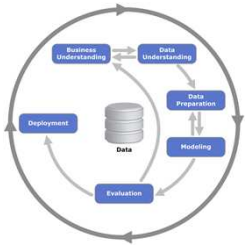
Institute of Electrical and Electronics Engineers

EZ Approach Requirements Syntax (EARS)

Richard Frederick, PMP (rfrederick.pmp@gmail.com; 214-755-7035 talk or text; www.linkedin.com/in/rfrederick)



CBDA CERTIFIED IIBA

ISO International Organization for Standardization

ANSI American National Standards Institute

sas

OMG OBJECT MANAGEMENT GROUP

INCOSE International Council on Systems Engineering

IEEE

PMI Project Management Institute

Scaled Agile Framework

IIBA International Institute of Business Analysis

UNIFIED MODELING LANGUAGE

W3COMPUTING A DEVELOPER REFERENCE WEBSITE

Requirements Development Management (RDM) for

Business Data Analytics

(The Language of Data)

Richard Frederick, PMP (rfrederick.pmp@gmail.com; 214-755-7035 talk or text; www.linkedin.com/in/rfrederick)

Requirements Agenda

Business Data Analytics "The Language of Data"

01-FOUNDATION

- **LAB-Introductions (Name, Job Title, Objectives)**
- Dashboards and Robots (Data Mining & Machine Learning)
- Requirements and Testing (Four Quadrants)
- What are Structured Language Requirements? (Structured English and Structured Query Language)
- Why Should You Care? (Primary Source of Project Problems)
- How Do They Work? (Discreet Intellectual Property Inventory)
- Types of Requirements (Product, Project, DATA)
- Natural Language Processing (Morphology, Semantics, Syntax and Linguistics)
- OMG-SBVR (Semantics of Business Vocabulary & Rules)
- IEEE-EARS (Easy Approach to Requirements Syntax)
- INCOSE (Rules for Writing Requirements) & QVscribe
- Waterfall and Agile (Assembly Methods)

02-ELICIT

- **LAB-Vision/Scope (Seek to Understand)**

Elicitation Techniques:

- Document Analysis (Low Hanging Fruit)
- Interface Analysis (Navigation & Functionality)
- Benchmarking (Actual Data)
- Brainstorming (Every Idea is a Good Idea until it becomes a Bad Idea)
- Prototyping (Minimum Viable Product)
- Reverse Engineering (Begin with the End in Mind)
- Interview (Thinking Questions)
- Workshop (Group Interviews)
- Observation (What do you See?)
- Survey Questionnaire (Paper equals proof)

03-ANALYZE

- What are Models? (Pictures of Language)
- **LAB-The Language of Modeling (GIVEN pre WHEN process THEN output-result)**
- Types of Models (Context-Structure, Usage, Data Behavior, Process Flow)
- Context-Structure (Vision, Roadmap, Scope WBS)
- Usage (EPIC, UseCase, UserStory, Feature)
- Data Behavior (ERD, JOIN-Denormalization, Star Schema, Dimensional OLAP, Dashboard, Intelligence)
- Data Behavior (Data Dictionary, DataFlow, Data Structure Instance, Data Element Attribute, Data Store)
- Data Behavior (Process Logic, Business Rules)
- Process Flow (Swimlane)

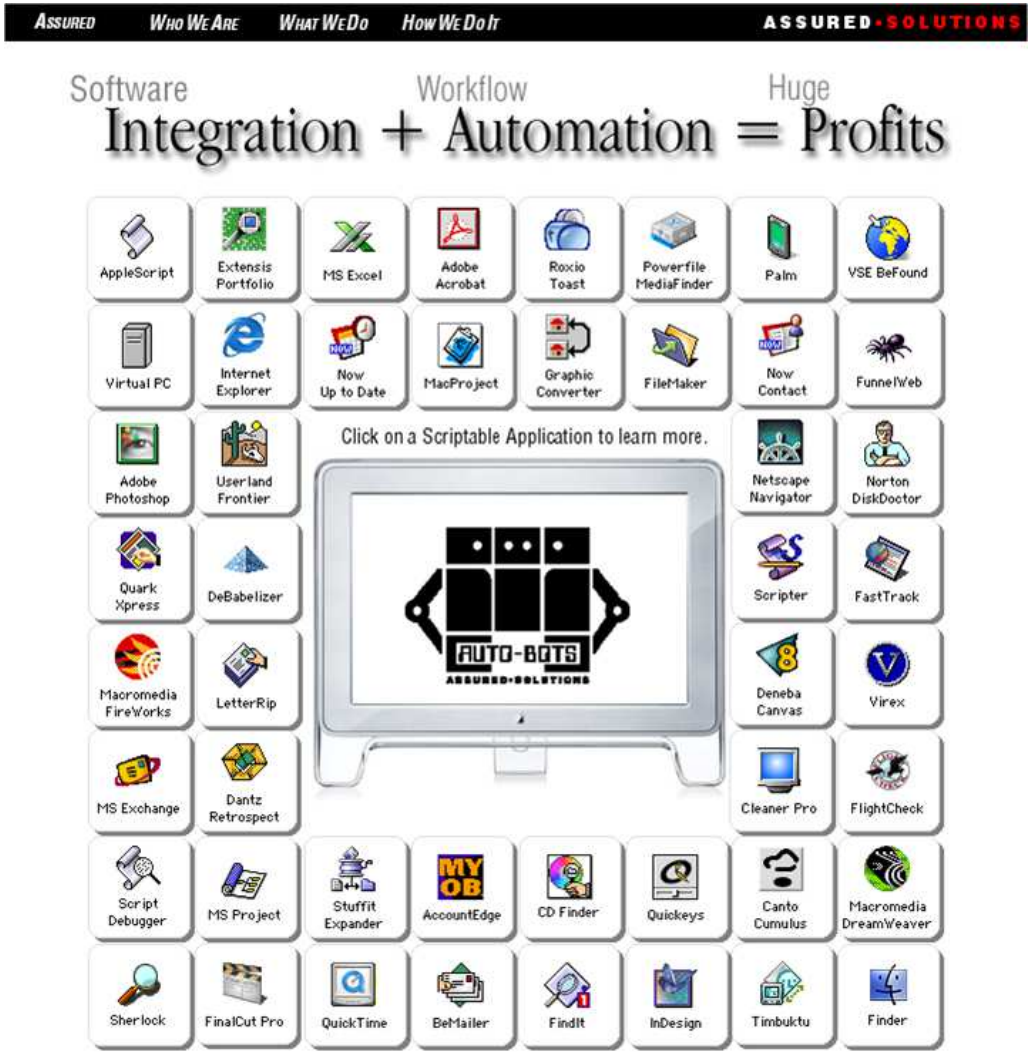
04-DOCUMENT

- Categorization, Organization, Documentation, Integration, Automation
- Making Documents Easy to Read (Fonts & Navigation)
- Document Types (BRD, TRD)
- **LAB-Business Requirement Document (Concept of Operation)**
- Technical Requirement Document (System Specification)

05-VALIDATE

- Validation thru Triangulation (Prep Drills)
- Traceability (Project Unique Identifier)
- Requirements Baseline (ROM Estimate, Planning Estimate, Definitive Estimate)
- **LAB-Estimating Story Points (Complexity and Risk)**
- Lessons Learned (Course Wrap-Up)

Assured Solutions

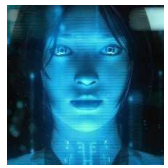


Overview "Seek to Understand"

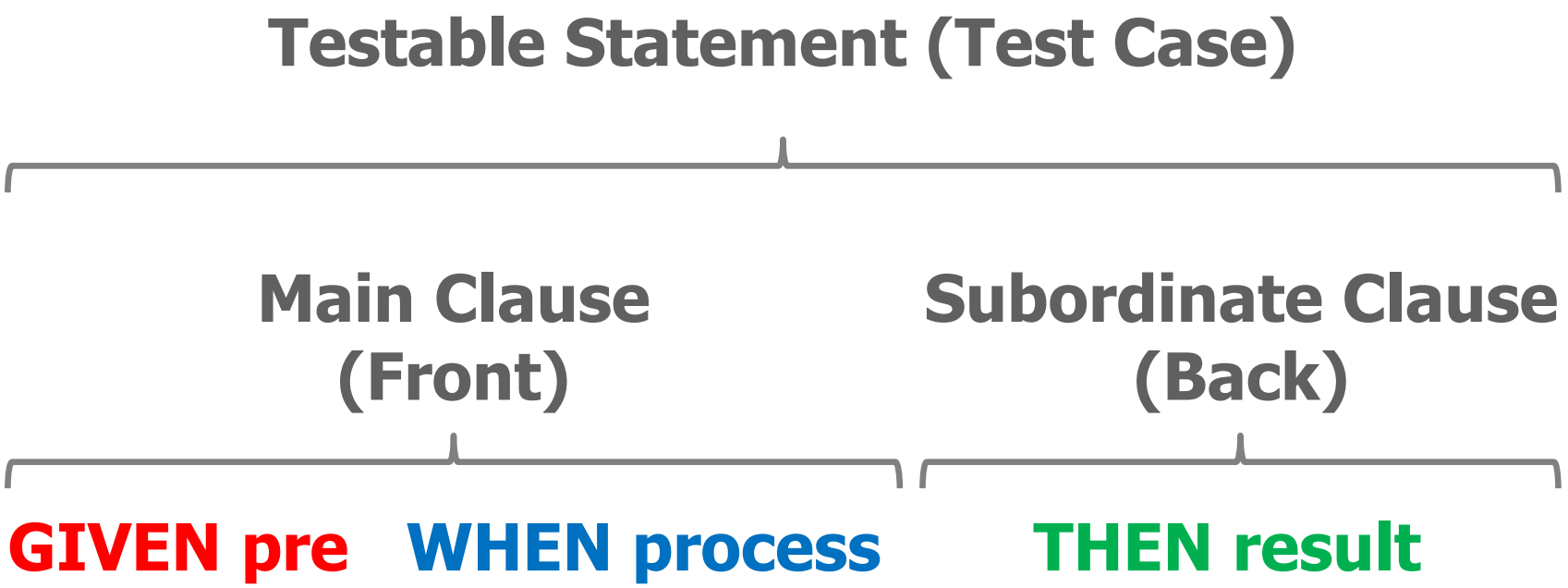
- What is a Requirement?
  "Requirement is a Testable Statement"
- IEEE-EARS
  "Easy Approach to Requirements Syntax





A Requirement is a Testable Statement



Institute of Electrical and Electronics Engineers

EZ Approach Requirements Syntax (EARS)

EZ Approach Requirements Syntax (EARS)

Requirement Type	• Syntax Pattern
Ubiquitous Universal (TSS)	• The <system> shall <system response>. [PUI]
Event-Driven (WHEN-THEN)	• WHEN <trigger> <optional precondition> THEN the <system> shall <system response>. [PUI]
Decision-Driven Extension & Risk Statement (IF-THEN)	• IF <condition or event occurs>, THEN the <system> shall <system response>. [PUI]
State-Driven (WHILE-THEN)	• WHILE <in a state>, the <system> shall <system response>. [PUI]
Optional Feature (WHERE-THEN)	• WHERE <feature is included>, THEN the <system> shall <system response>. [PUI]
Combined (AND)	• AND <Multiple Conditions>, the <system> shall <system response>. [PUI] • (combinations of the above patterns)

Ubiquitous-Universal Requirements (TSS)

Syntax Format for ubiquitous requirements.

- **The <system> shall <system response>. [PUI]**
- **The <Garage Door Opener> shall <display the Power Status>. [PUI]**

Examples of ubiquitous requirements.

- *The **FCC** shall control communication on the Avionics Bus in accordance with MIL-STD-1553B and Table 3.1 of the program ICD. [PUI]*
- *The software shall be written in C++. [PUI]*

Event-Driven requirements (WHEN-THEN)

- Syntax Format for event-driven requirements.
- **WHEN <trigger> <optional precondition> THEN the <system> shall <system response>. [PUI]**
- **WHEN <the Open Button is Pressed> THEN the <Garage Door Opener> shall <open the Garage Door>. [PUI]**

- Examples of event-driven requirements.
- ***WHEN*** *the power button is depressed while the system is off,* ***THEN*** *the system shall initiate its start-up sequence. [PUI]*
- ***WHEN*** *the water level falls below the Low_Water_Threshold,* ***THEN*** *the software shall open the water valve to fill the tank to the High_Water_Threshold. [PUI]*

Decision-Driven Extensions & Risk Statements (IF-THEN)

- Syntax Format for unwanted behaviour requirements.
- **IF <condition or event>, THEN the <system> shall <system response>. [PUI]**
- **IF <the Obstruction Sensor Fails>, THEN The <Garage Door Opener> shall <enter Emergency Mode>. [PUI]**

- Examples of unwanted behaviour requirements.
- ***IF*** *the battery charge level falls below 20% remaining*, ***THEN*** *the system shall go into Power Saver mode. [PUI]*
- ***IF*** *the input checksum is invalid*, ***THEN*** *the system shall reject the new data and retain the previous data in memory. [PUI]*

Decision-Driven Extensions & Risk Statements (IF-THEN)

EXTENSIONS:

- **Alternate (IF, THEN, RESUME)**
- **Exception (IF, THEN, USE CASE ENDS)**

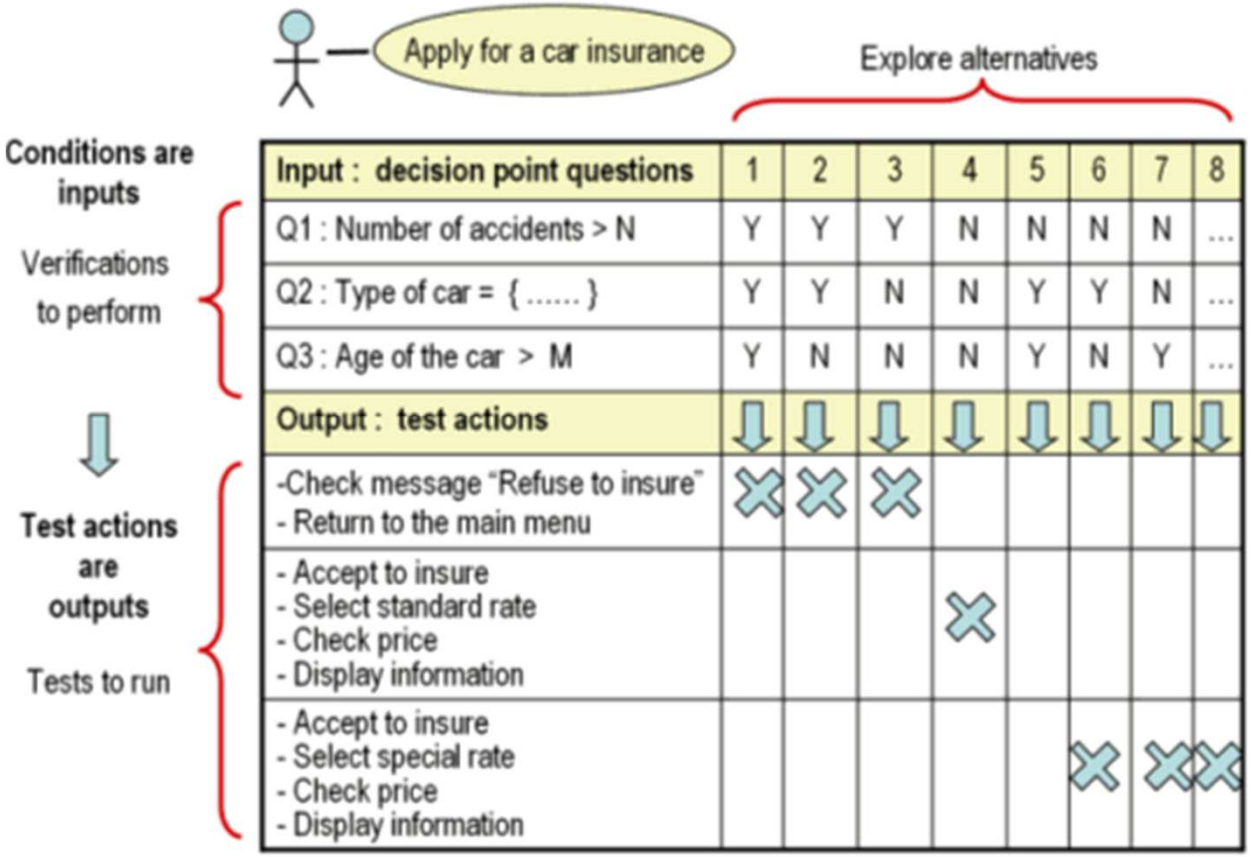
A Sample Decision Table

Condition	Requirement Number				
	1	2	3	4	5
Requester is authorized	F	T	T	T	T
Chemical is available	—	F	T	T	T
Chemical is hazardous	—	—	F	T	T
Requester is trained	—	—	—	F	T
Action					
Accept request			X		X
Reject request	X	X		X	

Decision-Driven Extensions & Risk Statements (IF-THEN)

EXTENSIONS:

- **Alternate (IF, THEN, RESUME)**
- **Exception (IF, THEN, USE CASE ENDS)**



Input : decision point questions	1	2	3	4	5	6	7	8
Q1 : Number of accidents > N	Y	Y	Y	N	N	N	N	...
Q2 : Type of car = { ..... }	Y	Y	N	N	Y	Y	N	...
Q3 : Age of the car > M	Y	N	N	N	Y	N	Y	...
Output : test actions	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩
-Check message "Refuse to insure" - Return to the main menu	X	X	X					
- Accept to insure - Select standard rate - Check price - Display information				X				
- Accept to insure - Select special rate - Check price - Display information						X	X	X

State-Driven Requirements (WHILE)

- Syntax Format for state-driven requirements.
- **WHILE <in a state>, the <system> shall <system response>. [PUI]**
- **WHILE <the Garage Door is Open>, the <Garage Door Opener> shall <turn the Garage Light On>. [PUI]**

- Examples of state-driven requirements.
- ***WHILE*** *in the Power Saver mode, the system shall limit screen brightness to a maximum of 60%. [PUI]*
- ***WHILE*** *the autopilot is engaged, the flight control panel shall display a visual indication that the aircraft is under autopilot control. [PUI]*

Optional Feature Requirements (WHERE-THEN)

- Syntax Format for optional feature requirements.
- **WHERE <feature is included>, THEN the <system> shall <system response>. [PUI]**
- **WHERE <the Keypad is Installed>, THEN the <Garage Door Opener> shall <require Passcode to Open Garage Door>. [PUI]**

- Examples of optional feature requirements.
- ***WHERE*** *the automobile is furnished with the GPS navigation system,* ***THEN*** *the automobile shall enable the driver to mute the navigation instructions via the steering wheel controls. [PUI]*
- ***WHERE*** *the encryption hardware is installed,* ***THEN*** *the system shall encrypt data using that encryption hardware, instead of using a software algorithm. [PUI]*

Combined Requirements (AND)

- Syntax Format for complex requirements.
- **<Multiple Conditions>, the <system> shall <system response>. [PUI]**
- **WHILE <in a state>, AND WHEN <trigger> THEN the <system> shall <system response>. [PUI]**
- **WHILE <the Garage Door is Closing>, AND WHEN <the Open Button is Pressed> THEN the <Garage Door Opener> shall <stop the Garage Door>. [PUI]**

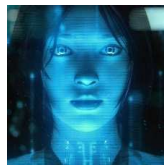
- Examples of complex requirements.
- ***WHEN*** *in A/A mode with an A/A missile selected and the Master Arm switch in ARM,* ***AND*** *the WEAPON RELEASE signal is received from the SSC,* ***THEN*** *the SMS shall send LAUNCH signal to the selected station. [PUI]*
- ***WHILE*** *in A/A mode,* ***IF*** *the selected station returns the WEAPON FAILED signal,* ***THEN*** *the SMS will set the FAIL signal for the selected station, and select the next station in the station selection sequence. [PUI]*

Summary “Train the Trainer”

- What is a Requirement?
  “Requirement is a Testable Statement”
- IEEE-EARS
  “Easy Approach to Requirements Syntax





Let’s stay in contact:

Richard Frederick, PMP

214-755-7035 (text or talk)

Rfrederick.pmp@gmail.com

www.linkedin.com/in/rfrederick

https://meetings.hubspot.com/rfrederick-pmp