



Solutions in the Next Era of Digital Transformation

Kyle Thomas – Industry Lead, Appian Public Sector
Gaurav Shekhar – Professor, UT Dallas

23rd September 2022

Start with the why.

- Shortage of developers
- Technical debt
- Great resignation
- People expect more

What can we do about it?

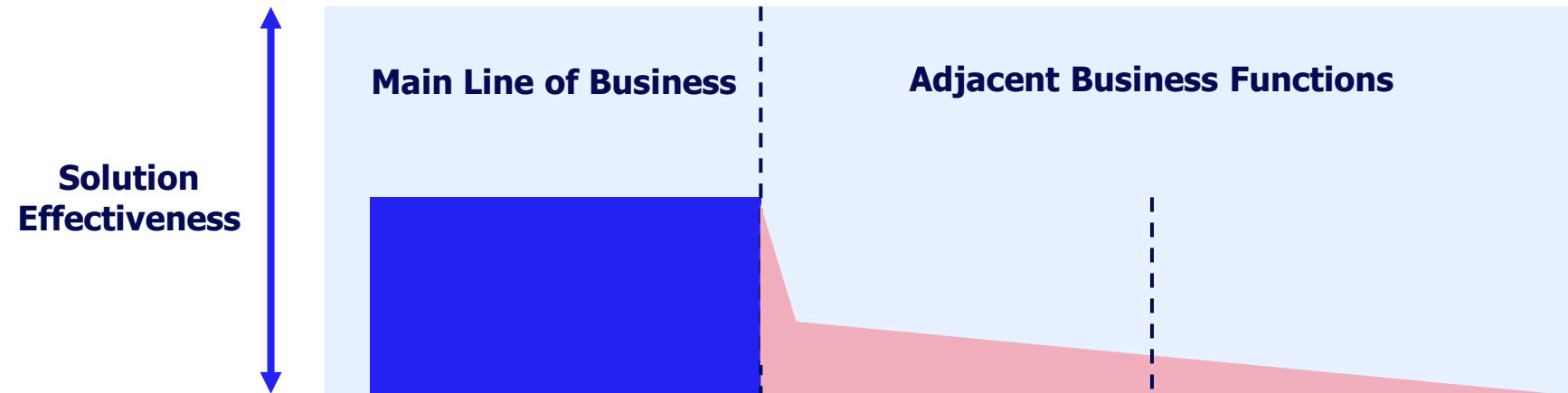
- Get low
- Get human
- Harness your data
- Get moving

Purpose and takeaways.

Brief look at digital transformation from the perspective of solutions – key components of any enterprise

- **Spot patterns** of newer and older solutions when procuring.
- Know **critical questions** to ask your vendors.
- Insights for **non-tech conversations** with business leaders.

Silo solutions.



- Very common: Historically, solutions are **siloed by design**
- **Business applicability drops rapidly** outside the silo
- Understand why this happens and **limit it in the future**

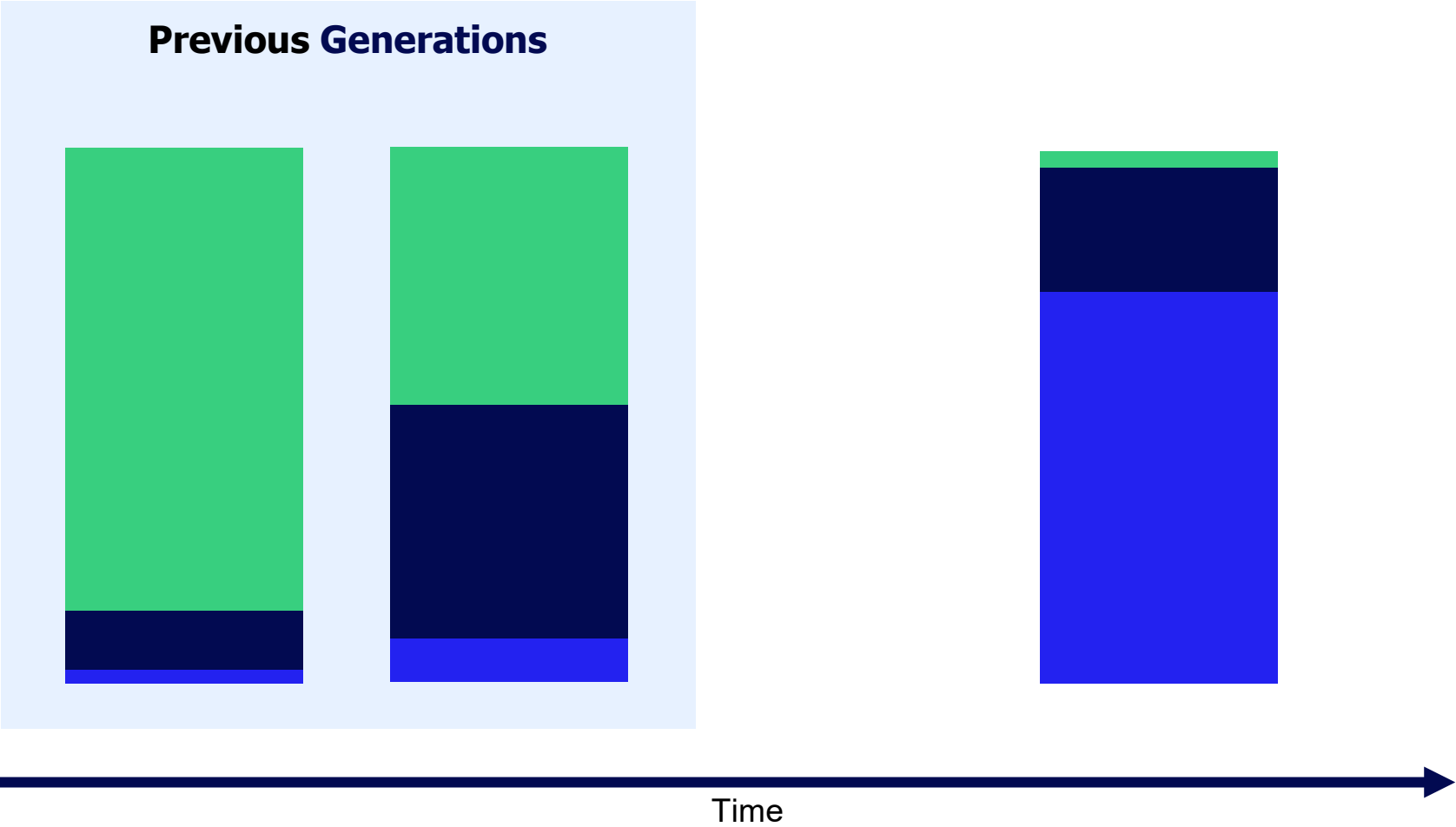
Evolution of solutions.



Evolution of solutions.

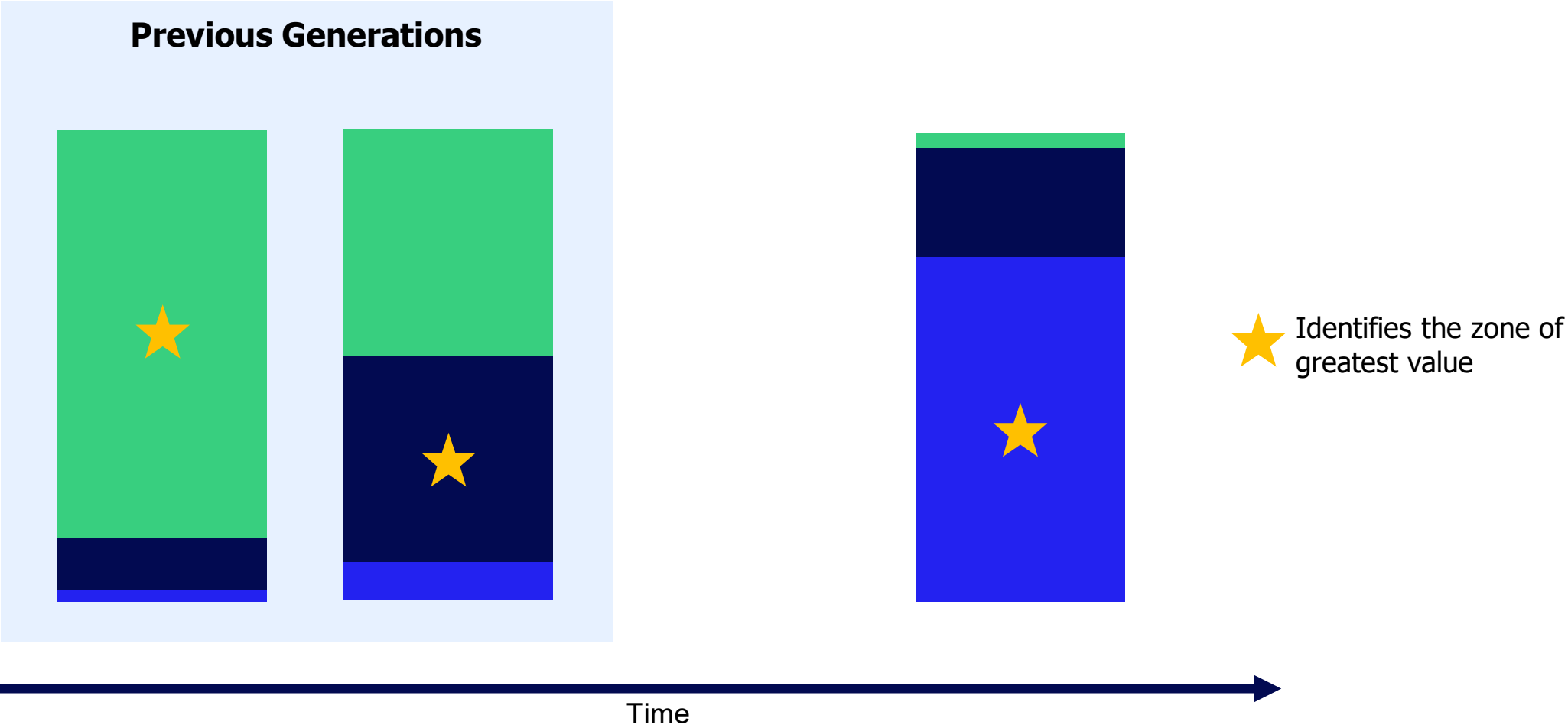
- General Characteristics over Time
- Early, Current, and Next Gen Solution Characteristics
 - Where are they seen most
 - Where is the “Value Zone”
 - Things to beware of when purchasing
- **What this is NOT:**
 - “Good” vs “Bad”
 - An endorsement of any one technology

Solution compositions over time.



■ LOB-Specific/Custom Code ■ Business Accelerators ■ Configurable Platform

Solution compositions over time.



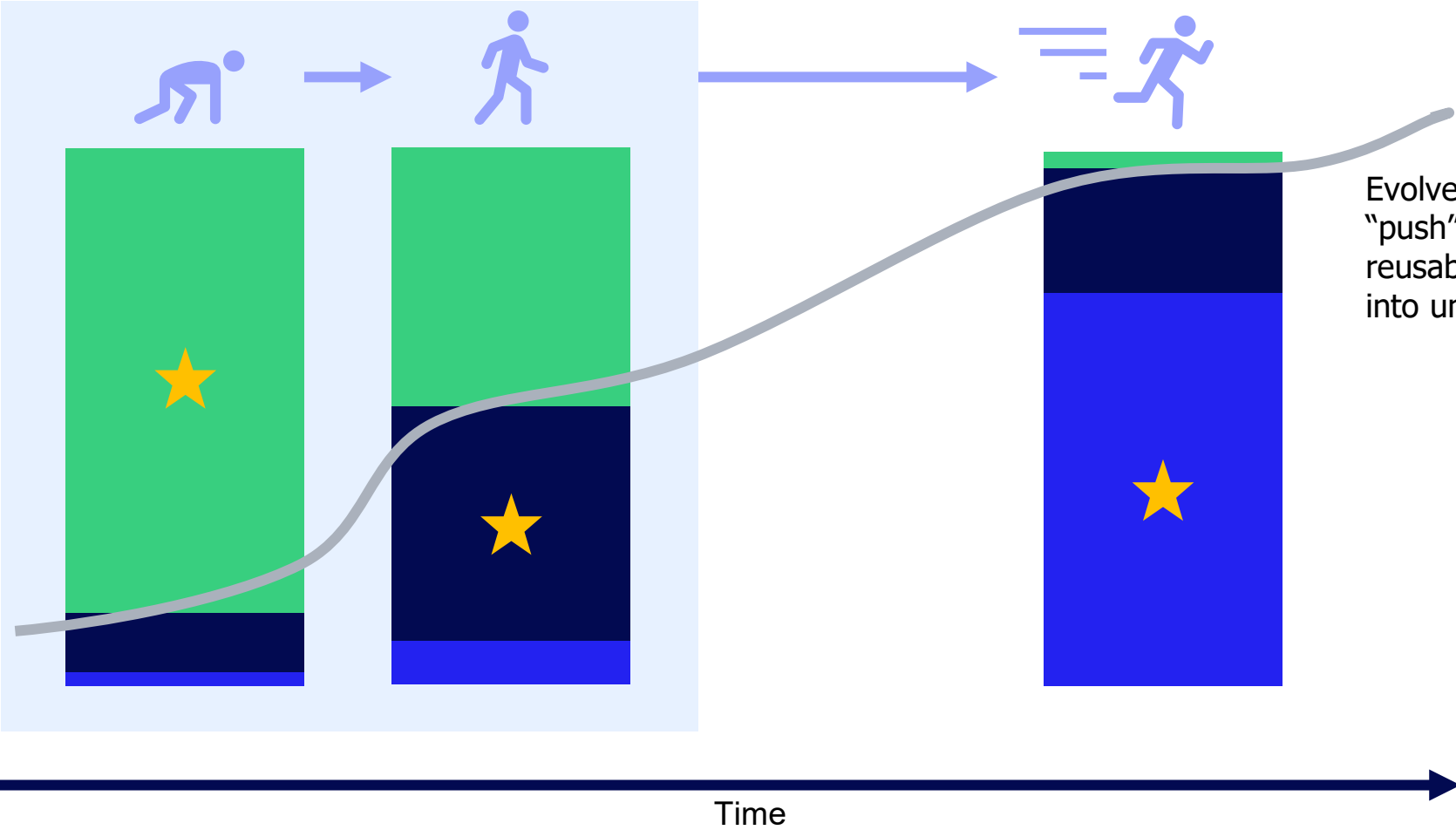
★ "Value Zone"

■ LOB-Specific/Custom Code

■ Business Accelerators

■ Configurable Platform

Solution compositions over time.



Evolved solutions tend over time to “push” features and functions into reusable business accelerators and into underlying technology

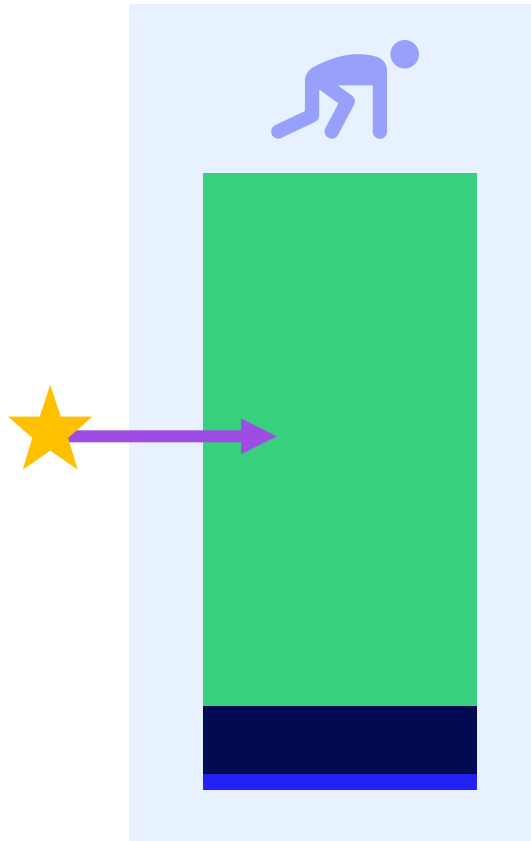
★ “Value Zone”

■ LOB-Specific/Custom Code

■ Business Accelerators

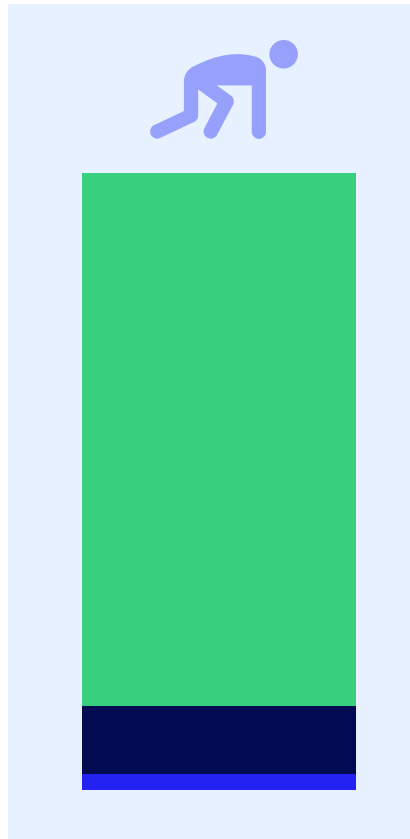
■ Configurable Platform

Earlier solutions.



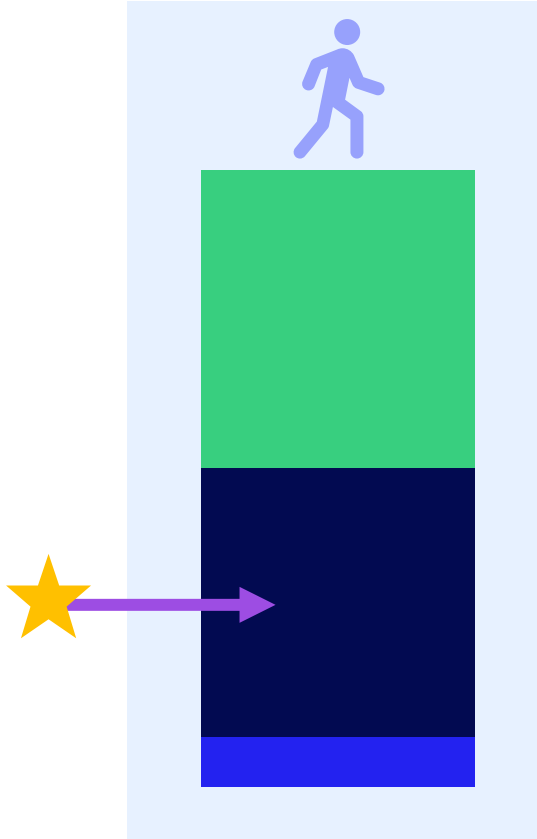
Characteristics		★ Value Zone
Custom Dev, Early COTS, Niche Use Cases		<ul style="list-style-type: none"> • Immediately identifiable • Highest customer/LOB specificity • Coded for each customer • Work for small, extremely stable business processes
Development Method	Traditional Code	
Configurability	Low/None	
Reusable Components	Low/None	
Automation (AI/RPA/PM)	None	
Data Sharing/Interfacing	Low/None	
Security Considerations	Non-standard; ad-hoc	
Change or Innovation	Very Costly; Risky	
Implementation Cycle	Long; Costly	

Earlier solutions.



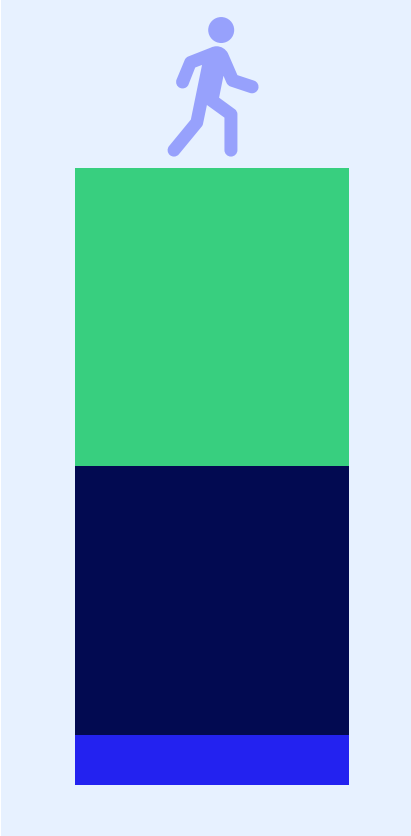
Characteristics		Beware of...
Custom Dev, Early COTS, Niche Use Cases		<ul style="list-style-type: none"> • Bespoke code/knowledge • Narrow business applicability • Manual code fixes • Rapid accrual of tech debt in the face of change
Development Method	Traditional Code	
Configurability	Low/None	
Reusable Components	Low/None	
Automation (AI/RPA/PM)	None	
Data Sharing/Interfacing	Low/None	
Security Considerations	Non-standard; ad-hoc	
Change or Innovation	Very Costly; Risky	
Implementation Cycle	Long; Costly	

Current generation.



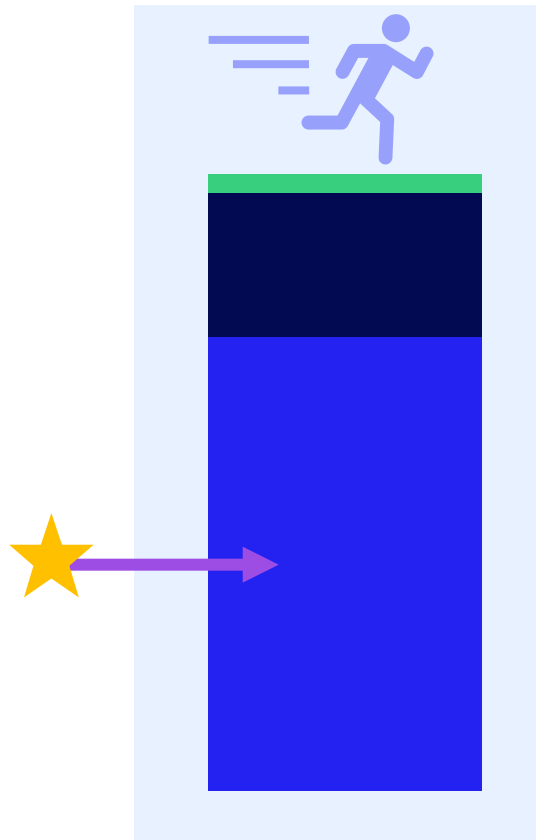
Characteristics		★ Value Zone
COTS/"MOTS", Mid-to-Large-Enterprise		<ul style="list-style-type: none"> • Baked-in business identity • Robust LOB framework • "Purpose-built", multiple customers • Evolved from variations <i>within</i> LOB/Industry
Development Method	Traditional Code/Scripting	
Configurability	Varies; Accelerator-level	
Reusable Components	LOB-specific; some general	
Automation (AI/RPA/PM)	None	
Data Sharing/Interfacing	Ad-Hoc	
Security Considerations	App-Specific	
Change or Innovation	Costly; Slow	
Implementation Cycle	Long; Costly	

Current generation.



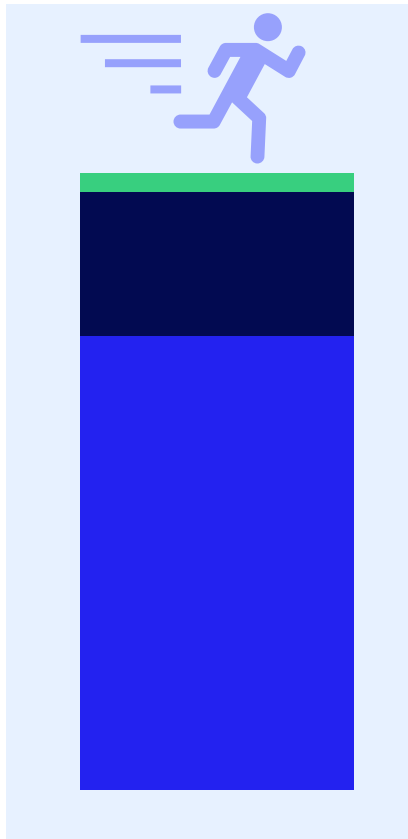
Characteristics		Beware of...
COTS/"MOTS", Mid-to-Large-Enterprise		<ul style="list-style-type: none"> Limited configurability More-than-meets-the-eye customization Surface level modernity
Development Method	Traditional Code	
Configurability	Varies; Accelerator-level	
Reusable Components	LOB-specific; some general	
Automation (AI/RPA/PM)	None	
Data Sharing/Interfacing	Ad-Hoc	
Security Considerations	App-Specific	
Change or Innovation	Costly; Slow	
Implementation Cycle	Long; Costly	

Next gen solutions.



Characteristics		★ Value Zone
Platform-Based Solutions		<ul style="list-style-type: none"> Automation: More than just human workers Configurability, Flexibility Evolved from variations <i>across</i> industries LOB needs arise from core components
Development Method	Low Code	
Configurability	Very High	
Reusable Components	Building blocks; accelerators	
Automation (AI/RPA/PM)	Native	
Data Sharing/Interfacing	Native	
Security Considerations	Native	
Change or Innovation	Cheap; Rapid	
Implementation Cycle	Short	

Next gen solutions.



Characteristics		Beware of...
Platform-Based Solutions		<ul style="list-style-type: none"> • Business identity not always immediate • Oversimplifying business needs • Pricing clarity
Development Method	Low Code	
Configurability	Very High	
Reusable Components	Building blocks; accelerators	
Automation (AI/RPA/PM)	Native	
Data Sharing/Interfacing	Native	
Security Considerations	Native	
Change or Innovation	Cheap; Rapid	
Implementation Cycle	Short	

What should I look for?

Find the Value Zone!

The greater the solution's footprint, the more critical this is to know

Challenge solutions against the future

Ensures that you aren't buying only for today

Don't be afraid of a "blank slate"

Look deeper than the EXACT business need

Ability to share data and integrate

Modern solutions don't always NEED to encapsulate ancillary functions

Do our humans NEED to do XYZ process like before?

You should have access to digital workers

Final thought.

Be Open to Innovation!

New technology, new solutions, new approaches

Solutions don't look the same as they once did

2022-2026 | State Strategic Plan for Information Resources Management

Goal 4: Proactive Approach to Emerging Technologies



As agencies face the next phase of modernizing legacy IT systems, they must plan for emerging technologies that are collaborative, scalable, and adaptive to a rapidly changing environment.

While a reliable telecommunications network infrastructure continues to be the foundation for data, video, and voice communications, long-term plans must consider advancements in technologies that fall outside of traditional telecommunication.

Solutions using artificial intelligence (AI), machine learning, robotic process automation (RPA), digital assistants, and low-code or no-code development can help incrementally phase out software, hardware, and services that are no longer supported.

Agencies should take proactive approaches now to increase readiness for the advanced technologies of tomorrow.

Why do we teach low code application development ?



1987



1990

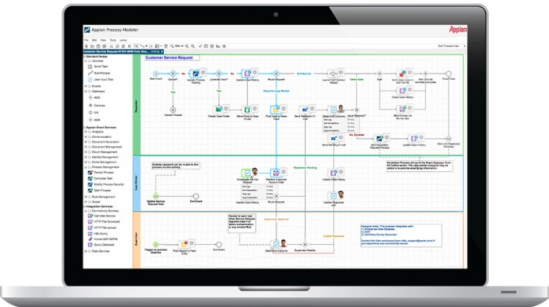
Why do we teach low code application development ?



We are in a crisis

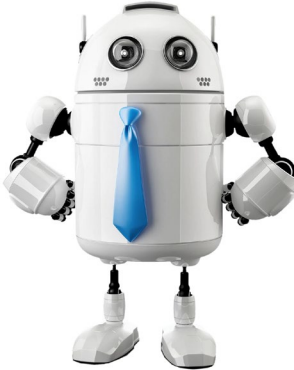
- 400 million jobs worldwide
- 13 million in the US
- Legacy Systems Failed

What do we teach at UT Dallas?



Application Development
6 weeks

+



Robotic Process Automation (RPA)
6 weeks

+



Test Automation
3 weeks

+



Three Industry Certifications



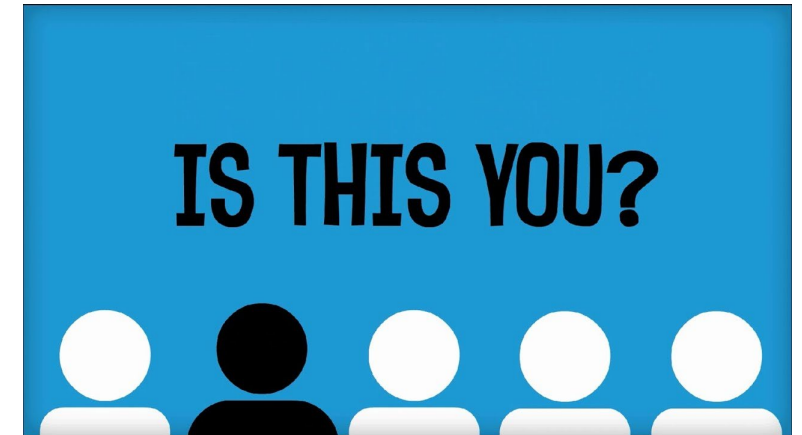
16 weeks long semester

Do I need low code?



More signs.

- Keeping up with demands from the business is difficult
- Reliance on legacy apps
- More time spent on maintenance than innovation
- Shadow IT
- Scarce development resources
- Inability to keep pace
- Evolving digital expectations



Think about the future.



The image shows a low-angle shot of a modern glass skyscraper against a clear blue sky. The Appian logo is mounted on the upper part of the building's facade. The logo consists of the word "app'ian" in a bold, sans-serif font, with the apostrophe being a small, curved mark. The letters are dark and have a slight 3D effect.

app'ian

A large, white, sans-serif version of the Appian logo is centered in the middle of the image. The font is clean and modern, with a slight shadow or depth to the letters. The apostrophe is a small, curved mark.

app'ian

A leader in low-code.