Unified and Stable Project: "Ushering in the Future"

Mohamed E. Fayad[1](✉), Gaurav Kuppa[1], and David Hamu[2]

[1] San Jose State University, San Jose, CA, USA
{m.fayad, gaurav.kuppa}@sjsu.edu
[2] Liberty Consulting, St. Petersburg, AZ, USA
dave.hamu@gmail.com

Abstract. Why do we need projects? What is the need to carry out a project? What is the ultimate-goal of any project? These are a few thoughts that crop up in one's mind while trying to develop and implement any project. The motivations of this paper are to establish the unified functional and non-functional requirements of AnyProject for the first time. Our goal is to enable unification for AnyProject pattern leading to the creation of a stable pattern language and to enable usability for applications across numerous domains and apply for an unlimited number of scenarios. The idea is to compare the existing or traditional model (TM) of a project and software stability model (SSM) of any project using non-functional requirements as criteria for comparison, evaluation, and measurements. The significant findings are: (1) The TM can only be applied to one scenario, unlike the AnyProject Pattern, which can be effectively applied to unlimited scenarios; (2) The TM requires high maintenance costs and limited scalability. On the other hand, the SSM definitively cuts down on costs because it is cognitive knowledge; and (3) In the measurability, the TM fails to model the essential properties of AnyProject. However, the stable model delineates essential properties. As a result, the SSM enables stability and unlimited applicability.

Keywords: Project definitions · TM · SSM · UML · Abstraction · Applicability · Impact · AnyProject stable design pattern

1 Introduction

A project is defined as an individual or collaborative enterprise that is carefully planned and designed to achieve a goal [1]. Projects have great importance in all domains because innovation and discovery are initiated by a large undertaking in the form of a project. The rapid integration of multidisciplinary fields with technology brings about a new era of collaboration and fusion of ideas that have never been experienced before. Due to the breadth of possible projects, the conventional definitions of "project" are incomplete. Not to worry, herein, we provide a complete and widely applicable definition for any project.

Existing or traditional models (TM) of a project are representations that are vastly different for each deployment in an application context. Consequently, TMs have

K. Arai et al. (Eds.): FTC 2019, AISC 1070, pp. 641–651, 2020.
https://doi.org/10.1007/978-3-030-32523-7_47

proven to be ineffective in achieving any reuse even across a very similar application context, and much of the work will need to be reinvented for subsequent projects. Unnecessary development and maintenance time are spent developing new models. In comparison, a software stability model (SSM) [2–5] defines and uses high-level relationships to illustrate relationships between components that can be used to describe the core aspects of any project. Using the concepts of Enduring Business Themes (EBTs) and Business Objects (BOs), we can construct the SSM. We compare the TM of a project and the SSM of any project pattern using non-functional requirements, the critical attributes and criteria of a system application, as criteria for comparison, evaluation, and measurements.

This paper is organized by 5 major sections: Sect. 2 provides scenarios of any project in use; Sect. 3 illustrates corresponding TM; Sects. 4 to 8 portrays a weighted comparison between the stable, unified model and traditional models based on non-functional requirements and qualitative and quantitative measurements; Sect. 9 presents further discussion about AnyProject, and Sect. 10 presents a conclusion.

2 The Problem

In a world driven by productivity, innovation is heavily project-based. Unfortunately, there is no clarity as to what a project is. Every definition of a project portrays a different aspect of a project. This leaves no cohesive, unified definition for any project. More specifically, all definitions of a project do not cohesively piece together what a project is, nor do the conventional definitions identify the functional requirements, and how to use a project, or convey the true nonfunctional requirements.

In contemporary business and science domains, a project is an individual or collaborative enterprise, possibly involving research or design, that is carefully planned to achieve an aim and goal [6]. On the other hand, a project is also defined as a set of interrelated tasks to be executed over a fixed period and within certain cost constraints and other limitations [7].

The subtle but glaring differences between definitions exist in many forms. Beyond the semantic differences, the two definitions are different in what they convey. The first definition identifies a project to be an “undertaking with a goal” while the second definition determines a project to be “a set of interrelated tasks with limitations.” We have not established a universal definition for all projects. The problem extends to all existing definitions of a project. More broadly, there is no unified and applicable definition for AnyProject and this paper aims to illustrate that.

3 Context

Scenario #1: The Manhattan Project

The Manhattan Project is a famous example. It was an effort during World War II in the United States to develop the first nuclear weapon. This top-secret project lasted from 1939 until 1945 and utilized various resources under the direction of General Leslie R. Groves, Deputy Chief of Construction of the U.S. Army Corps of Engineers.

It resulted successfully the production of nuclear bombs used in WW II The Manhattan Project was an effort to make the nuclear bomb done by the US Government and the US Army in the middle of the 20th century. After spending considerable time and money and by utilization of advanced laboratories and knowledgeable personnel, the project was completed successfully in 1945 [9].

Scenario #2: Boeing B-29 Super-fortress
“The Boeing B-29 Super-fortress is a four-engine propeller-driven heavy bomber designed by Boeing, which was flown primarily by the United States during World War II and the Korean War. It was one of the largest aircraft operational during World War II and featured state-of-the-art technology. Including design and production – at over $3 billion – it was the most expensive weapons project in the war, exceeding the $1.9 billion cost of the Manhattan Project—using the value of dollars in 1945. Innovations introduced included a pressurized cabin, dual-wheeled, tricycle landing gear, and an analog computer-controlled fire-control system directing four remote machine gun turrets that could be operated by a single gunner and a fire-control officer. A manned tail gun installation was semi-remote” [8].

4 Traditional Model

The following traditional models which are business as usual in Figs. 1 and 2 will illustrate the two specific projects one scenario per model.

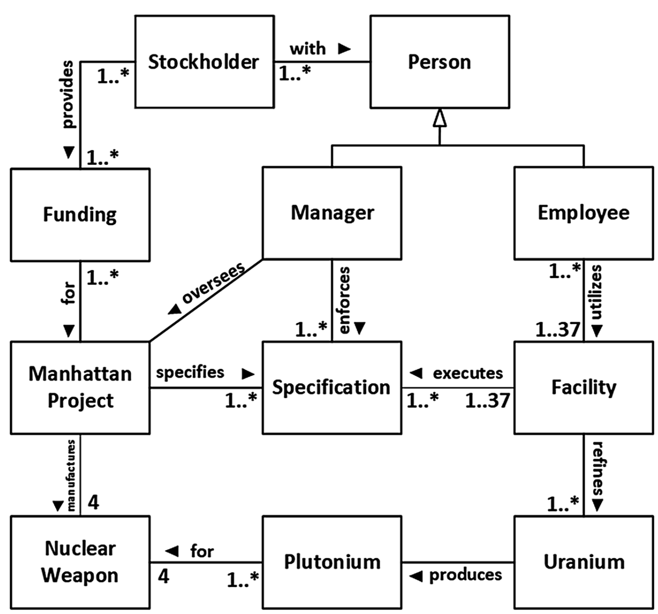

Fig. 1. The manhattan project traditional model

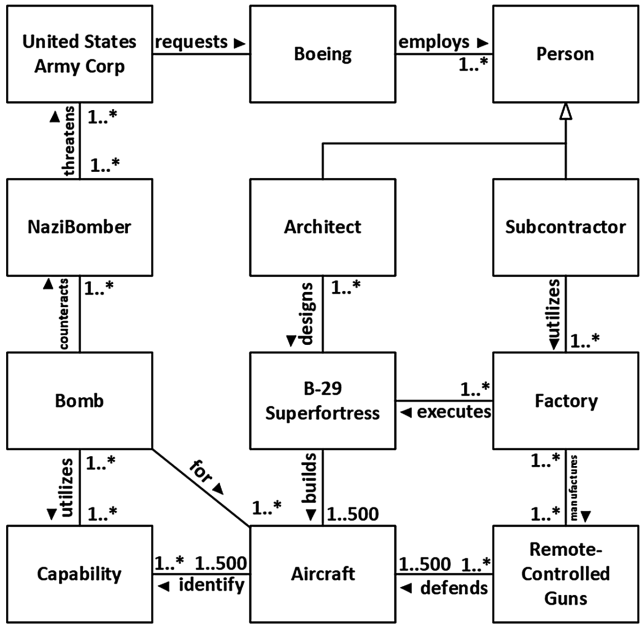

Fig. 2. Boeing B-29 superfortress traditional model

5 Stable Model

The stable project pattern is unified and stable to build on top of its unlimited applications, whereas TM is only suitable for a single application.

5.1 Functional Requirements

Figure 3 illustrates the functional requirements of any project:

1. Need is the goal of AnyProject, i.e., it satisfies one more many (AnyProject).
2. One or more project (AnyProject) contains one or more tasks (AnyTask).
3. One or more task (AnyTask) reaches one or more milestone (AnyMilestone).
4. One or more party (AnyParty) such as human, organization, etc., defines or manages Need for a project.
5. One or more party (AnyParty) defines one or more criteria (AnyCriteria) for the project.
6. One or more party (AnyParty) examines one or more milestones (AnyMilestone) defined for a project.
7. One or more project (AnyProject) requires one or more resources (AnyResource).
8. One or more party (AnyParty) provides/controls one or more resources (AnyResource).
9. One or more criteria (AnyCriteria) influences one or more (AnyResource).

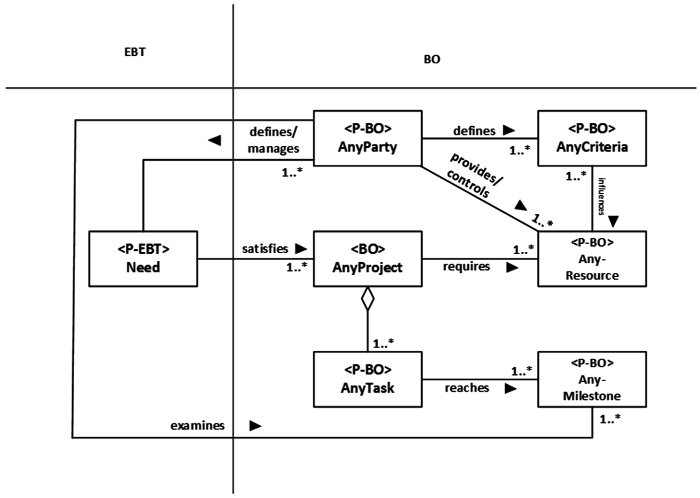

Fig. 3. AnyProject stable design pattern

5.2 Non-functional Requirements

In the following section, we present the non-functional requirements of AnyProject:

1. **Practical:** AnyProject should be practical in completing AnyTask as a part of satisfying the ultimate Need of AnyProject. A practical project is a very important quality factor because it highlights the usefulness of a project in the world and the innovation that the project embodies. Also, a project's practicality is correlated with its critical value to a larger movement. In total, a project must be practical, useful, and provide some critical value to a larger movement in the world.

2. **Doable:** AnyProject is doable if it has AnyTask and AnyMilestone laid out with AnyParty who are responsible for management and execution. In other words, a project must have a systematic approach to achieving a specific milestone. This results in a project that can be done through the given resources and limitations that inherently exist.

3. **Manageable:** AnyProject must have AnyParty who manages the project and enforces AnyCriteria to ensure quality output. A project is deemed manageable if its criteria are measurably satisfied. The constant evaluation and re-calibration of the project's goals, criteria, and outlook result in a manageable project.

4. **Growth:** AnyProject must lay out several AnyTasks that will progressively pile up and satisfy AnyMilestone. In other words, the advancement of a project must lie in the productive nature of completing tasks such that the larger milestone is met. In the same way, these completions of milestones will satisfy the need.

6 Applicability

Application 1: The Manhattan Project
The United States (AnyParty), United Kingdom (AnyParty), and Canada (AnyParty) worked on The Manhattan Project (Fig. 4) (AnyProject) from 1942 to 1946. This project was a $2 billion (AnyResource) undertaking during the World War II in attempts to create the world's first nuclear weapon (AnyMilestone) before Germany's nuclear weapon project was completed (AnyCriteria). The Military Major (AnyParty) managed this top-secret project and employed up to 130,000 people (AnyResource) at 37 facilities (AnyResource) across the nation. These employees conducted research on enriching and separating uranium (AnyTask), producing plutonium from uranium (AnyTask), and gathering intelligence on the German nuclear weapon project (AnyTask).

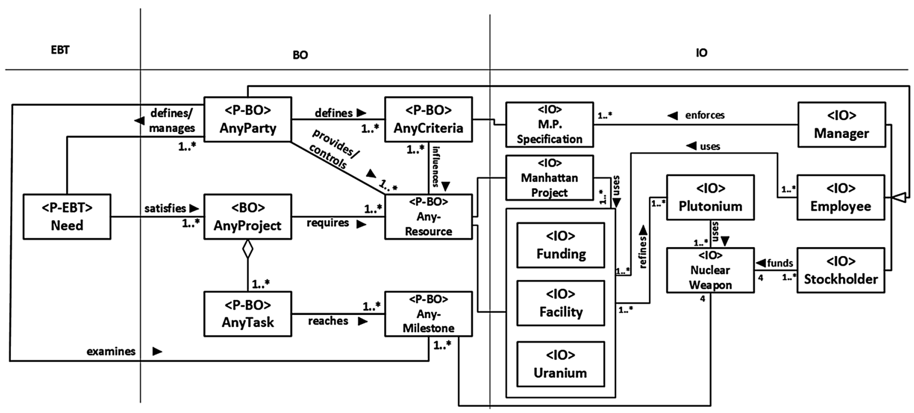

Fig. 4. The manhattan project application

Application 2: Boeing B-29 Superfortress Project
Boeing (AnyParty) began to design the B-29 Superfortress (AnyProject) (Fig. 5) in 1938 in response to a request from the United States Army Air Corps (AnyParty). This request detailed that the aircraft should be capable of delivering 20,000 lb of bombs (AnyCriteria) from a distance of 2667 miles (AnyCriteria), and is capable of flying at a speed of 400 mph (AnyCriteria). Henry H. Arnold (AnyParty) led the production of a new bomber flight to counter the production of Nazi nuclear bombers (AnyMilestone). After getting the proposal approved, Boeing received a production order for 500 aircraft (AnyMilestone). The manufacturing process involved operating plants at four different locations and employing thousands of subcontractors. The B-29 Superfortress had many new features such as remote-controlled guns (AnyCriteria) and pressurized and connected crew areas (AnyCriteria).

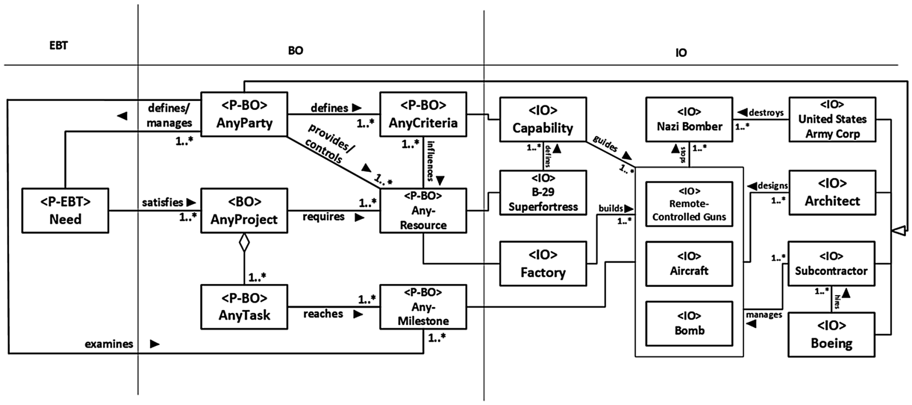

Fig. 5. Boeing B-29 superfortress project application

7 Scenarios in Table

See Table 1

Table 1. Five scenario table

EBT	BOs	App-1 – Directing a Movie	App-2 – Presidential Campaign	App-3 – Southern Border Wall	App-4 – Building Automatic Cars	App-5 – Hacking a Bank
Need	AnyParty	Director Pixar Studio	United States Voters Congress	President Senate Congress	Tesla United States	Wells Fargo Hacker
	AnyCriteria	Profit	Voter Popularity Political Experience	Protection Support	Air Bags	Stealth
	AnyProject	Movie	Campaign	Wall	Automatic Cars	Hack
	AnyResource	Performers Money Movie Set	Money Political Support Endorsements	Money Voter Support	Technology Car	Password
	AnyTask	Direct Movie	Voter Rally Town Hall	Construction	Testing	Virus
	AnyMilestone	Filming Production	Win Election	Approval	Release of Car	Infiltration

8 Weighted Comparison

The criteria were selected based on the non-functional requirements of any project which are shown in Table 2. Each of the non-functional requirements, practical, doable, manageable, growth, have an equal weighting.

Table 2. Weighted comparison of traditional and stable project model

Criteria/Property	W	Description for TM	TM W	Description for SSM	SSMW
Practical	25	TM has elements of practicality because it is constructed from tangible objects, which leads to the easy implementation of a scenario	20	SSM builds the foundation of core knowledge, which applies to all scenarios of AnyProject; therefore, the SSM is more practical	23
Doable	25	TM is doable because its construction is based on tangibility. Tangible objects can be constructed and changed	18	SSM builds on top of the defined core knowledge which extends to the IOs, which are tangible objects that can be easily implemented	25
Manageable	25	The TM is based on tangible objects, and the design of the model is incomplete because it is not based on the functional and nonfunctional requirements	2	As a result of the core knowledge, the SSM has a great conceptual understanding of AnyProject. Therefore, the SSM can control, execute and manage a project more effectively	24
Growth	25	The TM does not show the outlook and growth of the project over time. The TM cannot grow because it can only be applied to one scenario. Therefore, the high maintenance costs and the single-use patterns stunt the growth of the TM	2	SSM is designed to deal with dynamic parts because the BOs can seamlessly be implemented without disrupting the entire structure of the pattern language. Therefore, there are lower maintenance costs, which enable the system to grow more effectively	25
Total	100	–	42	–	97

8.1 Measurability Study

The total number of methods in any system can be calculated using the formula (Table 3):

Table 3. Summary of comparison of traditional and stable model

Feature	TM	SSM
Number of tangible classes	11–12	7
Number of inheritances	1	0
Number of attributes per class	1–7	3–6
Number of operations per class	0–6	1–5
Number of applications	1	Unlimited

T = C * M; where,
T = total number of operations
C = total number of classes
M = number of methods per class

Traditional Project Model 1
C = 11
M = 4
T = 11 * 4 = 44
Traditional Project Model 2
C = 12
M = 3
T = 12 * 3 = 36
Stable Project Model
C = 7
M = 3
T = 7 * 2 = 14

The Stable Project Design Pattern is more applicable compared to the TM because it is less complex, as evidenced by the fewer operations in the Stable Project Design Pattern. The stable model based on three level architecture achieves a more detailed understanding of the problem requirements. Moreover, it achieves this deep representation with far less complexity than TM. The Traditional Project Model includes tangible classes which make it vulnerable in the event of any change.

8.2 Qualitative Measurements

8.2.1 Growth

The following analysis will use growth as the criteria to determine the better model between TM and SSM. Since the SSM is completely applicable, it enables greater growth because it can adapt to situations, new technologies, and innovation much easier than a TM. Additionally, there will be greater growth because of reduced maintenance costs because the SSM core knowledge does not change. On the other hand, TM will not nurture growth because it is only useful for a single scenario. Just like a single-use plastic water bottle, a TM is built to serve its purpose only once. Therefore, the long-term costs of using the TM are higher than the Stable Project Design Pattern. It follows that the Stable Project Design Pattern has a higher ceiling and more potential than the TM.

9 Discussions and Analysis

9.1 Abstraction

The abstraction process emphasizes pattern reusability and usability. Reusability refers to utilizing the same model in various scenarios, reuse in different contexts, thus it proves to be cost-effective. As the qualitative and quantitative measurements clearly show, SSM is much more effective at creating a stable and reusable design pattern language for AnyProject. The reusability factor and the count of reused classes speak to how the SSM abstracts the patterns more efficiently than the TM. This study appeals to what we know about classes in the TM since all the classes are defined for a scenario. Hence, for each different scenario, TM must be developed anew. However, with software stability model, we have the freedom to plug in any number of IOs to the Bos to make the model fit into different scenarios.

Additionally, the nonfunctional requirements abstracted within each class in the SSM creates a functioning pattern language. The comparative study demonstrates the superiority of the SSM over the TM, as evidenced by the nonfunctional requirements, which serve as the evaluation and measurement metrics. The effective abstraction of the nonfunctional requirements and pattern language shows how the SSM is more stable and reusable.

9.2 Applicability

TM does not have a wide range of applicability. Since a TM is constructed in the business as usual fashion, it is created for a specific scenario, only. There is little to no consideration for wide applicability beyond the imminent scenario. In contrast, the SSM is constructed with applicability in mind. The process of constructing an SSM is rigorous in its attempt to encompass all scenarios of AnyProject. The core knowledge of the AnyProject SSM is extended to include the specific IOs of each scenario, as shown in the applicability. This core knowledge is extensible to any scenario of AnyProject. Applicability exists when you have less complexity in classes such that you can represent the entire system without having a large number of operations that are susceptible to error TM 1 and 2 have 33 and 36 classes, respectively, whereas SSM only has 14 methods. This analysis shows that the Stable Project Design Pattern is more applicable than the TM as it can be utilized in many more scenarios as a result of its applicability. This applicability leads to immense prosperity for the AnyProject implementation.

9.3 Impacts

Traditional Model has zero impact in terms of usability and costs. Since its applicability is very limited, and the idea of doing a project is not defined, to obtain more and exact information, considerable rework is required, which involves money as well as cost. On the other hand, the stability model proves to be a clear winner in terms of usability and costs. Its impact is much higher because costs are significantly reduced due to the design patterns involved and it supports reusability too. Thus, using the software stability model as a design model results in a higher success rate, which is targeted towards achieving goals of completing a project.

10 Conclusion

In conclusion, we could say that software stability model helps obtain the enduring concept dynamically, i.e., it has wide applicability for different scenario where it is applied. Thus, the enduring concepts help build a solid foundation for solutions and implement a better way of software stability model. Hence, the stability model is preferred as a design model for any application.

Also, we recognize that contemporary approaches to project execution tend to shun the creation of analytical artifacts and abundant documentation. This is referred to as to this as NBUA (No Big Upfront Anything) [10]. With the implicit reusability of AnyProject Stability Model, the approach will reduce the creation of redundant project artifacts while employing the requisite analysis to complete the project successfully.

References

1. "Project," Oxford English Dictionary, Oxford, United Kingdom, Oxford University Press, Oxford (2012)
2. Fayad, M.E., Altman, A.: An introduction to software stability. Commun. ACM **44**(9), 95–98 (2001)
3. Fayad, M.E.: Accomplishing software stability. Commun. ACM **45**(1), 95–98 (2002)
4. Fayad, M.E.: How to deal with software stability. Commun. ACM **45**(4), 109–112 (2002)
5. Fayad, M.E., Sanchez, H., Hegde, S., Basia, A., Vakil, A.: Software Patterns, Knowledge Maps, and Domain Analysis. Auerbach Publications, Boca Raton, 422 p., December 2015
6. Compare: An individual or collaborative enterprise that is carefully planned to achieve a particular aim. Oxford English Dictionary, Oxford, United Kingdom, Oxford University Press, Oxford (2012)
7. What is a project? definition and meaning, 19 April 2016. BusinessDictionary.com
8. Boeing B-29 Superfortress, 28 March 2019. https://en.wikipedia.org/wiki/Boeing_B-29_Superfortress. Accessed 08 April 2019
9. Kelly, C.C.: The Manhattan Project: The Birth of the Atomic Bomb in the Words of Its Creators, Eyewitnesses, and Historians. Black Dog & Leventhal Publishers, Incorporated - History - 495 p. (2009)
10. Big Design Up Front, 28 March 2019. https://en.wikipedia.org/wiki/Big_Design_Up_Front. Accessed 08 April 2019