

GUIDE

A Guide to Agile Application Lifecycle Management (ALM)

What Is Agile Application Lifecycle Management?

What Is ALM?

Application Lifecycle Management (ALM) describes software's governance, development, and maintenance throughout its lifecycle. ALM supports multiple software development and systems engineering methodologies ranging from Waterfall to Agile.

What Is Agile?

Agile software development emphasizes short development cycles, responsiveness to changing requirements, the incorporation of stakeholder feedback, team collaboration, and continuous application improvement.

There are numerous ways to implement Agile software development, but most implementations contain sprints and Program Increments (PIs). Sprints are the lowest level of iteration within Agile. Individual sprints have narrow goals and typically range in duration from one to four weeks. PIs are larger iterations involving multiple teams coordinating to provide key

features. PIs are generally broken down into four or five sprints. The relationship between Agile sprints and PIs is shown in Figure 1.

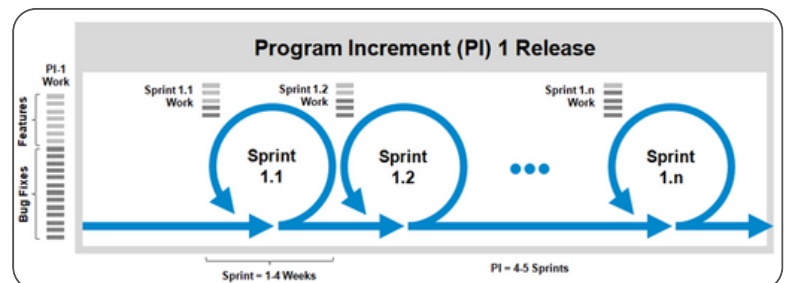


Figure 1. Agile Sprint to PI Relationship

The Agile ALM approach described in this document is best applied for Program Increment (PI) level releases, but it can be tailored to other release durations.

Agile ALM Tools Ecosystem

SPEC Innovations supplements an Innoslate-based Digital Engineering Environment (DEE) with other ALM tools to implement its approach to Agile ALM. The tools comprising the Innoslate-based Agile ALM ecosystem and their primary functions are shown in Table 1.

Tool	Purpose
Innoslate	Requirements Management, System Modeling, Model Simulation, Test Management, Program Management, Data Repository
GitHub	Application Development, Application Deployment
Selenium	Automated Testing
Sopatra	Operational Analysis

Table 1. Innoslate-Based Agile ALM Ecosystem

Innoslate

Innoslate is a full-lifecycle, Model-Based Systems Engineering (MBSE) tool that acts as the central digital engineering data repository within the DEE. It provides high-level technical and programmatic functions in support of the Agile ALM.

GitHub

In addition to performing its primary functions of application development and deployment, GitHub exchanges data with

Innoslate. Innoslate's GitHub View provides Innoslate users with limited access to GitHub repositories. Innoslate and GitHub can also be configured to synchronize issue data (i.e., GitHub Issues and Innoslate Issue class entities).

Selenium

Integrated with the JUnit test framework, Selenium is used for automated code testing. Innoslate Java SDK code is executed from within JUnit tests to update Innoslate Test Suites with the automated test results.

Sopatra

Sopatra, a SPEC Innovations tool for modeling and analyzing Standard Operating Procedures (SOPs), is used to evaluate user employment of deployed application releases and inform future requirements.

The Innoslate-based Agile ALM ecosystem can be customized with other tools (e.g., a development environment other than GitHub). The cost-benefit analysis for such modifications should account for the effort required to fully integrate these tools or the lost capabilities by failing to do so.

The Agile ALM Stages

SPEC Innovations' approach to Agile ALM is made up of five stages:

1. **Requirement Specification**
2. **Development**
3. **Testing and Quality Assurance**
4. **Deployment**
5. **Operations and Maintenance**

Additionally, Program Management occurs concurrently with the Agile ALM stages. The Agile ALM stages with their associated ecosystem tools are shown in Figure 2, described in the following subsections, and summarized in Appendix A.



Figure 2. Innoslate-Based Agile ALM Stages

The Agile ALM stages are executed sequentially in a loop, starting with the Requirement Specification stage. The loop terminates when the application is retired from use.

Multiple releases will often progress concurrently through different Agile ALM stages. For example, Release 1 may be in Operations and Maintenance, Release 2 is in Test and Quality Assurance, Release 3 is in Development, and Release 4 is in Requirement Specification.

Requirement Specification Stage

The Requirement Specification stage determines features and defect corrections to be implemented in the next release. Within SPEC Innovations' approach to Agile ALM, the following activities occur as part of this stage:

- **Application design documented** and/or updated using Action Diagrams.
- **Design completeness assessed** through Discrete Event Simulator runs.
- **Requirements (i.e., features) captured** and tagged with release labels in Innoslate's Documents View.

- **Requirement specification values determined** or supported by Monte Carlo Simulator runs.
- **Release goals established** for defects identified during the Operations and Maintenance stage; defects tagged with release labels.

Development Stage

The Development stage involves implementing software features and defect corrections identified in the Requirement Specification stage. Within SPEC Innovations' approach to Agile ALM, the following activities occur as part of this stage:

- **Features captured as issues;** either maintained in GitHub or synchronized between GitHub and Innoslate.
- **Application software feature and defect corrections** developed within GitHub.
- **Issues discovered, edited, or commented** upon from within GitHub (developers) or through Innoslate's GitHub Issues View (non-developers).
- **Development backlog, status, and assignments monitored** on Kanban Boards.
- **Productivity insights assessed** by viewing commits activity and pull requests.

Testing and Quality Assurance Stage

The Testing and Quality Assurance stage verifies that the application implements the features and defect corrections identified in the Requirement Specification stage. Preparatory activities for this stage may occur concurrently with the Development stage and include the following:

- **Test plan developed** in Innoslate's Documents View.
- **Test procedures developed** in Innoslate's Test Suite.
- **Automated application testing scripts created** in Selenium with JUnit.

Within SPEC Innovations' approach to Agile ALM, the following activities occur as part of this stage:

- **Test plans updated** in Innoslate's Documents View.
- **Test procedures updated** in Innoslate's Test Suite.
- **Manual tests conducted** in accordance with test procedures.
- **Automated application testing conducted** through Selenium.
- **Manual and automated test results captured** in Innoslate's Test Suite.
- **Reports generated** to document Test Case status and traceability.

Deployment Stage

The Deployment stage consists of issuing the application release to the target environment. Within SPEC Innovations' approach to Agile ALM, the following activities occur as part of this stage:

- **Application release deployed** with GitHub.

Operations and Maintenance Stage

The Operations and Maintenance (O&M) stage consists of monitoring, optimizing, and identifying issues with the deployed release. Within SPEC Innovations' approach to Agile ALM, the following activities occur as part of this stage:

- **User issues (i.e., software defects) discovered, edited, or commented** upon through Innoslate's GitHub View.
- **Operational analysis conducted** in Sopatra to inform future requirements.

Program Management Stage

Program Management consists of managing technical and programmatic aspects of the Agile ALM. Within SPEC Innovations' approach to Agile ALM, the following activities occur as part of Program Management:

- **Programmatic information** is maintained on Innoslate Project Dashboards to provide context for systems engineers, developers, testers, specialized support staff, and external stakeholders.
- **Development schedules** are tracked using Innoslate Timeline Diagrams or Gantt Charts; release details are viewable on development Kanban Boards.
- **Risk identification** and analysis are captured in Innoslate Risk Diagrams.
- **Risk mitigations** are maintained in Innoslate Risk Burn-Down Charts.
- **Collaborative** capabilities aid coordination and deconfliction.

Conclusion

SPEC Innovations' approach to Agile ALM consists of five stages plus Program Management. Innoslate, supplemented with other tools in the Innoslate-based Agile ALM

ecosystem, provides a tailorable mechanism to implement Agile methodologies throughout an application's lifecycle.

Appendix A

Innoslate-based Agile ALM stage activities with supporting ecosystem features:

Innoslate-Based Agile ALM Stage Activities With Supporting Ecosystem Features (X = Mandatory, O = Optional)		Innoslate-Based Agile ALM Ecosystem										
		Innoslate Collaborative Features	Innoslate Risk Management	Innoslate Timeline/Gantt Chart	Innoslate Project Dashboard	Innoslate Database View	Innoslate Documents View	Innoslate Test Center	Innoslate Modeling Capabilities	Innoslate Simulator	Github	Selenium
Requirement Specification												
Application Design	O					X						
Design Completeness Assessment							X					
Requirement (Feature) Capture	O			X								
Requirement Specification Value Determination							X					
Release Defect Goal Determination				O				X	X			
Development												
Features Captured as Issues				O				X	X			
Software Development									X			
Issue Updates								X	X			
Development Monitoring								X				
Productivity Insights Assessed								X				
Testing and Quality Assurance												
Test Plan Development				X								
Test Procedure Development	O				X							
Automated Application Test Script Creation											X	
Test Plan Updates				X								
Test Procedure Updates	O				X							
Automated Application Testing											X	
Test Results Capture					X							
Test Report Generation					X							
Deployment												
Application Release										X		
Operations and Maintenance												
Defects Identified or Updated				O				X	X			
Operational Analysis												X
Program Management												
Programmatic Information Posted				X								
Development Schedule Tracked			X					X				
Risk Identification and Analysis		X										
Risk Mitigation		X										
Collaboration	X											

