

NIYAM ITTM

Responsive MethodologyTM

Making Sense of Agile Projects

Why Responsive Methodology?

The Federal IT environment is undergoing a major transformation from traditional waterfall SDLC to Agile/DevOps. Most Federal IT implementations are mission-critical, involve legacy systems and have high security requirements. They must be cost effective in both time and material, and provide a high degree of transparency in the development process. Incremental and generative development processes are proving to be a challenge for project monitoring in such environments.

Responsive Methodology is drawn from successful practices that follow the Agile manifesto, and in sync with DevOps for tangible results. Implemented effectively, the outcomes of RM are transparency, high speed-to-market, and low risk. RM is based on 'Response' to change. The response is usually in the direction of improvement. A key element of measured response is metrics: what cannot be measured cannot be improved.



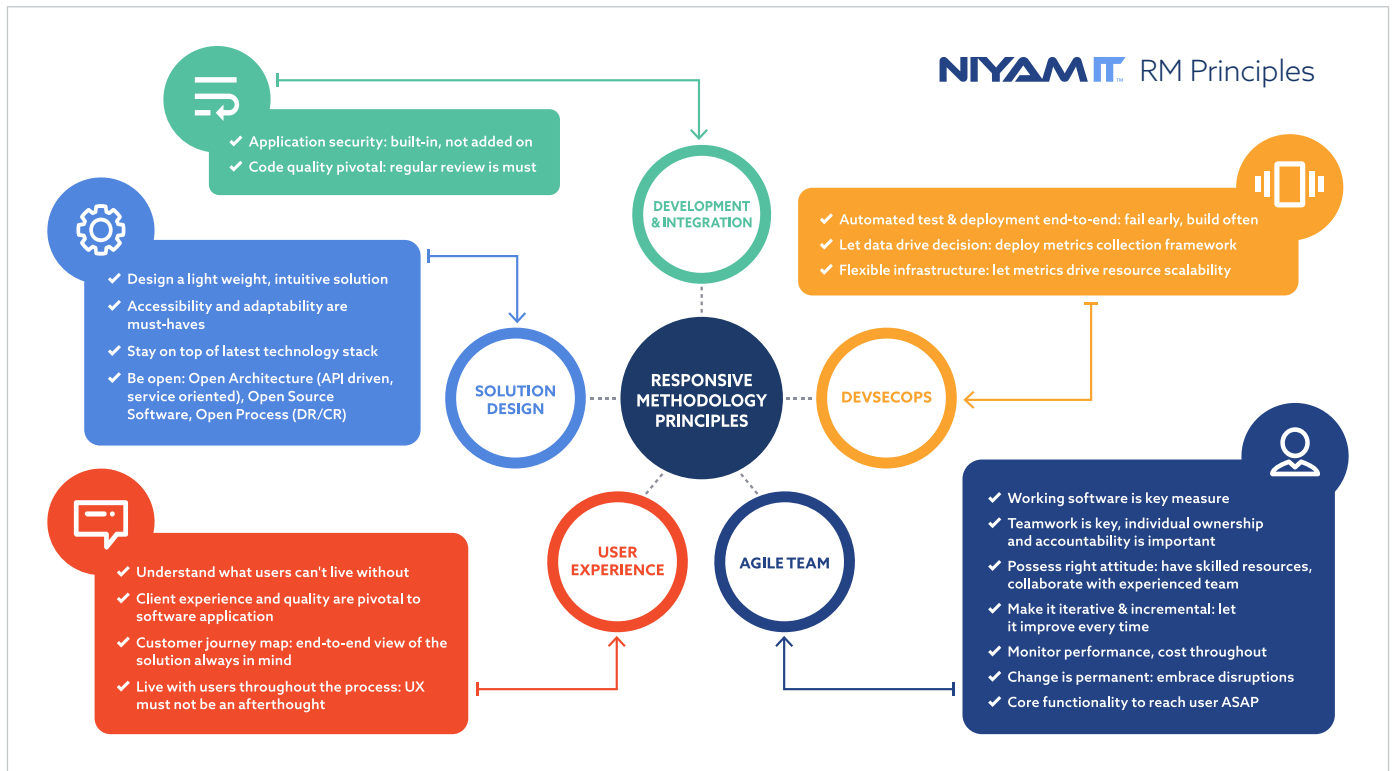
“Agile Processes are generative, not prescriptive. Processes need to evolve as needed, not prescribed up front. It begins with a set of simple processes and adds others as they are needed.”

Jim Highsmith
Agile Project Management

Contents

Section 1: Responsive Methodology Principles	3
Agile Team Principles: Mind Over Matter	3
User Experience Principles: Active Stakeholder Participation	3
Solution Design Principles: Response to Change	4
Development and Integration Principles: Developer’s Oath	4
DevOps Principles: Continuous Deployment Does Not Mean Continuous Delivery	4
Section 2: Implementation Flow	5
Incorporation of Agile Delivery Principles and DevOps Stage 1	5
UX, Solution Design Principles, and DevOps Stage 2 and 3	5
Development and Integration Principles and DevOps Stage 4	6
Section 3: Key Indicators	7
Extract Tool Specific Metrics: What to Measure	7
Analysis and Distribution: For Whom	7

Section 1: Responsive Methodology Principles



COMPONENT ONE

Agile Team Principles: Mind Over Matter

Team composition plays an important role in success stories. A defined technology stack serves as a guideline for skill requirements of team members. Team members should be ready to embrace change in any form – technology, requirements, or other people – with minimal disruptions. Responsive Methodology teams function on 3-point ideology at the grassroots level:

- Assess: Identify, isolate and prioritize high performance enhancing areas
- Strategize: Determine ROI on business change implementations
- Implement: Orchestrate rapid implementations and continuous improvement

Responsive Methodology team readiness involves the

degree of understanding these expectations:

- Working software is the primary measure of progress
- Teamwork is key but individual ownership and accountability is important
- Development of a customized action plan that is iterative and incremental, with improvement in every iteration
- Core functionality or defined minimum viable product (MVP) should reach the users as soon as possible
- Cut down on time spent on understanding the domain by staffing the team with adequate SMEs

COMPONENT TWO

User Experience Principles: Active Stakeholder Participation

Throughout the solution delivery process, active stakeholder participation is essential for top notch user experience. The team should foster a spirit of openness and mentor those

new to Agile with practical advice, resources, tools and community support. Team goals for this stage include:

- Understanding pain points and wish lists
- Maintaining an active customer journey map with end to end solutions always in mind
- Providing for accessibility requirements, including device interactions and 508 compliance checklists, to facilitate active stakeholder participation

Identify Key Stakeholders

Key stakeholders are part of the Integrated Product Team (IPT) and determine the definition of done, journey maps and product vision.

Determine Participant Style: Reactive/Proactive

Reactive stakeholders may cause a velocity lag when the team waits for responses.

Determine Best Times & Channels for Stakeholder Communication

Some people work best with emails, some via phone calls and some prefer a face-to-face interaction. Detecting these factors early on ensures fruitful stakeholder engagement.

COMPONENT THREE

Solution Design Principles: Response to Change

Agile implies that the cost of change is small, but this isn't the reality. The cost of change in real world enterprise systems is not small. The solution architecture should be flexible to accommodate new functionality and workloads. Flexibility is accomplished by forward thinking and planning for change. Team goals should include:

- Designing a light-weight solution to minimize impact of change
- Designing a simple and intuitive solution
- Designing for a purpose and stakeholder value
- Designing with scalability and adaptability in focus
- Designing with latest technology stack, frameworks and team learning curves in perspective
- Favoring open source software, open architecture (API driven, service oriented) and open processes for Change Request (CR) / Defect Report (DR)

COMPONENT FOUR

Development and Integration Principles: Developer's Oath

Federal services hold security as the highest priority. Continuous review for security should be conducted throughout iterative and incremental development by:

- Establishing a process to test vulnerabilities in components of each layer of the technology stack
- Certifying that these components may be reused for multiple services
- Maintaining code quality with automated reviews before merging upstream
- Developing continuous integration strategies for compiling code and potentially often rebuilding test bases
- Adopting the Infrastructure-as-Code principle
- Scheduling automated deployment of successful builds into appropriate environments on a regular basis

COMPONENT FIVE

DevOps Principles: Continuous Deployment Does Not Equal Continuous Delivery

Continuous delivery is the capability to deploy in any environment at any time. It is a must and is required to quickly recreate the environment, code version, and configuration to match those of the customer to fix a bug or defect.

Continuous deployment is an option, which is chosen based on the type of product, service or organizational need. RM suggests utilizing feedback from these items, after the first deployment cycle, to plan the frequency of subsequent continuous deployment:

- Code improvement
- Changes/improvement to environment
- Improvement to the process of delivery
- Automate deployments end-to-end for orchestration
- Deploy metrics collection framework
- Utilize scalability metrics to improve infrastructure flexibility

Section 2: Implementation Flow

INCEPTION

Incorporation of Agile Delivery Principles and DevOps Stage 1

Teams cannot work under the assumption that they can use any technology that they want. There is a risk of duplication in functionality. Added to the fact that each individual product may show stellar performance but the entire product/platform should work together. It should not conflict with other systems. It should take advantage and enhance shared infrastructure.

The key to high speed to market is Platformization. Platformization consists of predefined application frameworks, pre-configured tooling, test automation frameworks and ready to plug into environments. It provides a robust scaffolding to deploy risk free, quickly and easily to deliver business value.

Planning Checklist:

- Select a software stack based on reliable, proven

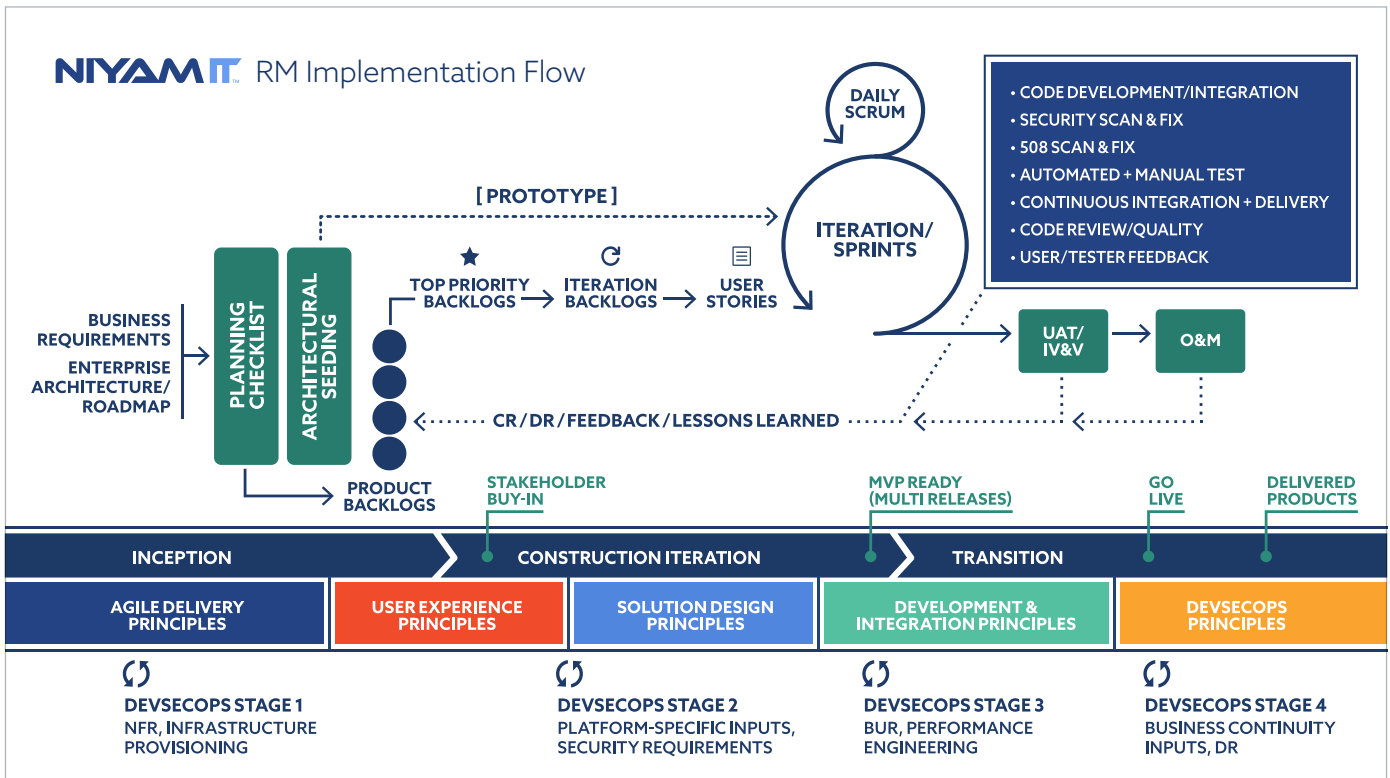
architecture and infrastructure. Adhere to the Technical Reference Model (TRM), a list of allowable COTS, GOTS and Open Source that can be deployed.

- Make certain that the system fits into existing environments and reflect future vision.
- Maximize reuse of components and services. These APIs and services should be field-tested, scalable for high volume users, and continually evolving to support the latest standards and regulations.
- Ensure that the chosen platform and tools comply with FISMA and FedRAMP for systems deployable on 'Federal Cloud.'

CONSTRUCTION ITERATION

UX, Solution Design Principles, and DevOps Stage 2 and 3

The epics and user stories identified by analysts constitute the product backlog. Prioritized items from the Value Effort





Matrix starting with low-effort high-value are loaded into the iteration backlog. The process repeats for each iteration.

Steps involved in every iteration:

- Establishing acceptance criteria with SMEs, stakeholders, and IPT, with help of screens or mock APIs without true functionality
- Developing an accepted prototype
- Performing automated unit testing and integration testing
- Performing regular code review

Architects plan for disaster recovery and risk mitigation at this stage. Developers have the entire code- base running locally. Full nightly builds on data and code in integration environment are captured and subject to automated test cycles. Depending upon the sensitivity of business operations, every piece of infrastructure in the delivery pipeline is provisioned in the cloud environment.

TRANSITION ITERATION AND DELIVERY

Development and Integration Principles and DevOps Stage 4

Team velocity is measured in each iteration, and feedback from retrospective meetings is incorporated. The committed backlog items are loaded into burndown charts and tracked automatically as developers update the completion of

stories. Issue management, collaboration, automated builds, and pipeline execution are entrusted to a delivery tool.

Operations Checklist:

- Teams are assigned delivery responsibility along with support operations
- Issues are tracked to completion in ticketed CR/DR system
- The CR/DR system is fully integrated with the development tool for traceability
- Compliance is incorporated with client software review gates
- Operational systems are managed under Application life cycle management principles
- Dedicated team roles assigned to manage user manuals, training assets, social communications, and outages

Security and Compliance are not add-ons or afterthoughts, but are implemented throughout the delivery pipeline.

Steps to Continuous Security Implementation in RM:

- Adopt secure coding practices (SEI CMU)
- Develop processes to deploy automated tools for security testing at different stages of the delivery pipeline for traceability and repeatability
- Automated security testing tools cover predetermined attacks, scan for common vulnerabilities, and enable writing of custom attacks for each application
- Perform security tests in the staging environment

Section 3: Key Indicators

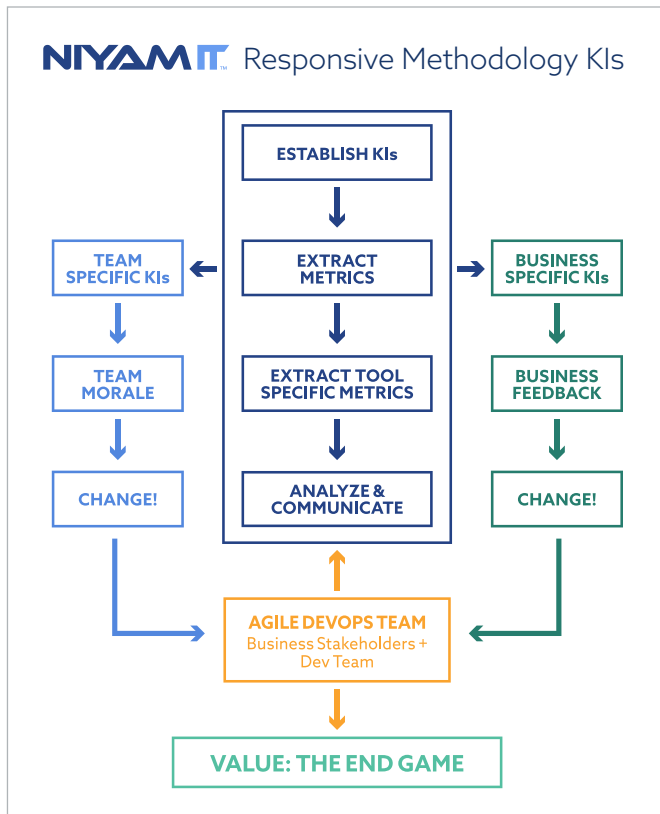
Agile projects involve a moving target. Some projects may never finish, given the emphasis on the 'continuous' process. Monitoring agile projects involves documentation of user stories, acceptance criteria, tasks to complete in fulfilling the "Definition of Done," and code quality/standards compliance.

In the Federal IT environment - where a shift is taking place from traditional waterfall SDLC to Agile/DevOps - reporting is a gray area for the PMO. With the shift to Agile, there is a need to address the reporting gap to provide meaningful metrics to all stakeholders for processes and delivery.

Responsive Methodology seeks to address this gap by highlighting key indicators at critical points in the Agile Dev cycles and enabling the appropriate 'response' to indicators, providing effective tracking for generative Agile processes.

RM key indicators enable:

- Frequent stakeholder feedback
- Gaining trust by demonstrating value after each iteration
- Tracking progress visibly



STEP ONE

Extract Tool Specific Metrics: What to Measure

This step involves establishing key indicators and scheduling extraction of metrics for analysis and communication. The metrics for extraction (tool specific) are:

- Sprint Burndown Chart
- Epic and Release Burndown Chart
- Velocity Chart
- Control Chart
- Cumulative Flow Diagram

STEP TWO

Analysis and Distribution: For Whom

Data analysis obtained from metrics involves interpretation of key indicators and classifying them into business-, stakeholder-, and agile team-specific KIs.

KIs for Business/Stakeholders:

1. Change of scope mid sprint
2. Scope grows faster
3. Business pressure
4. Funding pressure
5. Definition of done
6. Resolution of blockers
7. Customer support requests
8. Technical debt

Response to these KIs from the business/stakeholder unit should result in change, in the direction of improvement to implementation by the Agile-DevOps-Stakeholder team.

KIs for Agile Dev Team:

1. Time taken for team to finish sprint
2. Granularity of work breakdown
3. Scope creep
4. Progress made over several iterations
5. Release frequency, delivery speed
6. Unforeseen challenges to project estimates
7. Time taken from "In-progress" to "done"
8. Backlog growth
9. Automated testing coverage

Authors and Contributors

Niyam Agile/DevSecOps Team

Shweta Katre skatre@niyamit.com

Suman Biswas sbiswas@niyamit.com

Joseph Satchavarodom joseph@niyamit.com

References

Agile Project Management

Jim Highsmith, TechFAR Handbook

U.S Digital Services Playbook

The Missing Metrics of DevOps

www.insights.sei.cmu.edu

Secure Coding Publications

The CERT Division

www.cert.org

Adding Security to Your DevOps Pipeline

www.insights.sei.cmu.edu

Scaled Agile Framework (SAFe)

The Practices of Agile Modeling

Scott Wambler, IBM Disciplined Agile Delivery

Why Niyam?

Since our founding in 2007, Niyam has developed an impressive record of successful outcomes, earning our status as the go-to provider of innovative IT solutions. Today, Niyam sits at the vanguard of the industry, delivering state-of-the-art, mission-critical technologies to federal agencies and the public marketplace. Our solutions are proven to accelerate collaboration, increase efficiency, and consistently provide rapid, breakthrough results – all while remaining conscious of shifting timelines and budgets.

Our Technology Partners



Learn More

For more information on this topic, or to learn about our full range of Agile/DevSecOps capabilities, contact us at 703-429-2450 or email agile@niyamit.com.



PRIMARY HEADQUARTERS 202 Church St. SE, Suite 301 Leesburg, VA 20175

SOLUTION DEVELOPMENT CENTER 10201 Fairfax Blvd., Suite 224 Fairfax, VA 22030

+1 703.429.2450 info@niyamit.com www.niyamit.com

Schedule 70: GS-35F-144DA

SBA Certified 8(a) | SBA Certified HUBZone

CMMI DEV LVL 3

ISO 9001:2015 | ISO/IEC 20000-1:2018 | ISO/IEC 27001:2013

EFFECTIVE SOLUTIONS. MEASURABLE OUTCOMES.