SWEN 6301: Assignment #1

Due on September 28, 2019 at 2:00 PM

Problem 1

The following case study and accompanied references describes the Ariane 5 launch accident that occurred on the initial launch of the Ariane 5 rocket, a launcher developed by the European Space Agency. The rocket exploded shortly after take-off and the subsequent inquiry showed that this was due to a *fault in the software in the inertial navigation system.*

In June 1996, the new Ariane 5 rocket was launched on its maiden flight. It carried a payload of scientific satellites. Ariane 5 was commercially very significant for the European Space Agency as it could carry a much heavier payload than Ariane 4 series of launchers. Thirty seven seconds into the flight, software in the inertial navigation system, whose *software was reused from Ariane 4*, shut down causing incorrect signals to be sent to the engines. These swiveled in such a way that uncontrollable stresses were placed on the rocket and it started to break up. Ground controllers initiated self-destruct and the rocket and payload was destroyed.

A subsequent inquiry showed that the cause of the failure was that the software in the inertial reference system shut itself down because of an unhandled numeric exception (integer overflow). There was a backup software system but this was not diverse so it failed in the same way.

More technical details on the incident are available at: `http://www.rvs.uni-bielefeld.de/publications/Reports/ariane.html`

Use your understanding of the problem, its context, the technical details, and your software engineering experience gained so far to clearly and precisely answer the following questions:

1. What was the real cause of the Ariane 5 launch accident?
2. What can you say about the type of the software used in Ariane 5?
3. What software development paradigm is much suitable for developing Ariane 5 control software?
4. Which of the software development activities you think that it overlooked/missed the actual cause of Ariane 5 failure? Explain your idea with some details.
5. Explain how heterogeneity of the operating environment and software reuse played an important rule in Arian 5 failure?
6. List 3 possible detailed mitigation plans from the software development process perspective to avoid such disasters in the future?

Problem 2

1. Discuss how a team or a company can measure the success of its adopted software development process? *Software quality is not enough measure.* You can discuss any metrics widely used or come up with your own combination of important factors and how they relate the success of an adopted software development process.
2. In agile development, focusing on delivering new functionality may result in increased technical debt. Discuss in detail possible mitigation plan to avoid technical debt?

Problem 3

Kanban[1] and Scrum[2] are both frameworks for Agile software development. They both take large, complex tasks and break them down into smaller chunks. Kanban and Scrum also work toward continual improvement and optimization of the process, and want to keep work highly visible. While both Kanban and Scrum are very adaptive, Scrum is more rigid than Kanban. Scrum has more constraints, whereas Kanban is more flexible.
Read more on both frameworks online and answer the following questions:

1. List at least 2 software examples or types that can be best developed with Kanban and why?
2. Discuss 3 differences between Scrum and Kanban? and how each difference may affect the quality and the technical debt of the delivered software product?

[1] https://en.wikipedia.org/wiki/Kanban
[2] https://en.wikipedia.org/wiki/Scrum_(software_development)