incorporated market analysis and physical constraint analysis (or overview of physical architecture) of mobile terminals.

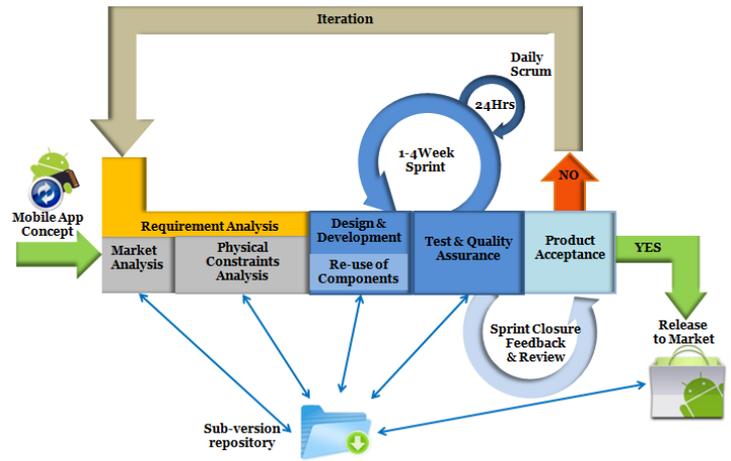


Figure 1. Proposed Agile-SCRUM Methodology for mobile Application development

These play important roles in mitigating uncertainties and technical risks respectively if they are included in the early stages of development, which is explained in the upcoming section. Phases 2 and 3 form a Sprint in our project that lasts for 1 to 4 weeks. The Re-use concept in software development is covered in Phase 2 of our methodology and is implemented in our Android application by means of Rsync over SSH. Daily SCRUM meetings are part of Sprint for the team to share updates on tasks accomplished, plan for the next course of action, foresee and address various issues. Phase 4 marks the closure of each sprint after which, effective feedback and retrospective meetings are held to document the best practices and shortcomings faced in the current sprint iteration.

Phase 5 is an important validation step which confirms the product's acceptance. On satisfying the customer's requirements, the product is set for official market release. If not, second iteration begins from Phase 1 paving way for another sprint cycle. Most of the phases in this methodology uses sub-version repository to control and maintain versioning of all project related documents as shown in Fig. 1. Our methodology provides a clear vision of Agility by its adaptive nature of re-framing requirements in subsequent iterations. This includes consideration to i) market demands (guarantees commercial success) and ii) changes to physical architecture of the mobile device (conducting knowledge sessions on new versions of the device) in every iteration to negate risks and ambiguities.

IV. SECURE BACKUP APPLICATION ARCHITECTURE

A. Components involved in developing Secure Backup Android Application

A detailed overview of components required to design the secure backup Android application using agile-scrum software development methods is provided here:

1) Google Android Operating System:

Open Handset Alliance (OHA) along with Google as the most active partner has given rise to development of Android OS. In 2010, Android market charged 25 USD to become a registered developer and a developer was charged 400 USD to purchase an Android developer's phone [12]. Android is based on Linux-kernel where the user space and Dalvik Virtual machine is written in C. Other royalty charges include download of non-free applications from the Android Market. An advantage of using Android OS is that the software is not tied to any particular mobile device like the iPhone OS. Other than Google, some notable OHA partners who produce mobile handsets that are compatible with Android OS are HTC, LG, Samsung, Sony Ericsson, Motorola and NTT Docomo. We have chosen Google Nexus as the mobile model on which the planned secure-backup application is developed. Even though the Android OS is open source, certain applications that access Google's web services are not open source to other mobile platforms. The Android OS is mainly focused on supporting multi-tasking, multi-touch and double tap zooming options. The modern graphical look of user interface is attractive and highly user-friendly. The OS is designed with an aim to always remain on and most of the processing would be done online making it cooperative and best suited for developing cloud and social computing applications. Other hardware manufacturers that deal with PC production have adapted to include Android enabled terminals (smart phones, tablets) like Dell and Acer in their agenda [16].

2) Android Architecture Overview:

The Android architecture consists of four distinct layers namely (from top to bottom): a) Application layer- includes software applications that run on the mobile phones (These applications are developed in Java using Android software development kit SDK), b) Application Framework Layer-provides services to applications like Activity Manager (controls activities for every application) and Content Provider (provides the application with the required content and controls data accessibility across other applications), c) Library / Virtual Machine Layer and d) Linux Kernel Layer. The third layer consists of Android run-time environment, libraries and also the Dalvik Virtual Machine which resides in Google's Android operating system. Dalvik VM provides instrumentation tools to help run the applications. Applications written in Java are converted to Dalvik Executable format before they are installed on mobiles, as this format was designed to support systems that have limited memory and processing speed.

Android applications are composed of four main components according to [14], namely:

a) *Activity*: It is a response to an event (generated by users or system) which is focused on user interaction. Each activity can represent a class in the source code.

b) *Broadband receivers*: Help applications to listen and react appropriately to broadcast announcements in the system.

c) *Content Providers*: Provides data for an application and controls data accessibility across applications.

d) *Services*: Certain services run in the background like an email client that constantly checks for new mails.

3) Python for SL4A:

Scripting Layer for Android (SL4A) is Android Scripting Environment that enables users to edit and execute scripts against multiple script interpreters on the Android device. These scripts have access to many Android APIs and SL4A enables users to install script interpreters into the application directly from SL4A homepage. It is available as an open source project, hosted on Google Code. This environment was designed to assist developers in creating new applications for Android operating system by providing support for many scripting languages such as Python, Linux shell, Ruby, Perl, PHP, Rhino, Lua, JavaScript or BeanShell. This system is applicable for accomplishing three kinds of tasks namely: Rapid Application Development (RAD), writing test scripts and building utilities. In this project, we use SL4A's architecture to build a secure back-up utility for Android OS.

Among other scripting languages, we have chosen Python for SL4A. To begin with, mobile computing was not easily compatible with Python scripting language, as most of the mobile platforms do not provide direct access to Python. Java/XML is mostly used for Android platforms. But, after the introduction of Python for SL4A by Google in 2011, this has changed and using Python for Android seemed quite challenging for us to undertake. Python is an easy to learn, interactive, object-oriented and multi-paradigm language. It is an open-source interpreted high-level programming language. Python does not come along with SL4A, unless downloaded and installed as an additional feature. In our project, this interpreter is used to develop our secure Backup application in Python scripting language for Android OS.

4) RSYNC over Secure Shell:

RSYNC utility is used to ensure availability of identical data at two different locations. It is used for taking backup or mirroring of files from one system to another. If the destination already has a different version of the file structure than the source, then backup of the difference in file structure only occurs. Secure Shell is a network protocol that enables cryptographic encryption of data communication between two systems that are connected via secure tunnel over an insecure network. RSA private key is used for encrypting the file transfer using SSH. Client connects to the remote backup server via internet and SSH tunnel. The Rsync process uses SSH to encrypt and transfer files from client to server. The server is loaded with corresponding RSA public key certificate to establish the connection with client and take back up of the specified directories

RSYNC over SSH has been used as an additional non-Python component in our architecture. The concept of *re-use* in

software engineering is thus highlighted in our project. The Rsync protocol avoids data redundancy and removes files from server during the next scheduled backup, if those files were already removed from the client.

5) *Sub-version control system:*

In order to maintain and manage different versions of project related documents like the source code, Config file, architecture diagram or framework and other readme files, we have used a sub-version control tool called Bitbucket, with Git as client-end. We have been developing and updating documents throughout the software development lifecycle of the secure-back-up application for Android. Bitbucket-Git helps in maintaining sub versions of all these documents and tracking each version and the changes that it went through with appropriate logs Fig. 9 is a snapshot of Git Client-end and Fig. 10 shows the sub-version repository that we have maintained on Bitbucket.

B. *Architecture of Secure Backup Android Application*

The architecture of our Android application is shown in Fig. 2. Android emulator is the client in our setup which is created using Android Virtual Devices (AVD). Google Nexus S is the chosen mobile model for developing the application. Script Layer for Android (SL4A) is installed on top of SDK. The second layer of our system architecture consists of the interpreter Python for SL4A which is embedded into SDK. In the same layer we have the Rsync over SSH tool for backup. This backup process will involve initial authentication between the client and server which is followed by connection establishment and secure file transfer process. The third layer consists of the Python script that performs secure backup. This layer consists of two important supporting files namely: a) CONFIG file, which contains server details, the directories to be considered for backup, location of the RSA files and such others and b) RSA Private Key: The certificate key for authenticating the client. Using the RSA key exchange SSH is activated and a connection is established between client and remote server, following which the secure back up process takes place.

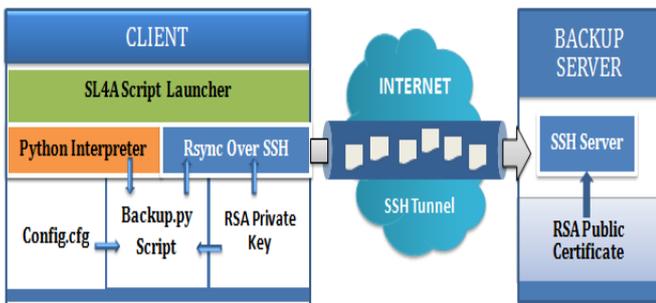


Figure 2. Architecture of Secure Backup Android Application

V. IMPLEMENTATION

For developing the proposed secure-backup Android application, we have installed Android Software Development Kit (SDK) in Windows Vista platform. After unpacking Android SDK; Android Virtual Device (AVD) is used to create mobile emulator for Google Nexus S model. Python for Android and SL4A need to be installed, as shown in Fig. 3 and Fig. 4; and the source code Backup.py script is created for performing the secure-incremental backup. RSA Keys are generated on the client with private key saved on client and public key saved on the server. This is to support Rsync to work over SSH. Variable details in the client-server setup such as the RSA key certificate location on the client, server details like IP address and domain name are configured in a Config file as seen in Fig. 9. The backup script is compiled and packaged along with required certificates, assets and resources into a single Android Application Package file (APK). With this APK format of our application, we can install the software on Android OS and run secure backup operation on the mobile device, as shown in Fig. 5, 6 and 7 (where, SecureBackup is the APK format of our application). A sample log on remote server is shown in Fig. 8.

The application is tested in Android emulator for Nexus S. The emulator is launched by using Android 4.1.2-API Level16 as target and CPU/ image as ARM (armeabi-v7a). Memory configuration: 343MB for RAM and 32MB for heap. Python version 3.2 has been used. We were able to launch Android Virtual Devices only for ARM.



Figure 3. SecureBackup APK in Android SDK

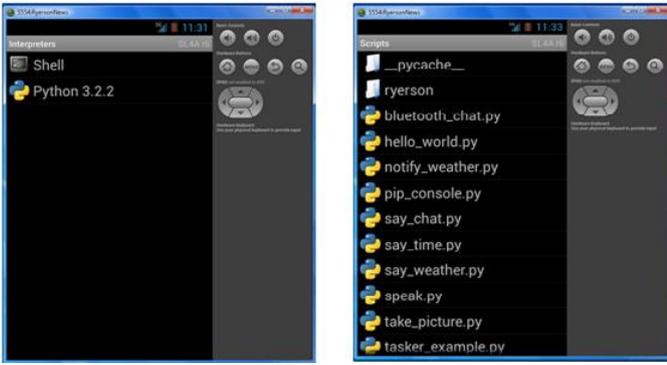


Figure 4. SL4A for Python

To maintain sub-versioning of the source code and related documentation, Bitbucket has been used. To access Bitbucket, we have used Git as the client end.



Figure 5. SecureBackup Script Execution

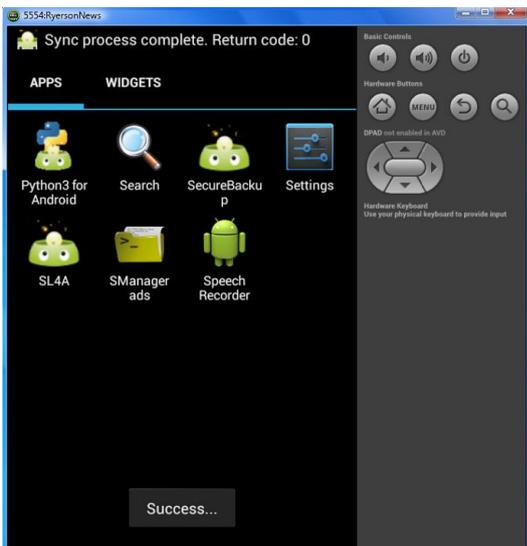


Figure 6. Completion of Execution Snapshot

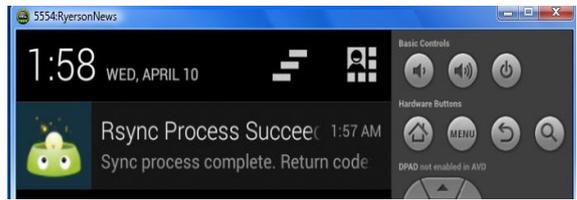


Figure 7. Post Execution Snapshot

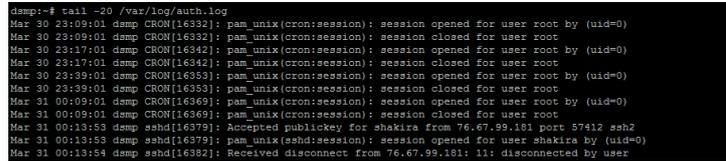


Figure 8. Remote Server Logs

Fig. 10 and Fig. 11 show sub-version commits of source code and various other files like Readme, Config file and Architecture diagram of secure-backup Android application maintained in Bitbucket.

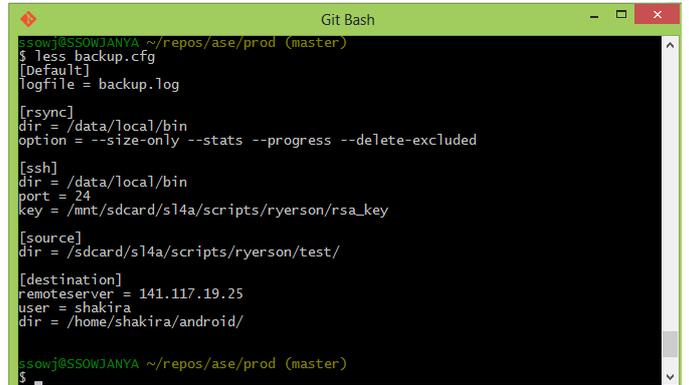


Figure 9. Config File Snapshot

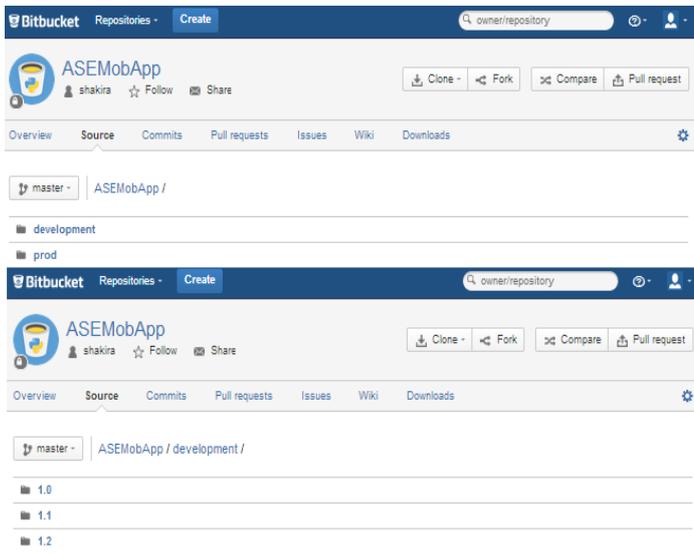


Figure 10. Bitbucket Project Repository

Shakira	5d2e0db	SL4A and Python-for-Android apk and instruction	4 days ago
Shakira	5fa71c2	Merge branch 'master' of https://bitbucket.org/shakira/asemobapp	4 days ago
Shakira	356ab44	AVD Source and instruction	4 days ago
Ssovjanya Ha...	8f3a32d	Merge branch 'master' of https://bitbucket.org/shakira/asemobapp	4 days ago
Ssovjanya Ha...	18e4d25	Minimal changes done to design of the Architecture Diagram	4 days ago
Shakira	e2b336c	Instructions to install Rsync Over SSH reuse components	4 days ago
Shakira	00e2959	Adding opkg installer package	4 days ago
Ssovjanya Ha...	3851236	Architecture of Android Backup App.pdf	11 days ago
Ssovjanya Ha...	88514de	remove Architecture(of Android) Backupl App.docx	11 days ago

Figure 11. Bitbucket Commit History

VI. OBSERVATION AND FINDINGS

This section discusses our observations in applying Agile-SCRUM methods in this project and the practical experience faced in developing the light-weight secure-backup Android application:

- Today's smart phones support many different programming options. We focused our work on the Android platform since it is one of the most prominent platforms by far [21], with the largest number of available devices.
- Use of Agile-SCRUM development methodology not only provides adaptability and flexibility to changing requirements, but also offers a strict and scheduled work model for the team to adhere to.
- In our experience, some benefits observed in using Agile-SCRUM methodology in Android application development are a) project risk minimization, b) better predictability, flexibility and adaptability, c) assurance of producing high quality software, d) enhanced project visibility and e) on-time software release.
- With relatively changing requirements and without a complete plan of execution, we were able to adapt and accommodate changes introduced during iterations.

Throughout the project, we have updated requirements, techniques-tools and maintained detailed documentation using sub-version control system. Since this was a small-scale project with highly volatile requirements and plan, it was quite impossible without the daily SCRUM meetings to achieve sync among team and work together to achieve these results.

- We began the initial iterations with the aim to produce a software application that would act as a back-up utility and progressed towards designing iterative sprints using SCRUM. Next, as an enhancement we introduced the concept of security in the application under development. After a brief research, we chose to use Rsync over SSH to cryptographically encrypt data transfer between client-server. Advantages of integrating Rsync over SSH in our backup application are: i) the concept of software engineering "Re-use" is practiced by using the pre-existing (non-Python) Rsync application; ii) Using different processes or modules like Rsync reduces overall CPU utilization of the application.
- The performance analysis of our application was done using Android Debug Bridge (ADB). Fig. 12 clearly shows that the application consumes only 12% of CPU utilization.
- Most of the Android applications are developed using JavaScript, PHP, Perl and such other scripting languages. Python generally has numerous libraries and provides options to integrate systems to achieve higher productivity and maintain lower maintenance costs.
- We came across some difficulties in using Python to develop the Android application. Python, supported by Android Development Environment has recently evolved in 2011, which is why we chose to use Python and impart our experience. Unfortunately, the vast number of Python library functions is not integrated in Android OS, so we had to explicitly build the required modules using instructions from [20].
- Regardless of the availability of feature-rich mobile phone hardware and powerful software platforms in abundance, developing reliable and strong applications continue to be a time consuming process which demands in-depth knowledge from amateur developers. Main restricting factor of using Python for android application development is that it is still in its early development phase and we do not have many applications developed or much help available in this context.
- We believe that implementing the concept of Software engineering *re-use* of existing components that are not natively developed in Python can help overcome issues that arise from using Python for Android. In this project, we have re-used Rsync utility and integrated it in our secure backup application, which addressed the challenge of developing backup functionality in Python.

- Most good quality and reliable Android backup applications are not available in market for free and needs to be purchased by users. As far as we know, secure backup application for Android devices using Rsync over SSH and Python scripting language has not been documented online. This could be because of the recent surfacing of Python for Android and also due to availability of many other GUI backup applications.
- This backup utility is meant to be developed by advanced Linux users. On the other hand, users with knowledge on RSA key generation alone can install our application to take a secure backup of Android devices with limited changes to the Config file.
- During implementation phase, we were able to launch the emulator for Google Nexus S device by configuring the Android Virtual device with ARM (armeabi-v7a) as system image and Target as Android 4.1.2-API Level 16. Other options of CPU or system image do not launch the emulator, due to which we were unable to check the feasibility of our application on any other Android architecture.

Apart from our practical experience in working for this project, we have learnt that importance to i) physical architecture of the mobile platform or device and ii) incorporation of commercialization right from the first iteration proves to be highly advantageous. Constraints in physical architecture of mobile devices need to be considered and elaborated from the early stages of SDLC process. These limitations constitute to a high degree of technical risk in the software design. Prototyping could help mitigate such threats. Mobile software development is majorly dependent on the market. Market consciousness needs to be incorporated into SDLC activities in order to help mitigate uncertainties and risks that arise from market. It is also imperative to analyze the market and customer's needs and meet a strict time-to-market schedule.

VII. RESULTS

It is observed that using Agile-SCRUM practices and incorporating ideas from market are highly essential in successfully developing a software application for Android or any mobile platforms. Our basic idea was to develop a light-weight Android application. Due to process-switched control mode of the application, we have achieved minimum CPU utilization at a given point in time.

We have obtained CPU performance statistics by using ADB. Maximum CPU utilization of the secure backup application is observed to be as low as 12% which is shown in Fig. 12. This proves that we have indeed developed a light-weight Android application. Another important feature to highlight is that this application performs incremental backup of data, i.e., it performs a backup of latest modifications since the last backup. Due to incremental backup, the amount of data transferred every time is minimized, which in turn reduces communication cost and bandwidth consumption of the network. Hence, we do believe that battery consumption of this application would be minimal. Unfortunately due to limitations

in using the emulator, we were not successful in testing the battery performance of our application.

```

===== CPU Utilization Statistics =====
PID PR CPU%S #THR VSS RSS PCY UID Name
25266 0 12% S 14 158148K 21580K fg u0_a52 com.ryerson.SecureBackup
..
25266 0 6% S 15 159188K 21488K fg u0_a52 com.ryerson.SecureBackup
..
25266 0 5% S 11 155080K 21604K fg u0_a52 com.ryerson.SecureBackup
..
25266 0 0% S 11 155028K 21552K bg u0_a52 com.ryerson.SecureBackup
=====

```

Figure 12. Performance of Backup Application

VIII. CONCLUSION

By using Agile and SCRUM practices in our initiative to develop a light-weight Android application, we have been able to i) construct the software, ii) accommodate changes in requirements and plans and iii) upgrade our technical skills to adapt to technology and tools used in the course of project. We have studied Agile-SCRUM practices that best suit requirements of android software development and applied them in designing a mobile software development methodology. Based on this methodology, we were able to successfully develop a secure backup application on Android platform. The application was developed using important features from Agile-SCRUM methodology such as adaptability to evolving requirements, technically strong development teams, test-driven development approach, effective communication through daily SCRUM meetings, enhanced quality assurance measurements, sub-versioning framework and process-wide reviews and feedback sessions. Use of incremental backup is believed to reduce the network's bandwidth utilization due to minimal data transfer. Performance analysis has resulted in low CPU utilization, which validates light-weight feature of our Android application. Apart from this, complete awareness of physical architecture (limitations) of the mobile terminals and market consciousness are considered essential in avoiding unprecedented risks. Mobile industries need to adapt software development methodologies that maintain a balance between market-oriented and technical activities in order to sustain in today's market.

REFERENCES

- [1] S.Agarwal and A.I.Wasserman, "Mobile application Development: A Developer Survey", submitted for publication, 2010
- [2] J.Dehlinger and J.Dixon, "Mobile application software engineering: Challenges and research directions", in *Workshop on Mobile Software Engineering*, Oct 2011
- [3] Yang-Jae Jeong, Ji-Hyeon Lee, and Gyu-Sang Shin, "Development Process of Mobile Application SW Based on Agile Methodology", *10th International Conference on Advanced Communication Technology*, 2008. ICACT 2008. vol.1, no., pp.362,366, 17-20 Feb. 2008
- [4] V.Rahimian and R.Ramsin, "Designing an agile methodology for mobile software development: A hybrid method engineering approach", *Second International Conference on Research Challenges in Information Science*, 2008. RCIS 2008.vol., no., pp.337,342, 3-6 June 2008
- [5] B.Boehm, "Get ready for agile methods, with care", *Computer*, vol.35, no.1, pp.64,69, Jan 2002 doi: 10.1109/2.976920

- [6] P.Abrahamsson, "Keynote: Mobile software development – the business opportunity of today", *Proceedings of the International Conference on Software Development.*, 2005, pp.20-23 May 2005
- [7] M.Taromirad and R.Ramsin, "Cefam: Comprehensive evaluation framework for agile methodologies", *Software Engineering Workshop*, 2008. SEW'08. 32nd Annual IEEE, pp.195-204, October 2008.
- [8] C.Scharff and R.Verma, "Scrum to support mobile application development projects in a just-in-time learning context", In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 25-31). ACM May 2010
- [9] P. Nicolas, "Introduction to SCRUM Agile Process for Global software development", *Coalesce Network*, pp. 1-9, 2007.
- [10] R. D. Austin, "CMM versus Agile: Methodology wars in software development", *Harvard Business School*, pp. 1-17, 2007
- [11] R.Sriram and S.K.Mathew, "Global software development using agile methodologies: A review of literature", *Management of Innovation and Technology*, ICMIT 2012, pp. 389-393, IEEE Conference, June 2012
- [12] A. Hammershoj, A. Sapuppo & R. Tadayoni, "Challenges for mobile application development", *Intelligence in Next Generation Networks (ICIN)*, 2010 14th International Conference on (pp. 1-8). IEEE
- [13] Feijoo, Pascu, Misuraca and Lusoli, "The Next Paradigm Shift in the Mobile Ecosystem: Mobile social computing and the increasing Relevance of users", *Communications & Strategies*, 2009, (75), 75.
- [14] Cuixiong Hu and Iulian Neamtiu. 2011 "Automating GUI testing for Android applications" *Proceedings of the 6th International Workshop on Automation of Software Test*. AST 2011). ACM, New York, NY, USA, 77-83.DOI=10.1145/1982595.1982612 <http://doi.acm.org/10.1145/1982595.1982612>
- [15] Cuixiong Hu and Iulian Neamtiu. 2011 "Automating GUI testing for Android applications" *Proceedings of the 6th International Workshop on Automation of Software Test*. AST 2011). ACM
- [16] L.Jordan, and P. Greyling, "Introducing SL4A: The Scripting Layer for Android", *Practical Android Projects* (pp. 193-220). Apress 2011
- [17] A.Chan, "Comparison of Python for Android and C-Python", *Review of Business and Technology Research*, Vol. 5, No.1, 2011, ISSN. pp.162-168
- [18] R.C.Martin and M.Martin, "Agile principles, patterns, and practices in C#," Prentice Hall PTR, June 2007
- [19] J.Sutherland and K.Schwaber, "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework", *SCRUM Training Institute*. 2010
- [20] M.Naranjo and R.J.Mathew, "Building Python Modules: Python for Android", Feb 2011, [online] Available: <https://code.google.com/p/python-for-android/wiki/BuildingModules>
- [21] Gartner Inc., "Gartner Report on Smartphone Sales", [online] Available: <http://www.gartner.com/it/page.jsp?id=1622614>
- [22] B.N. Javagal, Ratnakala, S.N. Rajeshwari and M. Nanditha, "Software Development using agile and scrum", in ICACEE. Nov 2012