SAN JOSÉ STATE
UNIVERSITY

AI Testing – A Tutorial

Presented by: Jerry Gao, Professor, and Director

San Jose State University – Excellence Research Center on Smart Technology, Computing, and Complex Systems

Date: 6/4/2018

Test Automation Market

The report "Automation Testing Market by Technology (IoT, AI, and Big Data), Testing Type (Functional, Performance, Compatibility, and Security), Service (Advisory & Consulting, Managed, and Implementation), Endpoint Interface, and Region - Global Forecast to 2023"

https://www.marketsandmarkets.com/Market-Reports/automation-testing-market-113583451.html

The automation testing market size is expected to grow from USD 8.52 Billion in 2018 to USD 19.27 Billion by 2023, at a Compound Annual Growth Rate (CAGR) of 17.7% during the forecast period (2018–2023).

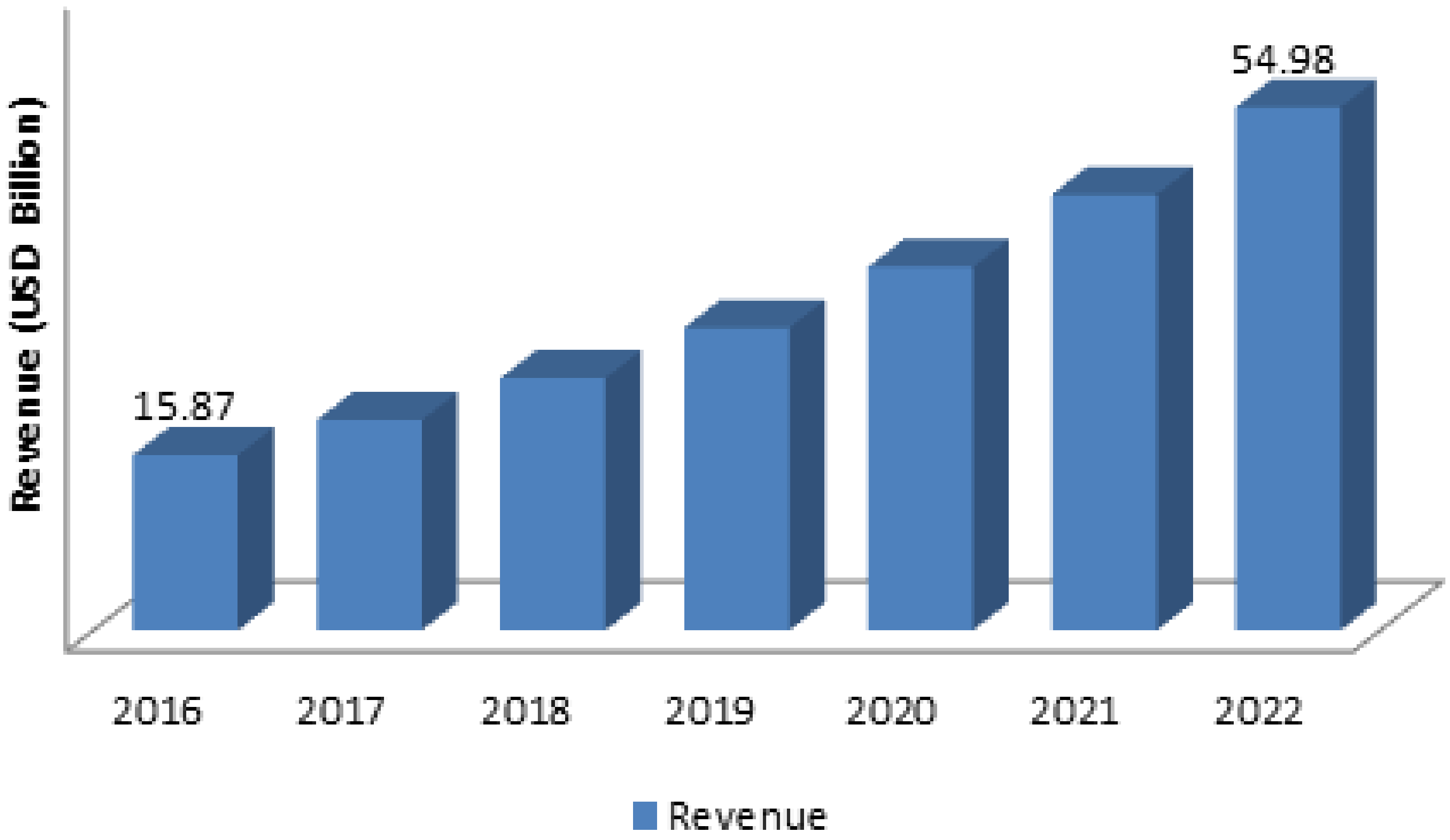
Global Test Automation Market Set For Rapid Growth, To Reach Around USD 54.98 Billion by 2022

https://www.zionmarketresearch.com/news/test-automation-market

As per a report by Transparency Market Research, the global test automation market will likely expand at a robust CAGR of 15.4% from 2017 to 2025 to become worth US$ 109.69 bn by 2025 from US$ 30.45 bn in 2016.

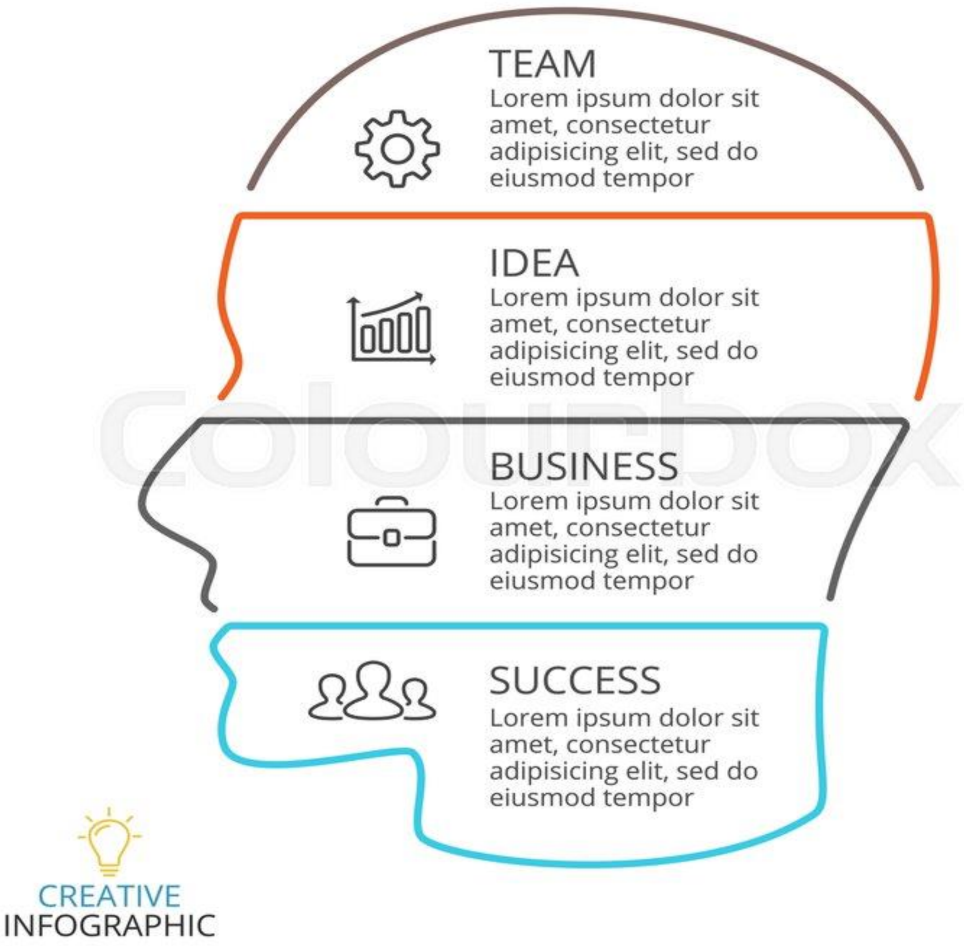
https://www.transparencymarketresearch.com/test-automation-market.html

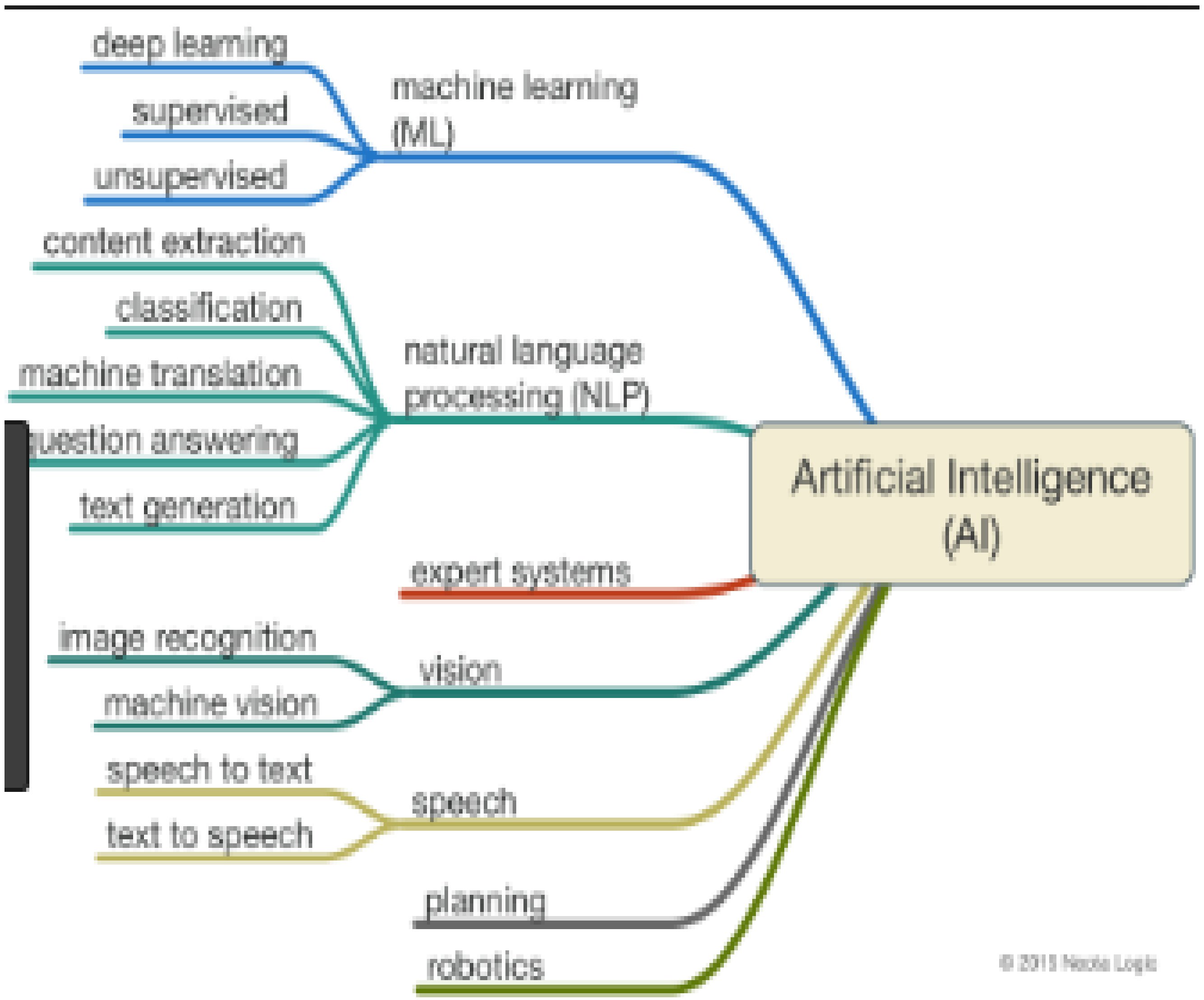
Global Test automation Market Revenue, 2015 - 2021 (USD Billion)



Source: Zion Market Research 2017

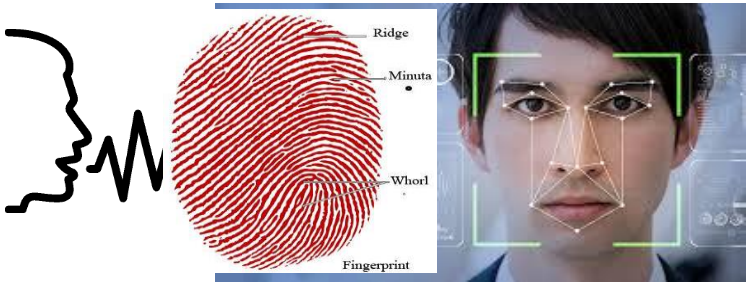
AI Classification


TEAM
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
IDEA
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
BUSINESS
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
SUCCESS
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
CREATIVE
INFOGRAPHIC

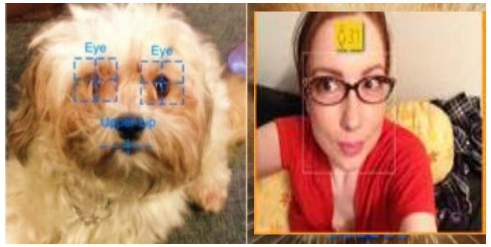


AI Software Classification

Expert System



Identification & recognition system



animal profiling & classification

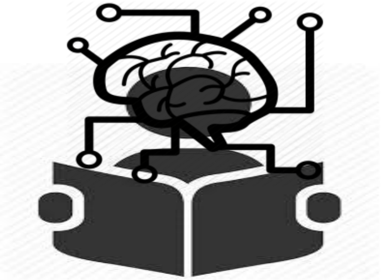
human profiling & classification

Recommendation System

Unmanned vehicle system

Translation system

weather
air/water/soil quality
disaster
transportation
……………………..

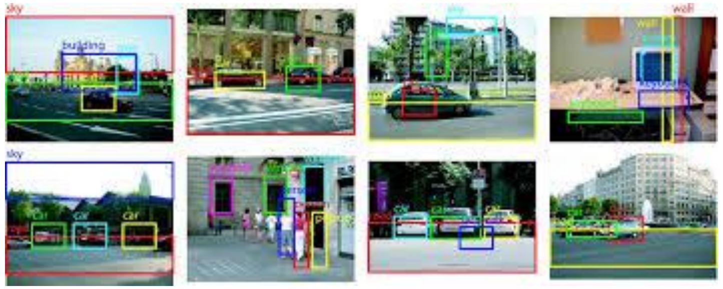


Learning-Based System

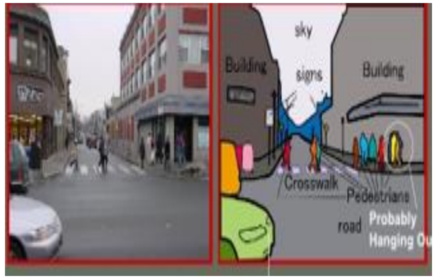
Behavior detection & classification

Q&A system

Business Intelligent system



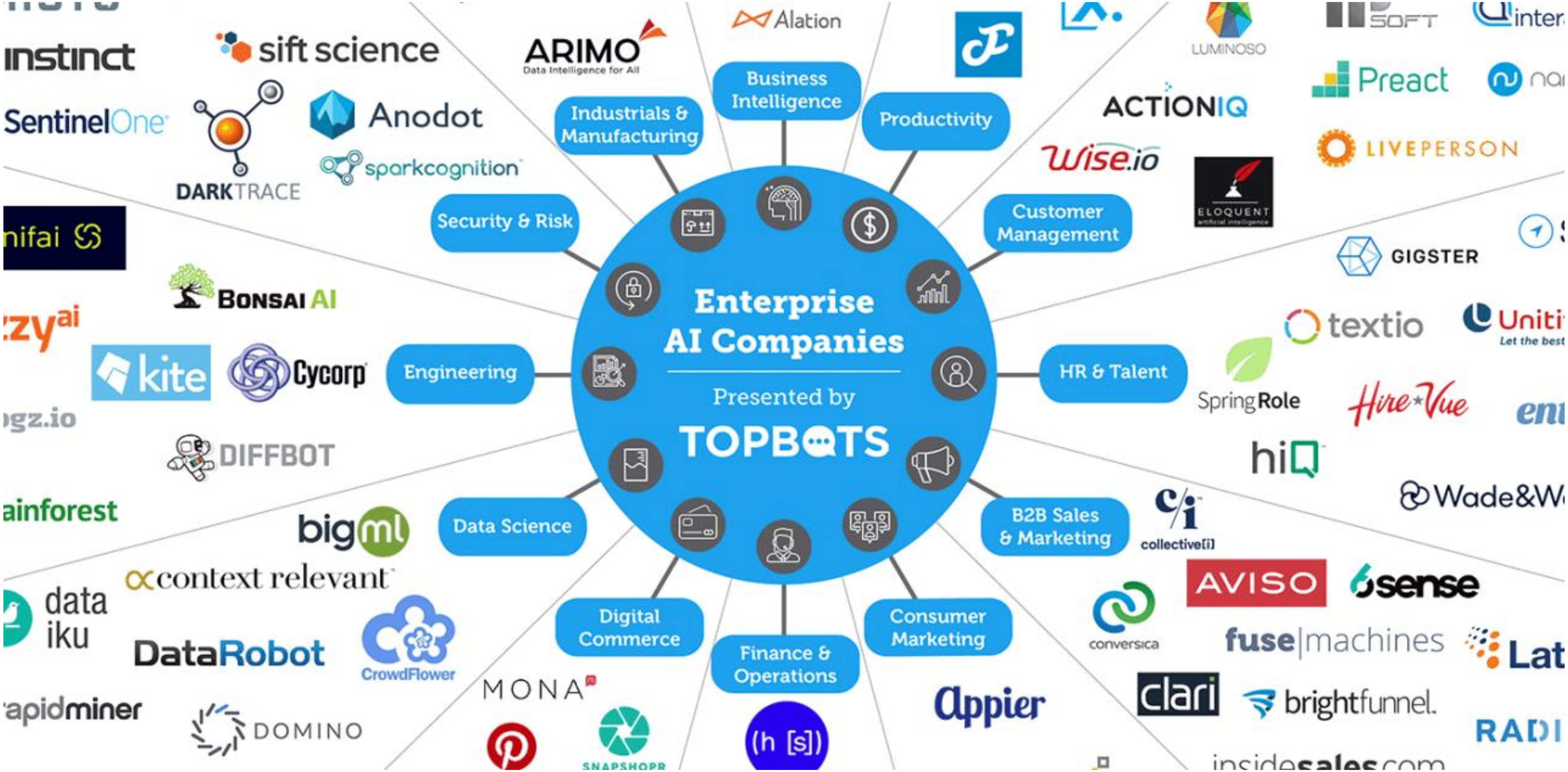
object detection & classification



context identification & classification

Command control & action

AI Enterprise Companies



Enterprise AI Companies

Presented by TOPBOTS

Business Intelligence

Productivity

Customer Management

HR & Talent

B2B Sales & Marketing

Consumer Marketing

Finance & Operations

Digital Commerce

Data Science

Engineering

Security & Risk

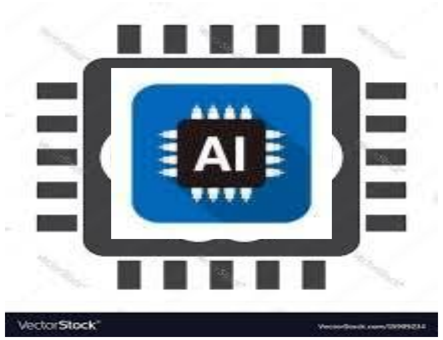
Industrials & Manufacturing

Current AI System Testing Special Features

Test cases based on
Input Data from Big Data with 6V

Example-Based
Learning System

Real-Time Based Application



Change Based-On Learning

Learning-Based System Evolution

Limited Knowledge Scope

Limited Training/Test Dataset

Uncertainty

Incorrect Decision

Incorrect Behavior & Actions

Common AI Testing Questions

- **What is AI software testing?**
- **How to test AI functions in a mobile app?**
- **How to identify and establish well-defined test requirements for AI functions in a mobile app?**
- **What and where are the cost-effective testing models and methods for testing AI functions?**
- **What and where are the adequate quality assessment criteria for AI functions?**
- **How to evaluate the training and test data sets?**
- **Where are the automatic tools supporting AI software testing?**

In addition, we also found following common facts and features of current AI mobile software apps after validating numerous ones.

1. **Limited data training and validation** – Most of our tested mobile apps with AI features are built based on machine learning models and techniques, and trained and validated with limited input data sets under ad-hoc contexts.
2. **Data-driven learning features –** Many mobile apps with learning features provide static and/or dynamic learning capabilities that affect the under-test software outcomes, results and actions.
3. **Uncertainty in system outputs, responses, or actions** – Since existing AI-based models are dependent on statistics algorithms, this brings the outcome uncertainty of AI software. We experienced many mobile apps with AI functions generated inconsistent outcomes for the same input test data when context conditions are changed.

What is AI Software Testing?

What Is AI Software Testing?

Intuitively, AI software testing refers to diverse quality testing activities for AI-based software systems using well-defined quality validation models, methods, and tools.

Its major objective is to validate system functions and features developed based machine learning models, techniques and technologies.

AI software testing includes the following primary goals:

- Establish AI function quality testing requirements and assessment criteria
- Detect AI function issues, limitations, and quantitative and quality problems
- Gain the quality confidence of AI functional features developed based on AI techniques and machine learning models.
- Evaluate AI system quality against well-established quality requirements and standards.

Major Testing Focuses and Scope

In the past two years, we have validated different types of AI mobile apps with diverse AI capabilities and features. Figure 1 shows the scope of AI software testing, which covers different types of artificial intelligent features and capabilities. Our students have tested numerous mobile apps powered with diverse machine learning models and AI algorithms. Here are some typical examples.

Figure 1. AI Software Testing Scope

Apple Siri - It is a built-in, voice-controlled personal assistant for Apple users. The idea isthat you talk to her as you would a friend and she aims to help you get things done, whether that be making a dinner reservation or sending a message.

- Calorie MAMA – It is a smart camera app that uses deep learning to track nutrition from food images.
-
- Seeing AI – It is a free app that narrates the world around you. Designed for the blind and low vision community, this ongoing research project with powered AI techniques. Its goal is to open up the visual world for users by describing nearby people, text and objects. - Check

- My Age - It is a biometric face detection and age estimation application. It uses world best Neurotechnology face recognition algorithms to find the age from the look of the face

Data quality validation:

Issue #1 - Domain-specific training data quality checking could be very costly and time consuming. For example, training data for medical machine learning projects require the quality validation and confirmation from medical doctors. This is not only costly but very time consuming.

Issue #2 - There is a lack of automatic unstructured data quality validation tools. Although some existing tools (such as AAAA, BBB) are available for raw data quality checking, we could not find automatic tools for validating annotated rich media training data (i.e. video, audio and images). This becomes a serious issue in training data quality validation.

Issue #3 – There is a lack of well-defined data quality evaluation models and assessment metrics for unstructured training data, including images, videos, and audios

Why Do We Need AI Software Testing?

Step #1: AI function test planning

In this step, testers need to perform several tasks:

- Task partitions and scheduling – In our project experience, students found that this is not easy task due to the complexity of AI function testing as well as the lack of AI knowledge and understanding.

- AI function identification, understanding and partitioning
  For many selected AI mobile APPs, students have difficulties in identifying and partitioning AI functions due to the lack of well-defined AI function requirements given by vendors. In many cases, several AI functions are aggregated together to generate system outcomes. ‘

- AI function test requirement analysis – Many CMPE 287 class students have encountered the difficulty in generate AI function test requirements in a system approach although most of them have used the scenario analysis approach. One of the major reasons is the lack of effective test requirement analysis approaches to assist testers to perform AI function test requirement analysis from data and context perspectives.

- Tool selection – Although many existing software testing tools are available for white-box testing, and system GUI and performance testing, students have problems to find any useful and practical tools for AI function validation

Step #2: AI function test modeling

– In this step, testers need to perform test modeling based on the established test requirements. Although many of student groups have selected existing test models, such as decision tables, state diagram or event-based GUI models, for their AI software testing, they have found that they are not suitable or adequate for them to come out sound and effective test models for their selected AI functions.

Step #3: AI function test design

- This step focuses on AI function test design and test case generation in terms of system input and expected system outcomes. In conventional function testing, test design is performed by applying existing testing methods, test cases are generated by identifying input test data and expected output data.

Step #4: AI function test execution

– In this step, testers need to perform and execute their test cases with input data to detect AI function quality issues by checking the actual outcomes and results. In addition, they need to report the detected problems (bugs) during their testing. One major difficulty in their bug reporting and analysis is to identify and locate the causes of AI function quality issues, such as correctness, consistency, and accuracy. Data scientist and AI software engineers may have encountered the same problem in identifying the isolating the cause of AI function quality issues.

Step #5: AI function test quality evaluation

– In this step, testers need to evaluate their testing quality and decide if they have done enough in function validation. Since AI software has special features such as non-oracles, timeliness and learning capability, here function test quality evaluation is added particularly as the final step of AI software testing process. In this step, different quality parameters are measured using the pre-defined quality metrics based on testing result analysis. If the evaluation results.

AI-Based Software Testing

Test selection and reduction using AI techniques

Now, let’s summarize the major causes to conduct AI testing:

- Cause #1 - Current existing software testing models and methods have limits to address AI software testing needs in supporting multi-models with unstructured input data, addressing large-scale classified inputs, and considering oracle problems, and quality accuracy, consistency, and correctness as well as relevance.

- Cause #2 – Most current AI software are built-in with machine learning models developed by data scientists through large-scale data training using scientific algorithmically approaches instead of engineering approaches. Hence, there is a big gap in considering quality validation and quality assurance from engineering perspectives. Hence, AI testing research is needed to study and develop new and effective quality standards and evaluation methods.

- Cause #3 – Building powerful AI software needs to use largescale training and test data sets. The current train methods and data generation lack of quality consideration, quality assessment, and certification. Hence, how to come out quality training data models, develop large-scale quality test data generation methods will be needed.

Therefore, AI function testing targets at built-in AI features in AI software applications. It refers to different testing activities to find AI software errors, verify evaluate quality parameters with well-defined testing models and quality assessment methods. The testing goal is to validate well-defined test requirements, meet pre-defined testing criteria, and standards of quality assurance of the under-test AI software.

TEST MODELING FOR AI FUNCTIONS

The basic test modeling procedure for each selected AI function consists of the following steps:

- Step #1: AI function context classification modeling

  In this step, a tester needs to identify and classified diverse context conditions and parameters, and present the classification results using a context classification model, known as a context classification tree.

  A context classification tree is a 3-tuples, denoted as GCT= (NCT, ECT, RCT), where NCT is a finite nonempty set of nodes with a node label. There are three types of nodes in NCT, including a root node (NR), intermediate nodes (NI) and leaf nodes (NL). Figure 5 shows a simple example. ECT consists of a set of edges, and each connects two nodes in the tree (GCT), and represents of different category semantic relations between them. These semantic relations are included in RCT as its elements. There are four types semantic relations: AND, XOR, SELET-1 and SELECTM. The table below shows the detailed descriptions.

Semantic Relations Descriptions

AND (NP,) NP has an AND relation with its n child nodes when all of its child nodes must be included. XOR (NP,) NP holds an XOR relation with its two child nodes if only one of its two child nodes could be selected.

SELECT-1 (NP) NP has a SELECT-1 relation with its child nodes when only of its child nodes could be selected.

SELECT-M (NP,) NP has a SELECT-M relation with its n child nodes if and only if m of n child nodes could be selected.

Step #2: AI function input classification modeling

In this step, a tester needs to focus on input classification to identify and classified diverse input data in terms of its category classes and their sub-classes. When an AI-based function accepts multiple input media formats (such as video, audio, and image, and text), each of them should be examined and classified. To effectively support a tester to conduct input classification, we used an input classification model (known as input classification tree) as our analysis and test model to facilitate and represent diverse input data classes and their sub-class using a category approach.

- Step #3: AI function outcome classification modeling

  In this step, a tester focuses on the classification of diverse AI function outputs, including texts, audio, video, and images, or events (or actions). Similar to input classification, an output classification tree model is generated as the outcome of this step.

- Step #4: Generate a 3D classification decision table

  In this step, a tester generates a new decision table for each under-test AI function feature, known as 3D classification decision table, to identify three dimensional mappings among disjoint classified context conditions, disjoint classified inputs, and disjoint classified outputs

AI Testing Approaches and Services

AI software testing could be carried out using the following approaches, shown in Figure 3.

- **Rule-based AI software testing**, in which pre-defined expert-based rules are established and used in AI test generation and validation. This approach has been reported in long time ago.
- **Classification-based AI software testing**, in which classification models for inputs, contexts, and outputs and events are setup for AI software testing to assure the adequate testing coverage of diverse input data classes, classified contexts and conditions, and corresponding outputs and classes.
- **Model-based AI software testing**, in which selected machine learning models are extended to be traceable and testable AI test models to facilitate AI software testing and operations in quality evaluation of training data and test data.
- **AI-based testing for AI software**, in which AI models and data-driven techniques are used to facilitate and optimize AI software testing in different perspective.
- **Metamorphic (Non-Oracle) testing**, in which a property-based software testing technique is used as an effective approach for addressing the test oracle problem and test case generation problem.
- **Testing robots for AI software,** where automatic software test robots are built and used to learn and follow experienced testers to perform user-oriented testing operation using collected user testing scenarios and test data.
- **Learning-based AI software testing** using the crowd-sourced approach, in which selected machine learning models and approaches are used to learn from crowd-sources testers in a service platform

CHALLENGES, ISSUES, AND NEEDS

AI software quality validation has a number of major challenges due to the lack of research work results and engineering experience reports. These challenges are summarized below.

- Challenge #1: How to establish the quality assurance requirements and testing coverage criteria for AI systems which are built using machine learning methods based on big data?

- Challenge #2: How to use systematic methods to establish and develop quality test models for learning-based AI systems?

- Challenge #3: How to use a systematic method to prepare quality training datasets and coverage-oriented test datasets for AI-based functional features in learning-based AI for todays' AI systems?

- Challenge #4: How to define quality assurance standards systems, and develop adequate quality test coverage? - Challenge #5: How to develop automatic solutions and tools to support AI-based system validation? In addition, there are a number of issues in AI software testing. Here are the primary ones summarized below

AI System Validation Problems

Problem #1:
Lack of well-defined and experience-approved AI system validation models and methods addressing the special features of today’s AI systems developed based on big data and using machine learning and deep learning techniques

Problem #2:
Lack of well-defined quality assurance standards and assessment methods for machine learning based AI systems based on big data

Problem #3:
Lack of efficient and cost-effective automatic quality validation tools for machine learning based AI systems

AI System Validation Challenges

Major Challenges:

Challenge #1:
How to establish the quality assurance and testing coverage criteria for AI systems using machine learning methods based on big data?

Challenge #2:
How to use a systematic method to prepare and generate quality training datasets and test datasets for today's AI systems developed based on big data using machine learning and deep learning techniques?

Challenge #3:
How to use systematic methods and models to develop learning-based quality assurance systems or tools for machine learning based AI systems?

AI System Validation Challenges

Major Challenges:

Challenge #4:
When should we stop testing for big data-based AI systems using machine learning methods? Or How could we know that we have enough testing for AI systems?

Challenge #5:
How to prepare a clear and effective problem/bug report and conduct problem/bug analysis for AI system developers?

AI System Validation Needs

Need #1 – Developing well-defined adequate validation models and criteria to address and present the special features and needs in testing AI-based functional features, such as object detection and classification, recommendation and prediction features, and so on

Need #2 – Establishing well-defined quality assurance programs and standards to address the special quality parameters relating to AI functional features in AI-based systems.

Need #3 –More innovative adequate testing methods and test automation tools to address the special needs and features of AI software and applications to deal with the coverage of big data spaces.

Unlike conventional software test automation tools, these expected test automation solutions must consider AI's special features and needs listed below:

- Large-scale big data inputs with diverse formats, and structured and non-structured data;
- Learning-based and knowledge-based system evolutions;
- Non-oracles problems and rich oracle functions with uncertainty;
- New QoS parameters, such as accuracy, consistency, correctness, accountability, usability, and
- Automated quality test data generation and discovery using crow-sourcing approaches and learning-based solutions.

SAN JOSÉ STATE UNIVERSITY

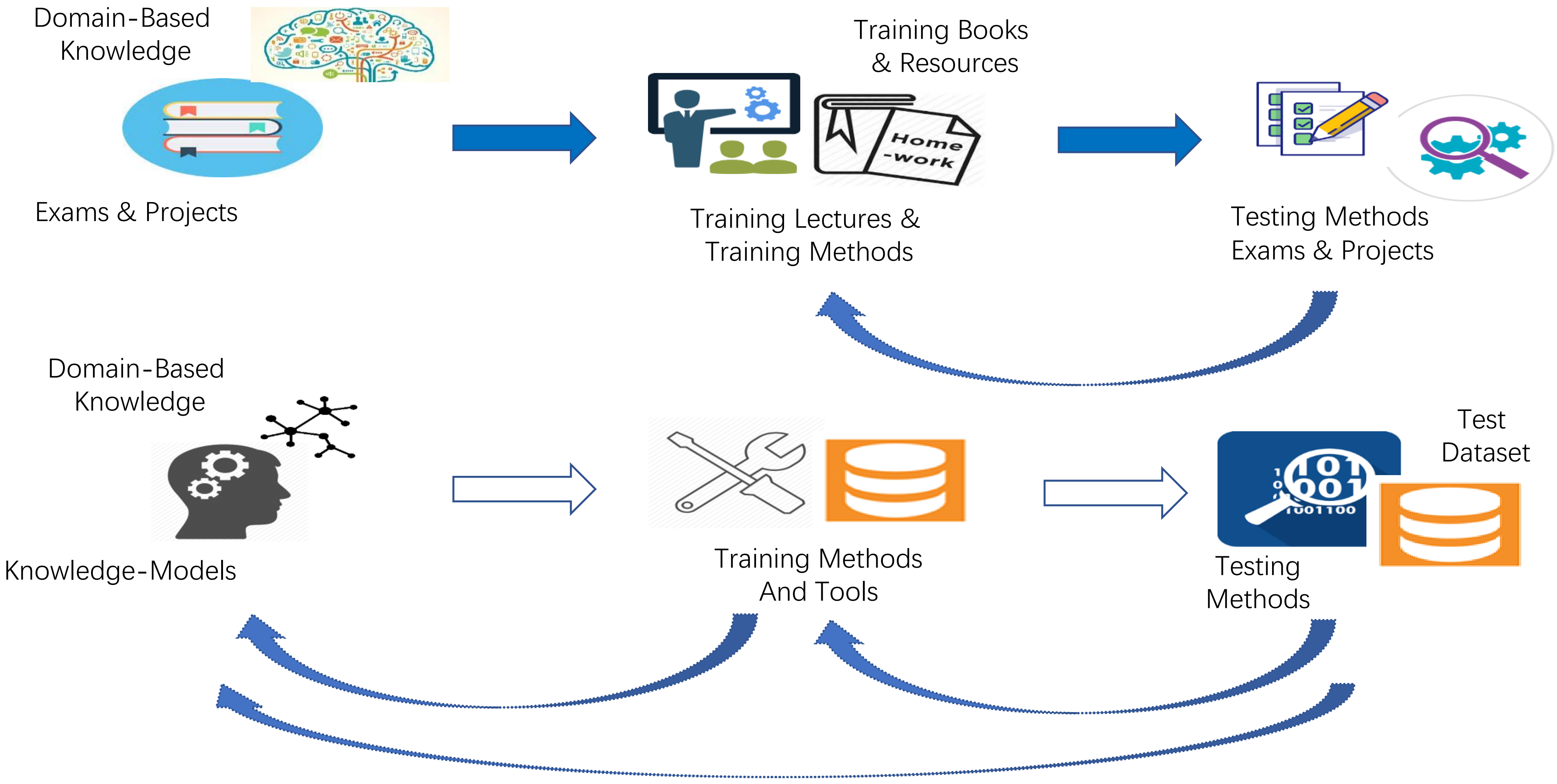
AI Testing – Training Process and Maturity Levels

Presented by: Jerry Gao, Professor, and Director

San Jose State University – Excellence Research Center on Smart Technology, Computing, and Complex Systems

Date: 6/4/2018

Knowledge-Based AI System Validation
– Validation Modeling, Methods, and Automation



Domain-Based Knowledge

Exams & Projects

Training Books & Resources

Training Lectures & Training Methods

Testing Methods Exams & Projects

Domain-Based Knowledge

Knowledge-Models

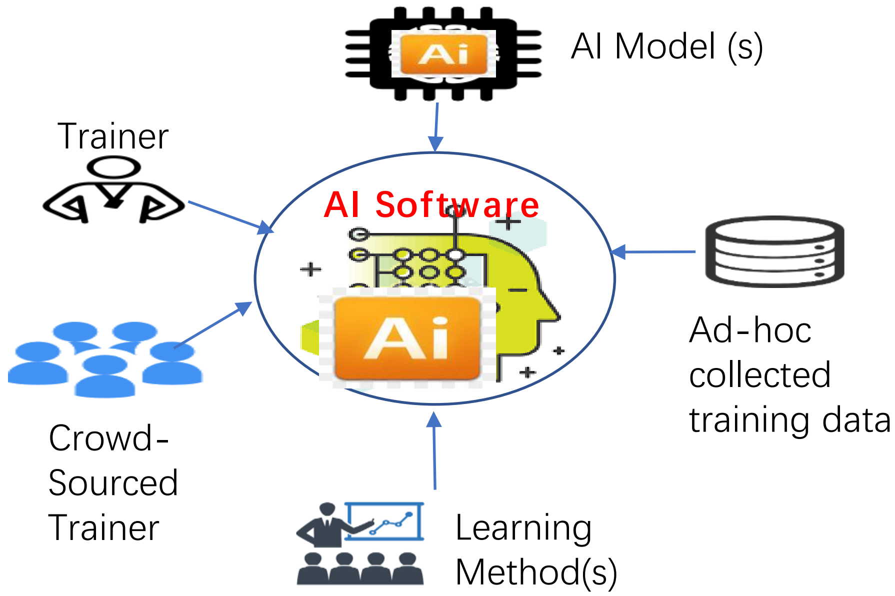
Training Methods And Tools

Test Dataset

Testing Methods

Training Modeling and Data Generation for AI Software

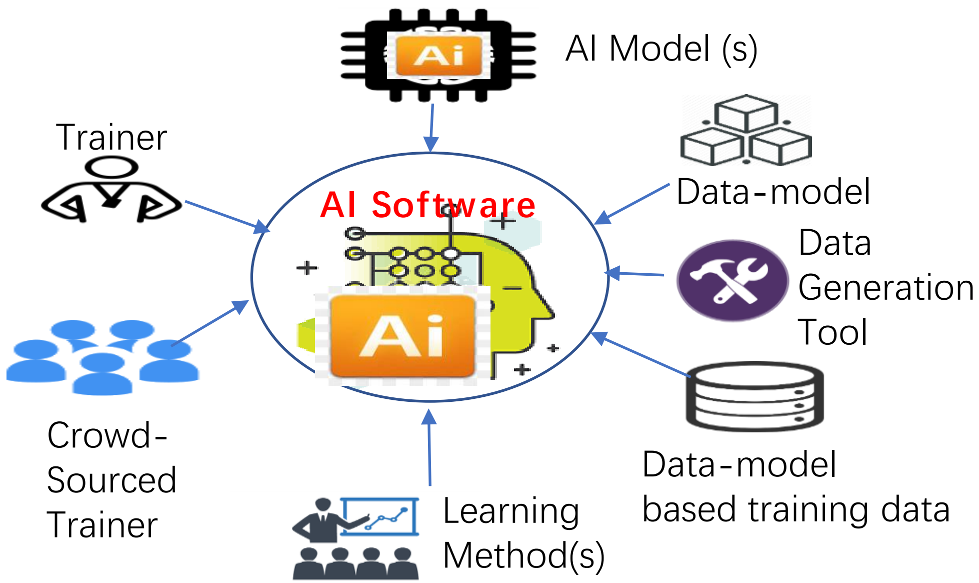
Ad-hoc Data Training



- Ad-hoc training data collection and preparation
- Ad-hoc data quality criteria for training data

Trainer: personal-based or crowd-sourced training
Approach: manual ad-hoc training approach

Data Model-based Training

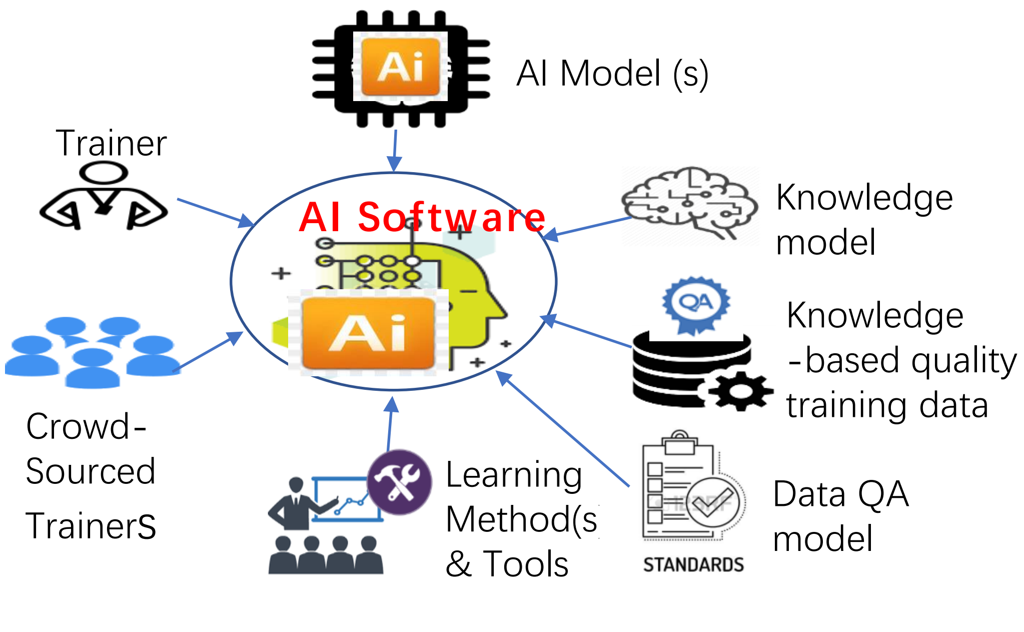


- Data model-based training data generation
- Ad-hoc data quality standards/criteria
- Model-based training data generation and simulation

Trainer: personal-based or crowd-sourced training
Approach: manual training approach
automatic training data simulation

Training Modeling and Data Generation for AI Software

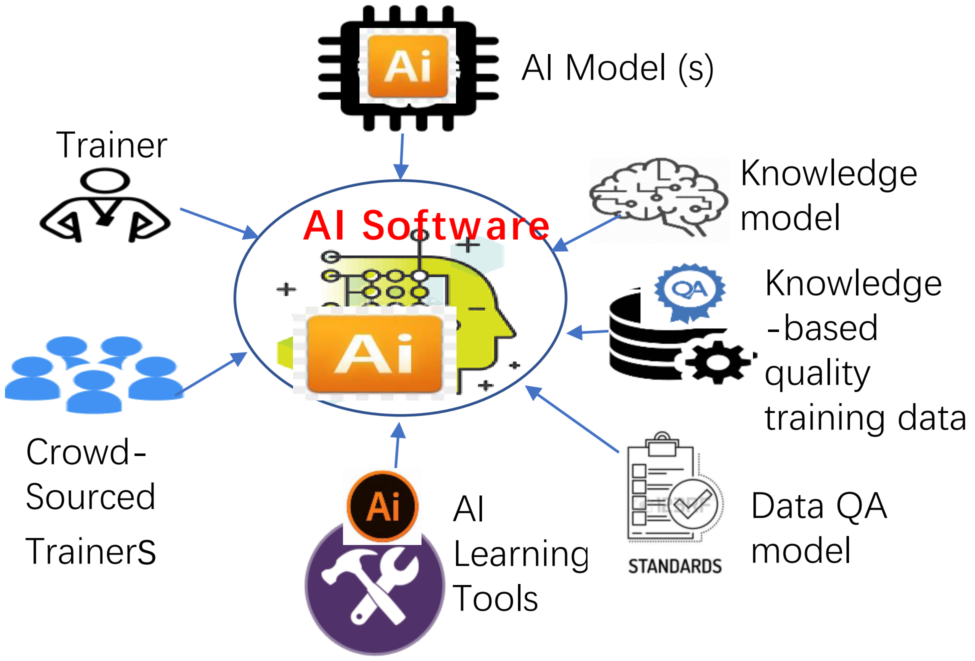
Knowledge-Based Quality Training



- Knowledge-based training data generation
- Quality-driven training data generation
- Model-based training data generation and simulation

Trainer: personal-based or crowd-sourced training
Approach: manual/automatic training approach

Knowledge-Based AI Quality Training



- Knowledge-based training data generation
- Quality-driven training data generation
- Model-based training data generation and simulation

Trainer: personal-based or crowd-sourced training
Approach: AI-based automatic training approach

SAN JOSÉ STATE UNIVERSITY

AI Testing – Test Automation

Presented by: Jerry Gao, Professor, and Director

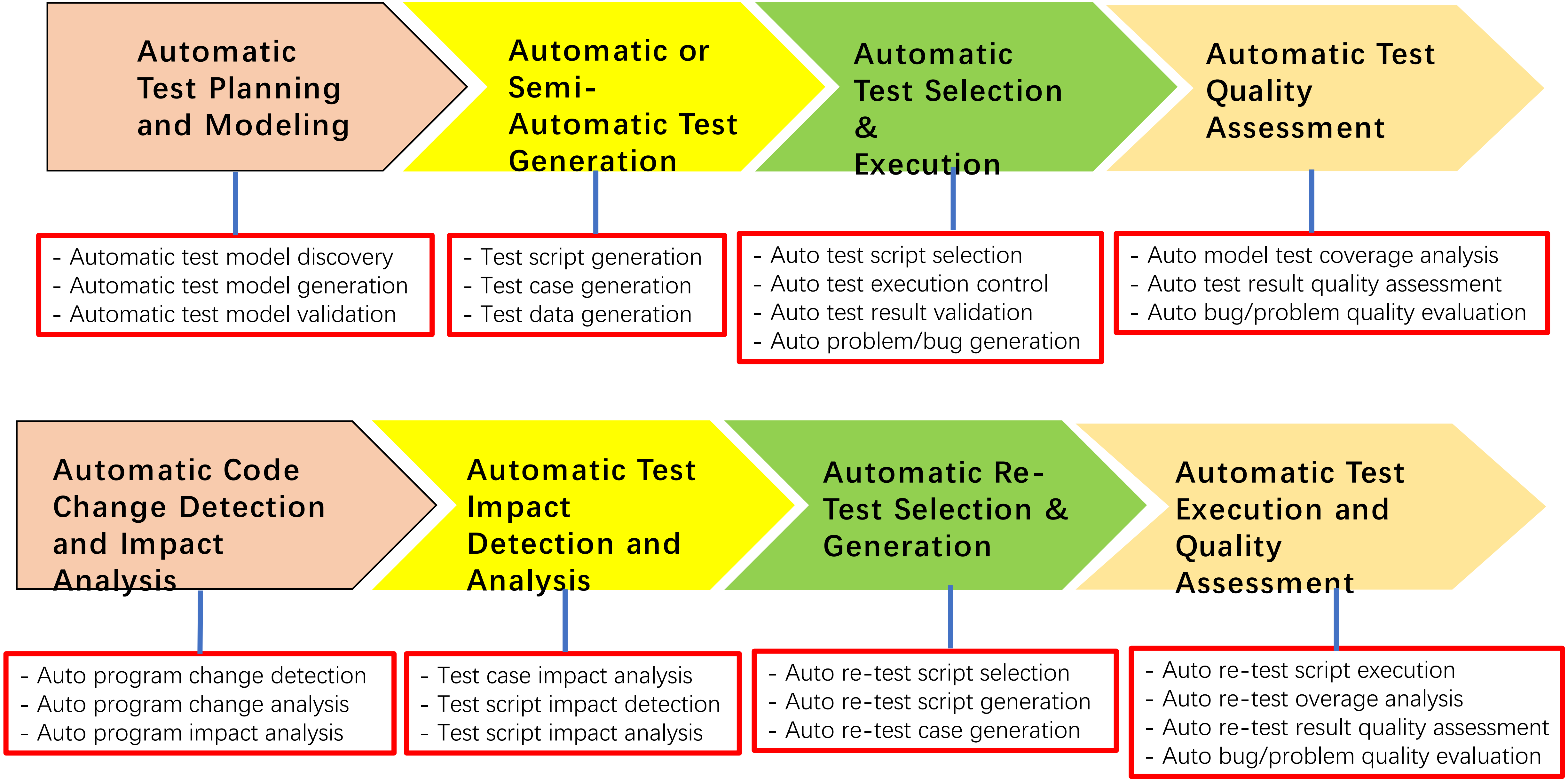
San Jose State University – Excellence Research Center on Smart Technology, Computing, and Complex Systems

Date: 6/4/2018

7 Innovative AI Test Automation Tools for the Future: The Third Wave

- Applitools
- SauceLabs
- Testim
- Sealights
- Test.AI
- Mabl
- ReTest

	Applitools	SauceLabs	Testim	Sealights	TestAI	Mabl	ReTest
Objectives	Ai-based automated maintenance		Ai-based test authorization and execution		Add an Ai brain to Selenium and Appium		
Approaches	Automatically understand the changes				Dynamic approach		
Test Format			Reducing flaky tests and test maintenance		Similar BDD syntax of Cucumber		
Test Execution	Auto visual test execution				Support GUI test execution		
Automatic Change Identification	GUI change and test identification				GUI screens and elements		
Platform/Tool	Visual Validation Tool	Cloud-Based	Cloud-Based		Tool		
Dependent Technology					Selenium and Appium		



Automatic Test Planning and Modeling

- Automatic test model discovery
- Automatic test model generation
- Automatic test model validation

Automatic or Semi-Automatic Test Generation

- Test script generation
- Test case generation
- Test data generation

Automatic Test Selection & Execution

- Auto test script selection
- Auto test execution control
- Auto test result validation
- Auto problem/bug generation

Automatic Test Quality Assessment

- Auto model test coverage analysis
- Auto test result quality assessment
- Auto bug/problem quality evaluation

Automatic Code Change Detection and Impact Analysis

- Auto program change detection
- Auto program change analysis
- Auto program impact analysis

Automatic Test Impact Detection and Analysis

- Test case impact analysis
- Test script impact detection
- Test script impact analysis

Automatic Re-Test Selection & Generation

- Auto re-test script selection
- Auto re-test script generation
- Auto re-test case generation

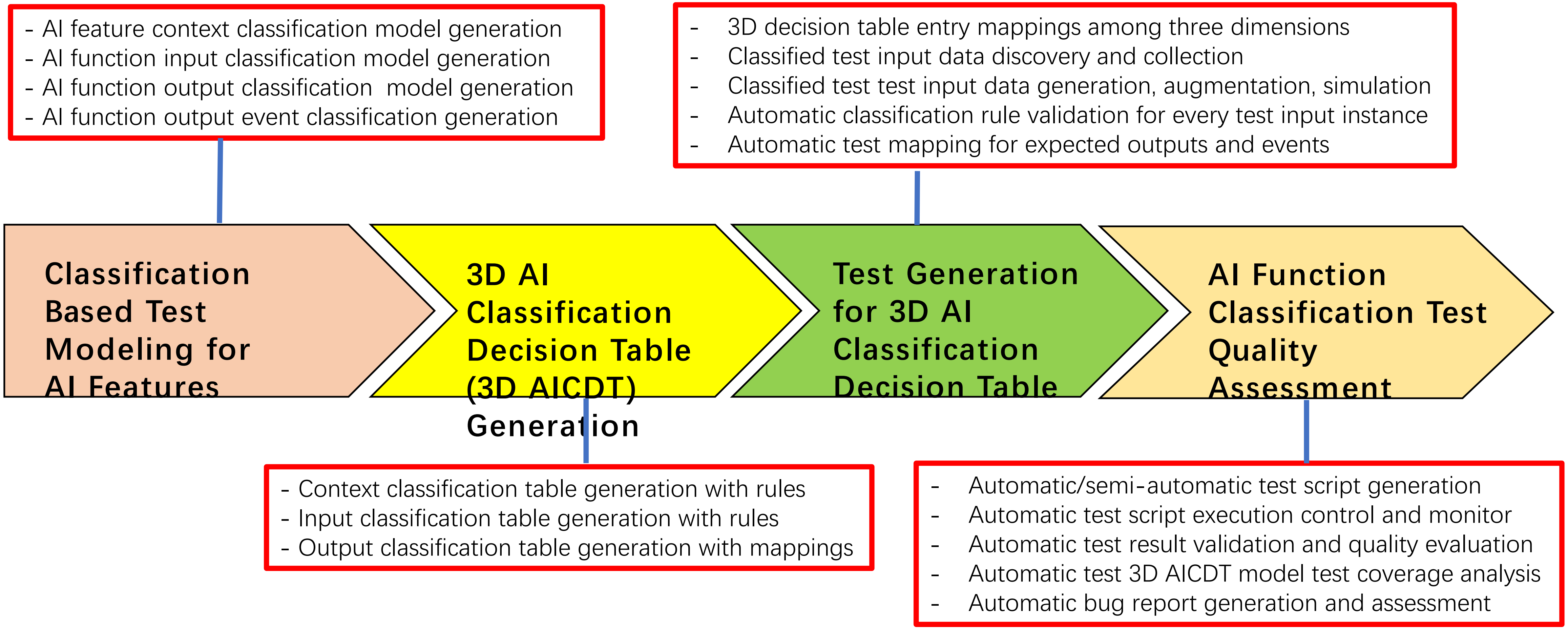
Automatic Test Execution and Quality Assessment

- Auto re-test script execution
- Auto re-test overage analysis
- Auto re-test result quality assessment
- Auto bug/problem quality evaluation

A Comparison Between AI Testing and Software Testing

Items	AI Testing	AI-Based Software Testing	Conventional Software Testing
Objectives	Validate and assure the quality of AI software and system by focusing on system AI functions and features	Leverage AI techniques and solutions to optimize a software testing process and its quality	Assure the system function quality for conventional software and its features
Primary AI testing focuses	AI feature quality factors: correctness, accuracy, consistency, timeless, completeness, and performance.	Optimize a test process in product quality increase, testing efficiency, and cost reduction.	Automate test operations for a conventional software process
Common system testing quality	System quality factors: performance, reliability, scalability, availability, security, throughput,	System quality factors: performance, reliability, scalability, availability, security, throughput,	System quality factors: performance, reliability, scalability, availability, security, throughput, ⋯
System function testing	AI system function testing: -Object detection & classification - Recommendation and prediction -Language translation	System functions, behaviors, user interfaces, ⋯.	System functions, behaviors, user interfaces, ⋯.
Test Selection	AI test model based test selection, classification, and recommendation	Test selection, classification, and recommendation using AI techniques	Rule-based and/or experience based test selection
Test Data Generation	AI test model based test data discovery, collection, generation, validation, and	AI-based test data collection, classification, and generation	Model-based and/or pattern based test generation
Bug Detection and Analysis	AI model-based bug detection, analysis, and report	Data-driven analysis for bug classification and detection, as well as prediction	Digital and systematic bug/problem management

A Classification-Based Test Automation



- AI feature context classification model generation
- AI function input classification model generation
- AI function output classification model generation
- AI function output event classification generation

Classification Based Test Modeling for AI Features

3D AI Classification Decision Table (3D AICDT) Generation

- Context classification table generation with rules
- Input classification table generation with rules
- Output classification table generation with mappings

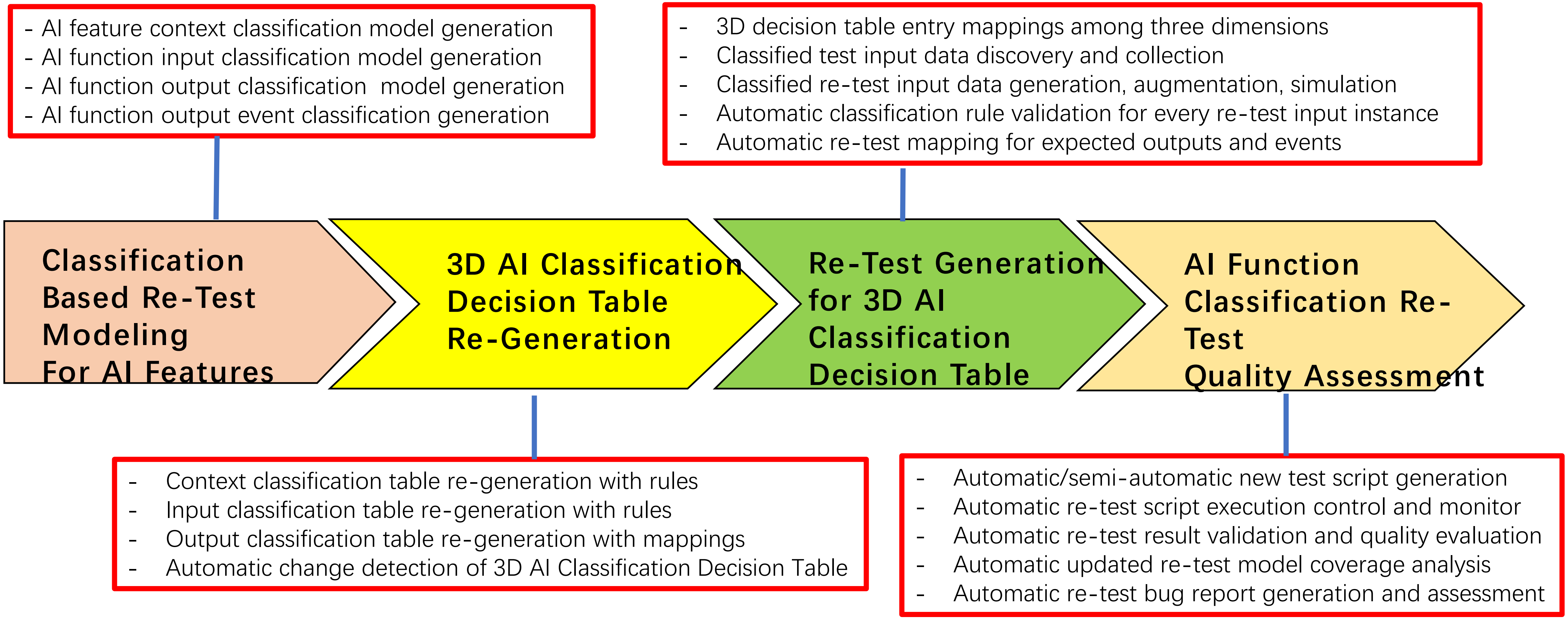
- 3D decision table entry mappings among three dimensions
- Classified test input data discovery and collection
- Classified test test input data generation, augmentation, simulation
- Automatic classification rule validation for every test input instance
- Automatic test mapping for expected outputs and events

Test Generation for 3D AI Classification Decision Table

AI Function Classification Test Quality Assessment

- Automatic/semi-automatic test script generation
- Automatic test script execution control and monitor
- Automatic test result validation and quality evaluation
- Automatic test 3D AICDT model test coverage analysis
- Automatic bug report generation and assessment

A Classification-Based Re-Test Automation



Classification Based Re-Test Modeling For AI Features

- AI feature context classification model generation
- AI function input classification model generation
- AI function output classification model generation
- AI function output event classification generation

3D AI Classification Decision Table Re-Generation

- Context classification table re-generation with rules
- Input classification table re-generation with rules
- Output classification table re-generation with mappings
- Automatic change detection of 3D AI Classification Decision Table

Re-Test Generation for 3D AI Classification Decision Table

- 3D decision table entry mappings among three dimensions
- Classified test input data discovery and collection
- Classified re-test input data generation, augmentation, simulation
- Automatic classification rule validation for every re-test input instance
- Automatic re-test mapping for expected outputs and events

AI Function Classification Re-Test Quality Assessment

- Automatic/semi-automatic new test script generation
- Automatic re-test script execution control and monitor
- Automatic re-test result validation and quality evaluation
- Automatic updated re-test model coverage analysis
- Automatic re-test bug report generation and assessment