

ECSE 321 - Intro to Software Engineering  
Release Pipeline Plan - Deliverable 4

Harley Wiltzer  
Camilo Garcia La Rotta  
Jake Shnaidman  
Robert Attard  
Matthew Lesko

March 25, 2017

# Contents

<b>I</b>	<b>Integration</b>	<b>2</b>
1	Description . . . . .	3
2	Tools . . . . .	3
3	Design . . . . .	3
4	Rationale . . . . .	3
<b>II</b>	<b>Build</b>	<b>4</b>
5	Description . . . . .	5
6	Tools . . . . .	5
7	Design . . . . .	5
8	Rationale . . . . .	5
<b>III</b>	<b>Deployment</b>	<b>6</b>
9	Description . . . . .	7
10	Tools . . . . .	7
11	Design . . . . .	7
12	Rationale . . . . .	8

# Part I

## Integration

- 1 Description**
- 2 Tools**
- 3 Design**
- 4 Rationale**

# Part II

## Build

**5 Description**

**6 Tools**

**7 Design**

**8 Rationale**

# Part III

## Deployment

## 9 Description

This phase will be activated once the integrated code has been tested and approved by the preceding phase. At this moment the code is ready to be released to the public, in the case of the Desktop application through a web market or physical means, for the Mobile application through the Android marketplace or an APK sharing forum and for the web application through a production server, such as a hosting web server to ensure its availability on the web. In this section we will go through the tools, design choices and specificities of each platform's deployment phase.

## 10 Tools

Within the paradigm of automatizing and avoiding manual repetitive tasks that is applied throughout the release pipeline, we look now at tools which can help with the process of deployment. These tools must be able to transfer the binaries, libraries and all other required files for the proper function of the product to each platform in a seamless, error-less automated way. All while monitoring the process, gathering metrics which can help troubleshoot any errors or improve future deployments.

TODO ADD TOOLS FOR EACH PLATFORM HERE WITH EXPLANATION (for web check out IIS web deploy as an example)

## 11 Design

The Deployment phase can be split into a set of well delimited activities which cover the entirety of the process. In this section we will describe them with relation to each available platform of the product.

- **Release:** Once the approved code is received from the Build Phase, the global repository containing the three platform specific applications will be tagged with an incremental numerical ID which helps mark the end of the addition of a concrete sequence of fixes and upgrades to the existent applications.

The sub-repositories containing only each target platform application will then be delegated to its specific tool (addressed in the **Tools** section) for it to be deployed to the client. We will discuss this process for each platform:

- **Desktop:** LESKO
  - **Mobile:** LESKO
  - **Web:** Because of the small size of our product, all the required libraries and executable can be found under the sub-repository containing the web application. It suffices to transfer this folder to the web server's repository. It can be achieved through the web deployment tools explained in the previous section LESKOOOO
- **Install/Activate:** This activity is related only to **Desktop** and **Mobile** targets, as the **Web** application is accessed through the web and doesn't host any permanent or temporary files client-side.
    - **Desktop:** The `.jar` file can be placed anywhere the client wants. To avoid the need to handle OS specific file system architectures and scripting languages, once the executable



is downloaded by the client the .jar will reside in the default download folder. From this point, it suffices for the client to run the executable through the method of his choice (Command Line or GUI) to access the application.

- **Mobile:** Once the client downloads the APK the Android application handler will automatically decompress it and install it. The client requires no further action in order to access the now installed application.
- **Uninstall/Deactivate:** In the given case that one of our platforms stops being supported by the development team, a deactivation process starts. This process involves notifying the client of the end of support followed by the deactivation/uninstall of the application module.
  - **Desktop:** LESKO
  - **Mobile:** LESKO
  - **Web:** The simplest target to deactivate, we only require to stop the web server daemon or the physical server itself.
- **Update:** Following the same logic of a full release, a subtag (i.e. x.1) would be given to the global repository containing the updates applications.
  - **Desktop:** The old .jar must be overwritten with the new compiled executable. If the user desires to keep an older version of the application, he simply needs to store the file under different name. Multiple versions can work concurrently.
  - **Mobile:** The android APK handler will receive the new APK and release a notification to all users who have the older version installed.
  - **Web:** Through a blue-green deployment procedure, one of the servers web daemon will be halted, the necessary configuration or script files updated and the service restarted. Once the application is up online again without problem the green server will be taken down to pass through the same procedure. Hence no downtime will be perceivable client-side
- **Version Tracking:** In order to provide a broader coverage of our application for the customers, an archive website will be kept from where all past versions of the application can be stored and retrieved from. This log of past versions will be linked to the GitHub repository containing the open-source code of the application.
- **Monitoring:** Based on the aforementioned monitoring tools, we would keep an extensive log of the current amount of installed applications on all platforms, including their up-time, their error-logs and their amount of transactions. This would help identify and prioritize the requirements for the next iteration of development. In this section of the deployment phase we enter a more continuous analysis of the production application's state. Log analysis tools such as loggly, Splunk, logstash among others would help retrieve useful, tangible metrics on the deployed applications.

## 12 Rationale

TODO : talk bout metrics, monitoring, why deployment phase important : <https://zachholman.com/posts/deploying-software-goals>