S-Square

TRUSTED . TESTED . COMMITTED

CERTIFIED NMSDC MBE 2018

S-Square - LowCode/NoCode (LC/NC) Enabling Technology Presentation

Jeff Friedman,
VP, Sales & Customer Success

Version - 20221215_V1

Current Challenges in Traditional Application Development

Long Development Timelines

- Custom development with standard SDLC processes
- Long incubation period before seeing a MVP
- Minor changes require long turn around time for design, build and testing.

High Capital Expenditure and Operating Costs

- Investment in Software platforms and Infrastructure for custom development
- Higher support costs due to diverse support requirements

Disparate Technology Landscape

- Multiple small projects using disparate technologies
- No uniform platform to manage small developments

Developer Shortages

- Developer shortages and skill-set challenges
- Multiple small productivity projects get deprioritized

6 Generations of Programming Languages

First generation (1GL) - machine-level programming language used to program first-generation computers
Examples: machine-level programming languages

Second generation (2GL) - assembly languages. Examples: Assembly

Third generation (3GL) - more machine-independent (portable) and more abstract therefore more programmer-friendly than previous generations of languages
Examples: Fortran, COBOL, BASIC, Pascal, C, C++, Perl, Python, Java, JavaScript, Ruby, PHP, C#

Fourth generation (4GL) - include support for database management, report generation, mathematical optimization, GUI development, or web development. Examples: ABAP, Unix Shell, SQL, PL/SQL, Oracle Reports, R

Fifth generation (5GL) - any programming language based on problem-solving using constraints given to the program to make the computer solve a given problem without the programmer, rather than using an algorithm written by a programmer. Examples: Prolog, OPS5, Mercury

Sixth generation (6GL) - programming language based on visual development. The overall umbrella term for these is "NoCode". Examples: Appian, WEM.io, Bubble.io

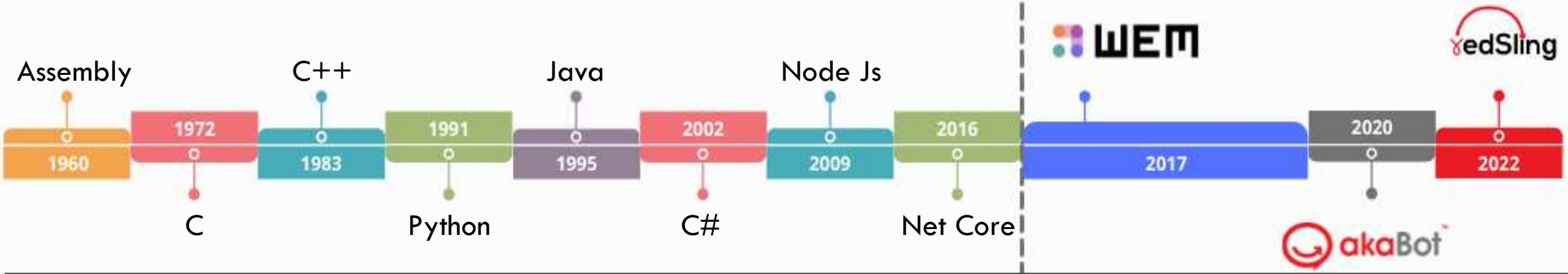
Reinventing Software Development

Traditional Coding

Requiring expensive, hard to retain code-linquists

No-Code

Empowering transforming support to employ business-knowledgeable techno-functional resources



Traditional computer languages require programmers to translate their thinking process into code built for the CPU and memory

Optimized for how we humans think. Converting natural thinking process into working software

Digital Transformation. Legacy Modernization. Business Velocity.

80%

COST REDUCTION

Empowers employing business knowledgeable (techno-functional) resources instead of costly, hard to retain code-linquists to build, deploy and maintain secure scalable enterprise-grade software.

10%

FASTER TIME-TO-MARKET

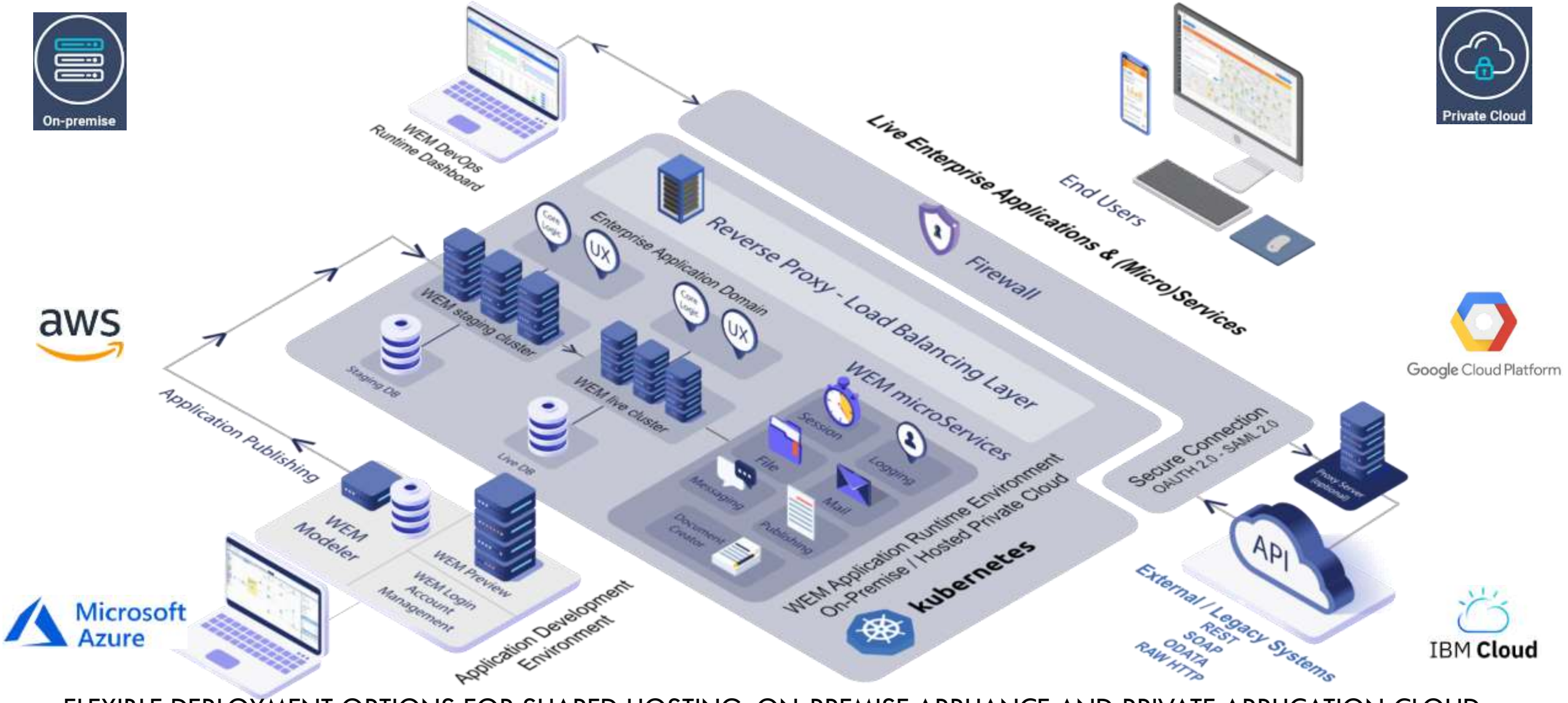
View app development in real-time. Deploy and update applications with a single click. Deliver software 10 times faster than traditional programming methods.

100%

ALIGNED TO BUSINESS

Translate innovative business ideas to custom software built with no code app builder at the speed of, and fully aligned with, business requirements.

SCALABLE, SECURE CLOUD ARCHITECTURE



FLEXIBLE DEPLOYMENT OPTIONS FOR SHARED HOSTING, ON-PREMISE APPLIANCE AND PRIVATE APPLICATION CLOUD

3rd Party LCNC Marketplace Product Evaluation

Criteria	WEM	Betty Blocks	Power Apps	OutSystems	Mendix
Category	No Code	Low code	Low code	Medium to high code	Low code
Platforms	Web, native apps	Web apps	Web, native apps	Web, native apps	Web, native apps
Data Model	Drag & Drop	Visual Editor	Tables	Visual Editor	Visual editor
Visual Editor	Web-based	For backend apps	Web-based	Many designer	Web-based, desktop-based
Workflows	Drag & Drop	Action Modeler	MS Flow	Visual modeler	Visual modeler
Look & Feel	Custom templates	Custom js/css/html	Customizable	Custom js/css	Custom js/css
Environment	Public, private cloud, on premise	Public cloud, on premise	Public, private cloud, on premise	Public, private cloud, on premise	Public, private cloud, on premise
Release Management	Fully	Fully	Partially	Fully	Fully
Integration	All API standards	JSON, SOAP/REST	Office365, REST	SOAP/REST	SOAP/REST

Use Case – DATA INTEGRATION

With a workforce size of more than 85,000 and a turnover of €49 billion, this global auto manufacturer of luxury cars worldwide, headquartered in Germany. has its presence in over 100 markets worldwide.

CUSTOMER CHALLENGES

- Joined with data about the vehicle and the knowledge expected to settle on moment driving choices, produces up to **4 terabytes of information each hour** for one test vehicle.
- That information is broken down for continuous occasions as overseen by the vehicle, yet later the information is investigated by producers, scene by scene, to distinguish novel driving conditions that can be utilized to illuminate the man-made intelligence that underlies the vehicle's independent tasks.
- Integration with various partner systems.
- The issue of scale becomes clear as many long stretches of recorded information should be looked for novel cases that would then be able to be utilized to "train" the independent driving calculations on the most proficient method to deal with those new circumstances. This can require a long time of exploration.

WEM ADVANTAGES

- **Quicker** information ingests rates (minutes as opposed to days), quicker advancement of calculations, more limited emphasis cycles, and huge algorithmic execution gains.
- **Integration** with various partner systems
- Faster agile development, short time to market (in production in a few months)
- Cloud solution for an extensive amount of data
- Easy to extend the application

PROBLEM

The auto manufacturer was tested with scaling advancement of **autonomous vehicles**, requiring a wide scope of abilities and capacities, and wanted to develop and implement the same into their vehicles. These advanced systems have less to do with how to make a vehicle go and everything to do with how to make it protected and keen, likeability in man-made consciousness (simulated intelligence), AI, cutting edge software engineering, and information the board.

SOLUTION

WEM helped the organization to accelerate its self-driving vehicle research and development, fabricating an application system for gathering and dealing with the monstrous information streams made by the car maker's test armada. AI and profound learning arrangements were made to dissect the information rapidly and to consequently signal fascinating experiences that could give important examples to the armada's independent driving programming. Effective use of the independent driving stage, tool stash, gas pedals, and skill helped produce various advantages for the car producer.

Representative WEM Enterprise Customers

MINISTERIO
DE TRANSPORTES, MOVILIDAD
Y AGENDA URBANA

Thank You

Jeff Friedman,
VP, Sales & Customer Success

S-Square Systems, Inc.
4225 Executive Square Suite 600
La Jolla, CA 92037
+1 858-213-7063, +1 858-764-4441

S-Square
TRUSTED . TESTED . COMMITTED