

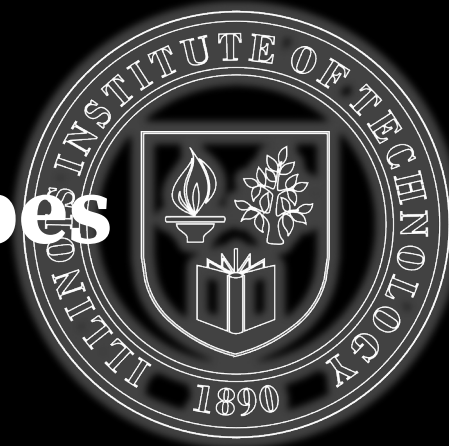
ILLINOIS TECH

College of Computing

ITMD 536 Software Testing & Maintenance

Chapter 3 & 4

Maintenance Pie and Ten Success Recipes



Objectives

- ◆ What is Work Breakdown Structure?
- ◆ What is software operations maintenance?
- ◆ How can you distribute the activities?
- ◆ How are resource needs handled?
- ◆ What is the success formula?
- ◆ What are the key process areas in software maintenance?(CMMi & S3m)



Objectives

- ◆ Why do you need to balance agility and discipline?
- ◆ How to emphasis on managing the work?
- ◆ How to establish proper infrastructure?
- ◆ How to deal with operational restrictions?
- ◆ What are the ten success recipes?
- ◆ What are the key process areas (KPA)?



3.1 Work Breakdown Structure

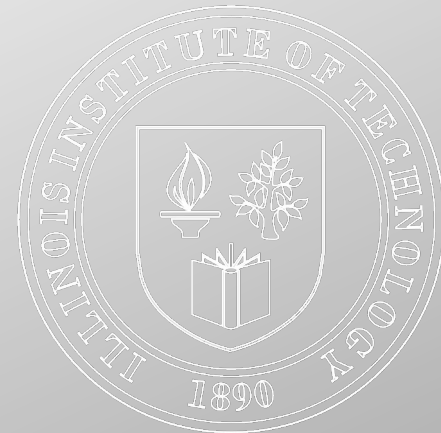
- ♦ **Work breakdown structure (WBS):** Family tree that organizes, defines and graphically illustrates the products, services, and tasks necessary to achieve project objectives.
- ♦ The goal to research and investigate was to find out more how the software maintenance was processed on day to basis.
- ♦ How it was planned, organized, staffed and executed?
- ♦ Findings were the system was not build with security in mind.
- ♦ To retrofit security workload will be huge.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.1 Maintenance

- 1.1.1 Release Requirements
- 1.1.2 Release Planning
- 1.1.3 Architecture Analyst
- 1.1.4 Hardware Defect Repair
- 1.1.5 Software Defect Repair
- 1.1.6 Hardware Enhancement
- 1.1.7 Software Enhancement
- 1.1.8 Release Integration and Test
- 1.1.9 Release Qualification and Delivery



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

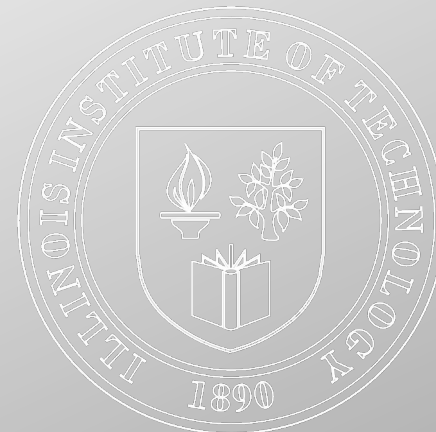
1.2 Sustaining Engineering

1.2.1 Analysis and Studies

1.2.2 Emergency Repair

1.2.3 User Training

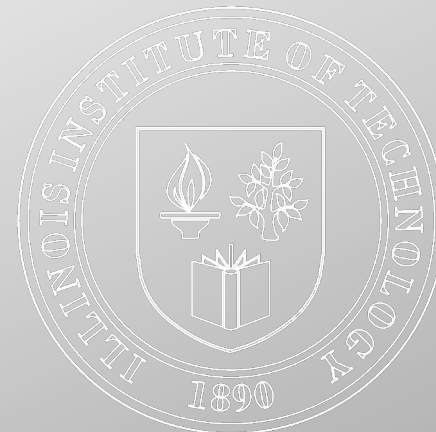
1.2.4 External Support



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.3 Independent Test and Verification

- 1.3.1 Test Planning
- 1.3.2 Test Preparation
- 1.3.3 Test Conduct
- 1.3.4 Independent Analysis and Verification



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.4 Product Support

1.4.1 Configuration Management

1.4.2 Quality Assurance

1.4.3 Peer Reviews

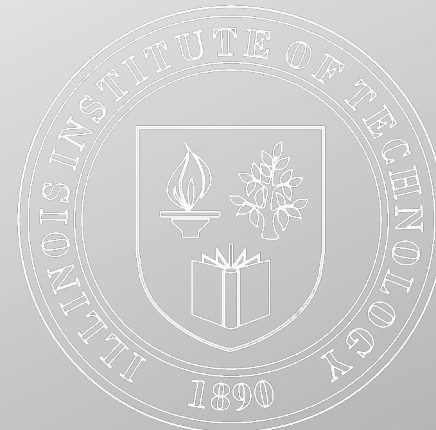
1.4.4 Supplier Management

1.4.5 Security

1.5 Information Assurance

1.5.1 Protection Services

1.5.2 Certification and Accreditation



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.6 Acquisition Support

1.7 Operations Support

1.8 Facility Support

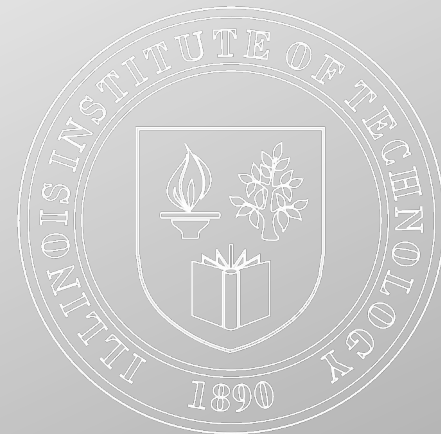
1.8.1 Maintenance Facility Sustainment

1.8.2 System Integration Lab Sustainment

1.8.3 Equipment Sustainment

1.8.4 Specialized Test Equipment and Tools Sustainment

1.8.5 Network Operations and Administration



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.9 Field Support

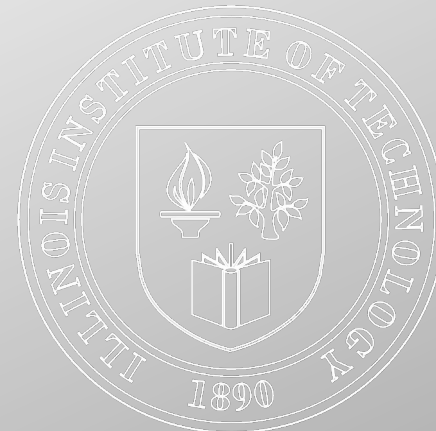
1.10 Management

1.10.1 Release Management

1.10.2 Sustaining Engineering Management

1.10.3 Risk Management

1.10.4 Measurement Analysis



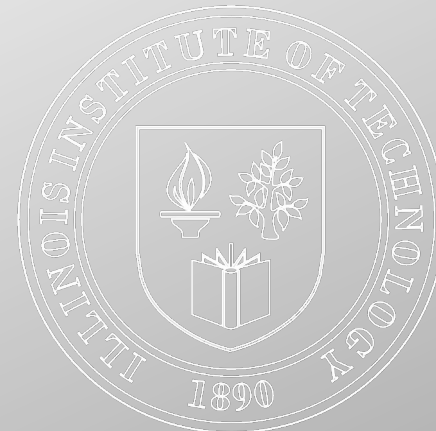
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.11 Parts

1.12 Spares

1.13 Licenses

1.14 Other



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.0 Operations, Maintenance, and Support

- This WBS identifies all of the possible work associated with maintaining a system after it has been transitioned and turned over to the maintenance organization once development has been completed.

1.1.1 Release Requirement

- This task develops block release requirements formulated based on user requests and problem analysis.

1.1.2 Release Planning

This task develops plans, budgets, and schedules for the block release.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.1.3 Architecture Analysis

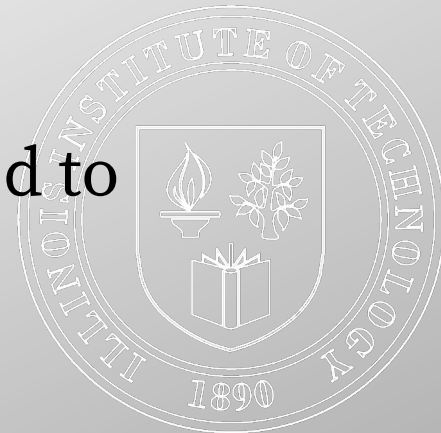
This task performs the architecture analysis to determine what architecture and design modifications are needed to satisfy the requirements.

1.1.4 Hardware Defect Repair

This task does all of the engineering and test work.

1.1.5 Software Defect Repair

- This task does all of the engineering and test work needed to make necessary software repairs as part of the release.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.1.6 Hardware Enhancement

This task makes hardware enhancements and perfective changes called out by the release plans. We have found that the software staff is often called upon to perform hardware tasks especially when COTS platforms are used. It includes all of the engineering and test work required to satisfy the release requirements.

1.1.7 Software Enhancements

- This task makes the software enhancements and perfective changes called out by the release plans. It includes all of the engineering and test work required to satisfy the release requirements.



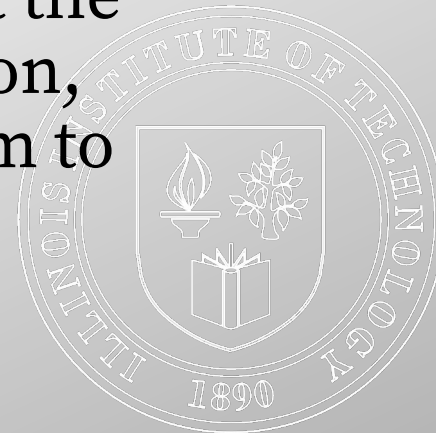
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.1.8 Release Integration and Test

- This task integrates the release and performs testing. Integration involves putting the pieces together and making sure that they work operationally as intended.

1.1.9 Release Qualification and Delivery

- This task qualifies the release and delivers it to the field. It performs some form of acceptance review to ensure that the release and all required support materials (documentation, configuration indices, etc.) are distributed in proper form to receiving sites.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.2 Sustain Engineering

- This activity refers to all of the works performed to sustain the release in the field. Such a work includes analysis and studies, emergency repairs, and user hand-holding and support.

1.2.1 Analysis and Studies

- This task conducts those analysis and studies needed to understand and provide fixes for operational issues and problems

1.2.2 Emergency Repairs

- This task makes those emergency repairs needed to keep the system operational. The task includes those efforts associated with developing and delivering patch releases to the field.



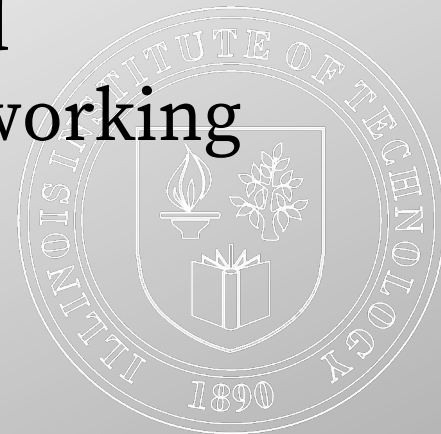
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.2.3 User Training

- This task provides mentoring and training for users. It may include developing training courses and related manuals.

1.2.4 External Support

- This task provides user, customer, and other forms of external support. It may also include developing and maintaining a website and some forms of social networking (Twitter, etc.).



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.3 Independent Test and Verification

This activity independently verifies and validates the system as releases are prepared typically by third parties (vendors, contractors, etc.). Such verification activities can range from independent testing to detailed analysis of both designs and code on a separately maintained test bench. The activity assumes that the test and verification of the block release is accomplished satisfactorily as part of the maintenance activity effort.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.3.1 Test Planning

- This task prepares test plans to perform independent test and verification activities.

1.3.2 Test Preparation

- This task develops test cases and scenarios for performing independent test and verification activities and the test tools needed to run them.



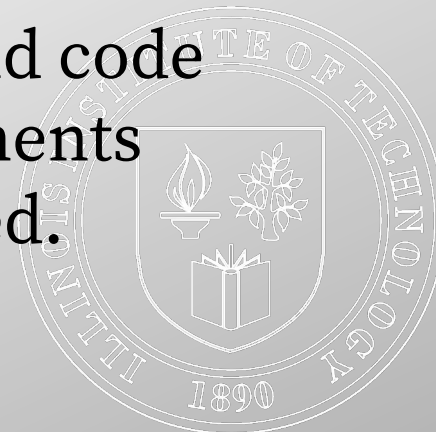
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.3.3 Test Conduct

- This task conducts the tests, captures results, verifies that release requirements are satisfied, and develops regression test baselines for use in revalidating the system when future changes are made.

1.3.4 Independent Analysis and Verification

- This activity performs the detailed analysis of designs and code needed to provide additional confirmation that requirements including those for security and safety have been satisfied.



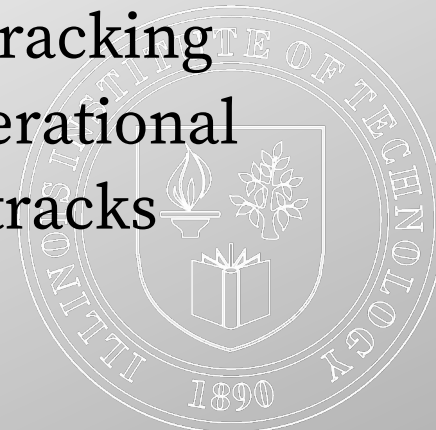
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.4 Product Support

- This activity maintains the overall integrity and quality of the processes, products, and supplier networks employed during the operations and maintenance phase

1.4.1 Configuration Management

- This task performs configuration management actions including those associated with Change Control Board (CCB) operations and tracking configurations, spares, licenses, and parts among various operational and support sites. It also distributes versions to the field and tracks site configurations.



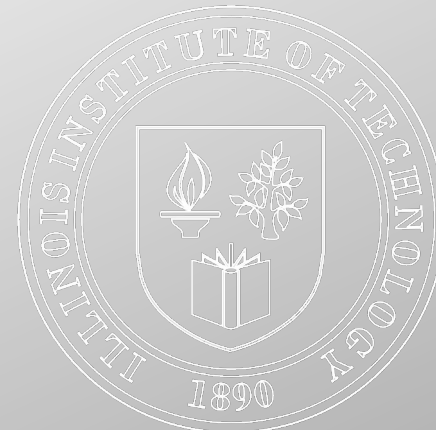
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.4.2 Quality Assurance

- This task performs quality assurance actions aimed at ensuring the quality of the products and integrity of the processes used for maintenance, operations, and support.

1.4.3 Peer Reviews

- This task conducts peer reviews on the products and processes including disposition of issues found.



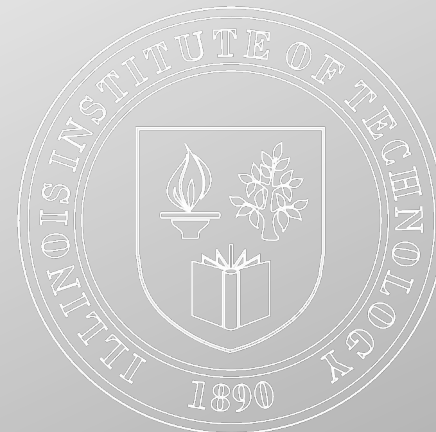
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.4.4 Supplier Management

- This task handles management of suppliers including those that provide parts, spares, and software licenses.

1.4.5 Security

- This task addresses security requirements for the project including those associated with planning, training, storage, administration, support, and security controls.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.5 Information Assurance

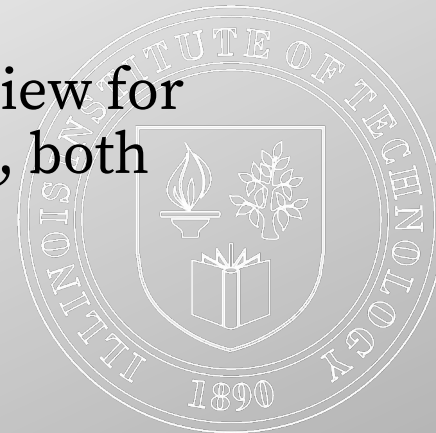
- This activity performs information assurance task including those associated with product and computer network protection.

1.5.1 Protection Services

- This task develops product protection including interesting any defenses associated with antitamper and maintaining any Secure Compartmented Information Facilities (SCIFs).

1.5.2 Certification and Accreditation (C&A)

- This task performs any required certification and accreditation review for those computer networks used to maintain and operate the system, both operational and support systems.



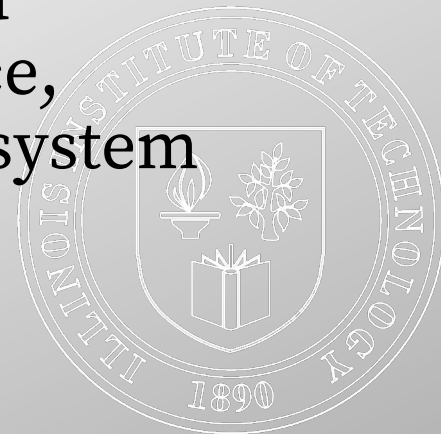
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.6 Acquisition and Management

- This activity provides acquisition management tasks. Such effort often occurs when managing third parties doing maintenance activities, both-on-site and at remote sites.

1.7 Operations Support

- This activity supports operations in the field including those efforts associated with supporting the “to be fielded” and “fielded” releases and with assessing system performance, database maintenance, configuration management and system administration.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.8 Facility Support

- This activity readies and maintains those development and test facilities needed to support and sustain the software in the field.

1.8.1 Maintenance Facility Sustainment

- This task readies and maintains a maintenance facility that is used to develop updates for and sustain the system once it is fielded.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.8.2 System Integration Lab (SIL)

- This task readies and maintains a System Integration Lab (SIL) that is used to test and evaluate new releases destined for the field under realistic operating conditions (using actual operational hardware in the loop and on-site user representatives, if possible).

1.8.3 Equipment Sustainment

- This entry task sets up, configures, and keeps the hardware used in the maintenance facility or SIL operational. Such equipment includes any specialized gear for security such as firewalls.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.8.4 Specialized Test Equipment and Tools

- This task sets up, configures, and keeps specialized test equipment and tools used in the maintenance facilities or SIL operational. Such equipment includes specialized gear like performance monitors.

1.8.5 Network Operations and Administration

- This task sets up, configures, manages, administers, and maintains the computer networks used for maintain, operating, and supporting the system in the field.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.9 Field Support

- This activity conducts field support including sending personnel to the field to investigate and fix problems.

1.10 Management

- This activity manages release and sustaining engineering activities and conducting risk and metric analysis.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.10.1 Release Management

- This task manages the generation and test of block releases and ensures that they satisfy user expectations.

1.10.2 Sustaining Engineering Management

- This task manages sustaining engineering efforts including those associated with independent testing; independent verification, acquisition, product, field, and operations facility support; and information assurance.



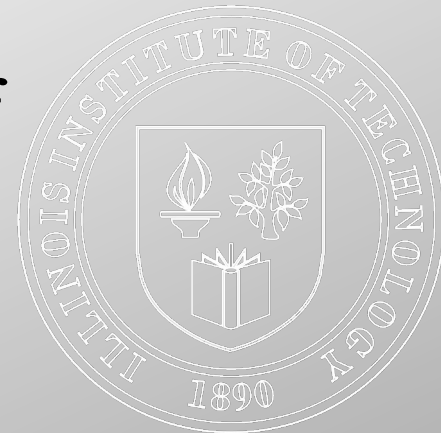
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.10.3 Risk and Opportunity Management

- This task plans for and performs risk and opportunity management activities for the project including mitigation actions.

1.10.4 Measurement Analysis

- This task collects analyzes, and reports the results of metrics analysis.



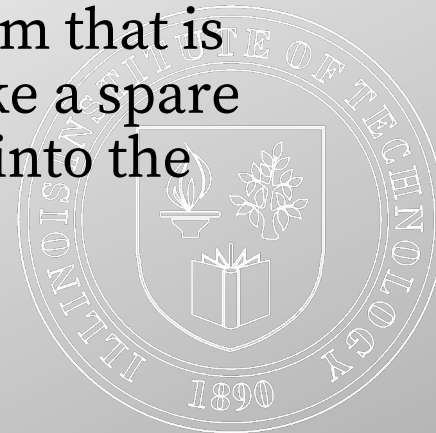
3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.11 Parts

- This activity acquires, packages, transports, and stores replacement parts, components, and subassemblies to field sites. Such parts can include hardware (extra hard drives, routers, etc.) and software components (recovery DVDs needed to rebuild the system, etc.)

1.12 Spares

- This activity acquires, packages, transports, and stores spares to field sites. Such spares include hardware and software that is part of the system that is essential for its included as assemblies. For example, you may make a spare processor with software loaded available for immediate switching into the system if it has to be operational twenty-four/seven.



3.1.1 Software Operations, Maintenance, and Support Work Breakdown Structure (WBS)

1.13 Licenses

- This activity manages software licenses and conducts those efforts needed to maintain market watch and vendor liaison functions during the course of maintenance.

1.14 Other

- This activity accounts for any other effort not included within this breakout. For example, process improvement support provided by the project would fit in this category as would the requirement to generate project summaries and lessons learned reports.

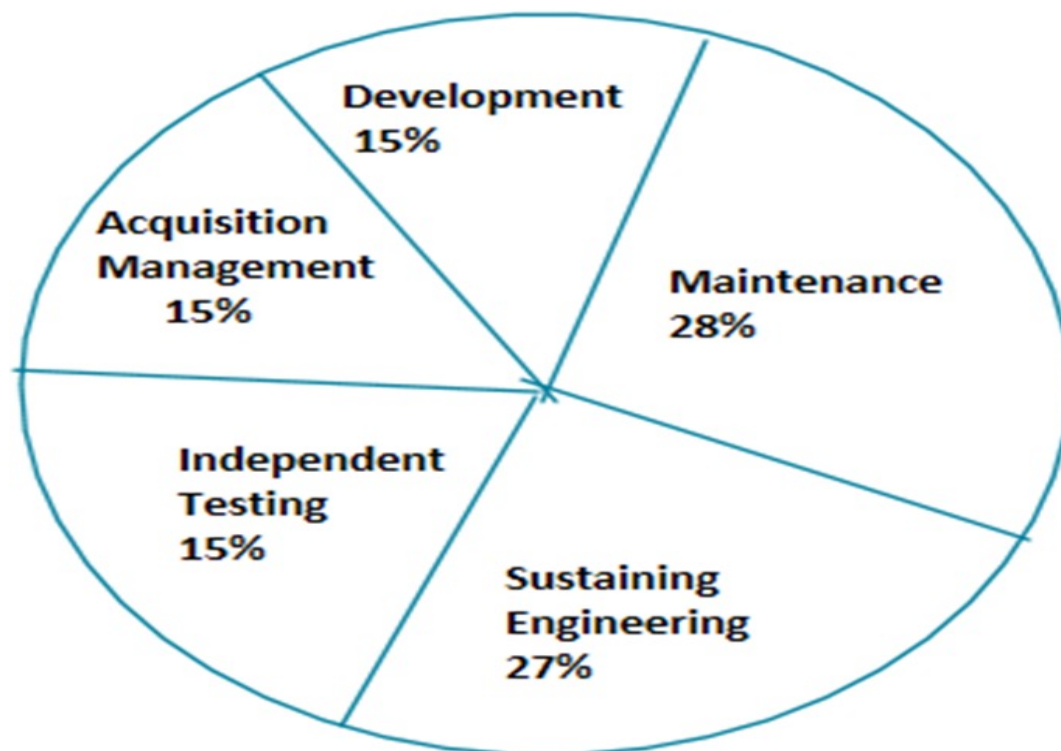


3.2 Activity Distributions

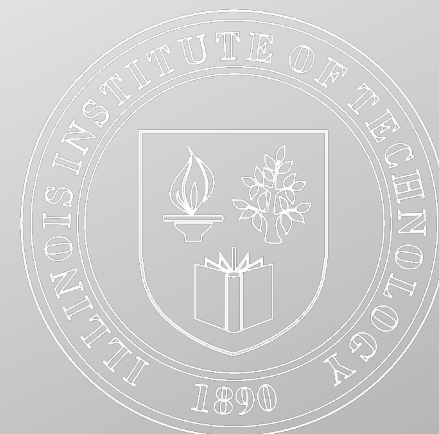
- ♦ The maintenance work dealt with supporting sustaining engineering, providing acquisition management, including contractor oversight, and performing independent test and analysis. Maintenance generate new release and maintaining the infrastructure and facilities required for the conduct of life-cycle support.
- ♦ 72% of work involves maintenance.



Effort Distribution (% Effort/Activity)



72% of work involves maintenance, Sustaining engineering and independent testing. Remaining work which can include development takes 28%

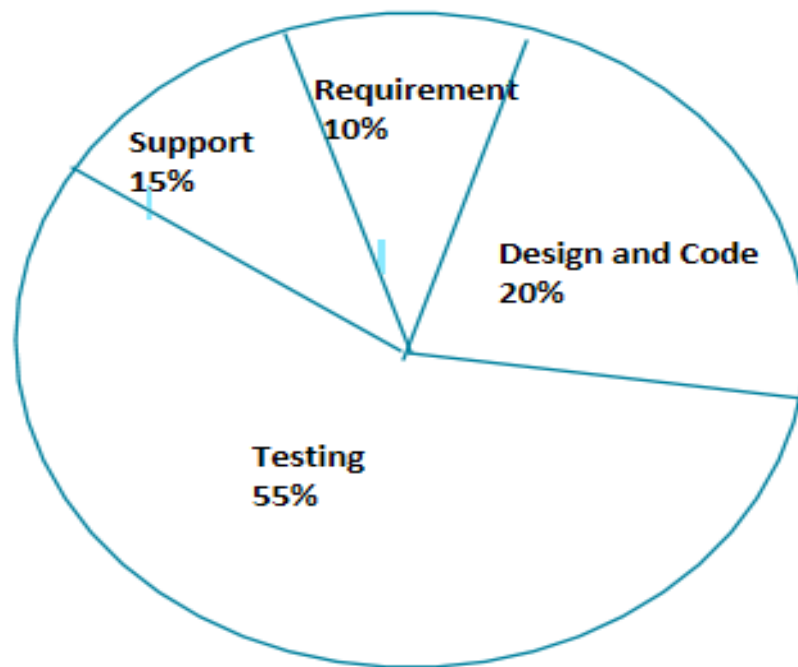


3.2 Activity Distributions

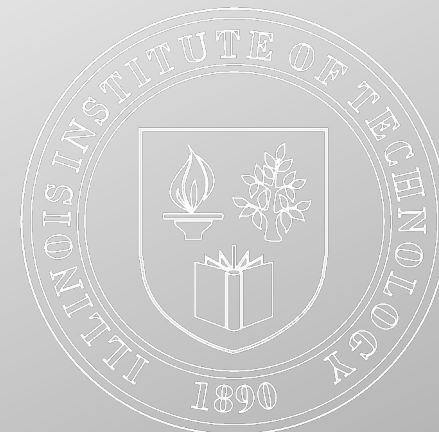
- ◆ Maintenance (% Work Done by Technical Task)
- ◆ Requirement 10%
- ◆ Design 20%
- ◆ Support 15%
- ◆ Testing 55%
- ◆ **55% -70%** or work done during maintenance supports **retesting.**



Maintenance (% Work Done by Technical Task)

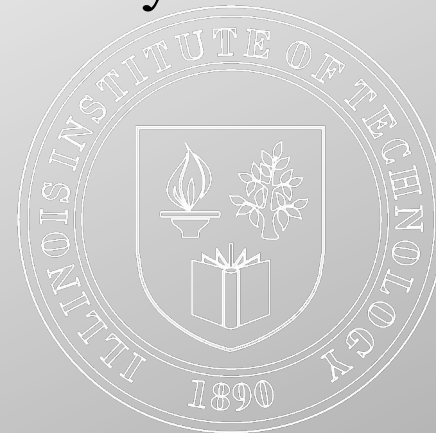


As much as 55 to 70% of the work done during Maintenance supports retesting
Support tasks are aimed at both the user and the field



3.2 Activity Distributions

- ♦ **Release Contents**
- ♦ **Enhancement:** Incorporate new functions and feature into the release based on approved change request.
- ♦ **Perfective Changes:** Making the software run more quicker or more efficiently.
- ♦ **Repairs:** Fixes incorporated to address outstanding software trouble reports.
- ♦ **Patch Releases:** Software releases sent to the field to correct minor problems.
- ♦ **Major Releases:** Software versions each released with different functionality.



3.3 Resources Needs

- ♦ Resources for maintenance projects are provided on a level-of-effort.
- ♦ 80% of maintenance projects are performed by teams of 10 or less.
- ♦ The team is split to manage block releases, sustaining engineering, independent test and evaluation, product support, field support, and other tasks that are needed to be performed.
- ♦ Licenses are needed for the application packages and systems operating systems, database managers, etc.



3.4 Success Formulas

1. Manage the software maintenance satisfactorily.
2. Structure the work
3. Recognize most of the maintenance projects are small.
4. Understand the Level Of Effort (LOE)
5. Appreciate you work with experienced associates
6. Recognize how it works operationally in targeted sites.
7. Understand you need a different mindset to succeed during software maintenance.



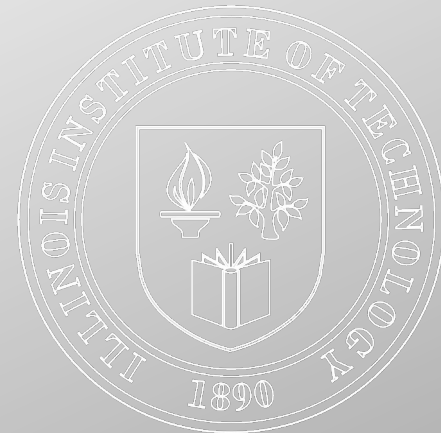
Foundations of the S3 Process Model

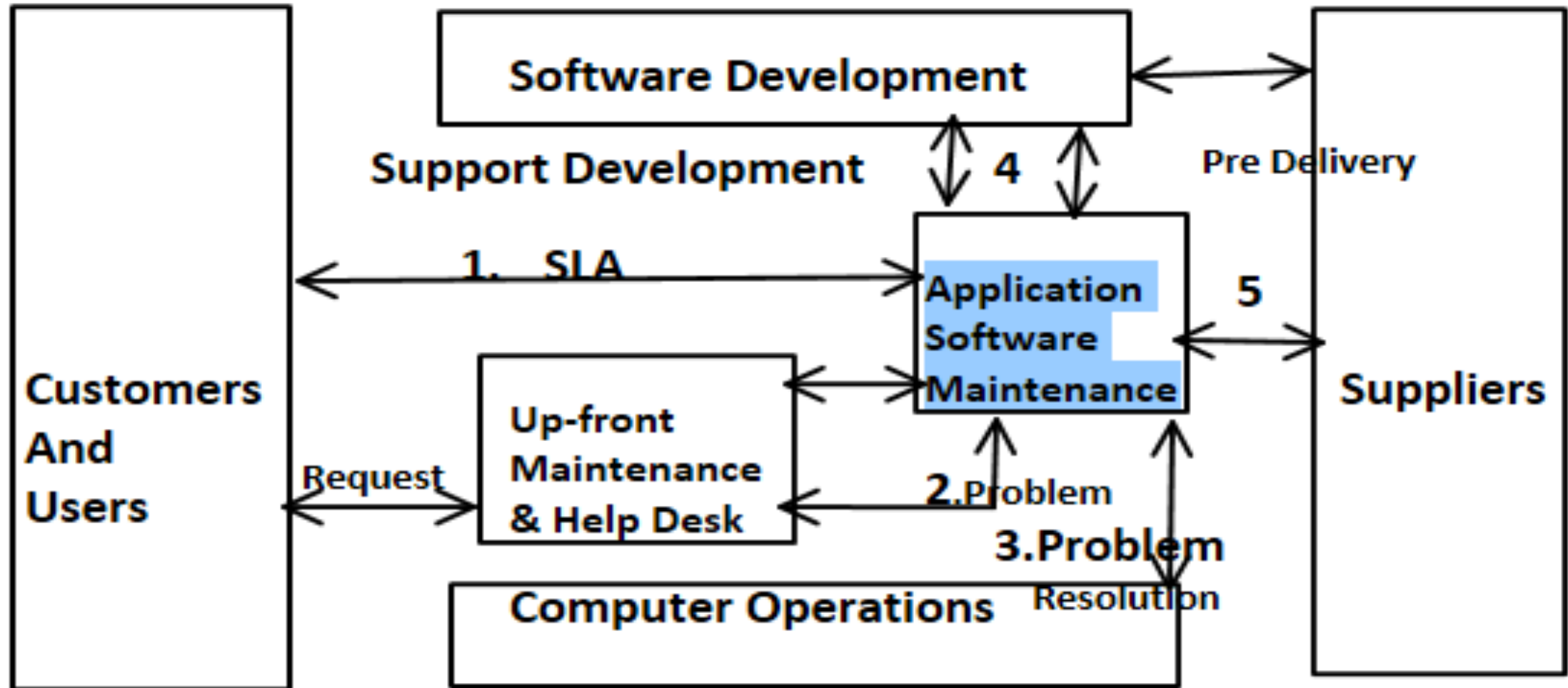
- ◆ Software Maintenance Maturity Model S3 is a structured way, organized by maturity levels.
- ◆ Organizations uses S3 model to sustain and a continuous improvement program.
- ◆ S3 is restricted to small maintenance activities. ISO 12207, 14764, 90003, and 15504 which also covers CMMi (ssm)



3.2 Context of Software Maintenance

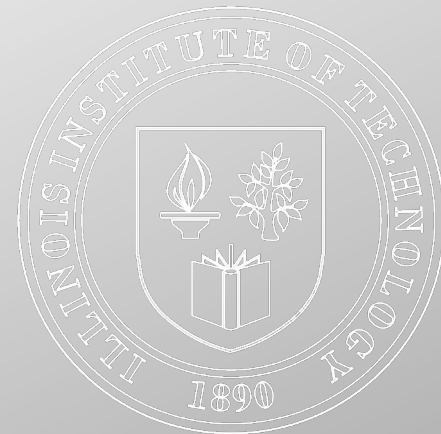
- ◆ Customers and users of a software maintenance interface (1)
- ◆ Upfront maintenance and help desk interface (2)
- ◆ Computer operations department interface (3)
- ◆ Developer interface (4)
- ◆ Supplier interface (5) (ssm)





3.3.2 Software Support Maintenance Process

- ◆ Documentation
- ◆ Configuration
- ◆ Product quality assurance
- ◆ Verification and validation
- ◆ Problem resolution process
- ◆ Reviews and audit processes



3.3 Process Domains: CMMi and S3

◆ CMMi4

- Process Domains
- Process Management
- Project Management
- Engineering
- Support

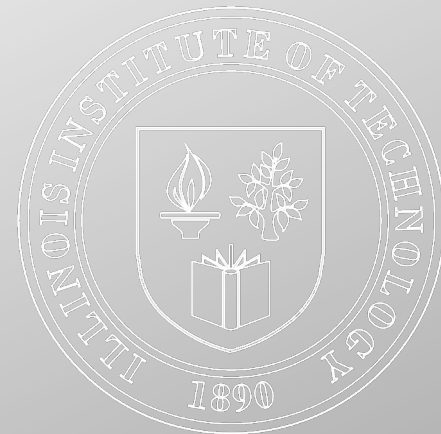
◆ S3M

- 4 Process Domains
- Process Management
- Event/Request Management
- Evolution Engineering
- Support to Evolution Engineering



3.4 Key Process Areas Software Maintenance

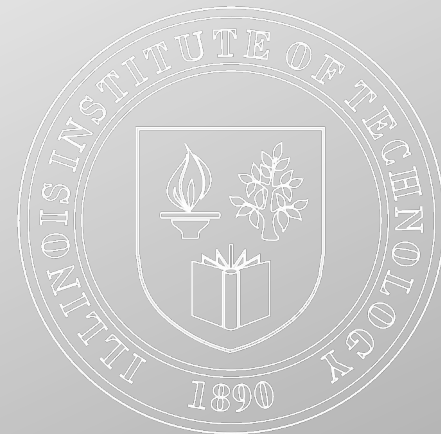
1. Operational – support and maintenance
2. Corrections process
3. Evolution process (adaptive, perfective and preventive)



3.5 Key Process Areas of Software Maintenance

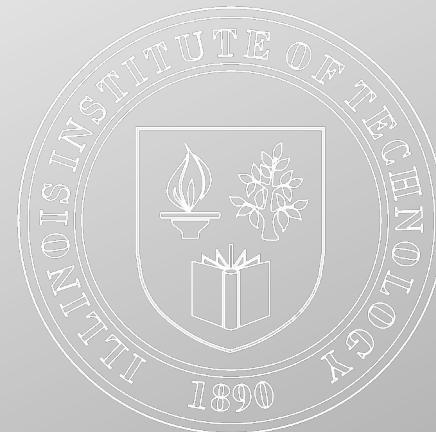
◆ Process Management

1. Maintenance process focus
2. Maintenance process/service definition
3. Maintenance training
4. Maintenance process performance
5. Maintenance innovation and deployment



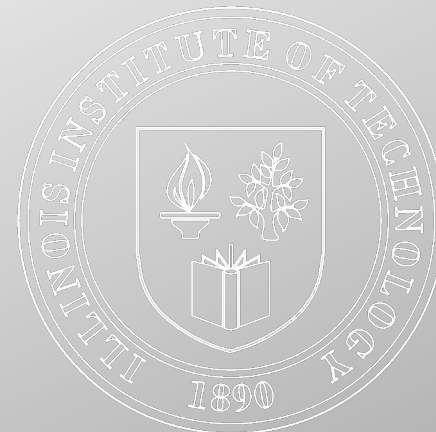
3.5 Key Process Areas of Software Maintenance

- ◆ Event/Request Management
 1. Event/request management
 2. Maintenance planning
 3. Requests/software monitoring & control
 4. SLA and supplier agreement management



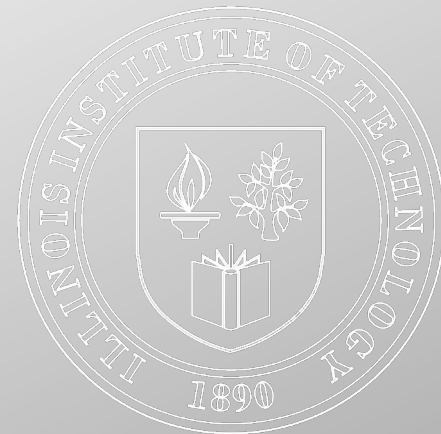
3.5 Key Process Areas of Software Maintenance

- ◆ Evolution Engineering
 1. Predelivery and transition services
 2. Operational support services
 3. Software evolution & correction services
 4. Verification and validation



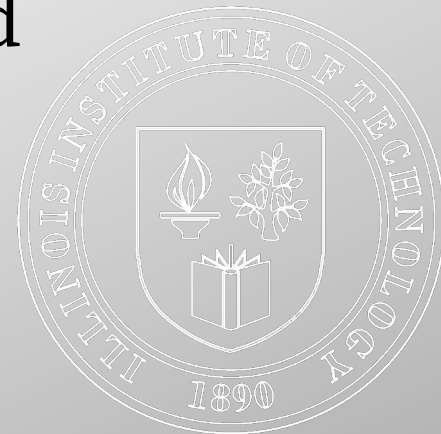
3.5 Key Process Areas of Software Maintenance

- ◆ Support to Evolution Engineering
 1. Configuration and version management
 2. Process, service and software quality assurance
 3. Maintenance measurement and analysis
 4. Causal analysis and problem resolution
 5. Software rejuvenation, migration and retirement

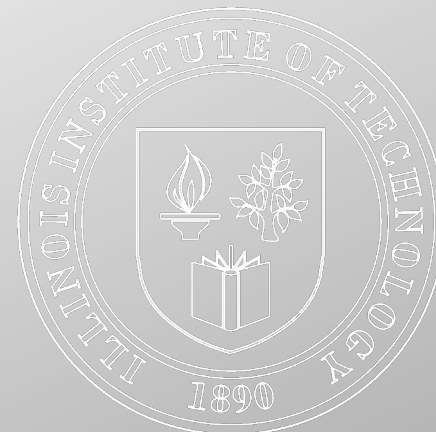
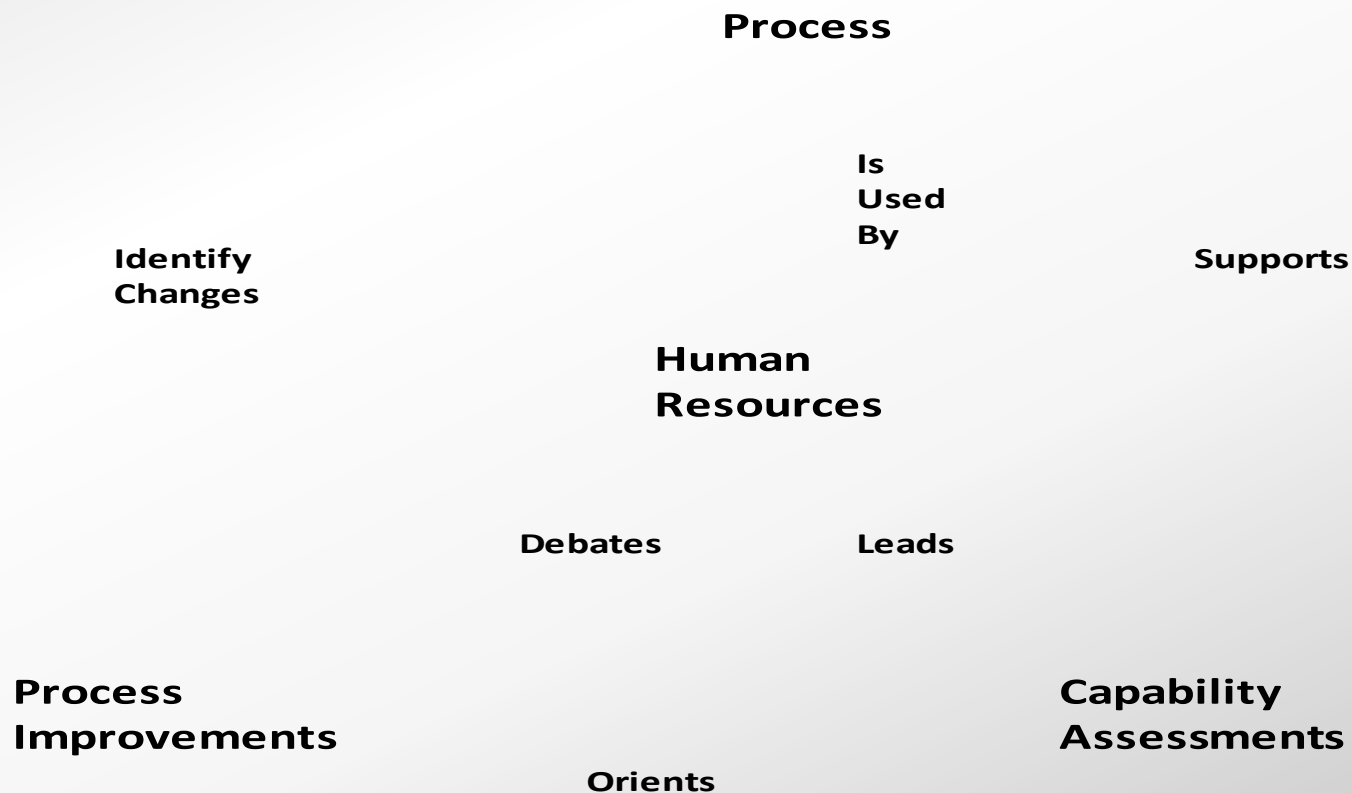


4. Process Management Domain

- ♦ The set of activities, methods, and tools applied to the definition, implementation, monitoring, and improvement of a process. Process management implies that a process is defined (since one cannot predict or control something that is undefined).
- ♦ The focus on process management implies that a program or organization takes into account both product and process-related factors in planning, performance, evaluation monitoring, and corrective action.



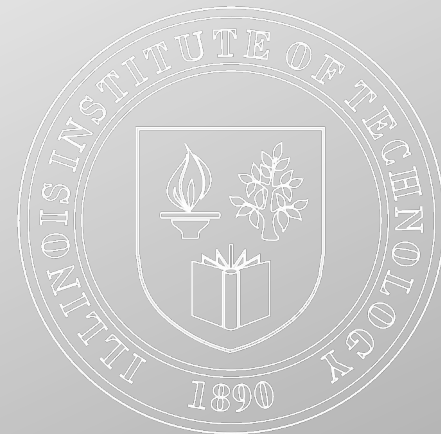
Process Management Context



Process Management Domain - 4

Process domain covers five KPAs:

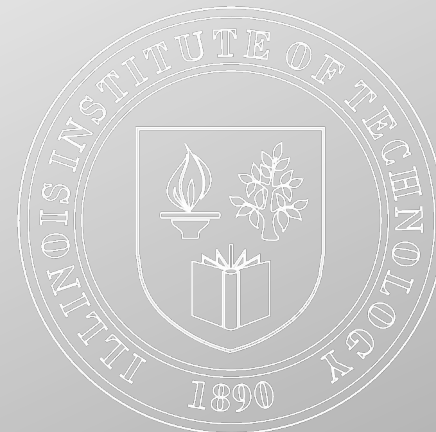
1. Maintenance Process Focus
2. Maintenance Process/Service Definition
3. Maintenance Training
4. Maintenance Process Performance
5. Maintenance Innovation and Deployment



Process Management Domain

1. Maintenance Process Focus-KPA

- Customers
- Its personnel
- Its personnel needs of additional knowledge and competencies
- The current performance of maintenance process
- The strengths/weaknesses of current maintenance techniques and tools
- The overall maintenance working environment
- Feedback



Process Management Domain

1. Maintenance Process Focus-KPA-4

Goals for this KPA

1. Identify improvements to the software maintenance processes by obtaining information from many sources, as well as comparative data.
2. Identify and deploy, in all types of maintenance services, exemplary practices that have proved successful in other organizations.
3. Establish improvement priorities, taking into account the priorities of customers, users maintenance personnel, and sponsors (IT management).



Process Management Domain

4. Establish an improvement plan that includes all the maintenance organizational units
5. Train personnel on improvement concepts and techniques.
6. Ensure that all personnel participate in the improvement efforts and ensure intergroup coordination.
7. Acknowledge contributions to improvement and quality in general



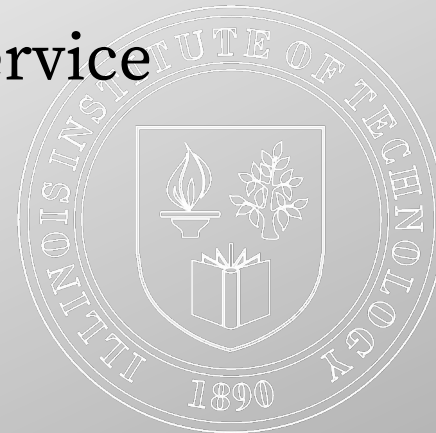
Process Management Domain

2. Maintenance Process/Service Definition

- KPA in which maintenance processes, techniques, and tools are assessed and reviewed to improve the performance of the maintenance engineer's daily tasks.

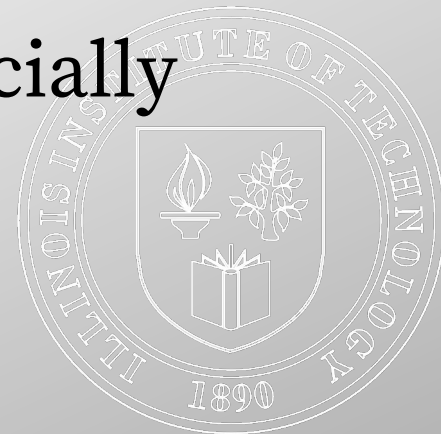
Goals for this KPA

1. Identify key software maintenance processes, activities, and services.
2. Generalized and standardize software maintenance process/services.
3. Establish guidelines for standardized maintenance process/service tailoring.



Process Management Domain

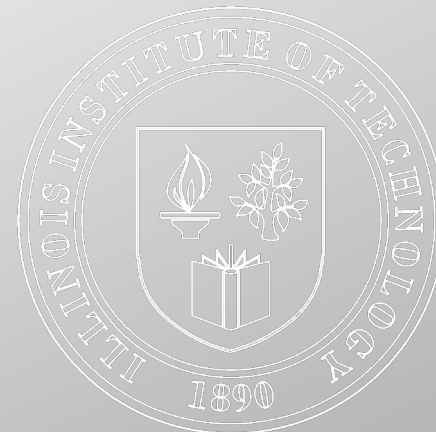
4. Communicate standardized software maintenance processes/services.
5. Establish a repository of standard maintenance software processes/services.
6. Integrate the software maintenance processes/services with those from other IS/IT organizational units especially where there is a direct interface.



Process Management Domain

3. Maintenance Training

- KPA identifies strategic needs for education and training, while focusing on the processes and also on the technical aspects. Training is developed internally or acquired by vendors/consultants with the objective of improving the competencies and knowledge needed for executing the maintenance process

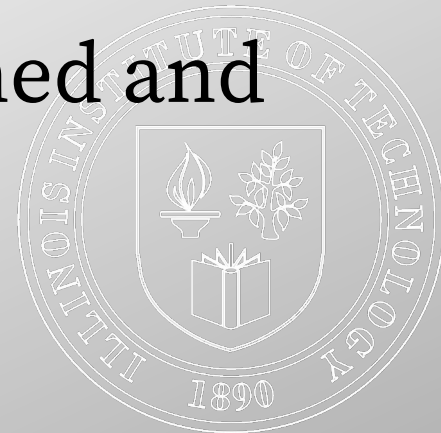


Process Management Domain

3. Maintenance Training – KPA

Goals of this KPA

1. Identify, request, and obtain the resources required for training and education of maintenance engineers.
2. Harmonize corporate training and locally planned and funded maintenance training.



Process Management Domain

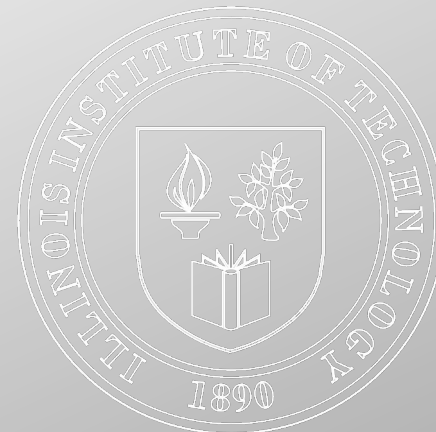
3. Ensure that there are competent and motivated maintenance personnel.
4. Motivate maintenance engineers by promoting education and training on processes, software, and technology.



Process Management Domain

4. Maintenance Process Performance

- KPA establishes quantitative goals for
 - The quality and performance levels of execution
 - Software products in operations
 - Intermediate products (artifacts)

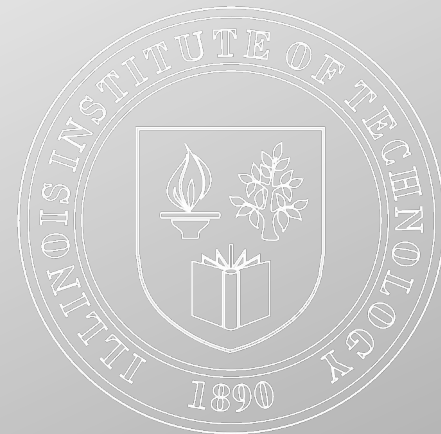


Process Management Domain

4. Maintenance Process Performance- KPA

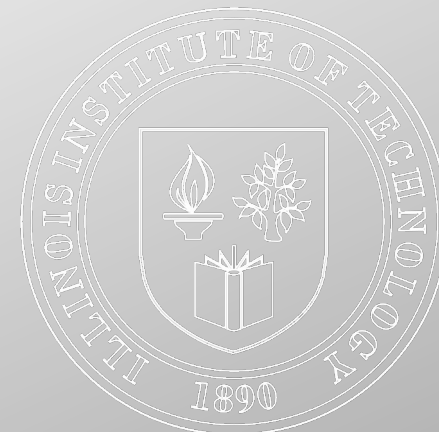
Goals of this KPA

1. Identify the processes and key activities of software maintenance that will be subject to performance analysis.
2. Set up a performance baseline for maintenance processes.



Process Management Domain

3. Identify and set up measures for software maintenance process performance.
4. Set up models for predicting process performance.



Process Management Domain

5. Maintenance Innovation and Deployment

- KPA which groups together practices to select and deploy innovations and improvement projects.

Decision on technological changes must be based on facts using data and cost/benefit studies, as well as conducting experiments and controlled deployments.



Process Management Domain

5. Maintenance Innovation and Deployment – KPA

Goals of this KPA

1. Identify the maintenance improvements and innovations with the greatest potential.
2. Conduct pilot projects to verify the performance of the most promising alternative.



Process Management Domain

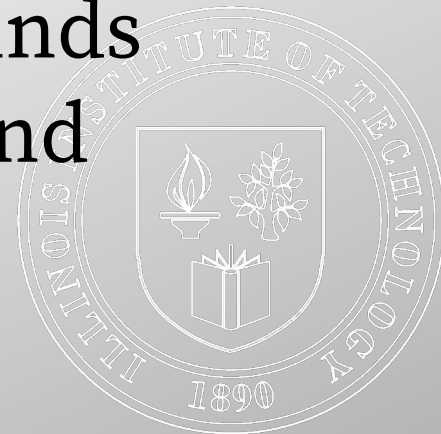
3. Identify, from the pilot projects, the improvements to be deployed.
4. Plan the improvements, manage their deployment, and measure the benefits.



Process Management Domain

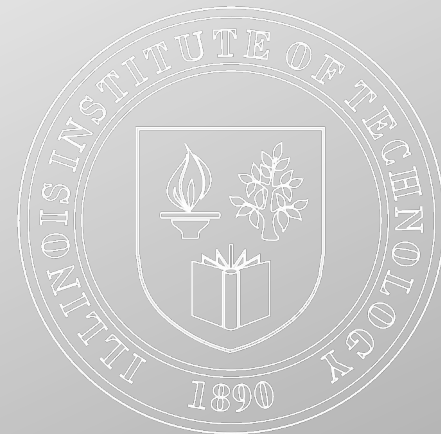
3.2 Activity Distributions

- ◆ Maintenance teams are spending more time making repairs.
- ◆ The root cause behind this trend seems to be the most maintenance projects were unable to reduce their repair backlog because of other demands and funds limitations. They fixed high-priority problems and deferred the others to later release.



Process Management Domain

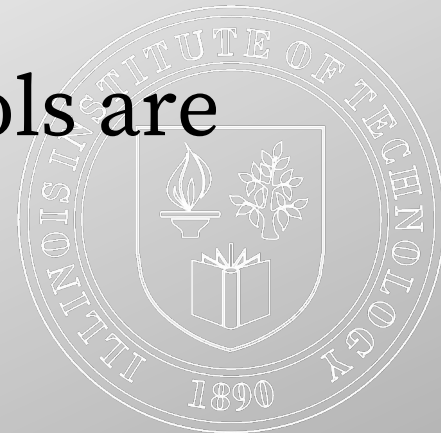
- ◆ We see 80 percent of problems come from 20 percent of the software modules.
- ◆ Bigger projects performed regression testing using test baselines to revalidate that the release was fit for operational use.



Process Management Domain

3.3 Resource Needs

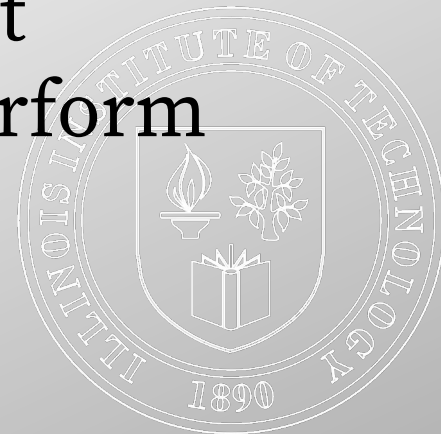
- ♦ The resources needed to accomplish the tasks in our WBS vary as a function of how the work is budgeted. 80 percent of maintenance projects are performed by teams of 10 or less.
- ♦ You need facilities, equipment, software and tools are needed to process changes and make fixes.



Process Management Domain

3.4 Success Formulas

1. To **manage** the software maintenance job properly, you first have to understand all of the work that needs to be done in order to complete it satisfactorily.
2. To **structure** the work involved in software maintenance so that it can be done most efficiently, you need to put processes in place and train your people in how to perform them.



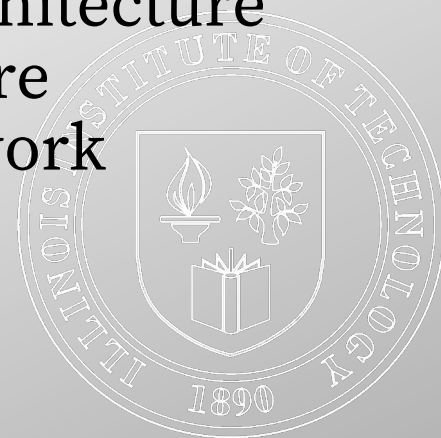
Process Management Domain

- ♦ 3. **Recognize** that most software maintenance projects are small. In response, make sure that your processes do not over-burden them with necessary effort.
- ♦ 4. **Understand** that most software maintenance projects are funded on a level-of-effort (LOE) basis. Unlike software development jobs where budgets vary, maintenance budgets are fixed. In response, you need to be able to figure out what you can do with what you are given.



Process Management Domain

- ♦ 5. **Appreciate** the fact that you are dealing with and experienced workforce whose skills are at a premium and who may be special circumstance employees (retirees, etc.). Put human resource practices and incentives in place that respond to the workforce's unique needs.
- ♦ 6. **Recognize** the product generated during maintenance is different from that provided by a software development shop. During software development, you worry about requirements satisfaction, architecture stability, and meeting cost and schedule goals. During software maintenance you worry more about content and how it will work operationally in targeted sites.



Process Management Domain

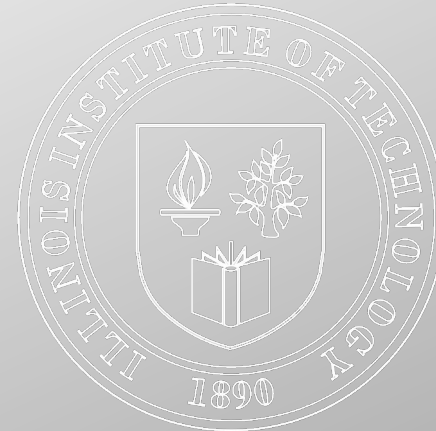
- ♦ **7. Understand** you need a different mind-set to succeed during software maintenance. During software development, you are geared to get a product out the door on time and per an agreed upon budget, schedule, and content. In contrast, during software maintenance, the product exists and your job is to keep it operational. In order to do this, you will have to focus more on the tactical decisions than on the strategic ones.



4 Ten Success Recipes for Surviving the Maintenance Battles

4.1 Balance between Agility and Discipline

- Maintenance projects try to be more agile than their development counterparts. Because, the software already exists and is being used, they have the luxury of trying newer techniques that focus more on the code than on the requirements.



4 Ten Success Recipes for Surviving the Maintenance Battles

- ♦ **Plan-driven** development assumes that requirements are the forcing function and that all subsequent activity is aimed at ensuring that they are satisfied.
- ♦ **Agile methods** focus attention on rapid prototyping and development because they believe the primary emphasis of the process should be code development.



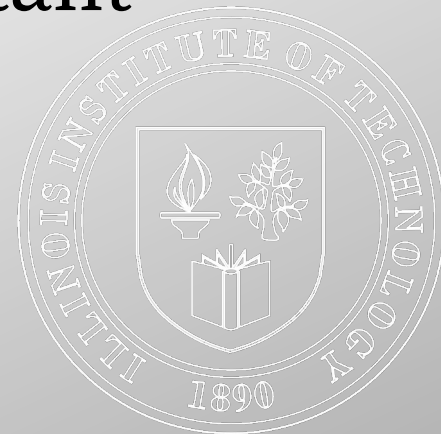
4 Ten Success Recipes for Surviving the Maintenance Battles

- ♦ **Agile Software Development:** Refers to a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.
- ♦ **Maintenance Plan:** Refer to a document that sets out the activities, schedules, practices, and resources to be used to maintain a software product and generate a software release.
- ♦ ISO and CMMi certified companies stand out from competition.
- ♦ More firms in India are rated CMMi level 5 than rest of the world combined.



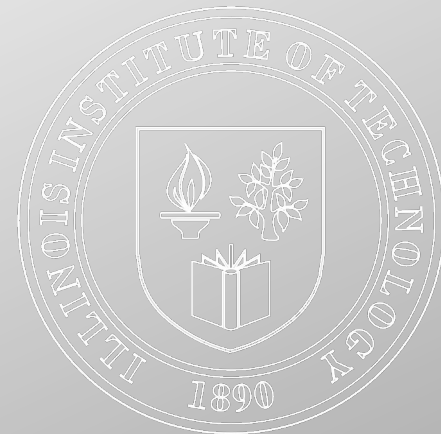
4.2 Emphasis on Managing the Work

- ♦ The emphasis makes tasks like technology refresh, build, regression test automation and management, distribution management, and support tasks like sustaining engineering (including user training and support), field support, and operations and maintenance (O&M) support much more important during maintenance than during software development.



4.3 Establish a Proper Infrastructure

- ◆ **Key Process Area (KPA):** Cluster of related activities that, when performed collectively, achieve a set of goals considered to be important for establishing process capability.

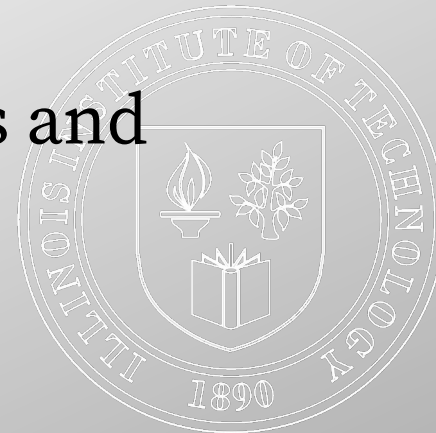


4.3 Establish a Proper Infrastructure

- ♦ **Engineering:**

Process Area (PA):

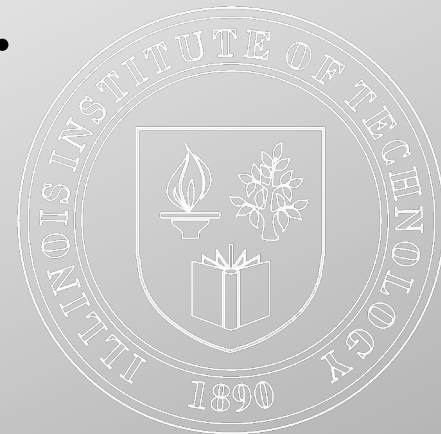
1. **Product Integration:** Build and integrate the product from its components and deliver it once it has been tested for delivery to customers in the field after it goes through various levels of acceptance and qualification testing.
- ♦ **2. Requirement Development:** Analyze customer needs (for new features, repairs, and perfective changes) and generate requirements in the form of engineering change requests and trouble reports aimed at satisfying them.



4.3 Establish a Proper Infrastructure

3. Technical Solution: Design, develop, and implement technical solutions to customer needs by integrating new and existing components into the architecture.

4. Validation: Ensure that the release satisfies its requirements and satisfies customer expectations.



4.3 Establish a Proper Infrastructure

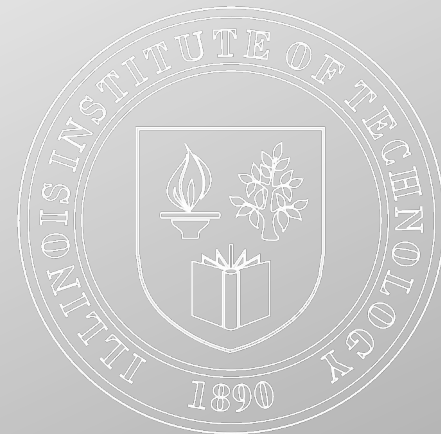
5. Verification: Ensure that the release works as intended in its operational environment.

6. Emergency Solution: Design, develop, and implement technical solutions to immediate problems that cannot tolerate delays. Interim solutions such as patches can be used if operational concerns make them necessary.



4.3 Establish a Proper Infrastructure

7. Test Management: Manage the test program to ensure that a regression test baseline can be used to revalidate the release once changes have been made to it. The actual regression tests used to make the determination should be automated, if possible.

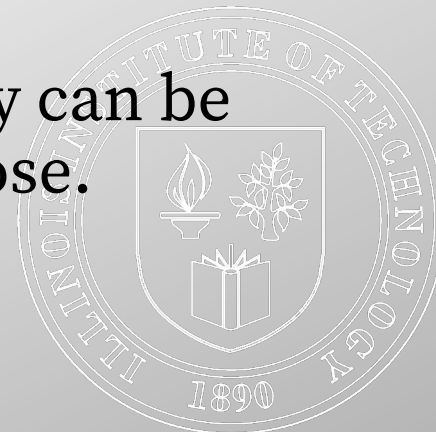


4.3 Establish a Proper Infrastructure

Project Management:

Process Area (PA):

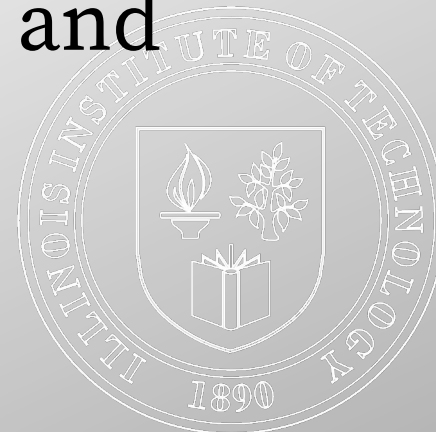
- 1. Integrated Project Management:** Manage release delivery and other tasks using processes that involve all stakeholders in planning, organizing, staffing, directing, and controlling the project.
- 2. Project Monitoring and Control:** Monitor and control performance of the release and other tasks such that if problems occur, the team can take the appropriate corrective actions.
- 3. Project Planning:** Plan the release and other tasks so that they can be accomplished using the resources made available for that purpose.



4.3 Establish a Proper Infrastructure

4. Requirement Management: Manage requirements as the project unfolds and ensure release plans and other release products are consistent with the plans.

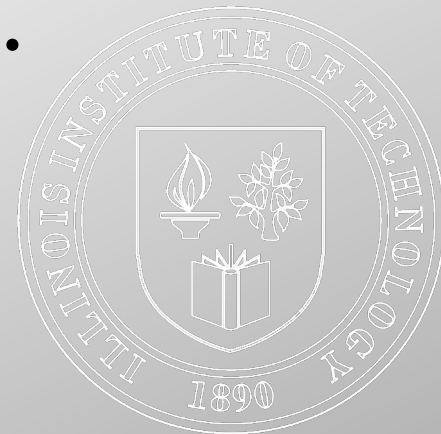
5. Quantitative Project Management: Quantitatively manage the project so that it can achieve its quality and process performance goals.



4.3 Establish a Proper Infrastructure

6. Risk Management: Identify and prioritize risks so that actions can be taken in a timely manner to mitigate their impacts.

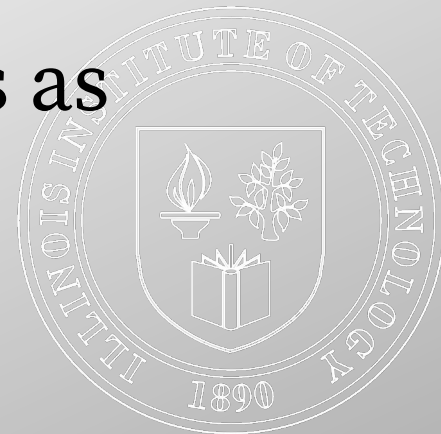
7. Supplier Agreement Management: Manage the acquisition of products and services from suppliers.



4.3 Establish a Proper Infrastructure

8. Facilities Management: Manage integration and test facilities so that they have the equipment, tools and licenses to optimally perform their functions.

9. Transition Management: Manage the transition and turnover of the products from development to maintenance shops such that the changeover goes as smoothly as possible.



4.3 Establish a Proper Infrastructure

Support:

Process Area (PA):

1. Casual Analysis and Resolution Configuration Management:

Identify causes of defects and put in place measures aimed at preventing them from reoccurring.

2. Configuration Management: Maintain integrity of work products and control changes being made to them.

3. Decision Analysis and Resolution: Make decisions using a process that assesses alternatives using established criteria.



4.3 Establish a Proper Infrastructure

4. Measurement and Analysis: Develop and sustain a measurement capability to support management information needs, including providing insight into process performance and product quality.

5. Process and Product Quality: Maintain integrity of processes and associated work products and focus on providing quality.



4.3 Establish a Proper Infrastructure

6. Customer Support: Provide high levels of customer support in the field and during operations including user training, help desk, hand-holding, and website assistance.

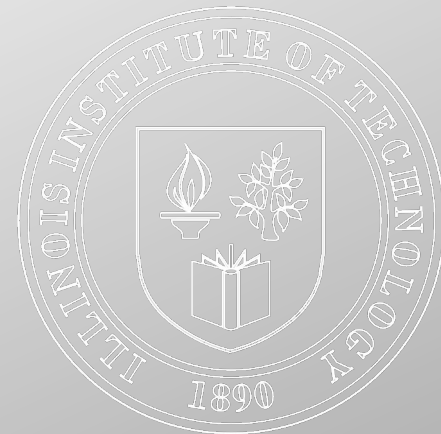
7. Distribution Management: Maintain integrity of the distribution process by assuring that products delivered are configured properly for the customer and work in their sites.



4.3 Establish a Proper Infrastructure

Top 5 Development Risks:

1. Personnel shortfalls
2. Unrealistic schedules and budgets
3. Requirements creep or volatility
4. Developing the wrong functions/interfaces
5. Gold-plating



4.4 Address Operational Restrictions

Operational Restriction:

- 1, Software Configured Improperly:** Old system used until new version shipped if not the first instance of use in the field. Field support may be needed to patch system so it does not fail in the field.
- 2. System Availability Limited Along with Test Time:** Time on system to install and configure software may be severely limited due to operational need. Delays will be common and test time limited.
- 3. Necessary Skills to Operate Software may be Missing:** Before the new release can be used, operators of the system must be trained. This may have to be done in the field. In response, self-paced instruction using the system as the training vehicle may have to be used.



4.4 Address Operational Restrictions

4. Necessary Prerequisite Software and Equipment may not be Present: Maintenance team may have to rapidly reconfigure a new version for the user and package and deliver it to the user with installation instructions at the operational site.

5. Patches buggy-cause Operational Failures: Maintenance team distributes binaries to prevent user from making patches. However, team may have to go to sites to make patches themselves when they are needed to preserve operational availability of the system.



4.4 Address Operational Restrictions

- ♦ **6. New Version Works but does not Perform as Advertised:** Maintenance team needs to determine root cause of problem. They need to analyze performance and other measurement data and make required fixes. In some cases, they have to go to the field to isolate issues.
- ♦ **7. Malicious code in delivery (especially commercial off-the-shelf (COTS) packages:** Malicious code may cause system to fall at just the wrong moment. Backups must be maintained along with a recovery plan.



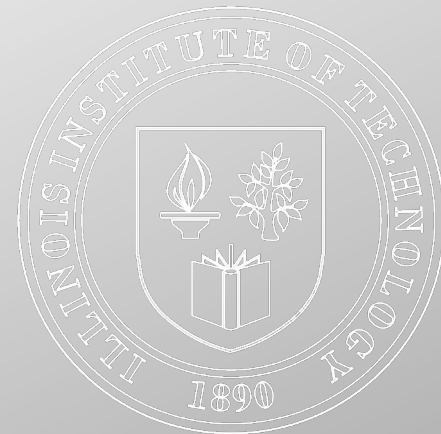
4.4 Address Operational Restrictions

- ♦ **8. Must address working 4 releases in parallel during Operations:** In parallel with preparing block releases, maintenance groups must support “fielded,” “to be fielded,” and a requirements release. These releases can take resources away from development and create conflicts that must be resolved before progress can be made.



4.5 Ten Success Recipes

1. Adequate transition and turnover planning – Chapter 5
2. Establishing solid management infrastructure – Chapter 6
3. Best-in-class facilities – Chapter 7
4. Responsive user support structure – Chapter 8
5. Focus on Regression Testing – Chapter 9



4.5 Ten Success Recipes

6. Content based annual releases – Chapter 10
7. Proper resources (staff and equipment) – Chapter 11
8. Effective measurement data utilization – Chapter 12
9. Being ready for the next upgrade – Chapter 13
10. Know when to retire the system – Chapter 14

