

RISK ANALYSIS IN SOFTWARE DEVELOPMENT BASED ON ARTIFICIAL INTELLIGENCE (A.I.): MODERN APPROACH

Prof. Dr. Bhoite Sudhakar D
(SIBER, India)

Prof. Mr. Budake M B
(K.H. College Gargoti, India)

Abstract:

Software development process is a very complex process. A more complex software applications have been built, time is becoming a critical factor to release applications that must be fully tested and comply with Business Requirements. In this paper presents the use of artificial intelligence in software development will definitely improve the automatic design of database, (Entity Relationship Diagram (ERD), UML diagrams, automated coding, automated testing software, and overall software development. Use of artificial intelligence in software development describes how to develop, design, coding, testing and maintain the software. This techniques improve the quality of the software project and over all the process of software development. It is important for companies in deciding how to apply AI in their software applications and to create strategies for its use.

Keywords: —Software Engineering, SDLC, Artificial Intelligence, Software Code Generation, Agile methodology, Machine Learning.

1. INTRODUCTION

Risk management should begin at the earliest stages of program planning and continue throughout the total life-cycle of the program. Additionally, risk management is most effective if it is supported with automated tool that ensures integration with the program's systems engineering and program management processes.

A] An entity relationship data model is a high level conceptual model that describes information as entities, attributes relationships and constraints. Entity relationship diagrams to design the database of the software. It involves a sequence of tasks including extracting the requirements, identifying the entities, their attributes, the relationship between the entities, constraints and finally drawing the diagram. As such entity relationship diagram design has become a tedious task for novice designer. The planning, requirement and design phase occur high risk in software development

B] It is not easy to design software architecture reflecting the intention of developers and implement the result of design modelling as a program while preserving the architectural correctness, because there is a gap between design and implementation.

C] The communication between developers and testers is not sufficiently effective through written communication and it has to be augmented by informal communication. One important observation is that the communication changes depending on the type of the tasks (new or old) and also on the experience of the testers and how much the developer knows about the experience of the tester. Regarding the distribution or collocation between testers

And developers, it is reasonable to conclude that distribution does affect the work of testing by creating more asynchronous communication and consequently less informal communication between developers and testers. The trade-off between "good documentation" vs. "asking questions" strategies is still an open question and needs to be better investigated to understand the real effects of it.

D] Nearly 70% of the time in software development is for maintenance and this involves huge costs for the developers. New software these days are very complex and the size of software has increased considerably, thereby making the software increasingly difficult to maintain. Software maintainability is directly related with the economic performance and success of a product or an organization. Using maintainability, we can predict what changes or failures that may occur in software after it has been deployed. Maintainability of the software is therefore a quality attribute for software that helps to determine the Performance of the software. Developers are assisted to determine probable amount of change that might occur in software modules during the maintenance phase. Owing to the correctness of the maintainability predictions, the design of software can be improved to determine changes that need to be made for development of software modules in the future.

It is clear that each software project is unique and needs adaptation and customization of selected automated tools to its practical implementation. This set of criteria should be defined for each specific IT project and consider the goals of implementation, objectives of risk management process, the size of IT team and their needs, the possibility of integration with required standards and methodologies.

A] Agile methodology:

Most of software development companies use agile methodology for develop products and projects. Agile methodology gives the responsivity to flexibility and speed of develop product or project. Almost every product and project which is being managed today

using agile methodology. Agile software development is use a user story tool to capture a description of software feature from end user perspective. A user story helps to create a simplified description of a requirement and can fit into agile frameworks like Scrum and Kanban.

User Story:

There are three (3) components of writing good story. These are as follows-

1) WHO, WHAT & WHY: This provides context of user story

Template:

As A—{WHO}

I Want—{WHAT}

So That—{WHY}

Example:

As a—Student

I Want—To take admission WHEN admission is open

So That—I can take admission

2) ACCEPTANCE CRITERIA:

Template:

GIVEN—{a context} →Preconditions

WHEN—{an event} →what happen

THEN—{an outcome} →describe outcomes

In order for a user story to be termed as deliver the acceptance criteria need to be satisfy. It is helpful for development team and testing team to work on their Unit test and QA test cases. It is also called behavior driven development approach .It is basically kind of document acceptance of criteria in the form of scenarios. So which gives clear-cut understanding what is require to both development and testing team

Example:

GIVEN—I am register user AND Login to software

WHEN—I select course AND select subjects for particular course

THEN—I can see the admission is confirm or not

3) SUPPORTING MODELS:

SCREEN MOCK UPS, PROCESS MODEL MAPS, DATA MODEL etc. which can help to support the user story to better understanding and more context for development team and testing team.

Agile doesn't handle database or system design. It tells you how to manage your project. With that in mind, you start by identifying all the features of the system and put them in the product backlog. Then you, along with the product owner, assign priorities to the features in the backlog. After you've done that you start taking features from the backlog. So somewhere in the backlog there should be a user story like above example. If story is rather large (epic) level. Then that has to be broken out into smaller stories.

Using user story system analyst predicate following table:

1) WHO, WHAT & WHY:

Entity	Attributes	Datatype
Student	S_Id	int
	S_name	varchar
	Email,Etc.	varchar

2) ACCEPTANCE CRITERIA:

In acceptance criteria system analysis broken out into small stories.

Entity	Attributes	Datatype
Course	C_Id	int
	C_name	varchar

Entity	Attributes	Datatype
Subject	Sub_Id	int
	Sub_name	varchar
	C_Id	int

Registration table will generated as follows:

Entity	Attributes	Datatype
Tbl_Reg	R_Id	int
	S_Id	varchar
	C_Id	Int

Problems in agile methodology:

Collecting Data and Requirements

1. This is one of the stages of the software development lifecycle in which human intervention is still a must.
2. Although agile development significantly accelerated the traditional software development lifecycle, all components, including features, functionalities, and integrations have to be manually managed and updated. Needless to say, this usually leads to numerous bugs and inconsistencies due to the complexity of the system.
3. Agile is based on the premise that teams won't always know their end result will look like (or the next few delivery cycles ahead for that matter) it also becomes difficult to accurately predict the cost, time, or resources needed at the beginning of a project.

Replace agile to AI:

1. With the help of natural language processing (NLP) and machine learning, the task of developers will be to collect and prepare domain-specific data and feed it to machine-learning algorithms.
2. This new approach will not only automate and facilitate the process but also result in models that are constantly trained and improved. Such a machine-learning model is capable of identifying software features that are important to users.
3. With machine-learning models, everything will be automated which means that human error will be eliminated.
4. It gives fragmented output.

B] Why Python in AI?

The most of the famous AI frameworks have primary support for python. It is also Multilanguage supported. Python comes with huge amount of inbuilt libraries. Many of the libraries are for Artificial Intelligence and Machine Learning. Some of the libraries are Tensorflow(Which is high level neural network library),Scikit-learn (For data mining, data analysis and machine learning etc).This list keeps going and never ends. The learner of AI and ML don't want to spend time on debugging the code for syntax errors, they want to spend more time on their algorithms and heuristics related to AI and ML.

We try to automating Software Development using AI from above example.

Python supports a number of powerful libraries you can use for analyzing, visualizing, and mining data. Small data analysis projects will familiarize you with a few popular Python libraries, such as Matplotlib, NumPy, and Pandas, for instance.

Create automatic database:

Simple example:

```
from sqlalchemy import create_engine
import pymysql
import openpyxl as op
import pandas as pd
c=op.load_workbook('C:\\Users\\Windows\\b.xlsx')
print(c.sheetnames)
s=c['Sheet1']
Tbl_Name=s.cell(row=2,column=1).value
print(Tbl_Name)
```

Output:

```
['Sheet1']
```

```
Student
```

```
cols=[1,2,3]
```

```

rows=[2]
df=pd.read_excel('b.xlsx',sheet_name="Sheet1",usecols=cols,header=None)
print(df)
print('\n')
Tbl=({ df.iloc[i][0]: df.iloc[i][1:].to_dict() for i in df.T})
print(Tbl)

```

Output:

```
{'Attributes': {1: 'Datatype'}, 'S_Id': {1: 'int'}, 'S_name': {1: 'varchar'}, 'Email': {1: 'varchar'}}
```

We can easily converted it into dictionary in python

```

Tbl_Name = {"S_Id": "Int", "S_Name": "Varchar(50)"};
tableName = " Tbl_Name "
dataFrame = pd.DataFrame(data= Tbl_Name)
sqlEngine = create_engine('mysql+pymysql://root:@127.0.0.1/test',
pool_recycle=3600)
dbConnection = sqlEngine.connect()
try:
    frame = dataFrame.to_sql(tableName, dbConnection, if_exists='fail');
except ValueError as vx:
    print(vx)
except Exception as ex:
    print(ex)
else:
    print("Table %s created successfully."%tableName);
finally:
    dbConnection.close()

```

Table will created automatically likewise all table will create using story writing.

Create automated Entity Relationship Diagram (ERD):**Simple Example:**

```
Import ConfigParser
```

```
From sqlalchemy import Metadata
```

```
From sqlalchemy_Schemadisplay import create_schema_graph
```

```
Config=ConfigParser.RawConfigParser()
```

```
Config.read() #Open config file
```

```
“””Read the flag ‘sql_connection’ and make connection to database or u can directly specify connection like above example”””
```

```
# Then generate graph of connected database
```

```
Graph=create_schema_graph(metadata=MeataData(connection),
```

```
    Show_datatype=False,
```

```
    Show_index=False,
```

```
    Concentrated=False)
```

```
#Genrate png image
```

```
Graph.write_png(‘dbStudent.png’)
```

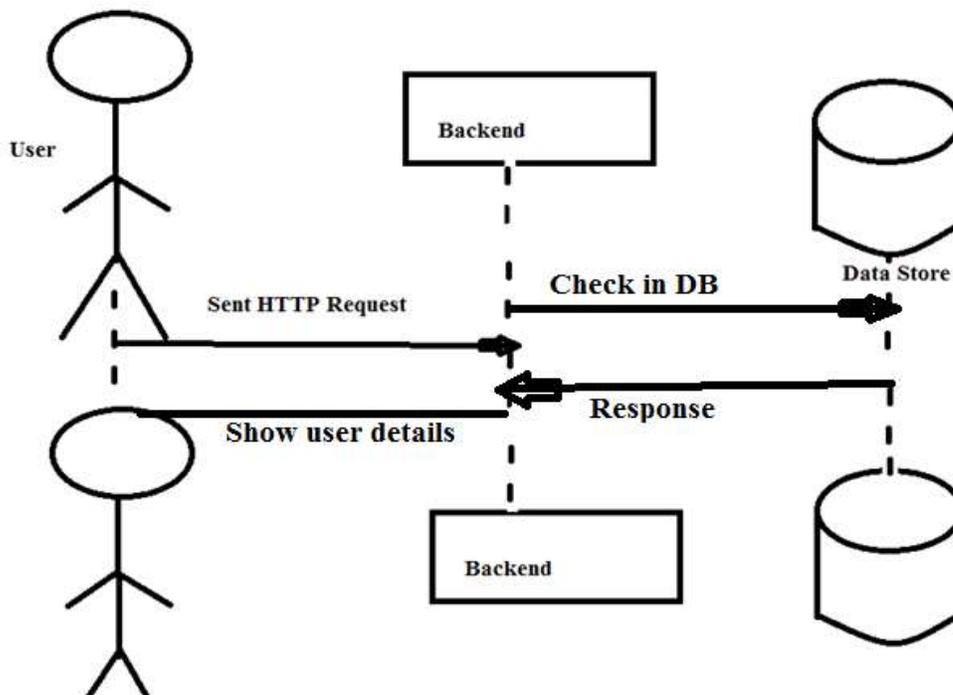
Output:

The ERD will show

C] Generate UML diagram:

Unified Modelling Language (UML) is a general purpose modelling language and a standard way to visualize a system. It helps software engineering, business professionals, system architects to do design and analysis. To generate the UML diagram we will construct a class diagram and then for this class diagram we will generate the code. PlantUML library in python helps us to draw diagram and file extension is wsd.

The fact that event driven systems can be modeled and implemented using Unified Modelling Language (UML) state chart diagrams has led to the development of code generation tools. These are tremendously helpful in making software system designs and can even generate skeletal source code from these designs. The implementation of such automatic code generation from state diagrams is not fully supported by the existing programming languages. The major down side is that there is no one-to-one correspondence between the elements in the state chart diagram and the programming constructs. The existing programming elements cannot effectively implement two main components of the state diagram namely, state hierarchy and concurrency.(Ref-_)



Code:

```
startuml
```

```
actor user
```

```
participant backend
```

```
database dataStore
```

```
user -> backend :Sent HTTP Request
```

backend -> datastore:Check in DB

datastore ->backend:Response

backend ->User:Show user details

D] Machine Learning:

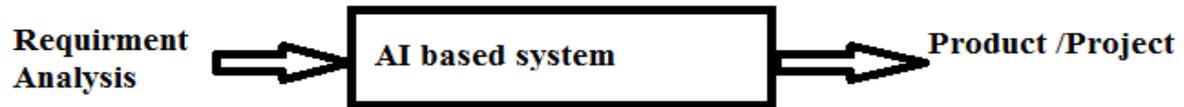
Machine Learning (ML) permeates into all related business processes to bring them to a qualitatively new level. To keep abreast with the times, any software development services provider seeks for the ways to accelerate software development lifecycle (SDLC) through using AI-driven approach. An ML-based development process presupposes a different role for a developer. It means an engineering team, unlike today, will be to a greater extent engaged in controlling and data-analyzing activities than in programming. There is a great hype around AI replacing software developers. In fact, AI will eliminate simple but strenuous programming, which will allow prioritizing more complex project tasks.

I. Automated Code Generation by ML:

In the Machine Learning (ML) proves the visual interface can be developed by applying ML algorithms for generating code from a graphical user interface screenshot. Automated code generation based on some given modules is not a new concept, and it can be used for process automation quite successfully.

II Testing:

Software testing, whatever the type, can benefit from the ML-based approach. Trained on some historical data, i.e., customer requirements, previous test cases, the algorithm can predict the nature and the source of bugs and defects throughout the SDLC. In the hands of skilled software development specialists, “smart” bug detection tools can be effectively applied for code refactoring and optimization.

Theme of paper:**Conclusion:**

4. To keep abreast with the times, any software development services provider seeks for the ways to accelerate software development lifecycle (SDLC) through using AI-driven approach.
5. Artificial Intelligence (A.I.) gives input such as Identity patterns across project, Discover dependencies, Suggest reuse opportunities, More precise requirements to SDLC phases and gives desire output with less defects, optimise performance, automatic deployment pipeline, less error and faster.
6. Artificial Intelligence (A.I.) provides automated tool with very limited planning is required to get started with the project. Software Development Company's assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
7. AI-based solutions can significantly improve the software testing process, making it faster and more efficient. Setting-up an AI system might have some initial investment, but on the long run it will surely help enterprises save effort and cost.
8. AI based system provides automating the code generation and detects the bugs from the code. Take an instance of a project where your idea in your natural language and your system can understand it by converting it into executable code.
9. AI offers a solution to estimate software that involves analyzing historical data from earlier projects of the enterprise to find correlations and statistics. It employs predictive analytics as well as business rules to offer accurate estimates of cost, time and effort.
10. AI based systems with the help of automated tool or automated programming tool we can eliminate risk assessment phase saving our time in software development.

ACKNOWLEDGMENT:

Thanks to all my friends who are working in various software development companies for helping me to write this paper.

REFERENCES:

11. Senthil Jayavel, S Arumugam, Brijendra Singh, Prashant Pandey, Akash Giri, Akshay Sharma: Use Of Artificial Intelligence In Automation Of Sequential Steps Of Software Development / Production:” Journal of Theoretical and Applied Information Technology”
12. Vipani Kumari, Sandeep Kulkarni: Use of Artificial Intelligence in Software Development Life Cycle Requirements and its Model, “International Research Journal of Engineering and Technology (IRJET)”.
13. Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, Thomas Zimmermann: Software Engineering for Machine Learning:A Case Study, “2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSESEIP)”.
14. Jerry Gao, Chuanqi Tao, Dou Jie, Shenqiang Lu: What is AI Software Testing? and Why:” 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)”.
15. Divanshi Priyadarshni Wangoo: Artificial Intelligence Techniques in Software Engineering for Automated Software Reuse and Design:” 2018 4th International Conference on Computing Communication and Automation (ICCCA)”.
16. Robert Feldt, Francisco G. de Oliveira Neto, and Richard Torkar: Ways of Applying Artificial Intelligence in Software Engineering:” RAISE’18, May 2018, Gothenburg, Sweden”.
17. Automatic Code Generation From UML StateChart Diagrams:-Sunitha E. V. and Philip Samuel:”IEE Access”
18. Extracting Entity Relationship Diagram (ERD) from English Sentences :-Amani Abdel-Salam Al-Btoush.
19. Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies: Gaurav Kumar and Pradeep Kumar Bhatia:” 2014 Fourth International Conference on Advanced Computing & Communication Technologies”

20. 6. Advantages of Agile Methodologies for Software and Product Development in a Capstone Design Project - by Diane Rover , Ryan Scheel , Julie Wegter and Cameron Whipple
21. Major Issues in Software Engineering Project Management:Richard h. Thayer, member, ieee, arthur B. Pyster, member, ieee, and Roger c. Wood, member, ieee
22. Risk management applied to software development projects in incubated technology-based companies: literature review, classification, and analysis:- Sandra Miranda Neves, Carlos Eduardo Sanches da Silva.