



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

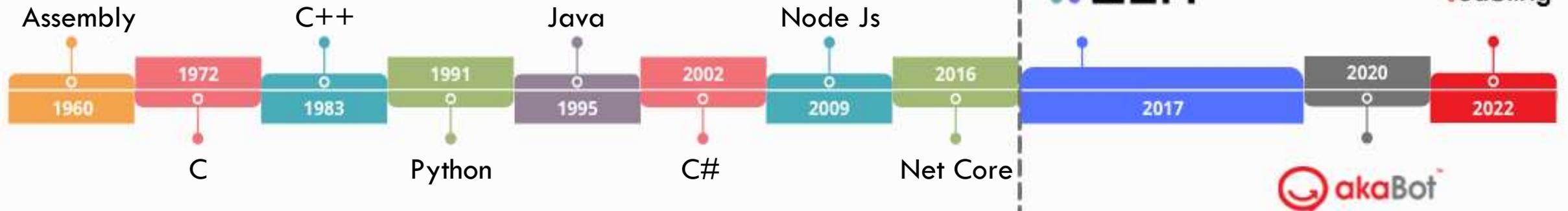
Reinventing Software Development

Traditional Coding

Requiring expensive, hard to retain code-linguists

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-linguists 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.



Banks,
Financial
Services and
Insurance >



Healthcare >



Telecommunication
>



Education &
Training >



Manufacturing
>



Public Sector
>



Automotive
>



Real Estate
>

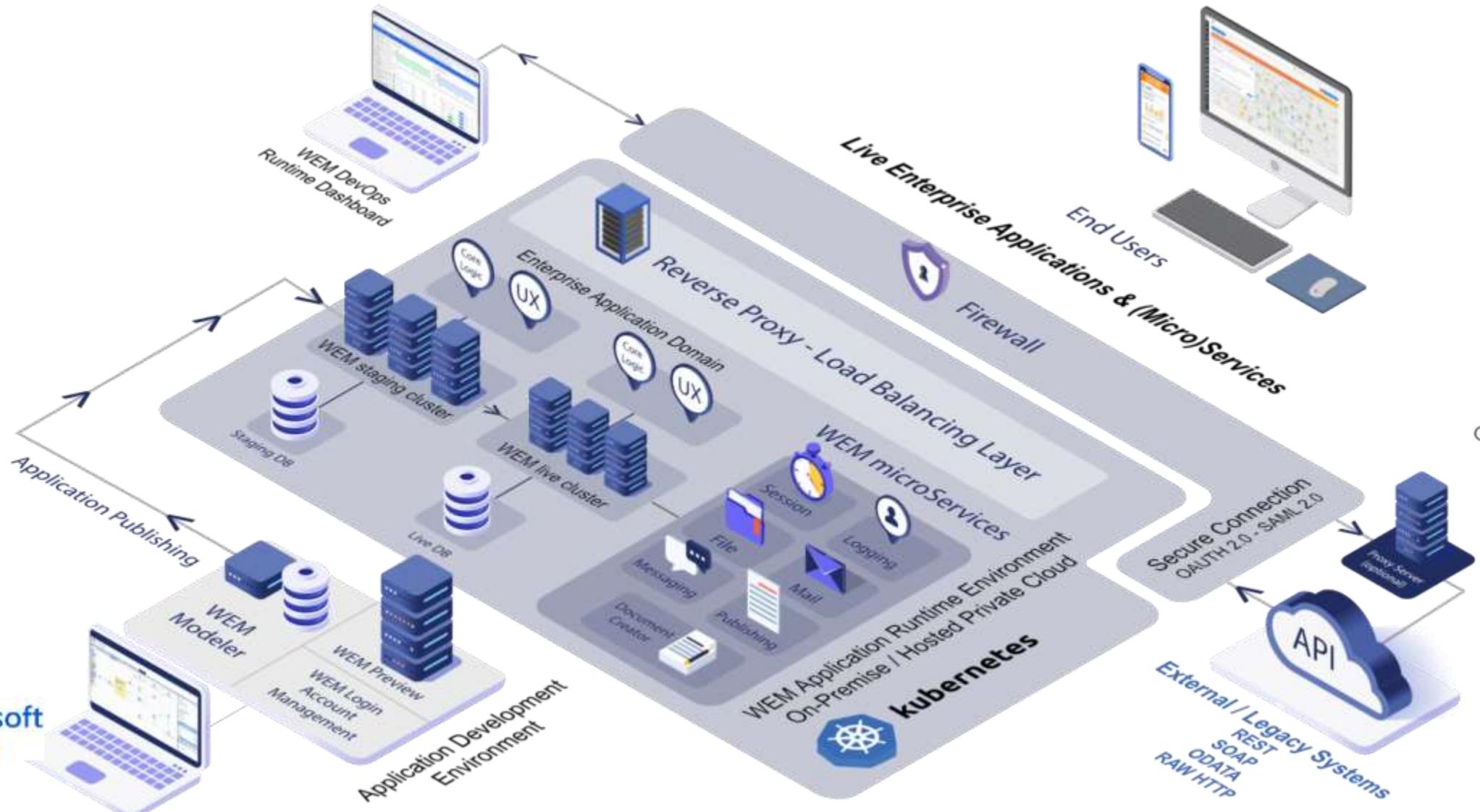
SCALABLE, SECURE CLOUD ARCHITECTURE



Google Cloud Platform



IBM Cloud



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 – ICU Mechanical Ventilation Application

This is a medical center and a teaching hospital located in the eastern-central part of the Netherlands. This hospital specializes in patient care, scientific research, teaching, and training. This is a 600 bedded hospital with nearly 12,000 employees and more than 3,000 students studying there.

PROBLEM

They wanted to build an ICU mechanical ventilation application for the Covid-19 patients. The motive behind such an app was to develop a solution to enable increased survival of Covid-19 patients in the ICU. It was realized that the Covid-19 patients developed hypoxemia due to ARDS but other respiratory acidosis like hypercapnia was developed due to the protective ventilation strategies of the ICU. Every patient has different pulmonary compliance therefore calculating respiratory compliance was realized to be of great value and therefore a system was needed for the same which could be used with a ventilator.

SOLUTION

An application for calculating the respiratory compliance for Covid-19 patients was developed with WEM. The application so developed is capable of making calculations of respiratory system compliance along with other respiratory mechanics like mechanical power, driving pressure, resistance, etc. The tool also allows for separate calculations for the lungs and the chest wall. The application works for volume control mode as well as the pressure support mode of the ventilator.

CUSTOMER CHALLENGES

- Integration of various complex respiratory compliance calculations in the system.
- The tool was to be developed for the Covid-19 patients on ventilator support therefore high accuracy was required.
- The tool was to be developed to be used upon every patient
- Separate calculations for lungs and chest wall were to be incorporated

WEM ADVANTAGES

- **An easy to use and no-code web environment to be maintained by non-IT skilled employees**
- **Agile development, week to week results, short time to market**
- **Cloud solution offers flexible workspaces (not tied to a location)**
- **An easy to extend application**
- **Fast return on investment**

Representative WEM Enterprise Customers



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