

Post-Mortem Report TOC

1. Project

- A. Description
 - A.i. Project Name:
 - A.ii. Client:
 - A.iii. Project Manager:
 - A.iv. Solutions Architect:
 - A.v. Start Date:
 - A.vi. Completion Date:
- B. Project Overview *[Describe the project in detail.*
 - B.i. Discuss the project charter
 - B.ii. What was the project success criterion?
 - B.iii. etc.

2. Performance

- A. Key Accomplishments *[List and describe key project accomplishments. Explain elements that worked well and why. Consider listing them in order of importance. Be specific.]*
 - A.i. What went right?
 - A.ii. What worked well?
 - A.iii. What was found to be particularly useful?
 - A.iv. Project highlights
- B. Key Problem Areas *[List problem areas experienced throughout the project. Be specific.]*
 - B.i. What went wrong?
 - B.ii. What project processes didn't work well?
 - B.iii. What specific processes caused problems?
 - B.iv. What were the effects of key problems areas (i.e. on budget, schedule, etc.)?
 - B.v. Technical challenges
- C. Risk Management *[List project risks that have been mitigated and those that are still outstanding and need to be managed.]*
 - C.i. Project risks that have been mitigated:
 - C.ii. Outstanding project risks that need to be managed:
- D. Overall Project Assessment *[Score/rank the overall project assessment according to the measures provided. A 10 indicates excellent, whereas a 1 indicates very poor.]*
- E. Additional Comments:
 - E.i. Other general comments about the project, project progress, etc.

3. Key Lessons Learned

- A. Lessons Learned *[Summarize and describe the key lessons and takeaways from the project. Be sure to include new processes or best practices that may have been developed as a result of this project and to discuss areas that could have been improved, as well as how (i.e. describe the problem and suggested solution for improvement).]*
- B. Post Project Tasks/Future Considerations *[List and describe, in detail, all future considerations and work that needs to be done with respect to the project.]*
 - B.i. Ongoing development and maintenance considerations
 - B.ii. What actions have yet to be completed and who is responsible for them?
 - B.iii. Is there anything still outstanding or that will take time to realize? (i.e. in some instances the full project deliverables will not be realized immediately)

Post Mortem Analysis: notSept (wednesday - Team 9 - Homy)

Project:

A. Description:

- A.I: Project Name: notSept.
- A.II: Client: Homy.
- A.III: Project Manager: Homy.
- A.IV: Solutions Architect: Liam Ryan.
- A.V: Start Date: March/2017
- A.VI: Completion Date: May/2017 (N/A depending on how you look at it)

B. Project Overview:

B.I: Project Charter:

Team members/Stakeholders:

Liam Ryan
Adam Simankowicz
Marcela Klocker
Jacob (Nhat) Ho
Homy.

Project outcomes:

The design and implementation of an appointment booking system, for further details see part A-C specification sheets.

B.II: Success criterion:

Functional software delivered.
Software easy to use, learn and navigate.
Everyone still alive at the end.
Product delivered on time.
Product delivered to spec.

Performance:

A. Key Accomplishments:

A.I: What went right:

A (semi) functional product was delivered. And everyone is still alive.
The product was delivered on time.

A.II: What worked well:

The database worked well due to the effort of the team member assigned to it. The tools used, specifically git hub and trello served their purpose well for those who made use of them, and any shortcoming was due to the human factor.

A.III: What was useful:

The tools used were useful for those who used them. The short meetings were also useful for those who attended. Project progress was easily shared and future planning was easy to communicate. Trello and github were also good for progress tracking and future planning.

B. Key problem areas:

B.I: What went wrong:

The main problem was teamwork, communication, lack of motivation and inadequate skill set. These shortcomings produced a sub standard end result, that would have been much improved with more input in the design process, and more willingness to learn the skills and tools required to complete the project.

B.II: processes that didn't work well:

While the stand up meetings were useful for those that turned up and participated, it was still lacking because some people chose not to attend and participate. This severely reduced the effectiveness of these meetings and in turn reduced the quality of the end product.

The original plan was to do the project following the TDD principles, this however did not end up working out. Because people were unfamiliar with unit testing and unwilling to learn or take responsibility when they were either assigned, or claimed the job the testing in the end not only ended up far from TDD but also far from adequate.

B.III: Specific processes that caused problems:

The processes themselves were generally fine, and may work quite well when used in a professional environment, but when used by unmotivated student with no fear or understanding of any negative consequences, they fall far short. One would assume that people actually working in the field are doing it because it what they want to do, but this is not the case for student. So when these processes are used by the unmotivated and unconcerned the fall somewhat short.

B.IV: Effects of key problem areas:

The effects are generally obvious, in that the end product was far below the standard it should be. This was because of a lack of input, and an increased load on the few people actually doing the work. These problems arose not because of any of the processes or tools used, but because of those who refused to use them, and participate.

B.V: Technical challenges:

Some technical challenges were getting familiar with the technologies and techniques used. Other issues were with the order of things. The project started off with the command line, migrated to a gui and finally was migrated to maven. This was the opposite of how the process should have been. The project should have been built using maven from the ground up, and should have been GUI based from the beginning. Taking these things on after the fact, and in the wrong order created issues with the project. Firstly converting from the command line to a gui, with little knowledge of the gui technology

being used made the project setup quite bad, and hard to deal with. This would have been avoided by starting with a GUI. The project was also set up fairly badly, which would have been avoided by making it a maven project from the beginning.

C: Risk Management:

C.I: Mitigated risks:

Poor team cohesion.

Poor team communication.

Poor time management.

C.II: Outstanding risks:

D: Overall project Assessment

Performance against project goals/objectives:	5
Performance against planned schedule:	6
Performance against quality goals:	3
Performance against planned budget:	2
Adherence to scope:	7
Project planning:	7
Resource management:	5
Project management:	4
Development:	6
Communication:	2
Team cooperation:	2
Project deliverable(s):	6

Key lessons learned:

A: lessons learned:

Don't work with students unless they are mature age.

B: Post project tasks/future considerations:

B.I: Ongoing development and maintenance:

Expand add business functionality

(times, services, multiple businesses)

Give owner ability to add/cancel bookings

Improve owner customisation options

Improve GUI.

B.II: Incomplete/responsibilities:

As above - No one.

Testing - no one.

B.III: anything outstanding:

As Above.