



Select *Computing*
I N C O R P O R A T E D

Systems Modernization Strategies

Best Practices for Systems Modernization

Presented to:

Medicaid Enterprise Systems Conference (MESC 2017)

Date: August 2017

Select Computing, Inc.

9841 Broken Land Parkway

Suite 209

Columbia, MD 21046

www.selectcomputing.com

© 2017, Select Computing, Inc. All rights reserved.

Systems Modernization Strategies

BEST PRACTICES FOR SYSTEMS MODERNIZATION

INTRODUCTION

The need for systems modernization is evident in the face of an ever-changing IT landscape and emerging business needs. Within the healthcare industry, the requirement for systems modernization was evident even before the Affordable Care Act (ACA) became law. Today, additional business impetus such as the Medicaid Information Technology Architecture (MITA), continue to drive the need to modernize systems. From our experience performing systems modernization projects for various Federal agencies (Commerce, HHS, VA) and the State of Minnesota, we identify three broad approaches to modernization: process-driven, data-driven, and codebase reconditioning.

Process-driven modernization is a top-down business architecture re-engineering effort. Organizations assess their current state, identify their desired state, and map the gap. Successful process-driven projects utilize architecture frameworks such as The Open Group Architecture Framework (TOGAF):

<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>

Data-driven modernization is a bottom-up approach. It centers around the creation of a Business Information Repository that serves as a data clearing house for the organization. Since the Repository is independent of existing systems, it does not interfere with ongoing operations. Legacy systems feed data that is mapped to a canonical data model that reflects current and future needs. Modernization projects can then proceed by integrating themselves with the Repository, thereby restoring organizational agility by decoupling systems.

Both approaches are valid and have distinct risks and benefits. It is common for modernization programs to utilize both approaches at different points in time. In this white paper, we will explore process-driven and data-driven modernization concepts. We will provide sufficient insight and tools to enable organizations to decide which approach is appropriate for their respective circumstances.

Modernization is the ideal time to embrace key technologies trends within the organization; specifically, the Cloud, Service Oriented Architecture, and Open Source. Introducing the Cloud lowers the entry cost of technologies and provides a well-established path for scaling up. Service Oriented Architecture enables concepts such as microservices and containerization that provide the foundation for Agile IT operations and DEVOPS. Open Source lowers operating costs by eliminating the need to license proprietary technology. During our white paper, we will discuss methods we have applied to incorporate these trends into our customers' modernization initiatives. The third strategy, Codebase Reconditioning, focuses on these types of improvements.

Modernization programs benefit from Agile because it allows the organization to focus its resources on the projects that return the best immediate value. There are several Agile scaling frameworks such as LeSS, DAD,

and SAFe that help organize and govern programs. We will discuss the pros and cons of these frameworks as they relate to modernization.

A solution provider's capability to execute the project and deliver value is best predicted by its CMMI appraised maturity level. We discuss the relationship between project problems, Agile, and CMMI capability.

STRATEGIES

The system modernization strategies are not mutually exclusive. They can be blended across a modernization project or program.

Code Reconditioning

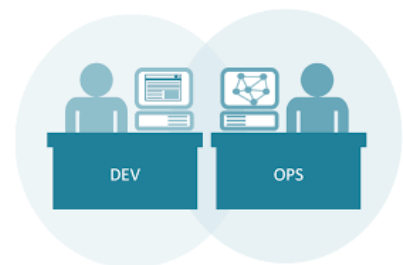
We begin with the code reconditioning strategy. The strategy affects the software codebase and is not intended to significantly change how the business operates. Often, the key drivers for this approach is the reduction or elimination of the maintenance debt burden, and, the reduction of the total cost of ownership. The strategy decomposes into several tactics.

Codebase Purification

Codebase purification involves analyzing and documenting the application programs, jobs, files, databases, and other artifacts. The objective is to identify and remove obsolete and redundant processing to eliminate waste. The codebase is refactored to isolate reusable functions and business rules into separate modules. The result is a lean codebase.

DEVOPS Enablement

A logical next step after codebase purification is to enable the codebase for DEVOPS. Given the lean codebase, we begin by creating a regression test suite to exercise all business rules and logical paths within the application. With regression testing established, the IT organization can implement a continuous integration or a continuous delivery approach (see Figure 1). In continuous integration, a software module is automatically tested as soon as the software developer attempts to check it into the configuration management system. If the tests fail, the module is not accepted and the developer is notified of the problems. In continuous delivery, a module that passes its tests is deployed into production as soon as possible.



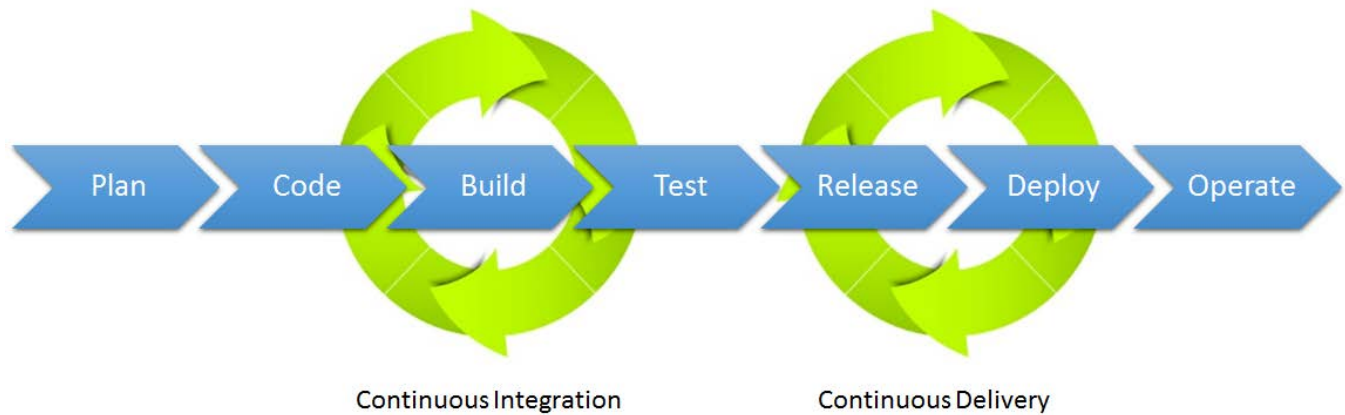


Figure 1. Continuous Integration and Continuous Delivery Approaches

The benefit of continuous integration is substantially improved software quality, with fewer production outages, and a significant increase in the productivity of the software development and maintenance organization. The benefit of continuous delivery is substantially reduced time-to-market for the organization.

Codebase Translation

Codebase translation converts the code base from one language to another. For example, the COBOL code base is translated into JAVA. While codebase purification is a prerequisite for DevOPS enablement, codebase translation should occur after codebase purification and before DevOPS enablement.



The benefit of codebase translation is to move the application from an older language into a newer language where more resources are available.

Codebase Porting

Codebase porting changes the underlying platform of the codebase. For example, the codebase moves from the mainframe to .NET or Linux. Codebase translation is not a prerequisite for this strategy. For example, COBOL applications can run on Windows or Linux or UNIX, in-house or in the Cloud.



The benefit of codebase porting is the reduction of total cost of ownership, without changing the necessary skillset of the software development and maintenance organization.

SOA Enablement

SOA enablement introduces the software infrastructure necessary to implement a service-oriented architecture. The objective may be to introduce a service bus, or microservices, or web services, for example. Code purification is the recommended prerequisite.

The benefit of SOA enablement is to make the legacy codebase accessible and open to new integration possibilities, thus extending the useful life of the legacy application while accelerating time-to-market for the organization.



Process-Driven

The process driven strategy dramatically impacts the organization outside of IT. It is suitable when the organization wants to change its business processes and automate the new business processes, while leveraging the investment in legacy applications. We analyze your business environment and work with your subject matter experts and business leaders to produce an enterprise architecture for the organization. We apply a methodology such as TOGAF. The business architecture drives the modernization program; introducing new technology, replacing applications with new custom or off-the-shelf solutions, and applying the codebase reconditioning strategy where appropriate.

By employing TOGAF's CMMI-based Business Transformation Readiness Assessment, Capability-Based Planning, and FEA's Business Reference Model, we analyze the organization to provide a feasible and sustainable plan for a complete Strategic Technology Plan that is driving operational and technical development forward to meet the newly re-engineered business processes (see Figure 2). This analysis was based on current capabilities of the organization and includes a plan for migration and implementation of automated technology driven processes. These are the primary processes which drive the operational mission success of the organization. The Strategic Technology Plan is segmented into 1-year, 3-year, and 5-year plans, all of which project the strategic vision forward in the outlying years.

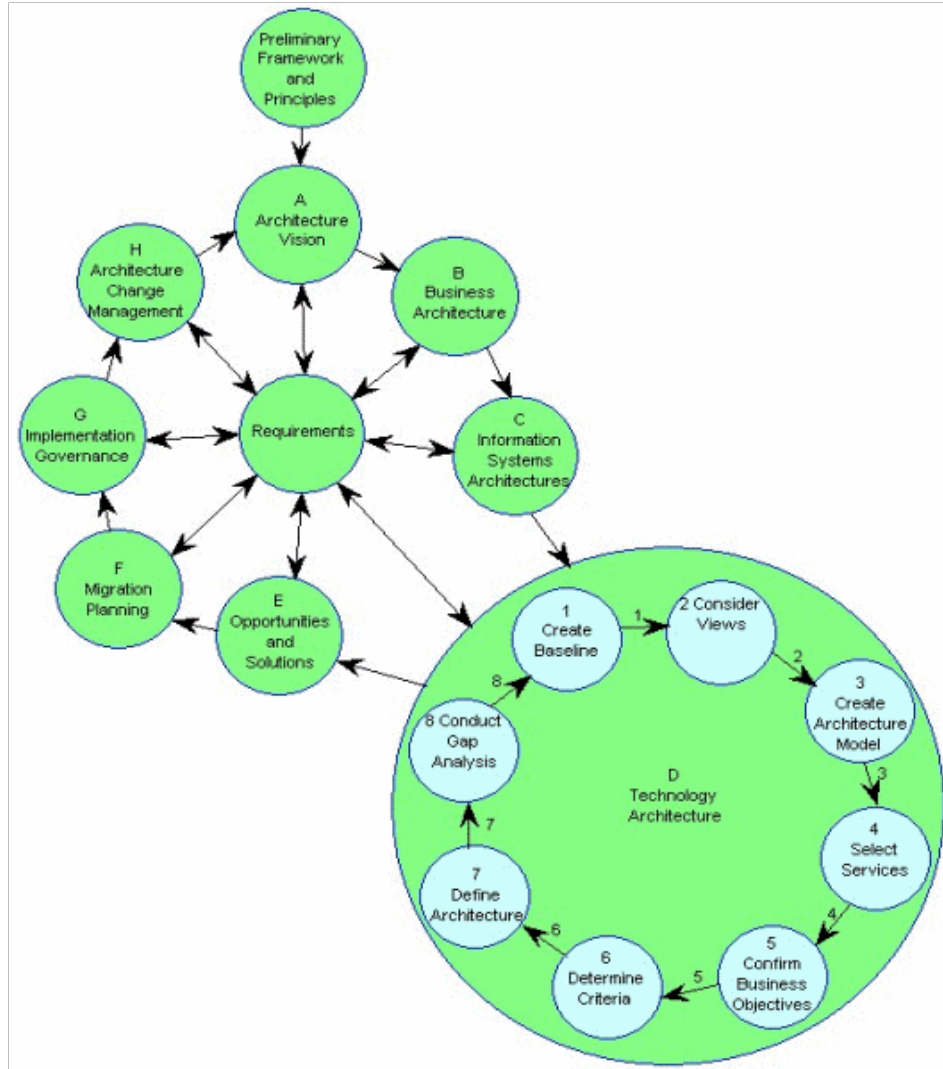
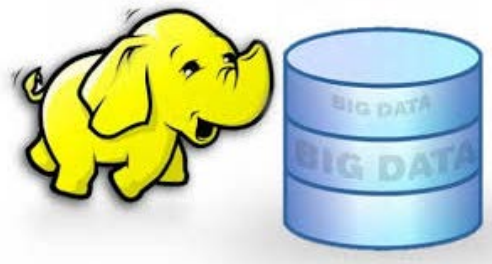


Figure 2. TOGAF Technology Architecture (The Open Group)

Data-Driven

The data-driven strategy does not immediately impact the organization. It can be thought of as an interim step of the process-driven strategy. It allows the organization to discover what its business architecture should be while receiving significant decision-making benefits.

As its name indicates, it focuses on the data the organization uses. At the core of the strategy is a corporate data store implemented using a canonical data model. The data store is normalized and it can support longitudinal views of data. It is not a data warehouse. A technology such as Cloudera Hadoop is ideal.



The data store can be fed using an SOA infrastructure that provides adapters for the legacy applications. These adapters read the legacy files and databases and transform the data into the canonical model for insertion into the corporate data store. With the data store in place, the organization's data maturity can be raised without disrupting operations. Again, as maturity rises, the appropriate modernization strategy can be applied to improve the business.

The characteristics the organization improves are:

- **Storage:** Where is the data stored?
- **Service:** Is data available through a service?
- **Governance:** Are data stewards managing the corporate stores with executive support?
- **Quality:** Are business rules known and appropriately applied to the data? Are defects (discrepancies between the corporate data store and the legacy systems) formally tracked and resolved?
- **Golden Record Identification:** Is duplicate data known? Are identifiers consistent across the corporate data store?
- **Time Phasing:** What is the delay of the appearance of data into the corporate data store? The ideal is no delay between the operational system and the corporate data store.

The following table describes the maturity levels the organization goes through. It is not necessary to reach all of the maturity levels; however, all prior levels are required.

| Maturity | Description |
|------------------|---|
| <i>Initial</i> | The data store is in place and is populated. Data reporting and visualization technologies are deployed. |
| <i>Managed</i> | The organization has a data stewardship council in place. The council consists of designated data stewards who manage their data subject areas within the corporate canonical data model. They insure the integrity of the corporate data store by identifying attribute characteristics and the business rules that apply to the data. They establish and enforce data standards in the corporate data store. Master data management principles are implemented and the master data plan is known. |
| <i>Defined</i> | Data dictionaries and data views are available and accessible to the organization based on roles. The master data plan is executed. Discrepancies between legacy data and corporate data store data are formally tracked and resolved. All information silos are integrated into the corporate data store. All data within the corporate data store complies with the business rules. |
| <i>Proactive</i> | The corporate data store drives and corrects information into the legacy systems. The adapters become bidirectional. Now, it is possible to replace legacy systems and integrate the replacement with the corporate data stores. Each system integrates with the corporate data store and organization-wide systems integration occurs |

| Maturity | Description |
|-------------------|--|
| <i>Innovative</i> | <p>through the corporate data store. The corporate data store publishes enterprise data services.</p> <p>The corporate data store drives innovation in the organization. The corporate data store is the master data store for the organization.</p> |

By focusing on the data, the data-driven modernization strategy enables the organization to focus on systems that do not meet its requirements or support its strategy. It provides a consistent and correct source of data for decision making. Eventually, the organization’s reporting and decision-making processes center on the corporate data store and less on the operational systems. Thus, the organization can reengineer itself around its data.

AGILE FRAMEWORKS

Modernization programs do not require Agile. However, modernization programs benefit from Agile in the same manner development programs do. They require the same flexibility and agility to respond to newly discovered requirements, functionality, and changes in scope. We strongly recommend using an Agile approach in modernization.

Modernization programs involve several concurrent projects that must be coordinated. There are three major Agile scaling frameworks; Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD), and Large Scale Scrum (LeSS). The frameworks are needed because Agile practices are defined at the project level, not at the program or portfolio level. How, then, can Agile principles be applied across multiple projects? How can multiple projects coordinate when each has the flexibility to change direction? Scrum is the project methodology. The question is how to scale Scrum up to a program or portfolio.

SAFe operates at the team (project), program, and portfolio levels (see Figure 3). The team level prescription is essentially Scrum, augmented with eXtreme Programming, Kanban, or other techniques. At the program level, SAFe is very prescriptive about aligning the teams with the organizational goals along the Agile Release Train. At the portfolio level, SAFe prescribes how to align goals and investments. The result is that management constrains the team’s agility.

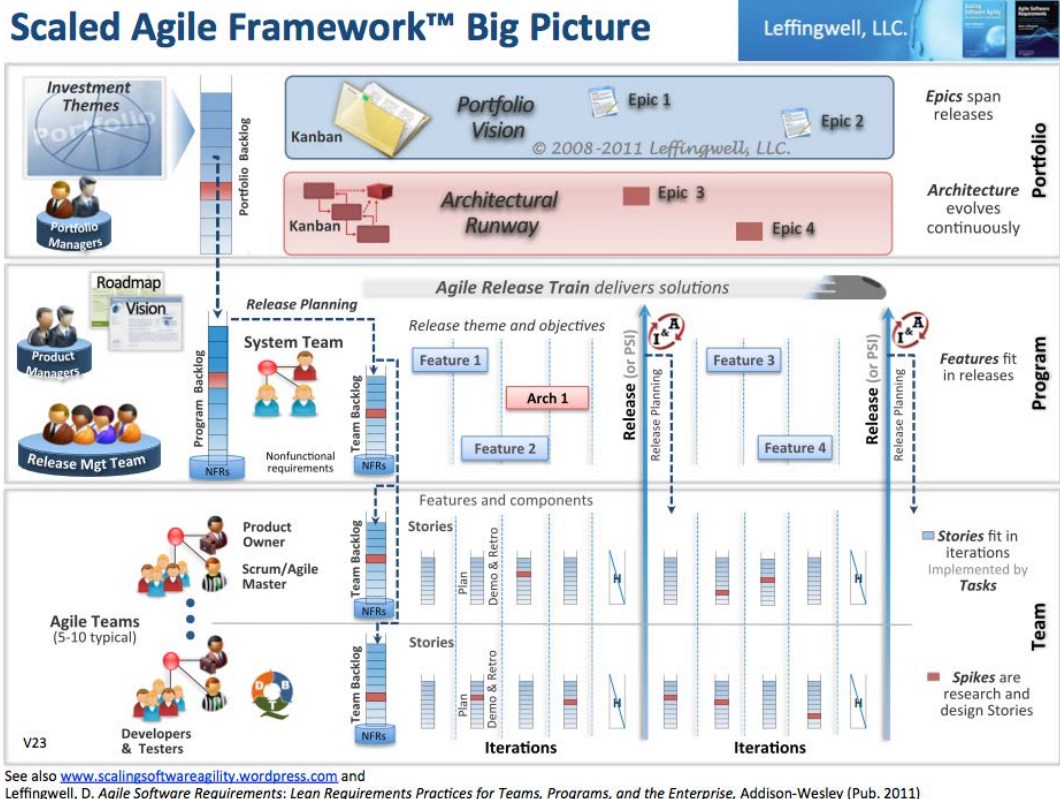


Figure 3. Scaled Agile Framework Big Picture™ (Leffingwell, LLC)

As hybrid approach, DAD blends other approaches (see Figure 4). For example, a project has three phases: Inception, Construction, and Transition. This structure is closer to the PMBoK. DAD introduces and redefines traditional Scrum roles. In particular, the Scrum Master is the Team Lead. A new role, Architecture Owner, reflects the importance DAD places on the architecture and on managing technical risk. DAD reflects its unified process roots making it easier for organization to adopt.

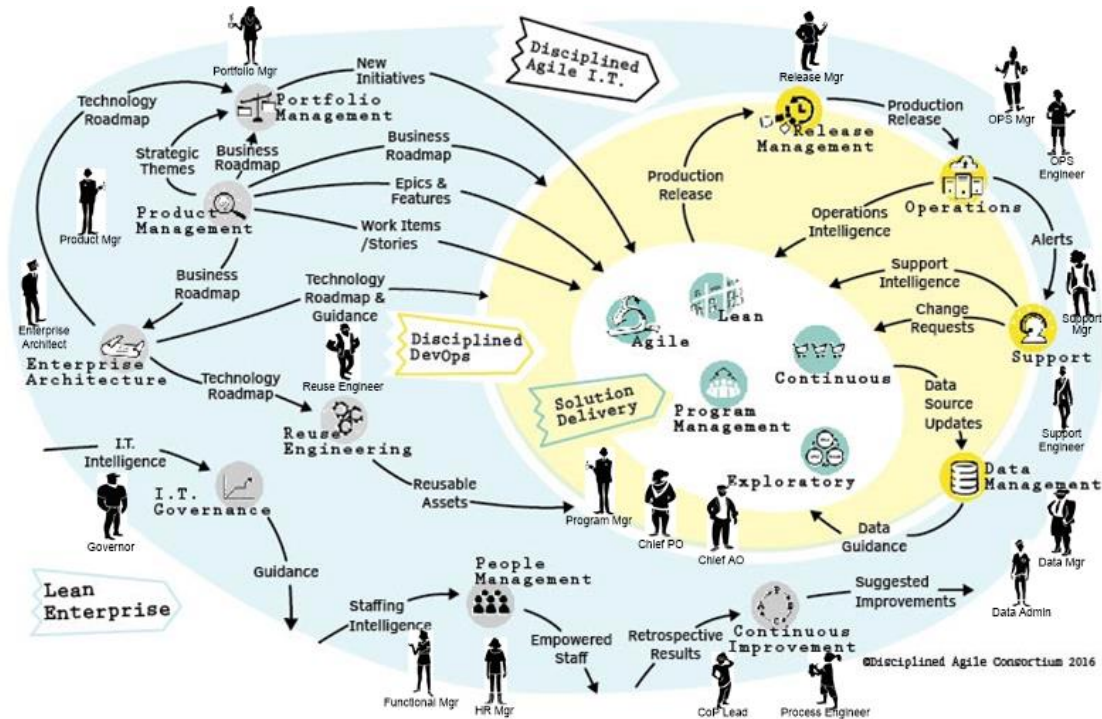


Figure 4. Disciplined Agile Delivery (Disciplined Agile Consortium 2016)

LeSS is an application of Scrum to large projects (see Figure 5). The approach distinguishes between teams of ten and teams with more than ten members. The basic Scrum roles remain unchanged, however, LeSS introduces new ceremonies that reflect the decomposition of the product. A new role, the Area Product Owner, is focused on product ownership responsibility for a major portion of the solution, which is further decomposed. As a result, the additional ceremonies mirror the planning and retrospective meetings at this higher level.

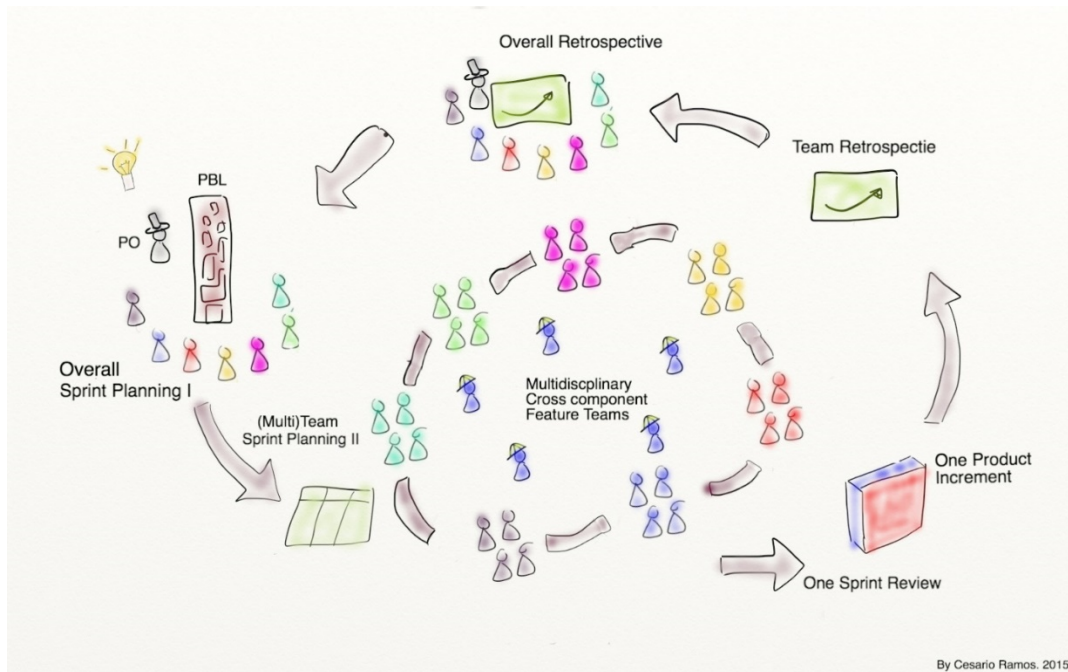


Figure 5. LeSS Framework (Cesario Ramos 2015)

In terms of the modernization strategies, SAFe works best for an organization applying the process-driven strategy. DAD is the best framework for the data-driven strategy. LeSS is the best framework for code reconditioning.

IMPACT OF CMMI MATURITY LEVELS

The **Capability Maturity Model Integration (CMMI)** is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organizational functions, set process improvement goals and priorities, guide quality processes, and provide a point of reference for appraising current processes.

Both Agile and CMMI are based on the Deming (**Plan-Do-Check [or Study]-Act**) Cycle and therefore share several values. In the simplest terms, Agile methodologies specify **HOW** things should be done while CMMI specifies **WHAT** should be done. The apparent conflict between CMMI and Agile stems from Agile's focus on deliverables that contribute directly to the product. Deliverables that do not contribute directly to the product are suspected waste. The best perspective from which to address the apparent conflict is risk management. Projects and programs should strive to deliver the product in a manner that minimizes risk to the project, program, and organization.

Managing Risk in Agile

Managing risk at the organizational level is critical and more difficult when different components use different Agile scaling frameworks. CMMI provides the answer through quantitative project management. The

Information Systems Audit and Control Association (ISACA) defines the Risk IT management framework as shown in Figure 6.

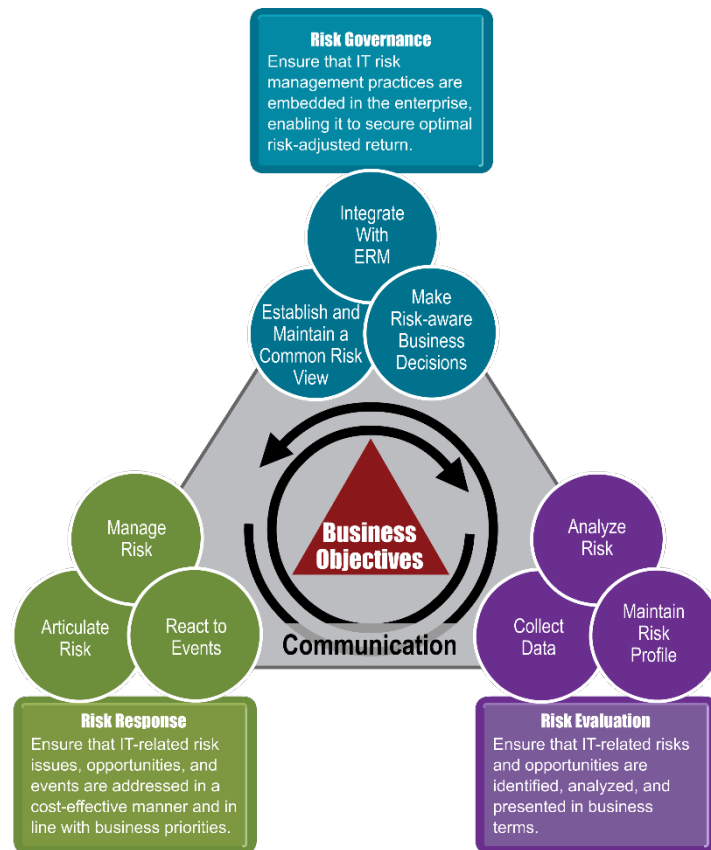


Figure 6. Risk IT Framework for Management of IT Related Business Risks (ISACA)

Below are the key CMMI process areas that appear to be waste but enable effective risk management (the CMMI maturity level, *ML*, is in parentheses):

Quantitative Project Management (QPM) (ML 4). Statistically demonstrate that the project will achieve its established quality and performance objectives. How can we believe the velocity metric of a given project? How comparable is it to the velocity of other projects? All projects operate with the constraints of the organization. Each organization has a normal velocity range. Burndown and velocity reflect process performance of the particular team. However, they do not reflect activity cost performance. Mitigating project risks requires that we compare project performance against the norms of the organization to determine whether the project is on track. Earned value is quite useful. Even for Agile projects, the questions remain: ***Has the value the project has produced so far commensurate with the costs to date? What will be the relationship between costs and value in the future?*** When implementing Agile at a large scale, managing the risk depends on solid measurements. While the measurements do not contribute to the product directly, they enable the organization to predict and detect when the project or program may veer off course.

Organizational Process Performance (OPP) (ML 4). Analyze organization performance and improve to close performance gaps. Use measurements to improve the organization's capability to execute the projects. From an Agile perspective, OPP is an organization-wide retrospective that should be focused on removing process waste. Having an organization-wide retrospective is as useful at the organizational level as it is at the team level. However, its effectiveness is a function of shared language and benchmarks across the organization. The implementation of Quantitative Project Management enables the necessary foundation for process improvement.

Risk Management (ML 3). Identify potential issues that can be prevented or mitigated. Risk identification, classification, and mitigation is built into most Agile methodologies. The focus tends to be on risks emanating from change since the risk of not meeting the requirements is low. Otherwise, there is not a focus on managing risk to organization objectives, such as financial performance for example. Agile risk management is inwardly focused as if the project is outside the context of the organization. Organizational governance is still necessary to protect organizational objectives. This focus on organizational objectives enables ML 4 Organization Process Performance.

Decision Analysis and Resolution (ML 3). Use a formal process for analyzing alternatives where selection criteria are defined and understood. The analysis of alternatives is important in Agile, for example, in estimating story points. In fact, decision making in Agile, by its collaborative nature, is much more complex than in waterfall projects. It is important not to leave the definition of selection criteria to the project team to make sure organizational objectives are represented within the criteria. For example, certain criteria or their definition can be made standard. Beginning at ML 3, the whole organization benefits. Formal criteria in decision making provide the foundation for effective ML 3 Risk Management.

Measurement and Analysis (ML 2). Ensure that sufficient measurement capability exists to make informed management decisions. Crucially, one of these management decisions is the prioritization of the backlog which relies on consistent point valuation of stories, and on the prioritization of the impediments list. This capability is leveraged in ML 3 Decision Analysis and Resolution.

Project Monitoring and Control (ML 2). Understand the actual project progress and correct deviations from objectives and schedules. In Scrum, the Scrum stand-ups and the burndown chart clearly exist for this purpose. The deviations at the project level begin risk identification that rolls up to organizational risk evaluation.

Process and Product Quality Assurance (ML 2). Evaluate processes and associated work products to gain insight in their quality. CMMI's emphasis is on processes and intermediate work products, more so than on the product. In Agile, the Scrum Master or the eXtreme Programming (XP) Coach are responsible for ensuring the team follow the methodology and are producing the intermediate deliverables the methodology requires.

When we examine the CMMI process areas on large-scale Agile implementation, we see that CMMI and Agile have much in common, due to their common foundation of the Deming Cycle. From the perspective of the Risk IT framework, we see that ML 2 process areas formalize Risk Response, ML 3 formalizes Risk Evaluation, and ML 4 formalizes Risk Governance. Agile frameworks rely on the Product Owners or Area Product Owners to reflect and represent Risk Governance within the project or program. As a result, it is critical that Product Owners

follow a common Risk Governance approach that pervades throughout projects, programs, and the organization. By applying CMMI, the organization can manage risk.

Improving Agile Performance with CMMI

The following table, adapted from CMMIInstitute.com, maps some common business problems where Agile and CMMI provide guidance for improvement. In the first column, the table identifies types of problems organizations face running projects. In the second column, the table identifies how Agile Scrum addresses the problem. The third column cross-references the CMMI process area that addresses the necessary capabilities the organization needs to apply the Scrum techniques effectively. An organization uses the CMMI Process Areas to strengthen the ability to perform the Agile ceremonies/ techniques to ensure scalability and resilience.

| Problems | Agile Ceremonies/ Techniques | CMMI Process Areas (Maturity Level) |
|--|--|---|
| <i>Project estimates are unrealistic or unknown.</i> | <ul style="list-style-type: none"> ▪ Team Estimating Game ▪ Planning Poker ▪ Sprint Planning ▪ Backlog Grooming ▪ Requirements Development ▪ Task Estimation ▪ Release Planning ▪ Sprint Backlog | <ul style="list-style-type: none"> ▪ Project Planning (ML 2) ▪ Integrated Project Management (ML 3) ▪ Requirements Management & Development (ML 2 and ML 3) ▪ Measurement and Analysis (ML 2) ▪ Quantitative Project Management (ML 4) |
| <i>Projects do not get delivered on schedule.</i> | <ul style="list-style-type: none"> ▪ Daily Standup/ Daily Scrum ▪ Release Burndown ▪ Sprint Burndown ▪ Task Estimation ▪ Release on Demand ▪ Incremental Release | <ul style="list-style-type: none"> ▪ Project Monitoring and Control (ML 2) ▪ Measurement and Analysis (ML 2) ▪ Quantitative Project Management (ML 4) |
| <i>Requirements are always changing.</i> | <ul style="list-style-type: none"> ▪ Backlog Grooming ▪ Sprint Planning ▪ Product Backlog ▪ User Stories/ Epics ▪ Definition of Done (for User Stories) ▪ Top “10” Features ▪ Release Planning | <ul style="list-style-type: none"> ▪ Requirements Management (ML 2) ▪ Project Management & Control (ML 2) ▪ Requirements Development (ML 3) ▪ Measurement and Analysis (ML 2) ▪ Verification (ML 3) |
| <i>Customer won’t commit to project.</i> | <ul style="list-style-type: none"> ▪ Release Planning ▪ Sprint Planning | <ul style="list-style-type: none"> ▪ Requirements Management (ML 2) ▪ Project Planning (ML 2) |
| <i>Customers are frustrated with progress.</i> | <ul style="list-style-type: none"> ▪ Release Planning ▪ Sprint Planning ▪ Sprint Demo ▪ Sprint Retrospective | <ul style="list-style-type: none"> ▪ Integrated Project Management (ML 3) ▪ Validation (ML 3) ▪ Project Planning (ML 2) ▪ Project Monitoring and Control (ML 2) ▪ Measurement and Analysis (ML 2) |

| Problems | Agile Ceremonies/ Techniques | CMMI Process Areas (Maturity Level) |
|--|--|--|
| <i>Customers are not satisfied with the end product.</i> | <ul style="list-style-type: none"> ▪ Definition of Done ▪ User Stories/ Epics ▪ Test Driven Development ▪ Sprint Demo | <ul style="list-style-type: none"> ▪ Validation (ML 3) ▪ Verification (ML 3) ▪ Requirements Development (ML 3) ▪ Requirements Management (ML 2) |
| <i>There are too many bugs during a project or post project.</i> | <ul style="list-style-type: none"> ▪ Test Driver Development ▪ Continuous Build/ Continuous Integration ▪ Refactoring | <ul style="list-style-type: none"> ▪ Validation (ML 3) ▪ Verification (ML 3) ▪ Product Integration (ML 3) ▪ Technical Solution (ML 3) |
| <i>Unable to secure or retain project resources.</i> | <ul style="list-style-type: none"> ▪ Team Agreements ▪ Release Planning ▪ Daily Standup ▪ Vision | <ul style="list-style-type: none"> ▪ Integrated Project Management (ML 3) ▪ Project Planning (ML 2) ▪ Risk Management (ML 3) ▪ Project Monitoring and Control (ML 2) ▪ Organizational Training (ML 3) |
| <i>Lack of internal and external support for project.</i> | <ul style="list-style-type: none"> ▪ Team Agreements ▪ Release Planning ▪ Sprint Planning ▪ Daily Standup ▪ Product Owner Meeting with Customer | <ul style="list-style-type: none"> ▪ Integrated Project Management (ML 3) ▪ Risk Management (ML 3) ▪ Project Planning (ML 2) ▪ Project Monitoring and Control (ML 2) |
| <i>Poor project communication.</i> | <ul style="list-style-type: none"> ▪ Daily Standup ▪ Release Planning ▪ Sprint Planning ▪ Sprint Demo ▪ Sprint Retrospective ▪ Product Owner Meeting with Customer | <ul style="list-style-type: none"> ▪ Project Monitoring and Control (ML 2) ▪ Project Planning (ML 2) ▪ Integrated Project Management (ML 3) |
| <i>Requirements are vague or open-ended.</i> | <ul style="list-style-type: none"> ▪ User Story/ Epic ▪ Definition of Done ▪ Backlog Grooming | <ul style="list-style-type: none"> ▪ Requirements Management (ML 2) ▪ Requirements Development (ML 3) |
| <i>Project team responsibilities are not clear.</i> | <ul style="list-style-type: none"> ▪ Team Agreements ▪ Sprint Planning ▪ Release Planning ▪ Incremental Release | <ul style="list-style-type: none"> ▪ Integrated Project Planning (ML 2) ▪ Project Planning (ML 2) |
| <i>Project team member training is inadequate for the task.</i> | <ul style="list-style-type: none"> ▪ Release Planning | <ul style="list-style-type: none"> ▪ Organizational Training (ML 3) ▪ Project Planning (ML 2) |

| Problems | Agile Ceremonies/ Techniques | CMMI Process Areas (Maturity Level) |
|---|---|---|
| <i>Failure to successfully plan the project.</i> | <ul style="list-style-type: none"> ▪ Release Planning ▪ Sprint Planning ▪ Backlog Grooming | <ul style="list-style-type: none"> ▪ Project Planning (ML 2) ▪ Integrated Project Planning (ML 2) ▪ Requirements Management (ML 2) |
| <i>Failure to foresee potential problems.</i> | <ul style="list-style-type: none"> ▪ Daily Standup ▪ Release Planning ▪ Sprint Retrospectives | <ul style="list-style-type: none"> ▪ Project Monitoring and Control (ML 2) ▪ Risk Management (ML 3) ▪ Integrated Project Management (ML 3) ▪ Quantitative Project Management (ML 4) |
| <i>Project information isn't available when needed.</i> | <ul style="list-style-type: none"> ▪ User Stories/ Epics ▪ Sprint Planning ▪ Release Planning ▪ Daily Standup ▪ Release Burndown ▪ Sprint Burndown ▪ Sprint Demo ▪ Vision | <ul style="list-style-type: none"> ▪ Project Monitoring (ML 2) ▪ Integrated Project Management (ML 3) ▪ Risk Management (ML 3) ▪ Measurement and Analysis (ML 2) ▪ Verification (ML 3) |
| <i>Code quality is poor.</i> | <ul style="list-style-type: none"> ▪ Pair Programming ▪ Test Driven Development ▪ Continuous Build/ Integration ▪ Retrospectives ▪ Definition of Done | <ul style="list-style-type: none"> ▪ Technical Solution (ML 3) ▪ Verification (ML 3) ▪ Validation (ML 3) ▪ Integrated Project Management (ML 3) ▪ Process & Product Quality Assurance (ML 2) |
| <i>Organizational performance isn't improving.</i> | <ul style="list-style-type: none"> ▪ Retrospectives | <ul style="list-style-type: none"> ▪ Integrated Project Management (ML 3) ▪ Organizational Process Focus (ML 3) ▪ Organizational Process Definition (ML 3) ▪ Organizational Training (ML 3) ▪ Process & Product Quality Assurance (ML 2) ▪ Organizational Process Performance (ML 4) ▪ Quantitative Project Management (ML 4) ▪ Causal Analysis and Resolution (ML 5) |

CONCLUSION

The key to success in modernization programs is to choose a strategy and execute it faithfully. The foundational enablers of success are Agile and CMMI (see Figure 7). Choosing an experienced partner to undertake this journey leads to fruitful and positive outcomes.



Figure 7. Keys to Success in Modernization Programs

About Select Computing, Inc.

Select Computing, Inc. (SCi) is an innovative CMMI level 4 appraised company, for both the development and services models. SCi specializes in blending CMMI level 4 process maturity and Agile methodology to produce successful outcomes. SCi can help an organization make the transition to Agile. Implementing Agile, using a contractor that is at CMMI level 4, can result in less rework and provide significant benefits. Implementing a CMMI compliant software development process that is also Agile will bring the repeatability and predictability offered by CMMI.

SCi is a registered business partner of Micro Focus, Cloudera, and Amazon Web Services.

With direct, practical, and proven experience integrating CMMI level 4 and Agile, SCi brings innovative approaches to our customers and facilitates their initiatives while relying on quantitative management and continuous process improvement to improve efficiency and realize significant cost savings.

Contact Us

Arnold Huff

Sr. Capture/Business Development Manager
Mobile: 301-509-6969 | Office: 410-381-0082 x127
BD@selectcomputing.com

Select Computing, Inc.
9841 Broken Land Pkwy, Suite 209
Columbia, MD. 21046
www.selectcomputing.com