Modern SAP Development with SAP Build: A Paradigm Shift from Traditional Development Techniques

Kumail Saifuddin Saif

SAP Reporting Analyst, Pure Storage Inc.
kumail.saif@gmail.com

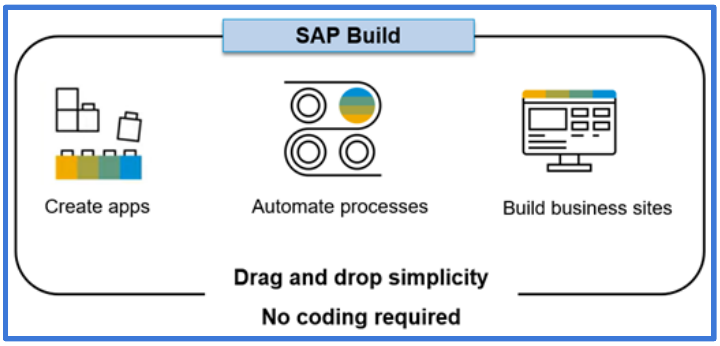
Abstract:
The landscape of SAP development has undergone a significant transformation with the introduction of SAP Build, a low-code/no-code (LCNC) tool designed to accelerate application development and democratize access to SAP's powerful business technology platform (BTP). This paper explores the evolution of SAP development, contrasting traditional development techniques with the modern, business-user-friendly approach enabled by SAP Build. We will examine the key features, benefits, and challenges of SAP Build and analyze how it redefines the roles of developers and business users. Finally, we will discuss the broader implications of this shift on enterprise IT strategy and future trends in SAP development.

Keywords: SAP S/4 HANA, SAP Business Technology Platform (BTP), SAP Build, No-Code/Low-Code (NCLC).

1. Introduction:

SAP has long been a leader in enterprise resource planning (ERP) and business process automation. Historically, SAP application development was a complex, code-intensive process requiring specialized skills in ABAP (Advanced Business Application Programming), Java, and other SAP-specific technologies. However, with digital transformation accelerating the demand for agility, efficiency, and innovation, businesses require faster, more accessible ways to build and deploy SAP-based applications.

SAP Build, introduced as part of the SAP Business Technology Platform (BTP), represents a paradigm shift in SAP development by enabling citizen developers (business users with little to no coding experience) to participate in application development. This shift moves away from traditional development models toward a more inclusive, agile, and user-centric approach.



2. Traditional SAP Development Techniques:

2.1 Characteristics of Traditional SAP Development

Traditional SAP development has been primarily developer-driven, requiring expertise in:

- ABAP: The primary programming language for SAP ERP applications.
- SAP Fiori/UI5: Used for developing modern web-based SAP applications.
- SAP NetWeaver and Java: Used in older SAP environments for middleware and application development.
- Integration with SAP HANA and S/4HANA: Required deep knowledge of SQL, data models, and backend optimizations.
- The traditional approach follows a waterfall or hybrid development model, emphasizing structured programming, extensive testing cycles, and IT-led governance.

2.2 Challenges of Traditional SAP Development

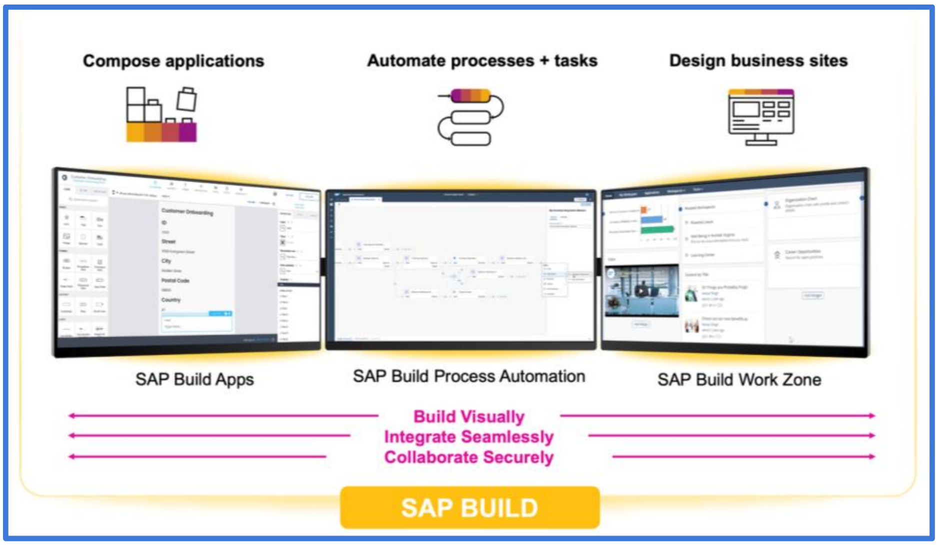
- Long Development Cycles: Custom SAP applications typically take months to develop due to rigorous requirements gathering, coding, testing, and deployment processes.
- Dependency on Specialized Developers: Only skilled ABAP or SAP UI5 developers can create and modify SAP applications, limiting agility.
- High Costs and Complexity: Custom development in SAP requires extensive coding, integration testing, and maintenance, increasing total cost of ownership (TCO).
- Limited Business User Involvement: Business users often struggle to translate their needs into technical specifications, leading to a gap between business requirements and technical implementation.

3. The Emergence of SAP Build: A Game Changer in SAP Development:

SAP Build is a low-code/no-code (LCNC) development platform that empowers business users and developers alike to rapidly create, modify, and deploy SAP applications with minimal coding effort. It is designed to streamline application development while ensuring compliance with SAP best practices.

3.1 Key Components of SAP Build

- **SAP Build Apps:** A visual, drag-and-drop application builder for creating enterprise apps without coding. Mobile and web apps can be built using 500+ templates and pre-built business logic components
- **SAP Build Process Automation:** Automates workflows and business processes using an intuitive, LCNC interface. It contains 340+ line-of-business and industry-specific process content packages, RPA bots and connectors
- **SAP Build Work Zone:** Provides a unified, customizable workspace to integrate apps, data, and collaboration tools. It speeds up the development process with pre- built line-of-business-specific content packages (for example, for SAP Success Factors)

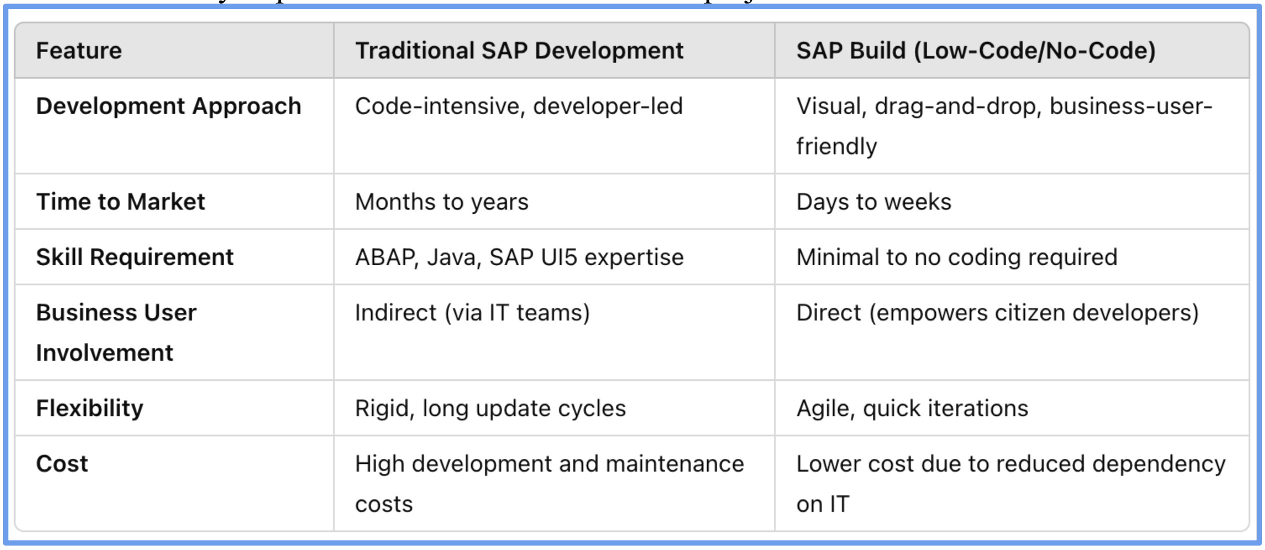


3.2 How SAP Build Transforms SAP Development

Business users and experienced developers can collaborate on projects using the development solutions that best fit their respective skills – these are called fusion teams.

Business users can use their domain expertise to create solutions themselves without writing any code. Content and artifacts can be published and shared amongst team members and across projects in an easy and secure way. As citizen developers, they can tackle the long tail of processes and activities and accelerate innovation

Experienced developers can leverage code-first solutions to develop complex logic and make it available for business users to easily import and utilize in their low-code projects.



Feature	Traditional SAP Development	SAP Build (Low-Code/No-Code)
Development Approach	Code-intensive, developer-led	Visual, drag-and-drop, business-user-friendly
Time to Market	Months to years	Days to weeks
Skill Requirement	ABAP, Java, SAP UI5 expertise	Minimal to no coding required
Business User Involvement	Indirect (via IT teams)	Direct (empowers citizen developers)
Flexibility	Rigid, long update cycles	Agile, quick iterations
Cost	High development and maintenance costs	Lower cost due to reduced dependency on IT

4. Benefits of SAP Build in Modern SAP Development:

- **Faster Development and Deployment:** SAP Build eliminates the need for extensive coding, allowing applications to be developed and deployed significantly faster. Business users can prototype and refine applications quickly, reducing development cycles.
- **Empowering Citizen Developers:** With an intuitive, no-code interface, SAP Build democratizes SAP development, enabling non-technical users to create applications without requiring IT intervention.
- **Improved Business-IT Collaboration**: By involving business users directly in application development, SAP Build reduces communication gaps between IT and business teams, ensuring applications align better with real-world business needs.
- **Cost Reduction and Resource Optimization**: Since SAP Build minimizes the need for specialized development skills, enterprises can reduce dependency on high-cost SAP developers while making better use of existing workforce capabilities.
- **Enhanced Agility and Innovation**: Organizations can quickly iterate and deploy applications, responding dynamically to changing business requirements. This enhances business agility and fosters innovation.

5. Challenges and Considerations in Adopting SAP Build:

While SAP Build offers significant advantages, organizations must also consider potential challenges:

- **5.1 Governance and Security**
  - Ensuring that citizen-developed applications adhere to enterprise security, compliance, and data governance policies.
  - Implementing role-based access controls to prevent unauthorized modifications.

- **5.2 Integration with Existing SAP and Non-SAP Systems**
  - Seamless integration with legacy SAP systems and third-party applications remains a challenge, requiring some technical expertise.
  - Need for pre-built connectors and APIs to facilitate integration.
- **5.3 Scalability and Performance Considerations**
  - LCNC tools may not be suitable for highly complex, large-scale applications requiring custom logic and deep system integration.
  - Performance optimizations might be limited compared to traditional coding approaches.
- **5.4 Change Management and Adoption**
  - Organizations need to train business users and ensure cultural alignment with the new development paradigm.
  - IT teams may resist LCNC adoption due to concerns about reduced control over application development.

6. The Future of SAP Development: What's Next?

SAP Build is a critical enabler of SAP's broader vision for intelligent enterprise automation. The future of SAP development is likely to be characterized by:

- AI-Driven Development: SAP will integrate AI-powered tools to further automate application building, making development even more intuitive.
- Hyper Automation: Combining SAP Build with RPA (Robotic Process Automation) and AI to enable end-to-end business process automation.
- Composable Enterprise Applications: Businesses will increasingly adopt a modular, drag-and-drop approach to SAP applications, assembling solutions dynamically.
- Deeper Integration with SAP BTP: SAP Build will continue to evolve, integrating more tightly with SAP's cloud ecosystem for enhanced analytics, automation, and predictive capabilities.

7. Conclusion:

SAP Build represents a transformational shift in SAP development, enabling faster, more agile, and business-user-driven application creation. Unlike traditional SAP development, which relies on specialized programming skills, SAP Build democratizes the process, making it accessible to non-technical users while still ensuring enterprise-grade security and integration. By embracing SAP Build, organizations can accelerate digital transformation, reduce costs, and improve collaboration between business and IT teams. However, successful adoption requires thoughtful governance, integration strategies, and change management efforts. As SAP continues to enhance its LCNC offerings, the future of SAP development will likely become even more intuitive, efficient, and innovation-driven.

REFERENCES:

1. SAP Build Help Portal [Online]. Available at: https://help.sap.com/docs/SAP_BUILD?locale=en-US
2. SAP Build - No-Code Development, Centralized Access, and Process Automation Book:https://www.sap-press.com/sap-build_5772/?utm_source=sappressblog&utm_medium=referral&utm_campaign=Blogs&utm_term=2481_chapter1&utm_content=2481
3. SAP Business Technology Platform (SAP BTP) [Online]. Available at: https://help.sap.com/docs/btp?locale=en-US
4. Get Free Access to SAP Build Process Automation [Online]. Available at: https://help.sap.com/docs/build-process-automation/sap-build-process-automation/get-free-access-to-sap-build-process-automation?locale=en-US
5. Install and Set Up Desktop Agent 3 [Online]. Available at: https://developers.sap.com/tutorials/spa-setup-desktop-3-0-agent.html

6. Create a Business Site Using SAP Build Work Zonehttps://help.sap.com/docs/build-process-automation/sap-build-process-automation/configure-sap-build-work-zone-for-sap-build-process-automation?locale=en-US
7. Subscribe to SAP Build Apps on SAP BTP [Online]. Available at: https://developers.sap.com/tutorials/appgyver-subscribe-service.html
8. Why SAP Build Code is a Game-Changer for SAP Developers [Online]. Available at: https://community.sap.com/t5/technology-blogs-by-sap/why-sap-build-code-is-a-game-changer-for-sap-developers/ba-p/13574173#:~:text=Supercharged%20development%20with%20generative%20AI,%2Dend%2C%20and%20mobile%20applications.
9. Developing Apps with SAP Build Apps using Drag-and-Drop Simplicity [Online]. Available at: https://learning.sap.com/learning-journeys/develop-apps-with-sap-build-apps-using-drag-and-drop-simplicity
10. Utilizing SAP Build for Low-Code/No-Code Applications and Automations for Citizen Developers [Online]. Available at: https://learning.sap.com/learning-journeys/utilize-sap-build-for-low-code-no-code-applications-and-automations-for-citizen-developers
11. Discovering SAP Business Technology Platform [Online]. Available at: https://learning.sap.com/learning-journeys/discover-sap-business-technology-platform