Erick Giovanni Guzmán Sierra
A01566171
Análisis y modelación de sistemas de software

Software Engineering Bibliography

1. **From the suggested books from the course presentations. list the top 3 that you would like to read, and explain why you selected them.**

Clean Code.- Coding is one of the things that I most enjoy doing and I want to improve as much as I can following some defined standards and utilizing the tips and tricks that Uncle Bob, a respected man in the industry, can provide.
Code Complete.- As stated before this book will help me improve in the craft and taking advice from different experts in the industry and taking the best of both worlds will only help me to become an even better coder.
Software Engineering by Ian Sommerville.- This book has been recommended to me by several teachers by now and I really want to read it because I'm aware that coding isn't all there is in a system. The way things are implemented, the documentation and good management of the team are what makes or breaks a project. I want to improve my skills not only as a coder but also as a software engineer and this book will help me accomplish that.

2. ***Suggest other 3 books related to the course topics that are not part of the suggested books, and indicate why***

Clean Architecture by "Uncle Bob".- I had this book on my wishlist for a long time (and I was browsing through it for this assignment) and I just realized that it's written by Uncle Bob. The book is a complement to his other two popular books, clean code and the cleaner coder, but it's aimed to aspiring or current software architectures. Looks like a good read to complement the other books I want to read.

The art of Game Design by Jesse Schell.- This is a book that I currently have but have procrastinated in reading. It focus on how to design a game depending on what you want to achieve in your players (users). I consider it related to the course topics because it has some similarities as the process of designing a "traditional" application but oriented to video games. Designing first based on the requirements then start creating/coding.

Dive Into Design Patterns.- I came across this book when I was researching my design pattern. It seems to explain design patterns and provide examples in 7 different programming languages. It has cartoony but very explanatory illustrations to convey what each design pattern is able to do. From my research it looks very beginner oriented but as a first dive into design patterns it looks like a fun and easy read.

3. ***List 3 youtube channels related to the course topics***

MIT OpenCourseWare (https://www.youtube.com/user/MIT/featured).- This is one of the MIT youtube channels that offers some of their classes for everyone to watch. The channel features material from every of their courses not only software but there's a lot of good material on SOLID, patterns, etc.

Christopher Okhravi
(https://www.youtube.com/channel/UcbF-4yQQAWw-UnuCd2Azfzg/featured).- This channel doesn't

have a lot of videos but its catalog is growing. This channel is from a PhD student and explain things as if he was your friend or peer. This approach may help tackle some daunting topics because it's explained in a "friendly" manner.

Web Dev Simplified (https://www.youtube.com/channel/UCFbNIlppjAuEX4znoulh0Cw/playlists).- This channels, as its name suggest, is more focused on Web development but it's pretty active and uploads videos on design patterns and software design. A great variety of content.

4. ***List 3 blogs related to the course topics***

https://refactoring.guru/ .- This page is the creator of the book "Dive into design patterns". In the page there are some design patterns with easy to understand illustrations and code examples in different languages.There's also an entire section dedicated to teach you how to refactor code and how to avoid bad coding in the first place.

https://sourcemaking.com/ .- This page is the brother of refactoring guru. This page also helped with the creation of the book "Dive into design patterns". This page also features some illustrations and easy to understand explanations but is a little bit more in depth. This page contains an entire section dedicated to UML and also has a section called the "antipatterns", which are common mistakes done and how to prevent them.

https://www.geeksforgeeks.org/ .- This is an all around page with an enormous quantity of content from various topics.

5. ***Read the chapter 3 from the Mythical Man-month (The surgical team) and give your thoughts about that chapter***

To start things of, I was amused by the way they implemented things that clearly work in other lines of work and apply them to software development. To have a single person do all the design and have a team supporting him on achieving the desired results sounds promising and productive to say the least. The only problem I see with this methodology is, in my opinion, the lack of aspiring for a better position (Ex. From toolsmith to surgeon). Since evetyone is just focused on a very specific task, and a supporting one that is, the odds of a particular person to get promoted are slim. The surgeon will always be surgeon, and the same goes for all the other positions. This may result in the waste of possible potential from members of a surgical team.