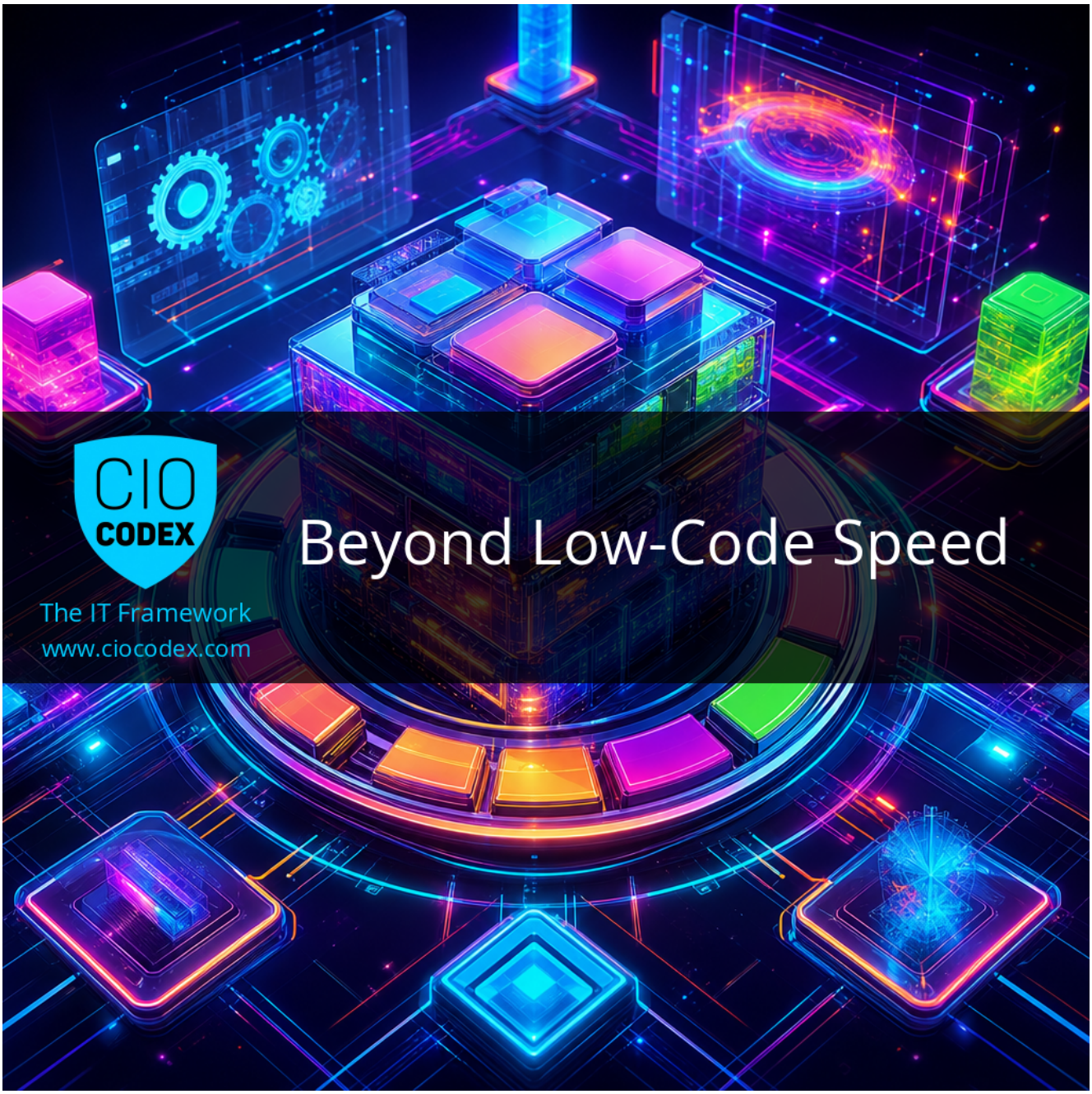


O avanço das plataformas low-code e no-code sempre me pareceu um movimento relevante para a evolução da Tecnologia da Informação. Não apenas porque essas tecnologias permitem criar aplicações mais rapidamente, mas porque modificam a relação entre as áreas de negócio e as equipes tradicionalmente responsáveis por desenvolver soluções. Ao reduzir parte das barreiras técnicas, torna-se possível aproximar a construção do software das pessoas que conhecem em profundidade os processos que serão digitalizados.

Essa perspectiva continua fazendo sentido, mas considero importante reconhecer que a discussão evoluiu significativamente desde os primeiros movimentos de popularização dessas plataformas. Aquilo que, há alguns anos, parecia concentrado em

formulários, workflows e pequenas automações passou a integrar soluções corporativas mais sofisticadas, serviços Cloud, interfaces conversacionais, integrações com APIs e, mais recentemente, agentes de Artificial Intelligence capazes de executar atividades sobre sistemas empresariais.

A disponibilidade de ferramentas melhores, entretanto, não elimina os desafios fundamentais da arquitetura corporativa. Uma aplicação pode ser criada sem escrever código tradicional e, ainda assim, depender de dados sensíveis, executar transações financeiras, integrar sistemas críticos ou afetar diretamente a experiência dos clientes. A facilidade de desenvolvimento não reduz automaticamente a responsabilidade sobre aquilo que foi construído.

A questão não é escolher entre permitir a inovação ou preservar os controles empresariais. A verdadeira oportunidade está em construir um modelo no qual a inovação possa acontecer com autonomia proporcional ao risco, sem transformar cada nova aplicação em um ativo desconhecido, sem responsáveis definidos ou sem condições adequadas de sustentação.

Ao revisitar uma matéria da InfoWorld sobre os riscos do low-code e do no-code em arquiteturas Cloud, percebo que as preocupações originais continuam pertinentes. Entretanto, o avanço da Generative AI, dos Citizen Developers e do desenvolvimento orientado por agentes tornou a discussão ainda mais abrangente. O desafio já não está apenas em controlar aplicações feitas com componentes visuais. Está em governar um ecossistema no qual praticamente qualquer área pode começar a produzir software e automações de maneira independente.

O alerta original da InfoWorld sobre low-code e no-code

O ponto de partida desta reflexão é o artigo de David Linthicum, publicado pela InfoWorld em 2023, sobre os riscos do desenvolvimento low-code e no-code em arquiteturas Cloud:

https://www.infoworld.com/article/3706892/the-risks-of-low-code-and-no-code-development-in-cloud-architecture.html

A matéria analisa quatro tensões particularmente relevantes: flexibilidade versus personalização, velocidade versus escalabilidade, facilidade de criação versus segurança e controle, além dos desafios de integração com sistemas existentes. A discussão parte do reconhecimento de que componentes pré-configurados e interfaces visuais reduzem esforços de implementação, mas podem introduzir limitações quando

aplicações precisam atender requisitos empresariais mais complexos.

A preocupação central não consiste em afirmar que o low-code é inadequado, mas em questionar a adoção indiscriminada de tecnologias cuja simplicidade inicial pode esconder dependências arquiteturais e operacionais. Uma solução criada rapidamente pode funcionar muito bem em seu primeiro caso de uso e encontrar dificuldades quando precisa atender mais usuários, implementar regras específicas ou integrar plataformas que não foram consideradas durante sua concepção.

É importante, entretanto, contextualizar esse diagnóstico. Desde 2023, o ecossistema amadureceu e várias plataformas ampliaram recursos de segurança, governança, integração, administração de ambientes e Application Lifecycle Management. Portanto, não seria correto generalizar que aplicações low-code são necessariamente inseguras ou incapazes de escalar. Suas capacidades variam conforme o fornecedor, a arquitetura, a modalidade de contratação e o modelo de utilização adotado.

Ao mesmo tempo, a facilidade de criação aumentou significativamente. Em 2026, a discussão já incorpora experiências nas quais usuários descrevem aplicações em linguagem natural, ferramentas geram componentes ou código e agentes podem executar workflows envolvendo múltiplos serviços. Essa evolução reduz determinadas barreiras, mas não resolve automaticamente questões relacionadas ao responsável pela aplicação, ao acesso aos dados, aos testes realizados ou à operação após a implantação.

O Gartner publicou, em setembro de 2026, pesquisas específicas sobre governança de Citizen Development em escala e sobre responsabilidades ao longo de todo o ciclo de vida de aplicações e agentes criados com apoio de AI. Esse movimento reforça que o mercado passou da pergunta sobre a viabilidade dessas tecnologias para uma discussão mais exigente: como permitir sua adoção em escala sem perder controle sobre o ambiente empresarial.

Minha leitura é que o alerta original permanece válido, mas precisa ser atualizado. A principal questão já não é apenas identificar os riscos das plataformas. É desenvolver mecanismos organizacionais e tecnológicos capazes de administrar os resultados de sua utilização, independentemente de quem os criou ou da quantidade de código escrita manualmente.

Onde a adoção realmente faz sentido

Acredito que existe um conjunto muito amplo de oportunidades para low-code e no-code, mas o sucesso depende de escolher adequadamente os casos de uso. A disponibilidade de uma ferramenta não deveria se transformar em justificativa

automática para utilizá-la em todos os problemas.

Uma estratégia madura começa pela identificação de necessidades nas quais a velocidade, a proximidade com o processo empresarial e a disponibilidade de componentes reutilizáveis produzem benefícios suficientemente relevantes. Depois, deve avaliar se a aplicação permanecerá sustentável diante de mudanças, crescimento e exigências de operação.

Produtividade e automação departamental

Formulários, aprovações, painéis operacionais, pequenas aplicações internas, coleta de informações e automação de tarefas repetitivas representam cenários comuns. Nessas situações, o benefício pode surgir da substituição de planilhas dispersas, controles por e-mail e processos manuais por mecanismos mais estruturados.

O ganho não deveria ser medido apenas pela velocidade de construção. Também importa avaliar redução de erros, melhoria da rastreabilidade, menor retrabalho e tempo economizado pelas equipes de negócio. Uma solução simples, criada em poucos dias e devidamente governada, pode gerar mais valor do que um projeto tecnicamente sofisticado destinado a resolver um problema de pouca relevância.

Extensão de plataformas corporativas

Outra oportunidade está na extensão de capacidades já disponíveis em sistemas empresariais. Muitas organizações utilizam ERP, CRM, HCM, ITSM e plataformas de colaboração que oferecem recursos de automação, configuração de interfaces e construção de aplicações complementares.

Nesses casos, low-code pode permitir adaptar processos sem necessariamente modificar o núcleo dos sistemas. Mas existe uma preocupação relevante: extensões precisam respeitar interfaces suportadas, modelos de dados e estratégias de atualização. Uma personalização aparentemente conveniente pode criar dependências que dificultam a evolução futura da plataforma principal.

O ideal é que aplicações complementares utilizem mecanismos de integração suportados, preservando separação clara de responsabilidades e evitando alterações desnecessárias em sistemas de missão crítica.

Casos em que outra abordagem pode ser

superior

Nem todo problema deve ser resolvido com low-code. Workloads que exigem desempenho extremo, algoritmos altamente específicos, grande liberdade arquitetural ou integração com componentes pouco suportados podem justificar desenvolvimento tradicional.

Também existem situações nas quais a aplicação já foi resolvida satisfatoriamente por um produto de mercado. Construir uma alternativa low-code apenas porque a ferramenta está disponível pode reproduzir funcionalidades existentes e aumentar a complexidade do portfolio.

A decisão adequada exige comparar alternativas considerando benefícios, limitações técnicas, custos de evolução, dependência de fornecedores e requisitos de segurança. A velocidade inicial deve ser considerada, mas não pode funcionar como critério exclusivo de escolha.

O principal risco continua sendo a arquitetura invisível

Uma das características mais atraentes das plataformas low-code está na capacidade de esconder detalhes técnicos através de componentes visuais. Entretanto, a simplicidade percebida pelo usuário pode dificultar a compreensão da arquitetura real que sustenta a aplicação.

Isso se torna relevante quando o produto precisa evoluir. Dependências entre componentes, limites de execução, comportamento de conectores, políticas de atualização e restrições de licenciamento podem não ser evidentes durante a construção inicial. Uma solução aparentemente simples pode acumular complexidade operacional relevante.

Personalização e dependência do fornecedor

A maioria das plataformas oferece mecanismos próprios de desenvolvimento, configuração e extensão. Isso permite acelerar entregas, mas pode criar dependência significativa de componentes proprietários, formatos específicos e recursos cujo comportamento é determinado pelo fornecedor.

Vendor Lock-in não é necessariamente motivo para rejeitar uma tecnologia. Muitos serviços gerenciados oferecem benefícios suficientes para justificar dependências. O problema surge quando essa escolha não é compreendida ou quando as dificuldades

de migração são descobertas apenas diante de uma mudança contratual, comercial ou tecnológica.

Uma avaliação estruturada deveria considerar possibilidades de exportação, integração, acesso aos dados, continuidade das aplicações e mecanismos de atualização. Quanto mais crítica for a solução, maior deverá ser a clareza sobre o que acontece se a plataforma deixar de atender aos requisitos empresariais.

Escalabilidade e limites reais

Uma interface visual pode permitir desenvolver rapidamente uma aplicação, mas sua escalabilidade dependerá da arquitetura subjacente. Restrições sobre chamadas de API, concorrência, processamento, volumes de dados, tempos de execução e serviços externos podem influenciar o comportamento em produção.

É importante distinguir capacidades disponibilizadas pela plataforma de características efetivamente demonstradas pela aplicação. Um fornecedor pode oferecer recursos escaláveis e, ainda assim, determinado desenho funcional produzir gargalos ou custos excessivos.

Testes de carga, análise de consumo e validação das integrações continuam necessários. A promessa de velocidade no desenvolvimento não substitui requisitos não funcionais, principalmente quando a solução passa a participar de processos essenciais.

Integração com sistemas existentes

A integração provavelmente continuará sendo um dos pontos mais desafiadores. O ambiente empresarial raramente é composto apenas por aplicações modernas, bem documentadas e acessíveis através de APIs padronizadas.

Sistemas legados, regras de negócio distribuídas, estruturas de dados inconsistentes e processos transversais introduzem complexidade que nenhuma plataforma visual consegue eliminar automaticamente. Conectores reduzem esforço, mas também possuem limites, políticas de autenticação, versões e dependências de terceiros.

Por isso, considero fundamental preservar uma arquitetura de integração corporativa. O low-code precisa utilizar interfaces governadas e mecanismos seguros, evitando que cada área crie conexões diretas e independentes com sistemas críticos. A soma de integrações locais aparentemente simples pode produzir um ambiente de manutenção extremamente difícil.

Citizen Development e Fusion Teams modificam o Operating Model

A democratização do desenvolvimento possui uma consequência organizacional importante: pessoas que anteriormente participavam apenas da definição dos requisitos passam a contribuir diretamente para a construção das soluções.

Esse movimento cria oportunidades de reduzir o intervalo entre identificar uma necessidade e disponibilizar uma resposta tecnológica. Entretanto, também exige redefinir responsabilidades e desenvolver competências que não estão necessariamente presentes em todas as áreas de negócio.

Citizen Developer não é sinônimo de ausência de competência

O conceito de Citizen Developer descreve profissionais que constroem aplicações ou automações sem exercer formalmente uma função tradicional de desenvolvimento de software. Isso não significa que essas pessoas possam trabalhar sem compreender riscos, dados, regras empresariais e responsabilidades associadas às soluções.

Quanto maior a autonomia, maior deverá ser a clareza sobre limites de atuação. Um usuário pode estar plenamente habilitado a automatizar tarefas pessoais e ainda não possuir competências necessárias para desenvolver uma aplicação que execute alterações em um sistema financeiro ou manipule informações sensíveis.

A capacitação precisa combinar utilização da ferramenta com fundamentos de segurança, qualidade, tratamento de dados, documentação e gestão do ciclo de vida. Sem essa formação, a facilidade de criação pode produzir um conjunto crescente de soluções frágeis ou dependentes de seus autores originais.

Fusion Teams aproximam tecnologia e negócio

Fusion Teams representam uma alternativa para conectar profissionais de negócio, desenvolvimento, arquitetura, segurança e outras competências em torno de objetivos comuns. A proposta não consiste em eliminar funções especializadas, mas em reduzir a distância entre conhecimento do processo e capacidade de implementação.

Em uma equipe multidisciplinar, o negócio contribui para definir prioridades e validar resultados, enquanto profissionais técnicos ajudam a estruturar integrações, controles e requisitos não funcionais. A composição deve variar conforme criticidade,

complexidade e maturidade do caso de uso.

Essa abordagem pode aumentar velocidade de aprendizado e reduzir retrabalho causado por interpretações inadequadas. Entretanto, exige responsabilidades bem definidas e incentivos coerentes. Apenas reunir pessoas de diferentes áreas não cria automaticamente um time integrado.

O papel de TI deixa de ser exclusivamente executar demandas

A expansão do Citizen Development pode modificar profundamente a atuação da TI. Em vez de concentrar todas as atividades de desenvolvimento, a função tecnológica passa a oferecer plataformas, padrões, serviços compartilhados, capacitação, arquitetura e mecanismos de governança.

Isso não significa abdicar da responsabilidade sobre o ambiente empresarial. A TI continua responsável por assegurar que aplicações críticas, identidades, dados e integrações sejam administrados adequadamente. O que muda é a forma como essa responsabilidade pode ser exercida, com maior ênfase em habilitação e controles incorporados às próprias plataformas.

A maturidade do modelo dependerá da capacidade de oferecer autonomia sem criar descontrole. Uma organização excessivamente restritiva pode estimular a utilização de soluções não autorizadas, enquanto a ausência de padrões pode produzir fragmentação e riscos crescentes.

De Shadow IT a um catálogo governado de soluções

O conceito de Shadow IT não nasceu com low-code, mas essas plataformas ampliaram a facilidade de criar aplicações fora dos processos tradicionais de Tecnologia. Muitas dessas iniciativas surgem de necessidades legítimas que não receberam atendimento dentro dos prazos esperados.

É importante reconhecer essa causa. Quando a TI não consegue oferecer alternativas proporcionais às necessidades dos usuários, surgem planilhas, bancos de dados locais, serviços externos e automações improvisadas. O low-code pode tanto reduzir esse problema, ao disponibilizar ferramentas corporativas adequadas, quanto ampliá-lo, quando é utilizado sem visibilidade e governança.

Descoberta e inventário são o ponto de partida

Uma organização não consegue administrar aquilo que desconhece. O primeiro passo consiste em identificar aplicações, automações, agentes e integrações existentes, independentemente de terem sido desenvolvidos por equipes tradicionais ou Citizen Developers.

O inventário deve contemplar finalidade, responsáveis, usuários, sistemas integrados, dados utilizados, licenciamento, criticidade e estágio do ciclo de vida. Também precisa identificar soluções abandonadas, duplicadas ou que continuem executando operações sem acompanhamento adequado.

Esse trabalho não deveria ser realizado apenas através de campanhas pontuais. Ferramentas de administração das plataformas podem permitir descoberta contínua e acompanhamento de mudanças. A evolução das capacidades de governança incorporadas aos produtos oferece uma oportunidade importante para industrializar esse processo.

Classificação de risco e autonomia proporcional

Nem todas as aplicações exigem o mesmo nível de controle. Um formulário de produtividade pessoal, sem acesso a informações sensíveis e com alcance limitado, representa risco muito diferente de um workflow que altera dados de clientes ou executa pagamentos.

Uma abordagem madura pode estabelecer categorias de utilização, definindo quem pode desenvolver, quais conectores são permitidos, quais validações são necessárias e quais ambientes poderão receber a solução. Quanto maior o risco, maior deverá ser a participação das competências especializadas.

Essa lógica também se tornou relevante para AI Agents. Em 2026, o Gartner reforçou que aplicar indiscriminadamente os mesmos controles a agentes com diferentes níveis de autonomia pode produzir tanto excesso de restrição quanto exposição desnecessária. O princípio de governança proporcional ajuda a equilibrar essas situações.

A gestão precisa continuar depois da implantação

Uma aplicação pode ser criada em poucas horas e permanecer em uso durante anos. Nesse período, seus responsáveis podem mudar de função, a plataforma pode sofrer

atualizações, requisitos podem evoluir e integrações podem se tornar obsoletas.

O Gartner destacou, em setembro de 2026, a importância da governança de todo o ciclo de vida para evitar aplicações e agentes sem responsáveis definidos. Essa preocupação é particularmente pertinente porque a facilidade de criação pode ser acompanhada por baixa disciplina de manutenção.

O modelo deve prever ownership, revisão periódica, suporte, recuperação, evolução e desativação. Uma aplicação sem responsável claro deveria ser tratada como risco organizacional, independentemente da ferramenta utilizada em seu desenvolvimento.

Segurança, privacidade e compliance não são opcionais

A facilidade de construir soluções cria a impressão de que o esforço de segurança também será reduzido. Entretanto, uma aplicação low-code continua sujeita a riscos relacionados a identidades, permissões, dados, integrações e comportamento funcional.

A plataforma pode oferecer controles robustos e ainda assim permitir configurações inadequadas. Da mesma forma, a existência de certificações e recursos de segurança do fornecedor não substitui a responsabilidade da empresa por administrar acessos, dados e processos.

Identidade, acesso e segregação de funções

O acesso às aplicações precisa considerar identidade, necessidade de utilização, privilégios e criticidade das transações. Ambientes corporativos devem utilizar mecanismos adequados de autenticação e autorização, integrando-se sempre que pertinente às políticas empresariais de Identity and Access Management.

Um risco recorrente aparece quando integrações utilizam credenciais compartilhadas ou permissões amplas demais. A aplicação pode parecer restrita a determinado grupo, mas executar operações sobre sistemas externos utilizando uma identidade com poderes muito superiores aos necessários.

Também é necessário considerar segregação de funções. Um usuário que cria ou modifica um workflow não deveria necessariamente possuir autorização para aprovar suas próprias operações em processos sujeitos a controles específicos. A facilidade de configuração precisa conviver com responsabilidades claramente estabelecidas.

Dados pessoais e informações sensíveis

No Brasil, a LGPD estabelece princípios como finalidade, adequação e necessidade no tratamento de dados pessoais. Aplicações desenvolvidas em plataformas low-code precisam observar esses princípios, além das demais obrigações aplicáveis à organização e ao setor.

A avaliação deve considerar quais informações serão coletadas, por quais motivos, onde serão armazenadas, quem poderá acessá-las e quais integrações participarão do processamento. Também é importante compreender mecanismos de retenção, exclusão, logs e proteção das informações.

A transferência de dados para serviços externos ou recursos de AI exige cuidado adicional. A disponibilidade de um conector não significa que sua utilização esteja automaticamente autorizada. As regras corporativas devem delimitar quais dados poderão ser processados por quais serviços e em quais condições.

Segurança incorporada à plataforma

Recursos de Data Loss Prevention, ambientes administrados, políticas sobre conectores e controles de compartilhamento representam exemplos de mecanismos disponíveis em determinados ecossistemas empresariais. A Microsoft, por exemplo, documenta capacidades de Managed Environments e controles de governança em Power Platform.

Esses mecanismos podem reduzir o esforço de aplicação de políticas e favorecer padronização, mas precisam ser configurados corretamente. Também devem ser acompanhados por procedimentos de auditoria, análise de riscos, avaliação de vulnerabilidades e resposta a incidentes.

O ideal é que os controles sejam incorporados à experiência de desenvolvimento sempre que possível. A segurança deve orientar comportamentos adequados por padrão, evitando depender exclusivamente de verificações manuais realizadas depois que a aplicação já se tornou crítica.

O próximo desafio é a convergência com AI e desenvolvimento agentic

A relação entre low-code, no-code e Artificial Intelligence evoluiu significativamente. Em 2023, boa parte das discussões estava relacionada à utilização de AI para facilitar a criação de interfaces, fórmulas e pequenas automações.

Em 2026, o cenário passou a incluir geração de aplicações a partir de instruções em linguagem natural, recursos de AI incorporados às plataformas e agentes capazes de executar atividades sobre sistemas empresariais. Essa transformação amplia consideravelmente o conjunto de pessoas habilitadas a criar soluções.

Generative AI reduz a distância entre intenção e implementação

A possibilidade de descrever uma necessidade e obter uma primeira versão funcional de uma aplicação modifica o processo tradicional de desenvolvimento. Em vez de começar necessariamente pela configuração manual de todos os componentes, o usuário pode partir de uma intenção expressa em linguagem natural e refinar o resultado.

Isso acelera experimentações e facilita a validação de conceitos. Entretanto, uma descrição aparentemente clara pode esconder ambiguidades relevantes sobre regras de negócio, exceções, permissões e integração. A capacidade de produzir uma interface rapidamente não garante que seus comportamentos estejam corretos.

Por isso, a revisão dos requisitos e a validação dos resultados permanecem essenciais. A AI pode reduzir o esforço de implementação, mas não substitui a responsabilidade de compreender o problema nem de assegurar que a aplicação atende aos critérios empresariais.

Vibe Coding e novas formas de Shadow IT

O termo Vibe Coding passou a ser utilizado para descrever abordagens nas quais aplicações são construídas principalmente através de interações com ferramentas de AI, com menor participação direta na escrita de código. Essas possibilidades ampliam a democratização do desenvolvimento para além das plataformas low-code tradicionais.

Em junho de 2026, o Gartner publicou recomendações específicas sobre como governar essa modalidade de criação entre Citizen Developers, considerando riscos de segurança, compliance e utilização de ferramentas não autorizadas.

A preocupação é semelhante àquela já conhecida no universo de low-code: a facilidade de construir algo funcional pode antecipar a implantação em relação à capacidade de compreender sua arquitetura e administrá-la. Isso reforça a necessidade de tratar soluções criadas com AI dentro das políticas corporativas de desenvolvimento, e não como uma categoria automaticamente isenta de controles.

Agentes executam ações, não apenas exibem informações

Existe uma mudança qualitativa quando passamos de aplicações que apresentam informações para agentes que podem consultar dados, tomar decisões delimitadas e executar ações. Um workflow tradicional costuma possuir regras explícitas, enquanto sistemas agentic podem incorporar mecanismos de interpretação e planejamento mais dinâmicos.

Essa característica cria novos desafios de autorização, rastreabilidade, validação e recuperação. É necessário estabelecer quais sistemas podem ser acessados, quais operações exigem aprovação, como registrar decisões e como interromper comportamentos inadequados.

O avanço dessas ferramentas não torna o low-code obsoleto. Pelo contrário, pode ampliar sua função como plataforma de composição de interfaces, workflows, integrações e agentes. Mas a governança precisa acompanhar essa evolução e considerar riscos que vão além da aplicação tradicional.

O Brasil possui oportunidades relevantes, mas exige pragmatismo

O mercado brasileiro apresenta características que podem favorecer a adoção dessas plataformas, incluindo grande diversidade de organizações, necessidade permanente de eficiência, ambientes tecnológicos heterogêneos e pressão por digitalização.

No texto original, comentei experiências positivas em eventos dedicados ao tema, observando uma comunidade engajada e interesse crescente entre profissionais de diferentes perfis. Essa percepção continua relevante como registro pessoal, mas não deveria ser convertida automaticamente em uma conclusão quantitativa sobre o tamanho ou a maturidade do mercado.

A situação varia significativamente entre setores e empresas. Existem organizações com programas estruturados de Citizen Development, enquanto outras ainda precisam resolver questões básicas de governança, integração e padronização. Por isso, comparações genéricas entre Brasil e mercados internacionais devem ser feitas com cautela.

O ambiente regulatório também precisa ser considerado. Instituições financeiras, seguradoras, empresas de saúde e outras organizações podem possuir exigências específicas de segurança, continuidade, rastreabilidade e tratamento de dados. Essas

obrigações precisam ser aplicadas conforme a natureza das soluções, independentemente da ferramenta utilizada.

Vejo um espaço importante para modelos que combinam capacitação, plataformas aprovadas e equipes multidisciplinares. O potencial não depende apenas da disponibilidade de produtos, mas da capacidade das organizações de estabelecer um caminho suficientemente simples para estimular a adoção e suficientemente rigoroso para preservar controles essenciais.

CIO Codex Framework – No & Low-code

Dentro do CIO Codex Framework, o tópico mais diretamente relacionado a essa discussão é **No & Low-code**, apresentado entre os componentes de New Technologies na dimensão HOW, dedicada às condições para que a Tecnologia da Informação seja bem-sucedida na era digital.

O framework descreve essas plataformas como mecanismos capazes de ampliar o acesso ao desenvolvimento de soluções, acelerar respostas às necessidades empresariais e favorecer a colaboração entre áreas de negócio e Tecnologia. Ao mesmo tempo, destaca a necessidade de alinhamento com arquitetura corporativa, segurança, governança de dados e uma estratégia clara de adoção.

Essa combinação está no centro de toda a reflexão. O valor do low-code não se limita à diminuição da quantidade de código escrito. Está na possibilidade de modificar a maneira como a organização mobiliza conhecimento, distribui capacidade de implementação e responde às necessidades de transformação.

No & Low-code como componente de New Technologies

A presença do tema na camada de New Technologies demonstra que o framework não trata essas plataformas apenas como outra ferramenta de desenvolvimento. Elas representam uma alternativa para ampliar as capabilities tecnológicas disponíveis à organização e modificar os mecanismos utilizados para criar soluções.

O ganho potencial decorre da abstração de atividades repetitivas, utilização de componentes pré-configurados e redução de dependências em determinados cenários. Isso pode acelerar a implementação de aplicações e automações quando existe aderência entre os requisitos empresariais e os recursos oferecidos pela plataforma.

Entretanto, o framework também pressupõe que novas tecnologias precisam ser

avaliadas dentro do contexto mais amplo da organização. Portanto, a simples disponibilidade de uma plataforma não constitui justificativa suficiente para sua adoção.

É necessário compreender sua contribuição para a estratégia, suas relações com os ativos existentes e os impactos sobre competências, processos e modelo operacional. A tecnologia deve ampliar a capacidade de gerar valor sem comprometer atributos fundamentais da arquitetura empresarial.

Essa abordagem evita tanto a rejeição automática de novas soluções quanto a adoção motivada exclusivamente pelo entusiasmo com suas funcionalidades.

Relação com Fusion Teams e Business Agility

A conexão com Fusion Teams é especialmente importante. O CIO Codex Framework apresenta esse conceito como uma evolução do modelo organizacional que aproxima competências de negócio e tecnologia, permitindo colaboração mais direta na construção de soluções.

Low-code pode funcionar como habilitador desse modelo, mas não o substitui. Uma ferramenta visual não elimina a necessidade de alinhamento, priorização e responsabilização. O potencial aumenta quando profissionais de negócio e especialistas técnicos compartilham objetivos e conseguem construir respostas de maneira coordenada.

Essa relação também se conecta a Business Agility. A capacidade de responder rapidamente às mudanças empresariais não depende apenas da velocidade de codificação, mas da redução do tempo necessário para identificar uma necessidade, avaliar alternativas, tomar decisões e disponibilizar uma solução adequada.

A aproximação entre quem compreende o processo e quem domina a tecnologia pode reduzir interpretações equivocadas e ciclos desnecessários de retrabalho. Entretanto, a organização precisa preservar mecanismos de arquitetura e segurança para evitar que a autonomia produza soluções desconectadas.

O resultado desejado é uma agilidade organizacional real, não apenas uma aceleração localizada de atividades de desenvolvimento.

Conexão com Solution Engineering

Embora o tópico principal seja No & Low-code, sua aplicação empresarial possui forte relação com Solution Engineering, responsável por transformar necessidades em soluções implementadas, mantidas e evoluídas.

Uma aplicação construída com componentes visuais continua exigindo compreensão dos requisitos, definição de interfaces, tratamento de exceções, testes, implantação e mecanismos de sustentação. A plataforma modifica a forma de implementação, mas não elimina essas responsabilidades.

Solution Engineering também ajuda a estabelecer padrões adequados para diferentes níveis de complexidade. Uma automação pessoal pode seguir um caminho simples, enquanto uma aplicação transacional integrada a sistemas críticos precisa de requisitos mais rigorosos.

Essa conexão reforça a importância do Application Ownership. Uma solução criada por Citizen Developers precisa possuir responsáveis claros e documentação suficiente para continuar funcionando quando seus autores mudarem de equipe ou deixarem a organização.

O desafio está em aplicar a disciplina de engenharia de maneira proporcional. Exigir de uma automação simples todos os controles de uma grande plataforma corporativa pode eliminar os benefícios. Ignorar completamente a engenharia, por outro lado, pode produzir riscos significativos.

Technology Visioning e Enterprise Architecture

A adoção de plataformas low-code precisa estar conectada à visão tecnológica corporativa. Technology Visioning oferece o contexto para avaliar tendências, estabelecer direcionadores arquiteturais e compreender como novas tecnologias devem se integrar ao ambiente existente.

O principal benefício dessa conexão está na possibilidade de evitar decisões isoladas. Uma área pode encontrar uma ferramenta adequada para sua necessidade imediata, mas a organização precisa avaliar como a escolha se relaciona com integrações, dados, segurança, custos e competências disponíveis.

Enterprise Architecture ajuda a estabelecer princípios comuns para seleção de plataformas, utilização de APIs, administração de componentes e definição de padrões de interoperabilidade. Esses princípios não devem impedir a experimentação, mas oferecer critérios para que novas iniciativas possam amadurecer de maneira sustentável.

A governança arquitetural também permite distinguir soluções temporárias de ativos estratégicos. Uma automação criada para uma necessidade pontual pode ser perfeitamente adequada durante determinado período, mas exigir revisão quando seu escopo e sua criticidade aumentam.

Essa relação demonstra que arquitetura e agilidade não precisam ser objetivos contraditórios. Quando bem estruturada, a arquitetura reduz a necessidade de repetir decisões e facilita a criação de soluções consistentes.

Cybersecurity e governança de dados

No CIO Codex Framework, Cybersecurity possui natureza transversal. Isso significa que seus princípios e mecanismos precisam estar presentes em todas as demais capacidades, incluindo a utilização de No & Low-code.

A facilidade de criação não diminui a necessidade de controlar identidades, privilégios, acesso a informações e integrações. Pelo contrário, o aumento do número de aplicações pode ampliar a superfície de risco caso a governança não acompanhe essa evolução.

Uma estratégia adequada precisa combinar recursos nativos das plataformas com políticas corporativas de segurança. Controles sobre conectores, ambientes, compartilhamento, tratamento de dados e registros de auditoria podem contribuir para reduzir exposições.

Também é importante reconhecer que a existência de recursos de segurança no produto não garante automaticamente sua utilização correta. Configurações inadequadas, permissões excessivas e falhas de desenho funcional podem comprometer soluções desenvolvidas em plataformas tecnicamente robustas.

A governança de dados deve acompanhar esse processo, considerando classificação das informações, finalidade de utilização, retenção e integração com sistemas corporativos. Isso se torna ainda mais relevante quando aplicações low-code incorporam Generative AI ou agentes capazes de executar ações.

Service Excellence e continuidade operacional

Outra conexão fundamental envolve Service Excellence. Depois que uma solução entra em produção, a organização precisa garantir condições adequadas de operação, suporte e evolução.

Isso é especialmente importante no universo de Citizen Development, no qual aplicações podem ser criadas por profissionais cujas responsabilidades principais não incluem sustentação tecnológica.

Uma aplicação utilizada apenas ocasionalmente pode exigir mecanismos simples de suporte. Já uma solução integrada a processos críticos precisa contemplar monitoramento, gestão de incidentes, continuidade e procedimentos de recuperação.

A classificação de criticidade ajuda a estabelecer essas diferenças. O importante é evitar que aplicações relevantes permaneçam sem responsáveis claros pela operação.

Service Excellence também contribui para administrar problemas decorrentes de atualizações da plataforma, falhas de conectores ou mudanças nos serviços externos. Como parte da infraestrutura é controlada pelo fornecedor, a organização precisa compreender os limites de sua atuação e os mecanismos disponíveis para diagnóstico e recuperação.

A sustentabilidade do low-code depende, portanto, de sua integração às capacidades normais de prestação de serviços de TI.

Concluindo

Continuo considerando que low-code e no-code possuem um papel relevante na evolução do desenvolvimento de software e na aproximação entre Tecnologia da Informação e áreas de negócio. As plataformas amadureceram, novos casos de uso surgiram e a integração com Artificial Intelligence ampliou significativamente as possibilidades de criação de aplicações e automações.

Uma aplicação criada rapidamente pode se transformar em componente essencial de um processo empresarial. Um workflow pode passar a movimentar informações sensíveis. Um agente desenvolvido por uma área de negócio pode receber permissões para executar operações sobre sistemas corporativos. Em todos esses casos, a responsabilidade organizacional permanece.

Fusion Teams, plataformas compartilhadas, mecanismos de governança e capacitação podem permitir que as áreas de negócio participem mais diretamente da construção das soluções, enquanto TI preserva a responsabilidade sobre arquitetura, segurança, integração e continuidade.

A introdução de AI e de agentes acrescenta uma nova dimensão a esse cenário. A quantidade de pessoas capazes de desenvolver software tende a crescer, mas a capacidade de uma organização de administrar seu ambiente não aumenta automaticamente na mesma proporção.

Por isso, considero que a principal prioridade deveria ser construir um modelo operacional capaz de transformar democratização em geração de valor sustentável. Isso exige abandonar tanto a ideia de que toda inovação precisa ser centralizada quanto a percepção de que ferramentas simples dispensam governança.

O CIO Codex Framework ajuda a organizar essa discussão ao conectar No & Low-code a Fusion Teams, Business Agility, Solution Engineering, Technology Visioning,

Cybersecurity, Service Excellence e IT Transformation.

No fim, a questão não é simplesmente quantas aplicações uma empresa consegue criar com pouco ou nenhum código. É quantas dessas soluções produzem resultados relevantes, permanecem seguras, podem ser sustentadas e continuam fazendo sentido quando o negócio evolui.

A verdadeira maturidade será alcançada quando desenvolver de maneira mais simples deixar de representar abrir mão de disciplina e passar a significar que a própria disciplina foi incorporada à plataforma, aos processos e à cultura da organização.