Proposal Regarding Addition For Newer Books At FAST NUCES Library

Saif Ul Islam, Taha Warsi
k180307@nu.edu.pk, k181174@nu.edu.pk

May 2021

1 Introduction

Proposed Administrator: Ms. Attya Shahid, Librarian, FAST NUCES, Karachi Main Campus.

One of the fastest growing trends in the technological world of today has been the advancement of the idea of building quicker products, building more suitable *"Minimum Viable Products"* (MVPs), and understanding how to build longer lasting products within a smaller team while maintaining the best of practices. It is all too common to hear of a group who is working on a new, great idea, only to succumb to pitfalls of (**a**) Technical Debt, (**b**) Poor Design In Software, (**c**) Unstructured Teams, (**d**) Unhealthy work environments due to lack of connection between employees, (**e**) Rush to act like a *"FANG"* company, despite not even experiencing the same set of problems as a *"FANG"* company, (**f**) Unable to understand how to take better design decisions based on requirement from users.

An ever growing amount of indirect contribution to a certain of these problems have led to lower customer satisfaction rates, consideration of building secure and reliable systems, falling behind due to overwhelming amount of technical complexity that can arise.

2 Statement Of Problem

This problem has often occurred due to a lack of certain misunderstandings within the technical community about how technical certain aspects of Software Engineering could do, and the myth that often arises is that bad products arise due to lower level of understanding of technical, whose only solution is to develop better technical skills. While this is true to some certain degree, it would be misleading to say that one factor entirely effects this whole process.

This mindset often leads to imbalanced sets of learning, where products are now built to demonstrate technical capabilities, rather than build on the emphasis on what the user would actually want or expect from certain piece of software.

Indirectly, this correlates to a lack of people and team understanding of how roles are organized, built, and developed to try to solve certain specific problems, and how important the involvement of customers is who will actually use the product. A balance lacks generally between building a product, an MVP, shipping it out through the front door, and between understanding what it takes to develop longer lasting products that can last years, and what decisions to make on the basis of that.

3 Proposed Solution

For the solution of these certain problems, it is our firm belief that the problem can be slowly solved over a period of a few years by starting at one of the main core roots of the problem, or where the misunderstanding rises from, and that would be from a University or College level perspective. The idea is to develop a mindset where students are more free and able to openly share their thoughts, ideas, and new innovative developments, without necessarily having to fear (**a**) competition or (**b**) going through self judgement of how others might think poorly of him/her.

A first step that can be taken could be about introducing a new set of books within the FAST NUCES Library at the Main Campus that go through some of the problems described, and by promoting these books maybe somewhere that can be seen at the main view entrances of the library. Our suggestions for these books are,

1. "Software Engineering at Google: Lessons Learned from Programming

Over Time 1st Edition", by Titus Winters, Tom Manshreck, Hyrum Wright.

This book is published by *"O'Reilly Media"*, as the 1st edition on the date **March 24, 2020**. Priced at $23.23 - $27.73, with $13.98 in shipping directly from Amazon

The main ideas of this book teach about,

(a) How time affects the sustainability of software and how to make your code resilient over time

(b) How scale affects the viability of software practices within an engineering organization

(c) What trade-offs a typical engineer needs to make when evaluating design and development decisions.

Details of this book are,

(a) Language : English

(b) Paperback : 602 pages

(c) ISBN-10 : 1492082791

(d) ISBN-13 : 978-1492082798

(e) Item Weight : 2.08 pounds

(f) Dimensions : 7 x 1.22 x 9.19 inches

2. "Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems 1st Edition", by Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea, Adam Stubblefield.

   This book is published by *"O'Reilly Media"*, as the 1st edition on the date **April 7, 2020**. Priced at $40.05- $45.66, with $13.98 in shipping directly from Amazon

   The main ideas of this book teach about,

   (a) Design strategies

   (b) Recommendations for coding, testing, and debugging practices

   (c) Strategies to prepare for, respond to, and recover from incidents

(d) Cultural best practices that help teams across your organization collaborate effectively

The details of this book are,

(a) Language : English
(b) Paperback : 558 pages
(c) ISBN-10 : 1492083127
(d) ISBN-13 : 978-1492083122
(e) Item Weight : 1.93 pounds
(f) Dimensions : 7 x 1.13 x 9.19 inches

3. "Mythical Man-Month, The: Essays on Software Engineering, Anniversary Edition Anniversary Edition", by Frederick Brooks Jr.

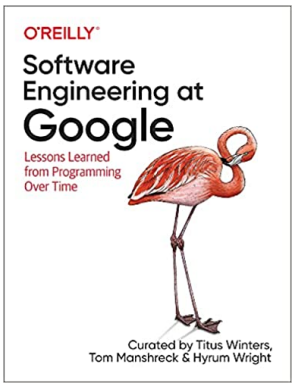
This book is published by *"Addison-Wesley Professional"*, as the **Anniversary edition** on the date **August 2, 1995**. Priced at $19.18, with $13.98 in shipping directly from Amazon

This book teaches about,

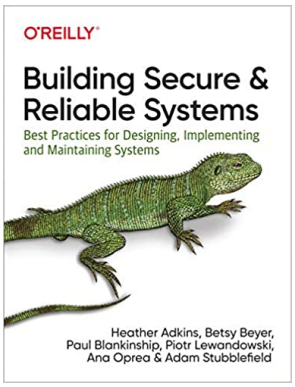
(a) The little aspects of human nature that effect the division of labor, and how to develop the time, environment, and space for that
(b) Psychological aspects of Software Engineering
(c) Possible management problems within the system design

The details of this book are,

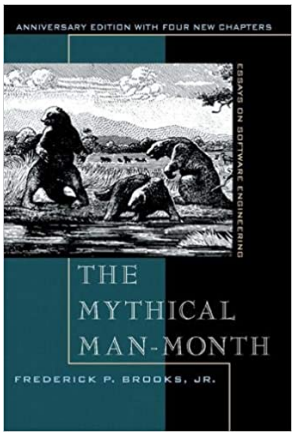
(a) Language : English
(b) Paperback : 336 pages
(c) ISBN-10 : 9780201835953
(d) ISBN-13 : 978-0201835953
(e) Item Weight : 1.03 pounds
(f) Dimensions : 9.07 x 6.11 x 0.75 inches



(a) Software Engineering at Google: Lessons Learned from Programming Over Time 1st Edition



(b) Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems 1st Edition



(c) Mythical Man-Month, The: Essays on Software Engineering, Anniversary Edition Anniversary Edition

4 Scope Of New Books

As stated with each book above, the main underlying idea is to help students understand the softer side of *Software Engineering* and *IT Development*. Each of the book touches on to some aspect on how to develop large scale infrastructure, either from a technical perspective, a team perspective, or a management perspective.

5 My Qualifications

We are undergrads at FAST NUCES, Karachi, currently enrolled in our 6th semester here. I have technical expertise with Data Science, Data Engineering, Web Development, Cloud, and DevOps, and I have spent some time reading at least of the books mentioned above, and I strongly that some of the ideas discussed above need to be shared and realized within other groups of students as well. It helped me to understand how to make project, assignments that not only "work", but also help others see easily what has been accomplished and what can be improved to showcase and develop the visibility of the built features and functionality.

6 Conclusion

In the end, clearly, action is needed to reduce the set of myths and set of ideas about what it takes to be a greater developer, and to ease the understanding that it takes time, effort, and the capacity to learn more innovative ideas from others and share your own perceptions of how things should work, or what you expect things to work out. We currently are in one of the most critical years of development, with a greater demand increasing every single day. While it may no longer be ideal to consider yourself to be always *"exceptional"*, in fact, that has hurt the IT industry more than it has helped, it is always useful to consider having a learning mentality in order to not flat line your career growth and expectations, and how to work on the next most ingenious products and objectives.