**Final Release Documentation preparation:**

Version 4.0 (added marking scheme ( x )

Version 4.1 – expanded external requirement (section 9.1.4)

Update the paragraph index from 9.1

Version4.2 – updated 9.1.3.6 for source code organization tree

Update 9.1.4 other specific requirements for (8) marks

The following is a guideline for preparing your project release report.You should be able to collect most of the material from the results of all your sprints (0 to final). You build this document incrementally based on the results from each of the sprint releases. More importantly, **get everyone involve** in writing and preparing this release document.

1. (1) Project Title, Group Name
   1. EZRyderz, Scrumdog Millionares
2. (1) Your Project website URL (if available)
   1. www.colinbernard.ca
3. (1) Version:
   1. 1.0
4. (2) Project Summary: (or project vision)

* What is the high-level overview of your project?
* What are you trying to accomplish?
* What will the system do when you are done?

1. (2) Problem statements:

Many students and workers commute long distances every day; often to the same location. In Kelowna, transit services are not an option for some students. This results in many commuters having to pay for a car and gas to drive on their own. Because of this there is more traffic and more parking lots are built (as evidenced by H-lot). Our solution will help connect drivers with prospective carpoolers. Drivers will benefit by paying less, or nothing for gas, and carpoolers will have a ride to school or work. We expect to create a usable and valuable resource to cut-down on unnecessary spending and pollution by April 6th.

* 1. Review and compare your original problem statement from the beginning of the project against your final releases
  2. Are there any changes to the desired deliverables?

(you can use the user story map from Sprint 1 vs the final Sprint as reference to illustrate. You don’t want to explain too much here)

* + - modification to the original items (change the nature),
    - exclusion (customer don’t need it)
    - additions (were not on the original list)

1. (3) Team Profile:
   1. Individual strengths, interests, and career goals ( management, presentation/documentation, testing, programming, design and analysis…etc)
      1. Wright your own shit here boys
      2. Zachary Grafton:
      3. Colin Bernard:
      4. Nishant Vyas
      5. Peter Dyczkowski:
      6. Farhan Chowdhury:
   2. Roles and responsibility (Product owner, developers, Scrum master) if the team is following the Agile/Scrum approach strictly. Since I (or your TA) will be your SCRUM master, the person with the most product domain knowledge representing the customer should be the product owner.
      1. Scrum Master: Ryan McQueen
      2. Developers:
         1. Zachary Grafton, Colin Bernard, Nishant Vyas, Peter Dyczkowski, Farhan Chowdhury.

6.2.3 Product Owner:

1. (5) Development Process: (Agile Scrum)
   1. Description of the Agile Scrum process that your team followed throughout the project development.
   2. JIRA has been used for the Project Managements along with GitHub for version control.
   3. Weekly scrum meetings were held. Confluence was used for documentation tracking. Work was logged on JIRA and code was pushed to GitHub for review/testing. Sprint demonstrations presented at the end of the scheduled sprint time (two weeks). This leads to the release build.
2. (3) Glossary of terms:

List important terms and their definitions to avoid ambiguity, ensure consistency, and boost readability.

1. Results:
   1. (25) Requirement gathering, analysis and design:
      1. (5) Product backlog (of the entire project) (Table of product backlog)
      2. (10) Functional requirement / Specification:

(What is functional requirement? be sure to read the documentation on Project Documentation flow during software development (Project Documentation preparation6.docx) )

Your GUI design and flow of operation development (from sprint 0 to sprint x (last) can be captured here.

* + - 1. Categorize all the user stories from Sprint 0 to X (final)
      2. You can include User Interface requirement specifications in this section. *Some companies keep it in a separate section from non-functional and functional.*
    1. (10) Non Functional requirement / specification
       1. Categorize all the user stories from Sprint 0 to X (final)
       2. Architecture / components / layers
  + DO NOT just show table, diagrams, UML (activities, class, sequence..) figures without a description.
    - 1. Development environments. (eg Php on Laravel, or JavaFX on Eclipse, Python on Django, …etc)
      2. Builds (procedure of integrating and building the final product.
      3. Installation (requirement, preparation, and steps)
      4. Source code organization tree: How are your source code folder organized?

The following is an example from Maven standard directory layout specific to Java language

src/main/java Application/Library sources

src/main/resources Application/Library resources

src/main/filters Resource filter files

src/main/assembly Assembly descriptors

src/main/config Configuration files

src/main/webapp Web application sources

src/test/java Test sources

src/test/resources Test resources

src/test/filters Test resource filter files

src/site Site

src/documentation documentation (design doc or release doc)

LICENSE.txt Project's license

NOTICE.txt Notices and attributions required by libraries

README.txt Project's readme

* + 1. (8) Other Specific Requirements
       1. External Requirements
  + Database type (SQL, MySQL, PostgreSQL, Oracle, …etc)
  + Support platforms /frameworks (AWS, JavaFX, DART, Oracle JRE, …etc)
  + OS (MacOS version, Windows version, Android xx, iOS, …)
  + Game platform…
    - 1. Hardware Requirements
  + Minimum RAM size (16 G?) to run efficiently?
  + Minimum disk spaces (20G?)
  + Any requirements for HD read/write performance.
    - 1. Communication Requirements
  + Web Browser (Chrome (latest), IE ( 10, 11,) MS Edge (Latest) …
  1. (15) Agile development planning

This section shows how you use the Agile development approach in planning your project:

* + 1. (10) Sprint planning summary

Use the ongoing user story map to show your user story planning and prioritization

* + 1. (10) RoadMap from Sprint 0 to Sprint (final)
  + Theme / goals of each sprint

use your sprint theme(s) to show your project planning direction aim towards the final product release

* + Table of user stories associated with each sprint
  + (it would be good to describe the discrepancy between the original and sprint release outcome) – you can get this from the JIRA sprint release reports.
  1. (5) Burn down chart history from Sprint 0 to final

With a brief descriptions (focusing on lessons learned )

* + Use the material from all your sprints release history
  1. (3) Sprint velocity and hours.
     + 1. With a brief description
       2. Summary of user points per sprint
       3. Summary of hours per sprint
  2. (5) Risk tracking table: (for each sprint)
  3. (3) Summary of Retrospectives from each sprint

What did we do well and should carry on doing?

What did we do that we should avoid in future?

What were the surprises (unusual items) during the sprint?

* + 1. Table of retrospectives
  1. (2) SCRUM meeting notes samples ( two from each sprint)

1. (14) Conclusions

Give a summary of your current sprint results focusing on:

(Your final project report should be the summary of the following from each of the sprint report collected incrementally)

* + Planning and estimation.
  + Requirement changes
  + Analysis and design changes
  + Surprise discovered and your approached of handling it.
  + Lessons learned

1. (2) References:
   * List of references and URL of material used in this project.

Other reference on writing Agile project reports:

Rick Freeman describes reports that project managers typically create at the end of each iteration of Agile Scrum Projects

<http://www.techrepublic.com/blog/tech-decision-maker/agile-reporting-methods-for-project-managers/>