

# BIMODAL IT

Bridging the bimodal divide



## TABLE OF CONTENTS

- 2 What is bimodal IT?
- 3 Characteristics and use cases
- 5 Getting started and bridging the bimodal divide
- 7 Summary

### **SANDY KEMSLEY,** KEMSLEY DESIGN LTD.

Sandy Kemsley is an independent analyst and application architect specializing in business process management and the social enterprise. During her career of more than 25 years, she founded and ran product and service companies in the area of content management, process management and e-commerce, and held the position of BPM evangelist for a major BPM vendor.

Currently, Sandy practices as a BPM industry analyst and process architect, performing engagements for end-user organizations and BPM vendors. She writes the popular "Column 2" BPM blog at [www.column2.com](http://www.column2.com) and is a featured conference speaker on BPM and digital transformation. She is a contributing author to books on social BPM and adaptive case management, and the winner of the 2016 Marvin L. Manheim award for significant contributions in the field of workflow.

The concept of bimodal IT—having two different application development streams with different techniques and goals—isn't new. Many organizations have development groups that are not part of the standard IT development structure, including developers embedded within business units creating applications that IT couldn't deliver quickly enough, and skunkworks development groups prototyping new ideas.

In many cases, this split didn't occur by design, but out of situational necessity when mainstream IT development groups were unable to service the needs of the business, especially transformational business units created specifically to drive innovation. However, in the past few years, analysts are positioning this split as a strategic action to boost innovation. By 2013, McKinsey & Company was proposing "greenfield IT"—technology managed independently of the legacy application development and infrastructure projects that typically consume most of the CIO's attention and budget—as a way to innovate more effectively. The advisor found a correlation between innovation and corporate growth, and greenfield IT as a way to achieve that innovation. By the end of 2014, the term "bimodal IT" was becoming popular, with Mode 1 being the traditional application development cycle focused on stability, well suited to infrastructure and legacy maintenance and Mode 2, focused on agility and innovation, similar to McKinsey's greenfield IT.

For example, an IT development group can manage a long-term ERP modernization project using a standard Mode 1 life cycle, since the stages and expected outcomes are well understood. The development life cycle and tools are more traditional, and portions of the development may be outsourced with

relatively low risk. To meet shorter-term objectives, such as a new regulatory requirement for product quality assurance, a Mode 2 development group can use low-code tools to create a case management application that can be quickly modified to meet changing regulations while leveraging information from the ERP system.

Is bimodal IT truly a new type of corporate IT structure for fostering innovation, or just a new label for the “citizen development” or “shadow IT” already existing in organizations? Do the benefits of establishing formal Mode 1 and Mode 2 development groups justify the disruption, or is this just the latest analyst fad? Is this a call to cast aside waterfall methods and allow agile development to flourish or an attempt to regulate citizen development?

In this white paper, we will look at the definition and use cases for bimodal IT, and the advantages and disadvantages of introducing it into your organization. If you decide to move ahead with bimodal IT, we’ll have some pointers on when to use Mode 1 versus Mode 2 development and how to bridge the divide between them.

## What is bimodal IT?

Gartner defines bimodal IT as “the practice of managing two separate, coherent modes of IT delivery, one focused on stability and the other on agility. Mode 1 is traditional and sequential, emphasizing safety and accuracy. Mode 2 is exploratory and nonlinear, emphasizing agility and speed.”<sup>1</sup> From our perspective, Mode 1 provides a stable and well-understood IT delivery model for infrastructure projects, legacy maintenance and other infrequently modified systems that require traditional development cycles to ensure security, efficiency and accuracy. Mode 2 provides a more fluid environment for rapid development and deployment of new ideas, where agile methods and a DevOps provisioning model allow for greater business involvement and fast iterations.

To generalize, Mode 1 is “slow IT” for systems of record while Mode 2 is “fast IT” for systems of engagement/differentiation. Both modes are necessary but require different skills, tools and techniques: The strengths of Mode 1 development teams are often weaknesses in Mode 2 development and vice versa. The goal of bimodal IT is not to make Mode 1 and Mode 2 teams more similar to each other, but to allow each type to focus on the projects best suited to their mode. The following section examines the defining characteristics of Mode 1 and Mode 2 projects to assist you with determining which approach will work best for a given situation.

Although there is general agreement that some development projects require a more traditional approach while others benefit from agile methods, bimodal IT takes this a step further by separating these capabilities and project types into completely separate groups within an organization. In addition to their differences in development methods and tools, Mode 1 and Mode 2 development groups don’t report through the same organizational structure and don’t use the same delivery mechanisms.

The rationale for this proposed split of Mode 1 and 2 into separate reporting and budget structures is to reduce conflicts for resources and budget within IT, which is currently tasked both with stable current operations and innovative future results. This presupposes that both modes are within IT, but in actual practice, Mode 2 capabilities are more often represented currently by shadow IT and citizen developers embedded within business units. Bimodal IT would have the effect of gathering these disparate development capabilities



<sup>1</sup> Gartner IT Glossary, Bimodal IT, <http://www.gartner.com/it-glossary/bimodal>

into a common reporting structure with standardized tools and disciplined methodologies, even if they remain decentralized, and create synchronicity between Mode 1 and Mode 2 efforts. Citizen development will become more IT-like and will also have the capability to become more strategic. As discussed later, bridging the divide between Mode 1 and Mode 2 is critical: Mode 2 relies on the information and services infrastructure provided by Mode 1, while Mode 1 relies on Mode 2 for testing both new product ideas and new development methods that may eventually be rolled back into Mode 1.

## Characteristics and use cases

In general, Mode 1 is applied to building and maintaining systems of record, while Mode 2 projects tend to be customer/employee-facing systems of engagement and innovation where there is little to lose and a lot to gain from experimentation.

Mode 1 IT is what exists in most large organizations today and has been used successfully to build and maintain applications and infrastructure to support predictable, long-term business needs. The development methodology is based on predefined requirements, deliverables and schedules, making it possible to outsource some portions of the development without significantly increasing risk. Although many organizations have moved away from a fully waterfall Software Development Life Cycle (SDLC) to a somewhat more agile approach, enterprise development projects are usually conducted in a waterfall-like manner: Create a plan, then execute on that plan. Operational stability is achieved through rigorous testing cycles and cautious production roll outs, with approval-based governance controlling the entire cycle.

Mode 2 IT, used where there is no legacy of infrastructure and applications to maintain, uses experiments and prototypes to try out new ideas in order to achieve specified business goals and outcomes. The project team may not have a predetermined vision of the solution's functionality, the technology required or the development schedule. The business needs are often uncertain, changing during the project life cycle as possible solutions emerge and as competitors expose their own solutions. This necessitates a fully agile development methodology, often using model-driven development where highly granular microservices are quickly assembled into solutions with a high degree of business participation. This often leverages a public cloud Infrastructure-as-a-Service (IaaS) for fast provisioning and scaling, bypassing on-premises infrastructure and operational bottlenecks. Governance of the SDLC is continuous and process-based, and relies on automating some of the development processes to support continuous delivery. Mode 2 typically embodies DevOps where building, testing, releasing and supporting software is done by a unified collaborative team.

Sandy Kemsley  
on **bimodal IT**



[Click here to listen to the full-length interview.](#)

The development environment for Mode 2, as mentioned above, is often visual and model-driven: “low-code” or “no-code” environments such as a Business Process Management System (BPMS) and other rapid application development tools. This provides a number of benefits for agile prototyping and development:

- Fast visual development of processes and user experience
- Pre-built interfaces to other systems and services
- Leverage existing business object, data and content metadata models
- Design, develop and deploy on the same platform
- Built-in security model and infrastructure
- Built-in collaboration for business and developers to exchange ideas
- Abstraction from code to allow less-technical developers and end-users to build applications
- Ability to extend functionality with code where required

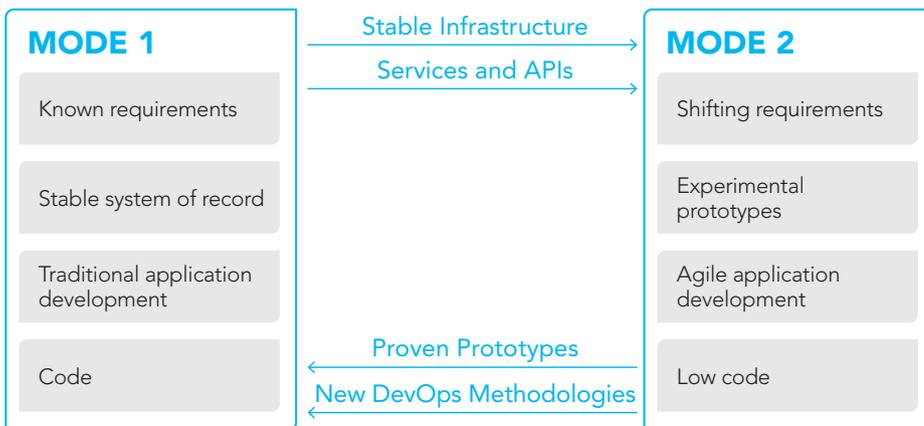
Some no-code BPMS and case management environments allow knowledge workers to build their own processes “on the fly” while working, which can then be examined in aggregate to determine emergent patterns to drive more structured work by Mode 2 or Mode 1 development teams.

Mode 2 development results in many more false starts and failed solutions than Mode 1. But the speed of the cycle allows Mode 2 development to “fail fast”: A solution may go from idea to operational prototype to the trash bin in a matter of days, while the team moves on to other potential solutions. This experimentation can include the entire DevOps infrastructure as well as the applications, such that the platforms, tools and methods evolve over time to suit the needs of the Mode 2 teams and the business users. This requires a very different culture than Mode 1 teams, who tend to be more cautious guardians versus risk-taking improvisers.

## Getting started and bridging the bimodal divide

Given that almost 50 percent of technology budget today is outside IT, most organizations have already started on bimodal IT without realizing it: Technology budgets controlled within business areas are almost certain to include some form of Mode 2 DevOps. There is no need to move decentralized Mode 2 IT under the control of central IT; in fact, there is a strong argument for leaving it within the business units in order to maintain the close ties to the business required for innovative and agile development methodologies. Instead, a Center of Excellence (CoE) for Mode 2 DevOps could provide a way to support and standardize those teams' efforts without compromising their agility.

It's not just a matter of standardizing Mode 2 tools and techniques, however; creating bimodal IT can have significant cultural implications within an organization. Care must be taken to create a healthy and desirable culture in both modes, to reduce friction between them, and to allow for an exchange of ideas, projects and technological innovations. Mode 1 and 2 development teams are not interchangeable due to different skills sets and personalities, but both are essential: equal but different. Mode 1 creates the digital core, while Mode 2 leverages those core capabilities in combination with new technologies. Mode 2 tests out new development methods and technologies, some of which may be adopted into Mode 1 development methodologies. Mode 2 solutions that stabilize and are broadly applicable within the organization can be refactored into the Mode 1 digital core, where they can be further evolved to improve robustness and scalability.



Start with bimodal IT by identifying the skills and methods required for each mode, then see how existing teams fall naturally into Mode 1 and Mode 2, whether centralized or decentralized. Note that decentralized citizen development is not necessarily Mode 2: Much of the business-led development of reporting and data manipulation, for example, is Mode 1 in their requirements and methods.

Identify initial Mode 2 projects that don't require changes to the digital core, but are strategically important and customer-facing, developed for short-term delivery with high business involvement.

Begin Mode 2 standardization by assembling a portfolio of experimental innovation techniques, development tools and agile/iterative development methodologies, and training the Mode 2 teams and related business areas on their use. Establish sourcing and procurement models that allow for cloud or internal provisioning and flexible funding based on goals rather than predefined deliverables. Provide collaboration capabilities for Mode 2 teams across different business areas in order to share experiences and knowledge. Ensure that sufficient discipline is in place to minimize the technical debt that can be created during overly rapid development. As team maturity increases, identify best practices for incorporation into the CoE. Don't wait to have this entire vision worked out before starting with Mode 2 projects: There will be iterations of the methodology and tools, not just the solutions.

Next, create the bridge between Mode 1 and Mode 2 teams. Initially, this will be to expose the Mode 1 systems and services to the Mode 2 teams for potential inclusion in solutions; as Mode 2 teams create workable solutions, some of these solutions may be taken over by Mode 1 for stabilization and integration into the digital core. Also consider applying methods—such as agile development and DevOps—that prove successful in Mode 2 IT back to Mode 1 teams and projects, within the limits of their stability-oriented requirements. Create predefined points for rolling Mode 2 solutions back to Mode 1, based on criteria such as code stability, size of user base or potential for reusability. Implementing a single development platform that provides both Mode 1 and Mode 2 development capabilities can make it easier to create this technology bridge, although the platform must provide both deeply technical capabilities to the Mode 1 developers and an easy-to-use low-code prototyping environment for Mode 2 developers.

Define the reporting structure for Mode 2 teams, both centralized and decentralized. Centralized Mode 2 may be a separate group from Mode 1 development but still reporting to the CIO. Or Mode 2 may report to a Chief Digital Officer (CDO), who is a peer of the CIO with more of a digital business transformation focus. Ensuring that the gap between Mode 1 and Mode 2 is bridged is the responsibility of architects in the office of the CIO/CDO, who identify the emergent Mode 2 solutions and methods that can be shifted over to Mode 1 teams.

Establish guidelines for assigning new development projects to Mode 1 or Mode 2. This will assist in allocation of budget and resources, and reduce territorial conflicts over systems that may be claimed by both modes.

Assess the ability of software and service providers to address your Mode 1 and Mode 2 tool and platform needs. In particular, understand platform providers' strategy for the peaceful coexistence of cloud and on-premises solutions. Consider the impact of moving core systems to the cloud on Mode 1 development, infrastructure and operations.

## Summary

Bimodal IT is already a fact of life in many organizations, even if it isn't formally identified: Many IT development teams are engaged in long-term projects to build and maintain core systems of record, while citizen developers fill the gaps for short-term business needs and prototyping new ideas for customer engagement. Providing the tools and techniques for successful coexistence of these two development modes can provide the advantages of both: maintaining stable infrastructure while fostering innovation.

The key to a successful bimodal IT strategy is to provide technology tools that support both modes—from full development to low-code environments—and allow solutions and ideas to flow easily between them. Mode 1 projects create the digital core, which is then leveraged as services and infrastructure by Mode 2 projects; Mode 2 applications may eventually stabilize and be refactored back into the digital core by a Mode 1 team.

There are also organizational issues with adopting bimodal IT, including resourcing, management structures and deciding which new projects are Mode 1 or Mode 2. Both modes are important to the success of an organization. But they thrive under quite different conditions.

### Take the next step

For more information, talk to your local Software AG representative.

---

## ABOUT SOFTWARE AG

The digital transformation is changing enterprise IT landscapes from inflexible application silos to modern software platform-driven IT architectures which deliver the openness, speed and agility needed to enable the digital real-time enterprise. Software AG offers the first end-to-end Digital Business Platform, based on open standards, with integration, process management, in-memory data, adaptive application development, real-time analytics and enterprise architecture management as core building blocks. The modular platform allows users to develop the next generation of application systems to build their digital future, today. With over 45 years of customer-centric innovation, Software AG is ranked as a leader in many innovative and digital technology categories. Learn more at [www.SoftwareAG.com](http://www.SoftwareAG.com).

© 2016 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

SAG\_Bimoda\_IT\_8PG\_WP\_Aug16

