

NO-CODE PLAYBOOK TOOLKIT: THE COMPANION GUIDE



Creatio

THE NO-CODE PLAYBOOK TIMELINE



Creatio



**GRAND
RELEASE
October 4th**



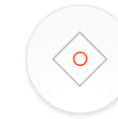
Principles of no-code development



12 stages of the no-code development lifecycle



Complexity assessment framework



Building the Center of Excellence



Governance framework



No-code application audit and monitoring



Fusion team roles and responsibilities

he No-Code Playbook downloads:

10,000+





NO-CODE PLAYBOOK

PODCAST FOR OPERATIONAL, DIGITAL, AND IT LEADERS

to learn insights, tips, and success stories from like-minded peers on how to leverage a no-code approach to transform businesses and deliver applications of any complexity



JASON MILLER
Head of Pre-Sales,
Americas, Creatio



BURLEY KAWASAKI
Founder, Tachyon
Solutions



ISAAC SACOLICK
Founder and President,
StarCIO



PHILIP LAKIN
Co-Founder & CEO,
NoCodeOps



PHIL SIMON
Award-Winning Author,
Keynote Speaker

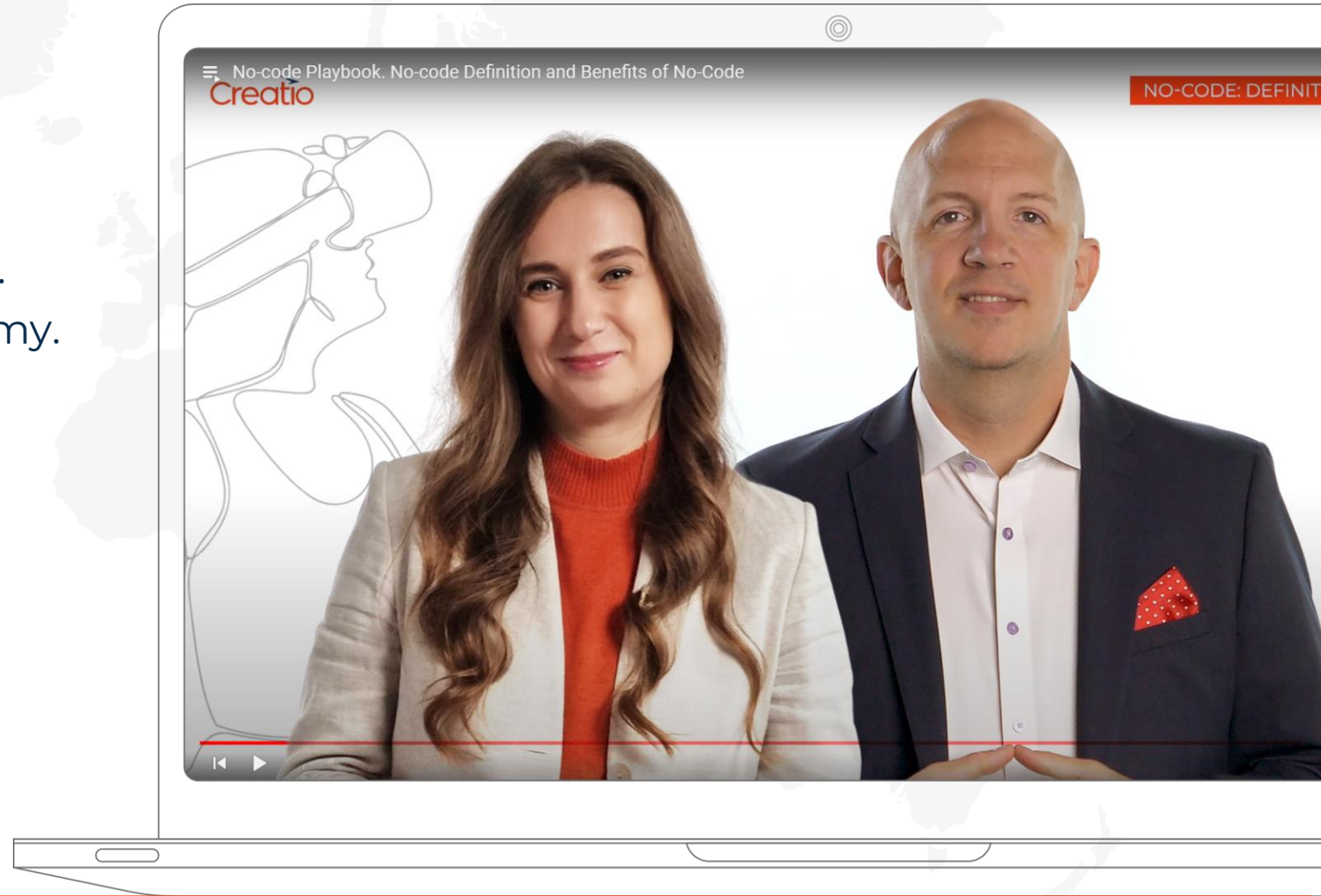
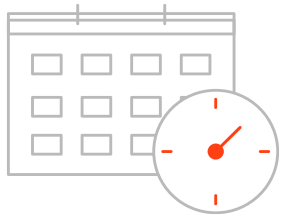


ANDIE DOVGAN
Chief Growth Officer,
Creatio

NO-CODE CREATOR

The core class with an overview of the No-Code Playbook key concepts. Available at Udemy and Creatio Academy.

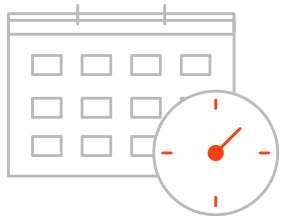
- **9 Modules**
- **8 hours**
- **Certification**



NO-CODE ARCHITECT

An advanced class for experienced no-code practitioners that is focused on practical aspects of designing, launching, and enhancing business applications of any complexity following three key delivery models: **DIY, CoE, and Fusion teams.**

- **6 Modules**
- **Practice**
- **Certification**



WHAT WE HEARD FROM YOU

Following the launch of the No-code Playbook, we listened to your feedback and comments to understand how we could make it even more effective to helping your No-code delivery efforts.

"The No-code Playbook is a great foundational, vendor-agnostic primer, but there is a need for more Creatio-specific deployment guidance to assist with project delivery."

"Please provide examples of delivery considerations and questions to ask in each stage of the lifecycle."

"Help us with guidance on estimating and planning."

"Help us with applying this at different levels of project complexity."

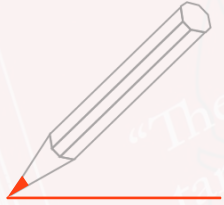
"We hired a group of no-code creators and need to train them how to use the No-Code Playbook in practice to run Creatio's projects. How can you help?"

NO-CODE PLAYBOOK TOOLKIT

The next step in our No-Code Playbook expansion process

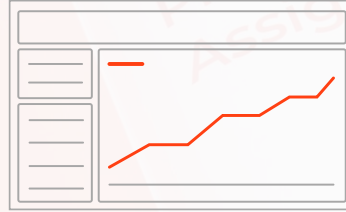
Set of tools, questionnaires, practical examples, and hints to complement the **12 stages of the no-code lifecycle** and provide our partners and customers with a step-by-step guide on how to deploy all your projects following the No-Code Playbook methodology.

- Creatio-specific addendum to the Playbook to support efficient No-code project delivery
- Primarily targets hands-on practitioners (No-code Creators and Architects)
- Organized by the No-code Lifecycle for easy reference
- Provides questions, best practices and tools for each stage of the No-code Lifecycle
- Tips and considerations organized by Delivery Models (DIY, CoE, Fusion Teams)
- Guidance for estimation and scoping of effort
- Tips on things to watch out for or avoid



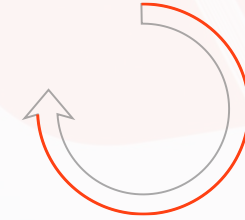
DESIGN

- ✓ Tips and Practices for defining Business Use Cases
- ✓ Defining your solution architecture (Buy vs. Build vs. No-code vs. No-code Templates)
- ✓ Tips and Practices for conducting a vendor evaluation or selecting partner assistance
- ✓ Guidance for effective design (UX/UI, Workflow and Logic, Integrations, Dashboards & Analytics)
- ✓ Best Practices for Effective Prototyping
- ✓ Guidance for defining MVP and estimation of effort



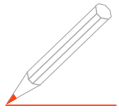
GO-LIVE

- ✓ Tips and Practices for evolving your prototype into your Go-Live app
- ✓ Guidance for effective feedback collection from users and stakeholders
- ✓ Examples and process guidance for change management
- ✓ Guidance for efficiently automating governance checks (including use of the new Creatio Governance app)
- ✓ Guidance on setting up environments
- ✓ Tips and Practices for an effective UAT
- ✓ Tips for effective user training sessions



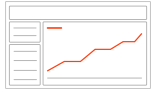
EVERYDAY DELIVERY

- ✓ Guidance for effective feedback collection once your app is in Production use
- ✓ How to run an effective stakeholder interview to collect feedback
- ✓ Tips and Practices for managing and prioritizing your request backlog
- ✓ Techniques to enable efficient Everyday Delivery including Loose coupling, Change isolation, Clear quality gates, Test automation and automated deployment.
- ✓ Tips and Practices for conducting an Application Audit



DESIGN

- Stage 1: **Business use case**
- Stage 2: **Options analysis**
- Stage 3: **Design and prototyping**
- Stage 4: **Project assignment**



GO-LIVE

- Stage 5: **Prototype-to-MVP**
- Stage 6: **Feedback loop**
- Stage 7: **Governance checks**
- Stage 8: **First release**



EVERYDAY DELIVERY

- Stage 9: **Feedback collection**
- Stage 10: **Incremental improvements**
- Stage 11: **Everyday delivery**
- Stage 12: **Application audit**





TOOLKIT: **Stage 1 - Business Use Case**



Pre-requisites
/ Inputs

Key Questions: DIY
Delivery Model

Key Questions: CoE,
Fusion Delivery
Model

Success
Criteria

Best Practice
Tips



TOOLKIT: **Stage 2 – Options Analysis**



Pre-requisites
/ Inputs

Key Questions: DIY
Delivery Model

Key Questions: CoE,
Fusion Delivery
Model

Vendor
Evaluation

Implementation
Assistance
Needs



Pre-requisites
/Inputs

Key Questions: DIY
Delivery Model

Key Questions: CoE,
Fusion Delivery
Model

Vendor
Evaluation

Implementation
Assistance
Needs

Key Questions: CoE, Fusion Delivery Model (Examples)

Step #1: Identify Packaged App Feasibility

- Start by identifying the possible fit of Packaged Applications to satisfy each process use-case. Is your process standardized for the domain and does not require amendments?
- Are there any key gaps in the “out of box” functionality that would require extension or customization? Are there any regulatory requirements that must be met by the solution?
- What is the cost (software licenses, implementation) of the packaged application and how does it compare to other options

Step #2: Identify Pre-built App Templates

- Identify the use of Pre-built App Templates as a starting point wherever there is a fit. Are there any existing [vertical solutions](#) or [applications](#) in the Marketplace that could be repurposed or modified to meet the business need?
- Do you consider the processes in scope for this no-code application to be unique to your organization or to be common to industry standards?

Step #3: No-code Platform Development

- After you have evaluated the availability of packaged apps or pre-built app templates, what are the remaining key processes that will need to be created from scratch using no-code development?
- Review the list of processes and sub-processes identified during the Business Use Case. Does the team have sufficient skills (both functional and technical) to support building out the target set of processes from scratch using no-code? What is the learning curve?

Step #4: Identify Composable Services

- The prior steps should have formed the core of the overall solutions architecture. You should now also revisit the elements of your application to see if there are narrow use cases that can also be covered by third-party services and composed into the core no-code application. (For example, the sales tax calculation use case can be automated by the third-party module.)

Creatio TOOLKIT: **Stage 3 – Design & Prototyping**



Pre-requisites
/ Inputs



Key Questions: DIY
Delivery Model



Key Questions: CoE,
Fusion Delivery
Model



Prototyping
Best Practices



TOOLKIT: **Stage 4 – Project Assignment**



Pre-requisites
/Inputs

“E2E Process”
Approach

“Complete Workflow”
Approach

Estimation
Guidelines

Schedule &
Resources

Pre-requisites
/Inputs“E2E Process”
Approach“Complete Workflow”
ApproachEstimation
GuidelinesSchedule &
Resources

Estimation Guidelines:

- Assess complexity using the Application Matrix guidelines & select Delivery Model
- Ensure Workflows and Features are sufficiently decomposed (especially for CoE and Fusion models)
- Use Workflow and Feature inventory to perform T-shirt sizing estimations

Size	Explanation	Suggested Time
S	Basic no-code customization (e.g. adding fields)	4-8 hours
M	Complex no-code customization (e.g. business process with many branches and conditions)	8-40 hours (varies based upon team experience & workflow complexity; should be < 1 week)
L	The low-code effort required (e.g. adding basic Javascript rules to UI)	40-80 hours (varies based upon team experience & complexity; should not be > 2 weeks)
XL	Complex coding is required (e.g. adding a new telephony integration feature to the telephony connector)	TBD (varies based upon pro dev complexity, development approach, etc).



TOOLKIT: **Stage 5 – Prototype-to-MVP**



Pre-requisites
/ Inputs



UI and
Data Model



Workflow/ User
Experience



Integrations &
Access Rights



Reports &
External Access



Pre-requisites
/Inputs

Select & Engage
Stakeholders

Build & Review
Workflows

Triage Go-Live
Scope

Address
Feedback



TOOLKIT: **Stage 7 – Governance Checks**



Pre-requisites
/ Inputs



Preparing
Governance Checks



Conducting
Governance



Automating
Governance



Pre-requisites
/Inputs

Preparing
Governance Checks

Conducting
Governance

Automating
Governance

Identifying Governance Requirements (Examples)

Compliance with Laws and Regulations

- Do we have specific laws and regulations the application needs to comply with?
- Have we identified all potential legal and regulatory risks associated with the application?
- Have we established/Do we need procedures for monitoring compliance with relevant laws and regulations?
- Have we ensured that all data collected and processed by the application comply with relevant data protection and privacy laws (GDPR, CCPA, etc.)?
- Are there any accessibility requirements that the application must meet?
- Is the application designed to comply with relevant security and confidentiality requirements?
- Have we conducted a thorough risk assessment of the application to identify any potential vulnerabilities or weaknesses?
- Are all third-party vendors and suppliers involved in the development and operation of the application in compliance with relevant laws and regulations?
- Have we implemented/Do we need appropriate controls and procedures to ensure ongoing compliance with relevant laws and regulations?

Internal Policies and Procedures

- What are the internal policies and procedures that the application needs to comply with?
- How does the application support and enforce the company's code of conduct and ethics?
- Does the application adhere to the company's standards for data protection, privacy, and confidentiality?
- How does the application ensure compliance with company policies related to security, such as access controls and authentication mechanisms?
- Are there any specific internal policies related to software development or application design that the application needs to comply with?
- How does the application support the company's disaster recovery and business continuity plans?
- Does the application support the company's change management policies and procedures?
- How does the application ensure compliance with the company's vendor management policies and procedures?
- Are there any internal compliance audits or assessments that the application needs to pass?



Pre-requisites
/ Inputs

Deployment
& Support

Provisioned
Environments

Final User
Acceptance

User Support
& Enablement



Pre-requisites
/ Inputs

Tips & Best Practices
for Feedback

User-generated
Feedback

System-Generated
Feedback

Evaluating
Feedback



Evaluating Feedback:

- It involves reviewing the feedback you have received and identifying patterns, trends, and insights that can be used to improve the project.
- Framework for categorizing based on the type of issue, severity, and priority.

Category	Feedback Type	Issue	Severity	Priority
User Experience	Usability	Navigation is confusing	High	High
User Experience	Design	Font size is too small	Medium	Medium
User Experience	Accessibility	Color contrast is low	High	High
Functionality	Performance	App is slow to load	High	High
Functionality	Features	Missing feature X	High	High
Functionality	Bugs	Button not working	High	High
Support	Documentation	User manual is outdated	Medium	Medium
Support	Customer Service	Long wait times on support calls	High	High
Support	Technical Support	Bug was not resolved	High	High
User Engagement	Content	Content is outdated	Medium	Medium
User Engagement	Interactivity	No way to interact with other users	High	High
Business Impact	Business value	Functionality is not generating enough revenue	High	High



TOOLKIT: **Stage 10 – Incremental Improvements**



Pre-requisites
/ Inputs

Backlog
Management

Prioritization
& Review

Delivery Process
(Build/Test)

Review
Governance Checks



TOOLKIT: **Stage 11 – Everyday Delivery**



Pre-requisites
/ Inputs

Loose
Coupling

Change
Isolation

Quality Gates /
Test Automation

Automated
Deployment



TOOLKIT: **Stage 12 – Application Audit**



Pre-requisites
/ Inputs



Audit Checks:
DIY Teams



Audit Checks:
CoE & Fusion Teams



Tips &
Best Practices



Automating
Audits

NEXT STEPS

May 5th

**Draft
Introduction and
Feedback
Collection**

July

**The Toolkit is
completed**

September

**Publication
and active
training**