State of Michigan (SOM)

Systems Engineering Methodology Version 1.4

The Systems Engineering Methodology (SEM) of the State Unified Information Technology Environment (SUITE)

February 2010

Michigan Department of Information Technology

PREFACE

The initial development of the *State of Michigan Systems Engineering Methodology (SEM)* was published in April 2007, and was performed as part of a continuing effort to improve the quality, performance, and productivity of State of Michigan information systems. Development of the SEM was governed by the Michigan *State Unified Information Technology Environment* (SUITE) initiative.

The purpose of SUITE is to standardize methodologies, procedures, training, and tools for project management and systems development lifecycle management throughout the Michigan Department of Information Technology (MDIT) in order to implement repeatable processes and conduct development activities according to Capability Maturity Model Integrated (CMMI) Level 3 requirements. A formal enterprise level support structure will be created to support, improve and administer SUITE, SEM, Project Management Methodology, and related enterprise initiatives. Until that structure is in place, questions regarding SEM should be sent to SUITE@michigan.gov where they will be addressed by a matrixed team.

This SEM replaced in total the former State of Michigan Systems Development Lifecycle (SDLC) document dated November 2001 and related templates.

ACKNOWLEDGEMENTS

The State of Michigan would like to thank the following individuals and organizations that made this version of the State of Michigan Systems Engineering Methodology possible. Without their input, dedication and hard work, this would not have been achieved.

MICHIGAN DEPARTMENT OF INFORMATION TECHNOLOGY (MDIT) PROJECT SPONSORS AND MANAGEMENT	
Ken Theis, Director, Michigan Department of Information Technology (Senior Executive Sponsor)	Lynn Draschil, Senior Deputy Director, MDIT Bureau of Agency Services (Executive Sponsor)
Linda Pung, Information Officer, MDIT Bureau of Agency Services	Diana Quintero, Director, Enterprise Portfolio Management Office
Jim Hogan, Information Officer, MDIT Bureau of Agency Services	Rob Surber, Assistant Director, Center for Shared Solutions, MDIT Bureau of Agency Services
Carol Steel-Sherman, Director, Data Center Services	Virginia Hambric, Director, Project Management Office, MDIT Agency Services DHS
Dan Buonodono, Project Management Specialist, MDIT Project Management Resource Center	

PRODUCTION RELEASE – Version 1 (April 2007)	
Dan Buonodono, Project Management Specialist, MDIT Project Management Resource Center	Virginia Hambric, State Division Administrator, MDIT Agency Services-Human Services and MiCSES
Leigh A. Scherzer, Account and Project Manager, Dedicated Customer Unit, MDIT Agency Services-Department of Labor & Economic Growth.	W. Steve Wensko, Operations Manager, Operations Systems Section, MDIT Agency Services-Transportation
Paul Perla, Team Lead, MDIT-Employee and Financial Services, Human Capital Management	James “Rock” Rakowski, State Administrative Manager, Agency Liaison Section, MDIT Office of Enterprise Security
Shawn M. Bauman, Special Projects Analyst, MDIT Agency Services-Department of Human Services and MiCSES	Donna Sivaraman, Programming Services Manager, MDIT Agency Services-Treasury
Kyle Wilson, IT Specialist, Quality Assurance Team, MDIT Agency Services-Transportation	Jerry Morey, State Administrative Manager, Planning and Solutions Development, MDIT Infrastructure Services – Data Center Operations
Samuel Roberts, IT Manager, Application Development, MDIT Agency Services-Natural Resources and History, Arts & Libraries	Mark Breithart, IT Specialist, Application Development, MDIT Agency Services-Environmental Quality
Fred Moye, IT Programmer/Analyst, MDIT Agency Services-Michigan State Police	Sandy Cain, Forms Officer, MDIT Office Automation-Technical Training
Robert Surber, State Division Administrator, MDIT-Center for Geographic Information	Dave Archer, Manager, Design Section, Infrastructure Services, MDIT Office Automation-Design and Delivery Division
Dave Reicosky, Project Manager, MDIT Infrastructure Services – Telecommunications	P. Michael Spagnuolo, IT Manager, Application Programming Support Services, MDIT Agency Services-Department of State

Randy Leyrer, Development Services Manager, MDIT Agency Services-Department of Treasury	Lucy Pline, Technical Manager, MDIT Agency Services-e-Michigan Web Development Division
Allan DeKoninck, PMP, IT Manager, Data Warehouse and Database Team, MDIT Agency Services-Community Health	Patty Whitlock, Departmental Analyst, Analysis & Review Team, MDIT Agency Services-Human Services and MiCSES
Tanis S. Lerash, PMP, IT Manager, MDIT Agency Services-Civil Service, DMB, DMB Retirement	Sue Tomes, Departmental Manager, Analysis & Review Team, MDIT Agency Services- Human Services and MiCSES
Rose Johnson-King, Information Security Specialist, MDIT Office of Enterprise Security-Communications, Awareness, Homeland Security	Michael Shanahan, Director, MDIT Agency Services-e-Michigan Web Development Division
Bryan Farr, IT Specialist, Application Development and Support, MDIT Agency Services-Michigan State Police and Department of Military and Veterans Affairs	Steve Ezzo, IT Specialist, MAIN-FACS Section, MDIT Agency Services-Management and Budget
Kirt Berwald, Director, Field Services Division, MDIT Infrastructure Services	Terry O'Neill, IT Manager, Business Services, MDIT Infrastructure Services
Aparna Agrawal, Director, Technical Services, MDIT Infrastructure Services	Sara Kanya, State Administrative Manager, Enterprise Platform Services, MDIT Infrastructure Services – Data Center Operations
Bob McDonough, State Administrative Manager, MDIT Office of Enterprise Architecture	Benjamin T. Kinsey, Information Systems Analyst, Project Development Section, MDIT Agency Services-Transportation

ORGANIZATIONS
STATE OF MICHIGAN – DEPARTMENT OF INFORMATION TECHNOLOGY
U.S. DEPARTMENT OF ENERGY – OFFICE OF THE CHIEF INFORMATION OFFICER

The SEM Development Teams owe a large debt to Brenda Coblentz of the U.S. Department of Energy (DOE) for both her encouragement in our efforts and for permitting us the free use of the DOE's own CMMI Level 3 compliant SEM as a basis for this document. In particular, much of this document draws directly from the DOE's Systems Engineering Methodology, which as of February 2010 can be found at (http://cio.energy.gov/documents/SEM3_1231.pdf).

Document Revisions

The following information is used to control and track modifications made to this document.

Revision Date	Section(s)	Summary
September 2007	Throughout the document	Fixed some grammatical errors. Other minor adjustments, including a few changes to the SEM Overview Diagram.
September 2008	Add Appendix E	Appendix E provides basic guidance on the roles responsible for completing SEM forms and other work products. Updated the title page and footers to reflect the new version number and date.
February 2009	Update Appendix E	Added Project Metrics Collection template, responsibilities.
November 2009	SEM Overview Diagram, Requirements Definition Chapter, and throughout the document	Updated SEM Overview Diagram to show updated touch points, removal of checklists (incorporated in templates). Updated Requirements Definition chapter to reflect changes in the Requirements templates. Added SEM, Express, and Maintenance process usage info from Systems Maintenance Guidebook. Removed references to DIT-0015A, as this form was replaced with ITRAC.
February 2010	SEM Overview Diagram, Testing Chapter, Testing Roles and Responsibilities	Updated SEM Overview Diagram to show updated testing templates. Updated Testing chapter to reflect changes in the Testing templates. Updated relevant chapters referencing revised testing templates.

Chapter Page

Exhibits **Page**

Industry Sites[1]

Carnegie Mellon University Software Engineering Institute............................ http://www.sei.cmu.edu

International Council on Systems Engineering...http://www.incose.org

Project Management Institute ....http://www.pmi.org

Quality Assurance Institute...http://www.qaiglobalinstitute.com

[1] Addresses are as of February 2010

Chapter: **1.0 Introduction**

Description:

The Systems Engineering Methodology (SEM) of the State Unified Information Technology Environment (SUITE) provides guidance for information systems engineering related project management activities and quality assurance practices and procedures. The primary purpose of the methodology is to promote the development of reliable, cost-effective, computer-based solutions while making efficient use of resources. Use of the methodology will also aid in the status tracking, management control, and documentation efforts of a project.

Development of the SEM was governed by the Michigan State Unified Information Technology Environment (SUITE) initiative.

The purpose of SUITE is to standardize methodologies, procedures, training, and tools for project and systems development lifecycle management throughout the Michigan Department of Information Technology (MDIT) in order to implement repeatable processes and conduct development activities according to Capability Maturity Model Integrated (CMMI) Level 3 requirements.

This information system engineering methodology is consistent with other methodologies used in State and Federal Governments and private industry. It complies with State of Michigan policy on project management, software configuration management, security, and records management. The SEM should be used in conjunction with all State of Michigan information management programs and initiatives.

It is important to differentiate between a project management methodology and a system engineering methodology. A project management methodology covers all the things a project manager needs to do regardless of whether the project is a software development, package selection, or relocation of a work unit.

The State of Michigan Project Management Methodology (PMM) covers standard areas of project management (Cost Management, Risk Management, Scope Management, Resource Management, Communications Management, Quality Management, Time Management, Procurement Management, and Integration Management) and purposely does not include the separate concepts and requirements of system engineering, leaving that to be included in the SEM. Conversely, this SEM does not reiterate the standards of project management, instead referring to the PMM as appropriate.

The PMM is the methodology for management of the work effort. The SEM is the step-by-step development of the software application.

The State of Michigan has a consistent project management methodology in place which can be used for all types of projects. The State of Michigan now also has a consistent system engineering methodology that is a companion to the project management methodology.

In this way, people can move comfortably from applications development, to infrastructure roll out, to software selection to even relocating to new buildings using the same approach throughout the organization.

Significant input for the methodology was obtained from information management programs throughout the State. The methodology integrates State of Michigan best practices and focuses on the quality of both the systems engineering process and the work products generated from the process.

The SEM is derived from the principles and standards advocated by information management industry leaders, such as The Institute of Electrical and Electronics Engineers (IEEE), and the Carnegie Melon Software Engineering Institute (SEI). ***This methodology is designed to enable State of Michigan project teams to achieve Level 3 maturity on the SEI Capability Maturity Model Integrated (CMMI) model.***

Quality assurance is integrated into the methodology, making quality the responsibility of all project managers and team members. To assure the development of quality products, the methodology prescribes reviews, inspections, and audits for the lifecycle processes and technical work products. To protect the integrity of information systems, the methodology also prescribes configuration controls over system components, data, and technical documentation.

The methodology encompasses the aspects of the information systems engineering project lifecycle, from project planning through production and maintenance, and integrates basic lifecycle management concepts (*Exhibit 2.0-1 SEM Overview Diagram* on page 13).

The SEM is intended to be used by individuals, project teams, and managers who are responsible for developing a new computer-based solution or effecting changes to an existing system. The methodology, including its templates, is reviewed on a regular basis and will be modified as needed to keep pace with the changing needs of State of Michigan information systems engineering environment and the continuing technical advances in the information systems industry. As a result of the reviews, it is anticipated that a new release of the SEM will be issued within three months of the initial release date, to coincide with the Software Engineering Process Group (SEPG) process improvement release process.

The following sections provide additional information about using the SEM.

1.1 Enterprise Implementation of the Methodology
1.2 Project Implementation of Methodology
1.3 Submitting Change Requests

Section: **1.1 Enterprise Implementation of the Methodology**

Description: While the focus of the SEM is at the project level (see Section 1.2, Project Implementation of Methodology), it is recognized that there must be an enterprise-wide ability for managing information systems development, integration, and maintenance processes and quality oversight to ensure the delivery of high-quality products. Within this context, an agency is a State of Michigan unit, (e.g., a State of Michigan department or bureau, within which, generally, many projects are managed). The SEM integrates systems and infrastructure project management and quality assurance practices and is designed to be flexible. It can be adapted to accommodate the specific needs of any information systems engineering organization and all computing platforms used in the State. With adoption of the SEM as the State of Michigan standard process for developing and maintaining systems, any additional specific or unique management processes should be integrated into the organization to help project managers and technical staff perform more effectively.

In a mature organization, the processes are institutionalized. They are documented, reusable, and consistent with the way the work is actually accomplished. The process definitions are updated when necessary, and improvements are applied when appropriate, with broad-based active involvement across the organization. Roles and responsibilities are clear and communicated throughout projects and across the organization. Enterprise-wide training ensures personnel are well trained so they can perform their roles effectively and efficiently.

The following tasks describe processes and activities complementary to those at the project level and aimed at maturing the entire enterprise in terms of capability to deliver high quality products.

Resource: Carnegie Mellon University, Software Engineering Institute, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley Longman, Inc., 1994

Tasks: The following tasks are involved in enterprise implementation of the SEM.

1.1.1 Enterprise Process Management
1.1.2 Enterprise Curriculum
1.1.3 Quality Oversight

Task: **1.1.1 Enterprise Process Management**

Description: The goal of this task is to establish the State's responsibility for lifecycle process activities that improve its overall capability. The State provides long-term commitments and resources to coordinate the development and maintenance of the process across current and future projects.

With MDIT's adoption of the SEM, it became the state-wide enterprise process for systems development. The Software Engineering Process Group (SEPG) will periodically assess its processes and develop an action plan for improvement. Changes to the process are then communicated to those individuals within the State responsible for implementing the process.

New processes, methods, and tools in limited use in the State are to be monitored, evaluated, and, where appropriate, transferred to other parts of the MDIT organization. The major component of Enterprise Process Management is the Enterprise Repository.

Enterprise Repository

It is anticipated that the SUITE Core Team will create, manage and control a repository to collect and make available data on the systems engineering process and resulting work products, (e.g., productivity data, quality measurements, and estimates of size, effort, and cost). It is anticipated that the repository will serve to improve project management planning and estimating by providing a resource for future systems engineering efforts. The repository will also establish and control a statewide library of systems process-related documentation including policies and procedures that will serve as a path to achieving CMMI Level 3. The library will be cataloged for easy reference and the contents made available for use by project teams and other systems-related groups. Library contents are to be updated as appropriate.

Work Products: An action plan is to be developed based on the periodic assessments. The action plan will identify guidelines for implementing the changes to address specified assessment findings and assigns responsibility for implementing changes.

An improvement plan is to be developed and maintained for process development and improvement activities. The plan uses the action plan and other improvement initiatives as primary input. The plan defines and schedules activities to be performed, assigns responsibility and identifies resources required for implementing the plan.

It is anticipated that a repository will be established for enterprise process and data (metrics) information. MDIT staff will be trained in the use of, and have controlled access to, the repository.

Review Process: Conduct structured walkthroughs for each of the written work products to remove as many defects as possible.

Task: **1.1.2 Enterprise Curriculum**

Description: An organization that is well prepared for the challenges posed by information systems engineering projects must ensure that its personnel are well trained to perform their roles effectively and efficiently. The goal of this task is to describe the areas of training that must be addressed to ensure the State of Michigan has a documented process in place to manage training activities on an ongoing basis.

The process shall be based on documented enterprise training standards. The standards should include how courses are to be developed (or standards that must be met where courses are procured) and how they are to be maintained according to these standards. Members of the training group (or vendors if training is acquired) need to have the necessary skills and knowledge to perform their training activities.

When determining the skills and knowledge needed for a project, the project teams are responsible for identifying their unique needs. Each project needs to evaluate its current and future skill needs and determine how these skills will be obtained. Some skills may be imparted through informal vehicles (e.g., on-the-job training, mentoring,) while other skills may need more formal training vehicles (e.g., classroom, self-study). Appropriate vehicles need to be selected and used.

Responsibility for training needs to be identified and communicated. It may lie with a single manager within the organization, or may be shared by several managers, each responsible for one or more knowledge areas or subjects. The specific enterprise responsibility for training needs to be identified, documented, and available for viewing by staff.

Waiver Process: A waiver procedure for required training needs to be established and used to determine whether staff already possesses the knowledge and skills to perform their jobs.

Measurements: Measurements need to be identified, collected, and used to assess the status of training activities. Measurements should address areas such as the quality of the training, and if it meets the needs of the staff. Measurements and the enterprise training should be reviewed with management on a regular basis. (In this context, “measurement” is not meant to be CMMI Level 4.)

Work Products: A written training policy describing how the State of Michigan will meet training requirements needs to be developed, communicated, and followed. The policy needs to be periodically reviewed and revised as appropriate based on feedback collected.

A written training plan that addresses how the training needs of the State of Michigan will be met. The plan should include information such as how training needs will be identified, what training is required, how training will be delivered, the cost and resources required, enterprise placement of the training function, who will be involved, when and how the plan will be reviewed and revised, and a work breakdown structure that identifies all of the activities involved.

Maintain records that training has been conducted and completed waivers, if and where appropriate.

Note: A written training plan should be developed for each project and a training program should be developed for system implementation and operation.

Review Process: Conduct structured walkthroughs for each of the written work products to remove as many defects as possible.

Task: **1.1.3 Quality Oversight**

Description: The goal of this task is to establish the enterprise responsibility for the quality oversight of information technology investments. While the lifecycle process activities for project implementation and maintenance are documented within the lifecycle stages of the SEM, an enterprise quality oversight program provides long-term commitments and resources to coordinate the quality activities across current and future projects.

The quality oversight program, referred to as the PPQA Process, should implement the appropriate level of management effort, and assume responsibility, accountability, and oversight for continued quality management process compliance within the organization. The quality oversight program should identify standards and best practices for product development, and ensure appropriate safety and security controls are in place, are effective, and reflect current accepted industry practices. The program should also ensure that project teams are aware of current State of Michigan computer and cyber security directives and have coordinated the project with computer security staff.

Work Products: A written quality oversight program describing how the organization will ensure the development of high quality information technology investments needs to be developed, communicated, and followed. The program needs to be periodically reviewed and revised as appropriate based on feedback collected.

The program should identify a point of contact for managing quality oversight and ensuring project risk assessments are conducted to determine the appropriate level of quality assurance activities to be applied. The program should ensure the level of quality assurance is tailored to the site and project needs. The oversight program should oversee the development and implementation of quality assurance processes and procedures, and ensure the development and implementation of project quality assurance plans and production and delivery of quality products.

Review Process: Conduct a structured walkthrough of the quality oversight program to remove as many defects as possible.

Section: **1.2 Project Implementation of Methodology**

Description: SUITE will integrate information systems engineering, project management and quality assurance practices and is designed to be flexible. It can be adapted to accommodate the specific needs of any information systems project and all computing platforms used in the State of Michigan including standalone and networked mainframes, servers, desktops, and other computers.

Projects that were initiated prior to the awareness or usage of this document should plan to implement the methodology at the earliest feasible stage or the next release of the product. If a Project Plan already exists, make the revisions necessary to integrate the systems engineering, project management, and quality assurance practices, as appropriate. If a Project Plan does not exist, develop a plan that summarizes the activities and deliverables of the previous stages and incorporates the methodology activities and products into the subsequent stages.

The information systems engineering methodology presented here does not supersede, replace, or override more stringent requirements that may apply to specific projects such as scientific and technical practices, and security and safety issues.

Questions: If specific questions are generated concerning the interpretation or applicability of portions of the methodology, the project team should attempt to resolve them during the project review activities built into the stages of the lifecycle. The system owner/user(s) and other project stakeholders must concur with any adaptations that are made.

When questions about interpretation or applicability of the guidance to a specific project cannot be resolved by the project team, the issue should be submitted to the site authority for information systems engineering, such as the team leader, supervisor, manager, or MDIT Client Services Director, for advice or resolution. Software Engineering Process Group (SEPG) staff may also be consulted on the interpretation or applicability of the methodology by sending e-mail to SUITE@michigan.gov.

Section: **1.3 Submitting Change Requests**

Description: The SEM environment is continuously changing as emerging technologies are integrated into projects, system owner/user requirements are expanded, and enterprise needs evolve. The SEM will be revised, as needed, to reflect changes in the environment, improvements suggested through user feedback, and the maturation of information systems engineering capabilities.

Users of the methodology are encouraged to submit suggestions for improving its content and to report any practices that are difficult to understand or create an implementation problem for a project team.

Suggestions and problems should be submitted via e-mail to suie@michigan.gov. All requests will be evaluated and the originator of the request will be notified of the action taken.

Some requests will be handled immediately while others may require investigation by an ad hoc working group (called a process action team, or PAT) of knowledgeable personnel. In some cases, a request may not be appropriate for the current environment, but will be retained for future consideration.

Chapter: **2.0 Lifecycle Model**

Description: This chapter describes the lifecycle model used for the SEM. This model partitions the information systems engineering lifecycle into seven major stages, as shown in *Exhibit 2.0-1, SEM Overview Diagram* on page 13. Each stage is divided into activities and tasks, and has a measurable end point (Stage Exit). The execution of all seven stages is based on the premise that the quality and success of the product depends on a feasible concept, comprehensive and participatory project planning, commitments to resources and schedules, complete and accurate requirements, a sound design, consistent and maintainable construction techniques, and a comprehensive testing program. The lifecycle stages and activities are described in the following chapters.

Intermediate work products are produced during the performance of the activities and tasks in each stage. These work products are inspected and can be used to assess system integrity, quality, and project status. As a result, adequacy of requirements, correctness of designs, and quality of the products become known early in the effort.

At least one time for each work product, a Structured Walkthrough (SWT) is performed. A Structured Walkthrough is an organized procedure for reviewing and discussing the technical aspects of systems or software engineering work products including documentation. The walkthrough is usually conducted by a group of peers and may include reviewers outside the developer's immediate peer group. The *Structured Walkthrough Process Guide* provides detailed process information. This document is available on the MDIT SUITE website.

At the conclusion of each stage, a Stage Exit is initiated to review the work products of that stage and to determine whether to proceed to the next stage, continue work in the current stage, or abandon the project. The approval of the system owner and other project stakeholders at the conclusion of each stage enables both the system owner and the project manager to remain in control of the project throughout its life, and prevents the project from proceeding beyond authorized milestones. The *Stage Exit Process Guide* provides detailed process information. This document is available on the MDIT SUITE website.

The end products of the lifecycle are the information system product, the data managed by the system, associated technical documentation, and user training and support. The end products and services are maintained throughout the remainder of the lifecycle in accordance with documented configuration management procedures.

The lifecycle model provides a method for performing the individual activities and tasks within an overall project framework. The stages and activities are designed to follow each other in an integrated fashion, whether the stages of development are accomplished sequentially, concurrently, or cyclically. Project teams have the flexibility to adapt the lifecycle model to accommodate a particular development method (e.g., spiral development,) information systems engineering technique (e.g., prototyping and rapid application development,) or other project constraints.

The amount of project and system documentation required throughout the lifecycle depends on the size and scope of the project. System documentation needs to be at a level that allows for full system operability, usability, and maintainability. Typically, projects that require at least one work-year of effort should have a full complement of documentation. For projects that require less than one work-year of effort, the project manager and system owner should determine the documentation requirements. In addition, the project's security and quality assurance criteria may require the performance of other activities and the generation of additional documentation.

The requirements for documentation should not be interpreted as mandating formal, standalone, printed documents in all cases. Progressive documents that continuously revise and expand existing documentation, online documents, forms, reports, electronic mail messages, and handwritten notes (e.g., informal conference records) are some examples of alternative documentation formats. Project managers should verify documentation standards within their sites.

The following sections provide additional information about the lifecycle model.

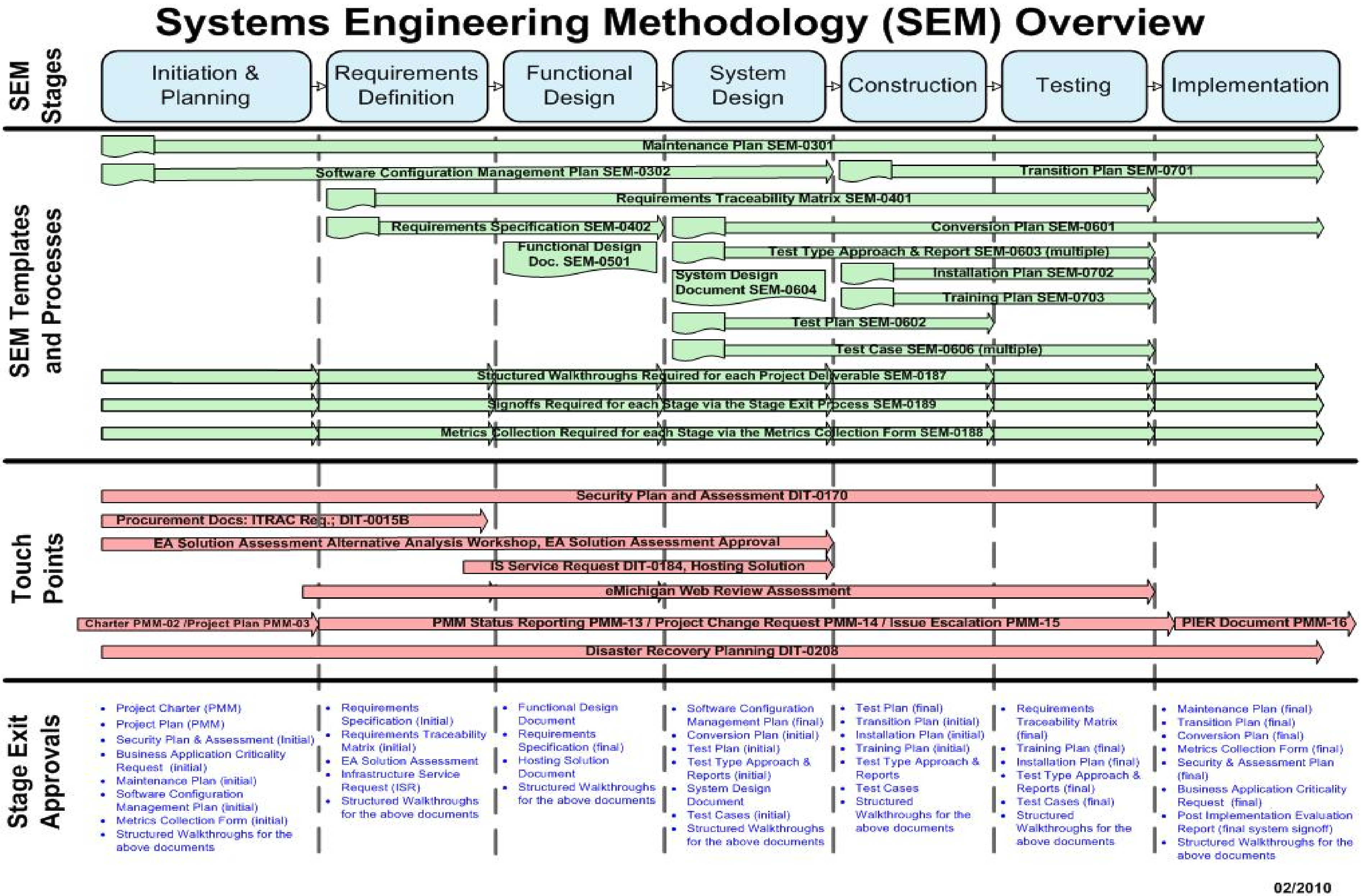
- 2.1 Project Sizes
- 2.2 Adapting the Lifecycle
  - 2.2.1 Tailoring Guidance
  - 2.2.2 Work Type Definitions
- 2.3 Development Techniques
- 2.4 Commercial-Off-The-Shelf (COTS) Products Based Projects
- 2.5 Quality Reviews

Bibliography: The following materials were referenced in the preparation of this chapter.

1. SLC Roadmap Document, Electronic Data Systems, Inc., 1991
2. Booch, G., *Object-Oriented Analysis and Design,* 2nd edition, Benjamin Cummings, 1994.
3. Budd, T., *An Introduction to Object-Oriented Programming,* 2nd edition, Addison-Wesley, 1996.
4. Carnegie Mellon University, Software Engineering Institute, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley Longman, Inc., 1994.
5. Carney, D, & Oberndorf, P. "The Commandments of COTS: Still Searching for the Promised Land." Crosstalk 10, 5 (May 1997): 25-30.
6. Federal Acquisition Regulations. Washington, DC: General Services Administration, 1996.
7. Jacobson, I., *Object-Oriented Software Engineering,* Addison-Wesley, 1992.

8. Meyers, Craig & Oberndorf, Tricia. Open Systems: The Promises and the Pitfalls. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.
9. Open Systems Joint Task Force Baseline Study, 1996 [online].
10. Open Systems Joint Task Force Case Study of U.S. Army Intelligence and Electronic Warfare Common Sensor (IEWCS), 1996 [online].
11. Pressman, Roger S., *Software Engineering - A Practitioner's Approach,* 4th edition, McGraw-Hill Companies, Inc., 1997.
12. Siy, Harvey, *Identifying the Mechanisms to Improve Code Inspections Costs and Benefits*, 1996 [online].
13. Yourdon, Edward, *Structured Walkthroughs*, second edition, Yourdon Inc., New York, 1978.

Exhibit 2.0-1 SEM Overview Diagram



Section: **2.1 Project Sizes**

Description: The lifecycle model used in this information systems engineering methodology can be applied to projects of varying sizes. In this model, projects are divided into four sizes: large, medium, small, and maintenance. Each project size, with the exception of maintenance, uses the same lifecycle stages. Medium and small projects may compress or combine stages and required documentation in direct proportion to the size of the development effort, as dictated in *SEM Express*. The major differences between project sizes are determined by the following items.

- The estimated total labor hours (the level of effort) required to complete the project
- The use of cutting edge or existing technology
- The type and extent of both user and system interface requirements
- The project's contribution to, and impact on, the activities carried out by the system users and other departmental organizations

The requirements, constraints, and risks associated with the project also influence the determination of project size. The project size and any plans for adapting the lifecycle model are documented in the Project Plan (PMM-03 / PMM-03 Exp), which is reviewed and approved by the system owner and other project stakeholders.

To the extent possible, all maintenance and operations activities should be managed as a project, utilizing the Systems Maintenance Template (SEM-0931), to gain the benefits inherent in project management and to enable tracking of activities and costs. The extent of project management activity will vary, and should be tailored according to the size, complexity, and impact of the change or enhancement. Refer to the Systems Maintenance Guidebook for more information on maintenance projects.

The following guidelines are meant to help MDIT staff determine which process to use and therefore which amount of project documentation and management is required:

Project Size (Effort)	Methodology To Use
0 – 200 hours	System Maintenance process
200 – 1000 hours	SEM Express
1001 hours or larger	SEM

The following subsections provide descriptions of the three project sizes used in this lifecycle model. *Exhibit 2.1-1, Information Systems Project Sizes*, shows the level of effort and complexity measures used to define the three sizes.

Large Projects: Large information systems engineering projects are included in the system owner's organizational long-range plans. Department-wide and site-specific projects are usually developed as large-sized projects and are likely to require a major acquisition of hardware and software. Typically, the larger the size and scope of the project, the greater the detail and coordination needed to manage the project. As risk factors and levels of effort increase, the scope of project management also increases and becomes a critical factor in the success of the project.

Medium Projects: Medium information systems engineering projects require less effort than large projects, typically use existing hardware and software, and might not be captured during the organizational long-range planning process. They are frequently developed to automate operations within a programmatic office or among a limited number of sites, and may be used to interface with other systems. Planning medium size projects within the context of the system owner organization's overall mission, and building in compatibility to the departmental IT environment can improve the product's ability to interface with other users, organizations, and applications; and increase the product's longevity.

Small Projects: Small information systems engineering projects require minimal effort and use existing hardware and software. The operational details of a small project can easily be managed by the project manager, so formal documentation requirements are limited. A project is small when the system being developed will have limited functionality and use, meets a one-time requirement, or is developed using reusable code.

Exhibit 2.1-1 Information Systems Project Sizes

Complexity (and associated characteristics)	**Effort Required** (in staff months)		
	0-8	**9-24**	**25-n**
Low: - Existing or known technology - Simple interfaces - Requirements well known - Skills are available	Small	Small	Medium
Medium: - Some new technology - Multiple interfaces - Requirements not well known - Skills not readily available	Small	Medium	Large
High: - New technology - Numerous complex interfaces - Numerous resources required - Skills must be acquired	Medium	Large	Large

Note: Size is used as a guide to help determine the appropriate degree of project management, and whether any stages may be combined for a given effort. Within this context, size is a combination of level of effort required (all activities) and complexity of the requirements. Attributes of complexity include technology, team skills, interfaces, and level of understanding of requirements. Other factors that can influence adaptation include risk, visibility, and business impact.

Maintenance Projects: The State of Michigan has added a System Maintenance process and template (SEM-0931) to the SUITE methodology in addition to the existing Systems Engineering Methodology (SEM) and SEM Express methodologies. The majority of system maintenance projects are typically smaller in size and will follow the maintenance methodology. Projects with a larger effort would follow SEM Express or the full SEM.

Section: **2.2 Adapting the Lifecycle**

Description: The SEM implements well-defined processes in a lifecycle model that can be adapted to meet the specific requirements or constraints of any project. This section provides guidelines for adapting the lifecycle processes to fit the characteristics of the project. These guidelines help ensure that there is a common basis across all projects for planning, implementing, tracking, and assuring the quality of the work products.

The lifecycle model has built-in flexibility. All of the stages and activities can be adapted to any size and scope information systems engineering project. The lifecycle can be successfully applied to development projects, maintenance or enhancements, and customization of commercial software. The lifecycle is appropriate for all types of administrative, business, manufacturing, laboratory, scientific, and technical applications. For scientific and technical projects, adaptations to the lifecycle may be dictated by the project stakeholders or the requirements for reporting technical results in formal reports or journal articles.

Adaptations: The lifecycle can be compressed to satisfy the needs of a small project, expanded to include additional activities or work products for a large or complex project, or supplemented to accommodate additional requirements, (e.g., security requirements). Any modifications to the lifecycle should be consistent with the established activities, documentation, and quality standards included in the methodology. Project teams are encouraged to adapt the lifecycle as long as the fundamental information systems engineering objectives are retained and quality is not compromised.

The following are some examples of lifecycle adaptations:

- Schedule stages and activities in concurrent or sequential order.
- Repeat, merge, or simplify stages, activities, or work products.
- Include additional activities, tasks, or work products in a stage.
- Change the sequence or implementation of lifecycle activities.
- Change the development schedule of the work products.
- Combine or expand activities and the timing of their execution.

The lifecycle forms the foundation for project planning, scheduling, risk management, and estimation. When a lifecycle stage, activity, or work product is adapted, the change must be identified, described, and justified in the Project Plan. The Project Plan is developed as a separate document and includes a description of the systems development lifecycle, which is the organization's standard process.

Exhibit 2.2-1, Adapting the Lifecycle, shows how stages can be combined to accommodate different size projects and information systems engineering techniques. Notes are provided throughout the lifecycle stage chapters to identify activities that have built-in project adaptation strategies. Adaptations should not introduce an unacceptable level of risk and require the approval of the system owner and other project stakeholders.

When adapting the lifecycle model, care must be taken to avoid the following pitfalls:

- Incomplete and inadequate project planning.
- Incomplete and inadequate definition of project objectives and requirements.
- Lack of a development methodology that is supported by information systems engineering preferred practices and tools.
- Insufficient time allocated to complete design before coding is started.
- Not defining and meeting criteria for completing one lifecycle stage before beginning the next.
- Compressing or eliminating testing activities to maintain an unrealistic schedule.

Sample Statements:

The following are sample statements that can be used in the Project Plan to describe different types of lifecycle adaptations. The first example shows a scenario where the Concept Document will not be developed in the Initiation and Planning Stage.

A Concept Document will not be developed for this project. The need for the product has been documented in several organizational reports and was included in the fiscal year long-range plans. The platform for the project is currently used for all applications owned by this organization. There are no known vendor packages that will satisfy the functional requirements described by the system owner.

The following is a sample that shows how the seven lifecycle stages can be compressed into five stages for a small project.

This project will require 10 staff months of effort to enhance an existing application. The seven stages in the lifecycle will be combined into five stages as follows: (1) Initiation and Planning, (2) Requirements and Design, (3) Construction, (4) Testing, and (5) Implementation.

The following deviations will occur for document deliverables:

- *A Concept Document and a Business Case will not be necessary due to the restricted software and hardware platform.*
- *The Requirements Specification will be limited to the statement of enhancement requirements.*
- *An amendment package will be developed for the existing Users Manual.*

Exhibit 2.2-1 Adapting the Lifecycle

Large Project						
Initiation & Planning	Requirements Definition	Functional Design	System Design	Construction	Testing	Implementation
			Iterative Development[1]			

Medium Project				
Initiation & Planning	Requirements Definition / Functional Design	System Design / Construction	Testing	Implementation
Rapid Prototyping[2]				

Small Project		
Initiation & Planning / Requirements Definition / Functional and System Designs	Construction and Testing	Implementation

← Project Management →

Less More

Structured Walkthroughs should be performed for each major deliverable. Stage Exits should occur at the end of each stage.

Note: Iterative development and rapid prototyping are optional techniques that can be used on any size project.

[1] Each iteration produces working function(s) from integrated program modules.
[2] Iterations may produce any or all of requirements, system architecture, functional design, system design.

***Section*:** **2.2.1 Tailoring Guidance**

Due to the large variation among system size and complexity, there is a need to offer guidance to the project / development manager regarding which components of the methodology, both project-based and product-based, are required.

The intent of this section is to provide flexibility in utilizing SEM components in the systems development process. The focus here is to ensure that adequate processes are used for each of the various types of systems engineering initiatives – “using the right tool for the job.”

A small project which meets the criteria for *SEM Express* is typically straight-forward in nature and estimated to be greater than 200 and less than 1000 effort hours (including both systems development related and project management related hours). A large project, which meets the criteria for the full SEM, is typically complex in nature and is estimated at more than 1000 effort hours. Projects that fall in the middle are considered medium projects, and will typically use a customized SEM for development of the system. Section 2.2.2 also offers guidance on customizing the SEM – giving guidance on which SEM templates to use, based on project work type. Systems maintenance projects are typically less than 200 effort hours.

The project manager has the discretion to use *SEM Express* for slightly larger projects if he/she feels the complexity is such that *SEM Express* is preferable.

If at any time the project manager feels he/she need to have more process guidance, he/she has the discretion to add processes and/or templates from the full SEM to meet the documentation/approval needs of the project. It is also acceptable to switch from *SEM Express* to a customized SEM mid-stream if the project warrants such a change, due to increased scope, inaccurate initial estimates, etc.

The following SEM Tailoring Matrix is designed to guide the project / development manager in selecting the relevant components of the Systems Engineering Methodology for use in their project.

This matrix is used to identify SEM templates and processes required for a given project size.

SEM Tailoring Matrix

NOTES:

1.) "If Applicable" means the template is required if the project has impact on that area, such as training, contract management, or infrastructure changes.

2.) It is assumed that if "master" documents exist for the system, those master documents will be updated and attached to the current SEM / SEM Express documents, with the new changes noted.

Template / Process	Document Reference	Small, Straight-Forward Project *-SEM Express-*	Medium Project -Customized SEM-	Large Project -SEM-	Guidance
EA Solution Assessment	SEM Touch Point (Solution Assessment Worksheet)	Not Applicable	Required if no existing EA Solution Assessment is on file with EA or if proposing changes to the one on file.	Required if no existing EA Solution Assessment is on file with EA or if proposing changes to the one on file.	Check with an Enterprise Architecture representative if unsure.
Maintenance Plan	SEM-0301	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	New plan required or updates to original plan, if available.	New plan required or updates to original plan, if available.	
Software Configuration Management Plan	SEM-0302	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	New plan required or updates to original plan, if available.	New plan required or updates to original plan, if available.	
Requirements Traceability Matrix	SEM-0401	Not Required. Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	Required	Required	
Requirements Specification	SEM-0402	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	New specification required or updates to original specification, if available	Required	
Functional Design Document	SEM-0501	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	New design required or updates to original design, if available	New design required or updates to original design, if available	
Conversion Plan	SEM-0601	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	Required if converting existing data	Required if converting existing data	
Test Plan	SEM-0602	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	Required	Required	
Test Type Approach and Reports (multiple)	SEM-0603	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	Required	Required	
Test Cases (multiple)	SEM-0606	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	Required	Required	

Template / Process	Document Reference	Small, Straight-Forward Project -*SEM Express*-	Medium Project -Customized SEM-	Large Project -SEM-	Guidance
System Design Document	SEM-0604	Integrated into *SEM Express* Initiation, Requirements, and Design Plan.	New design document required or updates to original design document, if available	Required	
Transition Plan	SEM-0701	Integrated into *SEM Express* Construction and Testing Plan.	If Applicable	Required	Required if new staffing or operational procedures are identified for operations staff, maintenance staff, or client staff
Installation Plan	SEM-0702	Integrated into *SEM Express* Construction and Testing Plan.	Required	Required	
Training Plan	SEM-0703	Integrated into *SEM Express* Construction and Testing Plan.	If Applicable	Required	Required if new staffing or training needs are identified
Structured Walkthrough process	Structured Walkthrough Process Guide	Required for both Initiation, Requirements, and Design Plan and Construction and Testing Plan	Required	Required	Structured Walkthroughs are required for all major deliverables
Stage Exit process	Stage Exit Process Guide	Required for each stage	Required for each stage	Required for each stage	
Security Plan	SEM Tough Point (DIT-0170 or DIT-0170 Exp)	If Applicable	If Applicable	Required	
Infrastructure Services Request	SEM Touch Point (DIT 184)	If Applicable	If Applicable	Required	
Contracts and Procurement documents	SEM Touch Point (ITRAC Req., DIT-0015B)	If Applicable	If Applicable	If Applicable	
Business Continuity Plan	SEM Touch Point	If Applicable	If Applicable	Required	

Section: **2.2.2 Work Type Definitions**

Description: There are 6 work types currently available within the SEM. Work types can be identified by either an alphabetic character designation or by a descriptive name. See *Exhibit 2.2.2 Work Type Selection Diagram* on page 27 to determine the work type that applies.

The work types documented in the SEM include the following:

(A) **Break/Fix**

The Break/Fix work type is used if there has been an interruption of a critical service to a client. An action is required and a solution must be put in place, even if the solution is temporary. The problem must be investigated to determine the root cause. The permanent solution to the problem may result in the initiation of another work type.

Examples:

Production abnormal end
Loss of on-line production system
Incorrect or missing customer data
Hardware malfunction
Network lines down

Use your current process to handle this work type.

(B) **Enhancement/Maintenance**

The Enhancement/Maintenance work type applies to an application system modification involving process changes and/or data structure changes. There are no changes to hardware or software platforms. This work type assumes that programs will be changed or created.

Examples:

Changing a business rule
Adding or changing edit checks for validating data
Changing or creating a report layout
Changing or creating an update/display screen
Creating a new data structure
Changing the field length of a data structure

For this work type, the following SEM templates need to be completed:

System Maintenance Document (SEM-0931)
Test Plan (SEM-0602)
Test Type Approach and Report (SEM-0603) [multiple]
Test Cases (SEM-0606) [multiple]
Transition Plan (SEM-0701)

For this work type the, following SEM templates will need to be revised, or created if they do not currently exist:

Requirements Specification (SEM-0402)
Functional Design (SEM-0501)
System Design (SEM-0604)
Security Plan and Assessment (DIT-0170 or DIT-0170 Exp)

For this work type, the following SEM templates may be revised or created as needed:

Requirements Traceability Matrix (SEM-401)
PMM Charter (PMM-02) / Project Plan (PMM-03 or PMM-03 Exp)
Training Plan (SEM-0703)

(C) **New Development**
The New Development work type applies to the development of a new application system.

Examples:
Developing a new Web application system
Developing a new Desktop application system

For this work type, the following SEM templates need to be completed:

Software Configuration Management Plan (SEM-0302)
Security Plan and Assessment (DIT-0170 or DIT-0170 Exp)
Solution Assessment
PMM Charter (PMM-02) / Project Plan (PMM-03 or PMM-03 Exp)
Requirements Traceability Matrix (SEM-0401)
Requirements Specification (SEM-0402)
Functional Design Document (SEM-0501)
System Design Document (SEM-0604)
Test Plan (SEM-0602)
Test Type Approach and Reports (SEM-0603) [multiple]
Test Cases (SEM-0606) [multiple]
Transition Plan (SEM-0701)
Training Plan (SEM-0703)

For this work type the, following SEM templates will need to be revised, or created if they do not currently exist:

Business Continuity Planning documentation

For this work type, the following SEM templates may be revised or created as needed:

Maintenance Plan (SEM-0301)
Procurement Docs (ITRAC Request, DIT-0015B)
Conversion Plan (SEM-0601)

(D) **Commercial Off The Shelf (COTS) Implementation**
The COTS Implementation work type applies to the implementation of an existing application system. This includes the implementation of vendor provided "turn key" applications.

Examples:
COTS Application
ERNIE (DEQ)

For this work type, the following SEM templates need to be completed:

Security Plan and Assessment (DIT-0170 or DIT-0170 Exp)
Procurement Documents (ITRAC Request, DIT-0015B)
Solution Assessment
Requirements Traceability Matrix (SEM-0401)
Requirements Specification (SEM-0402)
Test Plan (SEM-0602)
Test Type Approach and Reports (SEM-0603)
Test Cases (SEM-0606)
Installation Plan (SEM-0702)
Training Plan (SEM-0703)

For this work type the, following SEM templates will need to be revised, or created if they do not currently exist:

Business Continuity Planning documentation

For this work type, the following SEM templates may be revised or created as needed:

Maintenance Plan (SEM-0301)
Software Configuration Management Plan (SEM-0302)
PMM Charter (PMM-02) / Project Plan (PMM-03)
Conversion Plan (SEM-0601)

(E) **Application Migration**
The Application Migration work type applies to the migration of an application to a new hardware or software platform. Conversion programs may be necessary. There are no changes to processes or data structures.

Examples:
Migrating an application from one data center to another data center
Porting an application from a PC UNIX platform to a Sun UNIX platform

Porting an application from a NT 4 server to Windows 2003 server cluster
Porting a database from Oracle to SQL Server

For this work type, the following SEM templates need to be completed:

Solution Assessment
Conversion Plan (SEM-0601)
Test Plan (SEM-0602)
Test Type Approach and Reports (SEM-0603)
Test Cases (SEM-0606)

For this work type the, following SEM templates will need to be revised, or created if they do not currently exist:

Security Plan and Assessment (DIT-0170 or DIT-0170 Exp)
Business Continuity Planning documentation
Requirements Traceability Matrix (SEM-0401)
Requirements Specification (SEM-0402)

For this work type, the following SEM templates may be revised or created as needed:

Procurement Documents (ITRAC Request, DIT-0015B)
PMM Charter (PMM-02) / Project Plan (PMM-03 or PMM-03 Exp)

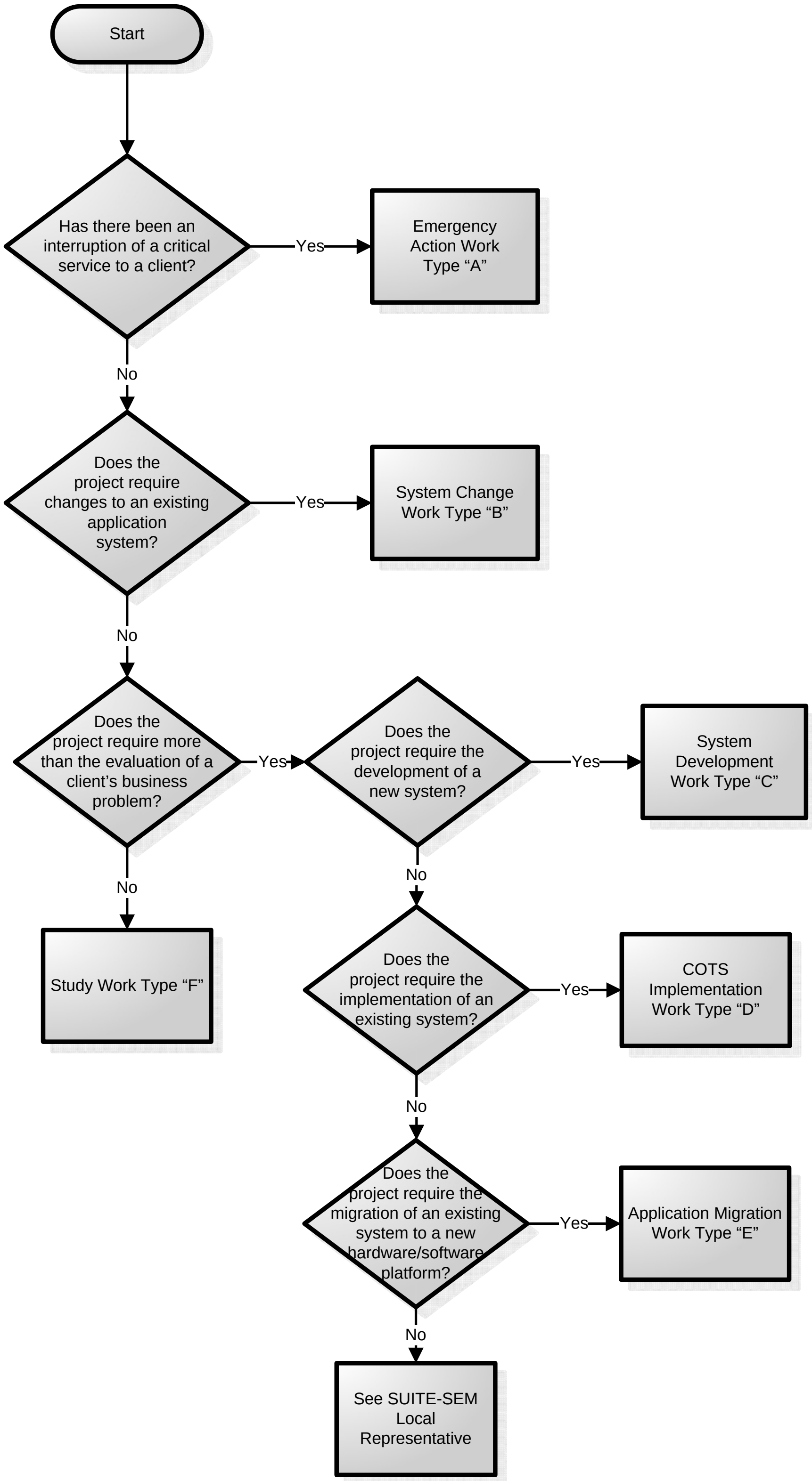
Study
The Study work type is used to evaluate a client's business problem or opportunity which result in recommended solutions. These solutions may not always result in system related work.

Examples:
- Providing a recommendation to decrease turn around time for accounts payable invoices
- Providing a recommendation for new application systems to replace old application systems.

For this work type, the following SEM templates need to be completed:

Requirements Specification (SEM-0402)
Solution Assessment

Exhibit 2.2.2 Work Type Selection Diagram



Section: **2.3 Development Techniques**

Description: This section provides descriptions of some development techniques that can be used with the SEM. The descriptions include high-level instructions on how to adapt the lifecycle stages to accommodate the development technique. The descriptions provided here are not intended to be a comprehensive list of development techniques.

Segmented Development: NOTE: The term "segment" is used here to avoid confusion between project and production phases.

Segmented development is most often applied to large information systems engineering projects where the project requirements can be divided into functional segments. Each segment becomes a separate project and provides a useful subset of the total capabilities of the full product. This segmenting serves two purposes: to break a large development effort into manageable pieces for easier project management and control; and to provide intermediate work products that form the building blocks for the complete product.

The lifecycle processes and activities are applied to each segment. The overall system and software objectives are defined, the system architecture is selected for the overall project, and a Project Plan for development of the first segment is written and approved by the system owner.

Segments are delivered to the system owner for evaluation or actual operation. The results of the evaluation or operation are then used to refine the content of the next segment. The next segment provides additional capabilities. This process is repeated until the entire product has been developed. If significant problems are encountered with a segment, it may be necessary to reexamine and revise the project objectives, modify the system architecture, update the overall schedule, or change how the segments are divided.

Two major advantages of this approach are: the project manager can demonstrate concrete evidence that the final product will work as specified; and users will have access to, and use of, segments or functions prior to the delivery of the entire product.

Spiral Development: Spiral development repeats the planning, requirements, and functional design stages in a succession of cycles in which the project's objectives are clarified, alternatives are defined, risks and constraints are identified, and a prototype is constructed. The prototype is evaluated and the next cycle is planned.

The project objectives, alternatives, constraints, and risks are refined based on this evaluation, and then an improved prototype is constructed. This process of refinement and prototyping is repeated as many times as necessary to provide an incrementally firm foundation on which to proceed with the project.

The lifecycle activities for the Initiation and Planning, Requirements Definition, and Functional Design Stages are repeated in each cycle. Once the design is firm, the lifecycle stages for System Design, Construction, and Testing are followed to produce the final product.

Rapid Prototyping:

Rapid prototyping can be applied to any information systems development methodology (e.g., segmented, spiral). Rapid prototyping is recommended for systems development that is based on a new technology or evolutionary requirements.

With the rapid prototyping technique, the most important and critical requirements are defined based on current knowledge and experience. A quick design addressing those requirements is prepared, and a prototype is coded and tested. The purpose of the prototype is to gain preliminary information about the total requirements and confidence in the correctness of the design approach. Characteristics needed in the final product, such as efficiency, maintainability, capacity, and adaptability might be ignored in the prototype.

The prototype is evaluated, preferably with extensive user participation, to refine the initial requirements and design. After confidence in the requirements and design approach is achieved, the final product is developed. The prototype might be discarded, or a portion of it used to develop the final product.

The normal documentation requirements are usually postponed with prototyping efforts. Typically, the project team, project stakeholders, and system owner agree that the prototype will be replaced with the actual product and required support documentation after proof of the model. The system that replaces the prototype should be developed using the lifecycle processes and activities.

Iterative Technique:

The iterative technique is normally used to develop products piece by piece. Once the system architecture and functional or conceptual design are defined and approved, system functionality can be divided into logically related pieces called "drivers."

In iterative fashion, the project team performs system design, code, unit test, and integration test activities for each driver, thereby delivering a working function of the product. These working functions or pieces of the product are designed to fit together as they are developed. This technique allows functions to be delivered incrementally for testing so that they can work in parallel with the project team. It also enables other functional areas, such as documentation and training, to begin performing their activities earlier and in a more parallel effort. In addition, the iterative technique enables progress to be visible earlier, and problems to be contained to a smaller scope.

With each iterative step of the development effort, the project team performs the lifecycle processes and activities.

Rapid Application Development: Rapid Application Development (RAD) is a method for developing systems incrementally and delivering working pieces every 3 to 4 months, rather than waiting until the entire project is constructed before implementation. Over the years, many information technology projects failed because by the time the implementation took place, the business had changed.

RAD employs a variety of automated design and development tools, including Computer-Aided Software Engineering (CASE), advanced generation languages, visual development, and graphical user interface (GUI) builders, which get prototypes up and running quickly. RAD focuses on personnel management and user involvement as much as on technology.

Joint Application Development: Joint Application Development (JAD) is a RAD concept that involves cooperation between the designer of a computer system and the end user to develop a system that meets the user's needs exactly. It complements other system analysis and design techniques by emphasizing participative development among system owners, users, designers, and builders. During JAD sessions for system design, the system designer will take on the role of facilitator for possibly several full-day workshops intended to address different design issues and deliverables.

Object-Oriented Development: Object-oriented development focuses on the design of components that mimic the real world. A component that adequately mimics the real world is much more likely to be used and reused. The approach emphasizes how a system operates, as opposed to analysis, which is concerned with what a system is capable of doing. One of the most important advantages in using an object-oriented approach is the ability to reuse components. Traditional practices surrounding development often mitigate against reuse. Short-term goals are stressed because today's milestones must be achieved before any thought can be given to milestones that may be months or years away.

Borrowed or reused software code is often code that has already been tested, and in the end, may translate into cost savings. Object-oriented development may make code reuse much easier but, the amount of actual reuse may still depend on the motivation of the project managers, designers and developers involved. Code reuse can also lead to faster development. Object-oriented systems are easier to maintain because their structures are inherently decoupled. This usually leads to fewer side effects when changes have to be made. In addition, object-oriented systems may be easier to adapt and scale (i.e., large systems can be created by assembling reusable subsystems).

Typically, the object-oriented process follows an evolutionary spiral that starts with customer communication, where the problem is defined. The technical work associated with the process follows the iterative path of analysis, design, construction, and testing. The fundamental core concepts in object-oriented design involve the elements of classes, objects, and attributes. Understanding the definition and relationships of these elements is crucial in the application of object-oriented technologies.

It is recommended that the following object-oriented issues be well understood in order to form a knowledge base for the analysis, design, testing, and implementation of systems using object-oriented techniques.

- What are the basic concepts and principles that are applicable to object-oriented thinking?
- How should object-oriented projects be planned and managed?
- What is object-oriented analysis and how do its various models enable a systems engineer to understand classes, their relationships and behavior?
- What is a “use case” and how can it be applied to analyze the requirements of a system?
- How do conventional and object-oriented approaches differ?
- What are the components of an object-oriented design model?
- How are “patterns” used in the creation of an object-oriented design?
- What are the basic concepts and principles that are applicable for testing of object-oriented systems?
- How do testing strategies and test case design methods change when an object-oriented system is considered?
- What technical metrics are available for assessing the quality of object-oriented systems?

Work Product: The work products described in the SEM will be the same for many of the development techniques and it is the responsibility of the project manager to adapt the work products accordingly and document adaptations in the Project Plan.

References: SOM Project Management Methodology:
http://www.michigan.gov/projectmanagement
See Section 3 – Project Planning Phase

Section: **2.4 Commercial-Off-The-Shelf (COTS) Products Based Projects**

Description: There is a current trend in information systems development to make greater use of Commercial-Off-The-Shelf (COTS) products, that is, to buy a ready-made system from a software manufacturer rather than developing it in-house from scratch. This carries with it a sense of getting a system that can do the job at a reasonable cost, and getting new functions in subsequent releases over time. This practice is especially encouraged and sometimes mandated in government agencies. There can be many benefits in using COTS products including improving quality and performance, developing and delivering solutions more quickly, maintaining systems more cost effectively, and standardizing across the organization. The main characteristics of a COTS product are that it exists, is known to be proven, is available to the general public, and can be bought, leased, or licensed.

COTS and Open Systems: Many initiatives are under way in both private industry and government agencies including the State of Michigan to promote the use of an open systems approach, thereby anticipating even greater benefits than can be obtained from the use of COTS products alone. These initiatives are occurring because just buying COTS does not necessarily result in an "open" system. COTS products are not necessarily open, and they do not necessarily conform to any recognized interface standards. Therefore, it is possible that using a COTS product commits the user to proprietary interfaces and solutions that are not common with any other product, component, or system.

If the sole objective is the ability to capture new technology more cheaply, then the use of COTS products that are not open may satisfy requirements. However, considering that the average COTS component is upgraded every 6 to 12 months and new technology appears on the scene about every 18 to 24 months, any money that is saved by procuring a COTS product with proprietary interfaces may quickly be lost in maintenance as products and interfaces change.

In the midst of all this, interface standards provide a source of stability. Without such standards every change in the marketplace can impose an unanticipated and unpredictable change to systems that use products found in the marketplace.

COTS Planning Considerations**:** A COTS-based systems solution approach requires new and different investments including market research on available and emerging products and technologies, and COTS product evaluation and selection. The key to determining if the best solution is one which includes COTS products is to weigh the risks of straying from the three basic criteria - fully-defined, available to the public, and maintained according to group consensus - against what is to be gained over the long-term. An open systems approach requires investments in the following areas early in a project's lifecycle and on an ongoing basis:

- Market surveys to determine the availability of standards
- Selection of appropriate applicable standards
- Selection of standards-compliant implementations

These costs/activities are the necessary foundation for creating systems that serve current needs and yet can grow and advance as technology advances and the marketplace changes. On an ongoing basis, it is important for project teams to stay informed in this area, with particular focus on:

- When revisions to specific standards are scheduled for release
- What changes are proposed in the new revision
- When ballots on the revisions are going to occur
- Where the implementations are headed

Skills Considerations: The depth of understanding and technical and management skills required on a project team are not necessarily diminished or decreased because of the use of COTS or open systems. The skills and understanding needed increase because of the potential complexity of integration issues, the need to seriously consider longer-term system evolution as part of initial development, and the need to make informed decisions about which products and standards are best.

Types of COTS Solutions: COTS products can be applied to a spectrum of system solutions, including (but not limited to) the following:

- Neatly packaged solutions such as Microsoft Office that require no integration with other components.
- COTS products that support the information management domain, such as Oracle or SQL Server. These systems typically consist of both COTS products and customized components, with some "glue" code to enable them to work cooperatively.
- Systems comprised of a mix of COTS products and non-commercial products that provide large-scale functionality that is otherwise not available. Such systems typically require larger amounts of "glue" code to integrate the various components.

COTS Impact on the Project Lifecycle:

All systems engineering projects include planning, requirements definition, architecture definition, system design, code, test, and system integration activities. The use of COTS products has an impact on project lifecycle activities. The most fundamental change is that the system is now composed from building blocks that may or may not work cooperatively directly out of the box. The project team will require skilled engineering expertise to determine how to make a set of components work cooperatively - and at what cost.

This fundamental shift from development to composition causes numerous technical, enterprise, management, and business changes. Some of these changes are obvious, whereas others are quite subtle.

Requirements Definition

For a COTS-based system, the specified requirements must be sufficiently flexible to accommodate a variety of available commercial products and their evolution. To write such requirements, the author should be sufficiently familiar with the commercial marketplace to describe functional features for which actual commercial products exist.

There is a critical relationship among technology and product selection, requirement specification, and architecture definition. If the architecture is defined to fulfill the requirements and then the COTS product is selected, there may be only a few or no available products that fit within the chosen architecture. Pragmatically, three essential elements--requirements, architecture, and product selection--must be worked in parallel with constant trade-offs among them.

Adaptation/Integration

Assembling COTS products presents new challenges. Although COTS products are attempting to simulate the "plug and play" capability of the hardware world, in reality, they seldom plug into anything easily. Most products require some amount of adaptation and integration to work harmoniously with other commercial or custom components in the system. The typical solution is to adapt each COTS product through the use of "wrappers," "bridges," or other "glueware." It is important to note that adaptation does not imply modification of the COTS product. Adaptation can be a complex activity that requires technical expertise at the detailed system and specific COTS component levels. Adaptation and integration must take into account the interactions among custom components, COTS products, any non-developmental item components, any legacy code, and the architecture including infrastructure and middleware elements.

Testing

As the testing of COTS-based systems is considered, it must be determined what levels of testing are possible and needed. A COTS product is a "black box" and therefore changes the nature of testing. A system may use only a partial set of features of a given COTS product. In developing a test strategy and test plans, consideration should be given to issues such as should only the features used in the system be tested, and how does one test for failures in used features that may have abnormal behavior due to unknown dependencies between the used and unused features of a COTS product?

Maintenance

Maintenance also changes in very fundamental ways; it is no longer solely concerned with fixing existing functionality or incorporating new mission needs. Vendors update their COTS products on their schedules and at differing intervals. Also, a vendor may elect to eliminate, change, add, or combine features for a release. Updates to one COTS product, such as new file formats or naming convention changes, can have unforeseen consequences for other COTS products in the system. To further complicate maintenance, all COTS products will require continual attention to license expirations and changes. All of these events routinely occur. All of these activities may (and typically do) start well before an organization installs the system or a major upgrade. Pragmatically, the distinction between development and maintenance all but disappears.

Adapting the SEM for COTS Projects:

All systems engineering projects have a project lifecycle, require project management activities such as project planning, requirements definition, project tracking, software configuration management, and quality assurance; and produce deliverables such as project plans, requirements specifications, software configuration management plans, and test plans. At the same time, each project, whether COTS or traditional, can vary in scope, duration, technology used or operating platform. The SEM can be used as the project lifecycle for COTS-based projects as well as for traditional systems development and maintenance projects where all of the code is developed “in-house.”

The key to using the SEM effectively for COTS projects lies in adapting the lifecycle stages and deliverables to best suit the individual needs and characteristics of each particular project. See *Exhibit 2.4-1, Example of SEM Adapted for COTS Projects*, for an example of how to adapt the SEM for a COTS project. Stages should be combined as appropriate if, for example, a project will have a relatively small scope, and/or short duration, and/or will use known technology. On the other hand, the traditional number of stages may be appropriate for large projects with new technology and long duration.

Deliverables may be added to, or deleted from the standard list prescribed by the SEM (see *Exhibit 2.0-1, SEM Overview Diagram* on page 13). For COTS-based projects, the lifecycles stages will typically include "evaluation," "selection," "customization," and "integration," and the project deliverables will typically include documents such as "Products to be Evaluated," and "COTS Solution Recommendations." See *Exhibit 2.4-1 Example of SEM Adapted for COTS Projects.*

Documenting Deviations: The adaptation (or deltas) from the standard SEM prescribed stages and deliverables are known as deviations. These deviations should be documented with an explanation in the project plan. Deviations from prescribed project deliverables should be documented with an explanation, and a statement, which describes how project risk is not elevated if a prescribed deliverable will not be produced.

Resources: The following references are from the features section of the Carnegie Mellon University Software Engineering Institute Website and were used in the preparation of this chapter:

- Software Technology Review: COTS and Open Systems
- Monthly Features: The Opportunities and Complexities of Applying COTS
- Monthly Features: Discussion with Members of the SEI COTS-Based Systems Initiative
- Software Technology Review: Components-based Software Development/COTS integration

Exhibit 2.4-1 Example of SEM Adapted for COTS Projects

SEM Stages	SEM/COTS Project Stages	Deliverables: SEM/COTS Project Planned Deliverables	Deliverables: Adaptation vs. SEM Deliverables
Initiation & Planning	Initiation & Planning	• Security Plan (PMM) • Project Plan (includes WBS) (PMM)	• Prototype instead of Concept Document (PMM) • Software Configuration Management Plan moved to Requirements Definition
Requirements Definition	Requirements Definition	• Functional Requirements Document • Business Continuity Plan • Products to be Evaluated • Software Configuration Mgmt Plan • Preliminary Data Requirements • Requirements Traceability Matrix	• The System and Acceptance Test Requirements will be developed in the COTS Evaluation and Selection Stage
Functional Design System Design	Evaluation and Selection	• COTS Solution/Recommendation • System Architecture • System/Acceptance Test Requirements • Conversion Plan	• Functional Design and System Design stages are combined • System Architecture document replaces System Design document • Logical Model, Physical Model, Construction Specifications, Coding Practices not applicable
Construction Testing	Customization, Testing	• Solution Baseline • Training Plan • User Documentation • System Maint. Documentation • Transition Plan • Test Type Approach and Reports • Test Cases • Installation Plan	• The Construction and Testing stages are combined • Integration portion of the Test Plan not required
Implementation	Implementation	• User Training Materials • Operational System	• No Deviations

Section: **2.5 Quality Reviews**

Description: This section describes the quality review and assurance mechanisms that are used with the SEM. The purpose of quality reviews is to assure that the established information systems development and project management processes and procedures are being followed effectively, and that exposures and risks to the current project plan are identified and addressed. The quality reviews facilitate the early detection of problems that could affect the reliability, maintainability, availability, integrity, safety, security, or usability of the software product. The quality reviews enhance the quality of the end work products and deliverables of a project.

Work products are subject to quality reviews. Quality reviews are conducted as Peer Reviews, Structured Walkthroughs (SWT) and Stage Exits. The quality review used depends on the work product being reviewed, the point of time within the project stage, and the purpose of the review.

Review Process: **Peer Review**

A peer review is an informal review of information systems engineering work products, including documentation, which can be conducted at any time at the discretion of the work product developer. These informal reviews are performed by the developer's "peers"-- frequently other developers working on the same project. Informal reviews can be held with relatively little preparation and follow up activity. Review comments are informally collected and the product developer determines which comments require future action. Some of the work products prepared are considered interim work products as they feed into a major deliverable or into another stage. Interim work products are ideal candidates for peer review; however, all work products benefit from peer reviews.

Responsibility

Team Members

Review Process: **Structured Walkthrough**

The Structured Walkthrough (SWT) is a more formal review and is prescribed by the SEM for all project deliverables. SWTs are used to find and remove errors from work products early and efficiently, and to develop a better understanding of defects that might be prevented. They are very effective in identifying design flaws, errors in analysis or requirements definition, and validating the accuracy and completeness of deliverable work products.

SWTs are conducted during all stages of the project lifecycle. They are used during the development of work products identified as deliverables for each stage (see *Exhibit 2.0-1 SEM Overview Diagram* on page 13), such as requirements, specifications, design, code, test cases (scripts), and documentation. SWTs are used after the work products have been completed to verify the correctness and the quality of the finished product. They should be scheduled in the work breakdown structure developed for the project plan, where, in practice, they are sometimes referred to generically as reviews. SWTs should also be scheduled to review small, meaningful pieces of work. The progress made in each lifecycle stage should determine the frequency of the walkthroughs; however, they may be conducted multiple times on a work product to ensure that it is free of defects.

SWTs can be conducted at various times in the development process, in various formats, with various levels of formality, and with different types of participants. They typically require some advance planning activities, a formal procedure for collecting comments, specific roles and responsibilities for participants, and have prescribed follow-up action and reporting procedures. Frequently reviewers include people outside of the developer's immediate peer group.

Responsibility
Project Manager, Team Members, Work Product Author, Reviewers

Work Products
A SWT Meeting Record (DIT-0187) and the Error Tracking Log (SEM-0186) are available for the reviewers to record errors found prior to the walkthrough session, and for the scribe to record information discussed during the walkthrough.

Reference
The State of Michigan guidance document titled *Structured Walkthrough Process Guide* provides a procedure and sample forms that can be used for SWTs. This document is available on the MDIT SUITE website.

Review Process:

Stage Exit
The Stage Exit is a process for ensuring a project meets the project standards and milestones identified in the project plan. The Stage Exit is conducted by the project manager with the project stakeholders (e.g., system owner and the following points of contact: user, quality assurance, security, architecture and standards, project manager’s manager, and platform). It is a high-level evaluation of all work products developed in a lifecycle stage. It is assumed that each deliverable has undergone several peer reviews and/or SWTs as appropriate prior to the Stage Exit process. The Stage Exit focuses on the satisfaction of all requirements for the stage of the lifecycle, rather than the specific content of each deliverable.

The goal of a Stage Exit is to secure the approval of designated key individuals to continue with the project and to move forward into the next lifecycle stage. The approval is a sign-off of the deliverables for the current stage of development including the updated project plan. It indicates that all qualifications (issues and concerns) have been closed or have an acceptable plan for resolution. At a Stage

Exit meeting, the project manager communicates the positions of the key personnel, along with qualifications raised during the stage exit process, and the action plan for resolution to the project team, stakeholders, and other interested meeting participants. The Stage Exit meeting is documented in summary form. Only one Stage Exit for each stage should be necessary to obtain approval assuming all deliverables have been accepted as identified in the project plan.

A Stage Exit is an effective project management tool that is required for all projects regardless of size. For small projects, stages can be combined and addressed during one Stage Exit.

Responsibility
Project Manager.

Work Products
A Stage Exit Approvals document (DIT-0189) is completed. A summary of the Stage Exit meeting is prepared by the project manager or a designee and distributed to the meeting attendees. The summary identifies any issues and action items needed to obtain concurrence prior to proceeding to the next lifecycle stage.

Reference: The MDIT guidance document titled *Stage Exit Process Guide* provides a procedure and sample report form that can be used for Stage Exits. This document is available on the MDIT SUITE website.

Refer to the PPQA Process Manual for external project reviews.