

**ILLINOIS TECH**

College of Computing

# ITMD 536 Software Testing & Maintenance

Chapter 7 and 8  
Best-in-Class Facilities &  
Responsive User Support Structure



# Objectives

- ♦ What is the difference between development and maintenance facilities?
- ♦ What are integration laboratories?
- ♦ How maintenance facilities are managed?
- ♦ What are the methods and tools?
- ♦ Where investments are needed?
- ♦ What are maintenance releases?
- ♦ What are emergency fixes?
- ♦ What is help desk?



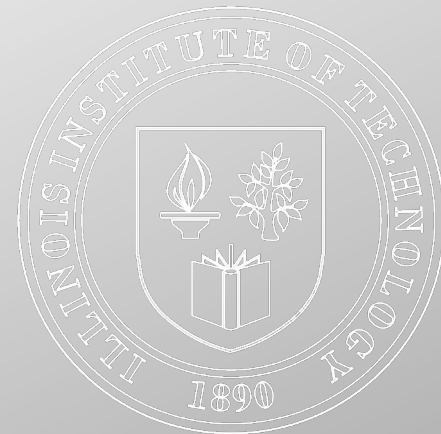
# Maintenance

- ◆ What is acquisition?
- ◆ What is maintainability?
- ◆ Who is the maintainer?
- ◆ What is maintenance?
- ◆ What are the maintenance actions?
- ◆ Why is maintenance plan required?
- ◆ Maintenance process?
- ◆ Maintenance schedule or duration?



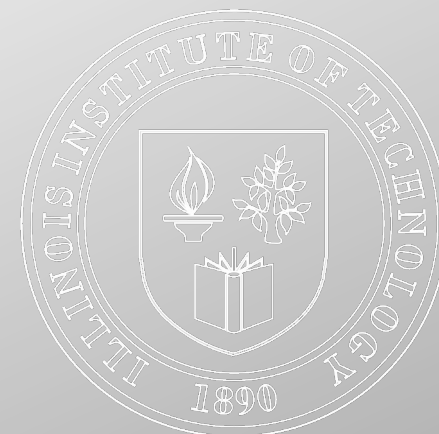
## 7. Best-in-Class Facilities

- ♦ **7.1 Facilities Overview:**
- ♦ Developers get to choose underlying technology where as maintainers have to live with developers choices.
- ♦ Developers get to choose the equipment they will need and size its computational resources where as maintainers monitor actual use.



## 7.1 Facilities Overview

- ◆ Developers get to choose underlying technology where as maintainers have to live with developers choices.
- ◆ Developers get to choose the equipment they will need and size its computational resources where as maintainers monitor actual use.



## 7.1 Facilities Overview

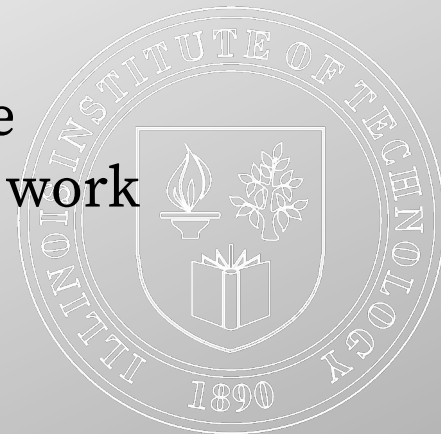
- ◆ Developers get to select software toolsets that make their work easier, while maintainers may have to use some of these tools to do work for which they were not designed.
- ◆ Developers get to negotiate for platform and tool software, while maintainers have to live with the agreements.
- ◆ Developers build their products without having to worry about computational limitations on development resources, while maintainers have to develop their updates on equipment geared for operations (i.e., often results in conflicts for computational resources like central processing unit [CPU] time and memory).



## Table 7.1 Differences between Development and Maintenance – (Development Facilities)

### ♦ Development Facilities:

- *Primary Use:* Developing new software to satisfy requirements
- *Facilities:* Software factory with lots of meeting space and office space geared for workforce productivity
- *Equipment:* Server(s) for storing work-in-progress and workstations for developing applications tied together via capable networks and linked to emulators or models of equipment to be used in the field
- *Tools:* Compatible set of methodology-driven software design, code integration, and test tools linked to some repository for storing the work products including work in progress





## Table 7.1 Differences between Development and Maintenance – (Development Facilities)

- *Software Licenses:* Development licenses for tools and platform software (operating systems, database systems, etc.) for both laboratories and operational systems used for software development
- *Facility Staff:* Services needed to keep facilities, equipment, and software operational. As a minimum, includes network and security administrators, tool experts, technicians, and representatives (users, maintainer, etc.) who basically serve as liaisons with other staff





## Table 7.1 Differences between Development and Maintenance – (Maintenance Facilities)

### ♦ Maintenance Facilities:

- *Primary Use:* Changing existing software to add features and repair bugs
- *Facilities:* Software factory in existing facility with offices situated so that there is ease of access to equipment and laboratories
- *Laboratory:* Maintenance laboratories loaded with lots of gear whose purpose is to facilitate making changes and fixes to existing software and to package and deliver release in as representative an operational environment as possible



## Table 7.1 Differences between Development and Maintenance – (Maintenance Facilities)

- *Equipment:* Server(s) for storing baselined configurations and workstations for preparing releases tied together via networks and linked to the actual operational equipment configured as used in the field
- *Tools:* Compatible set of both hardware and software maintenance tools that works on baselined products and is linked to some repository for storing works products including work in progress



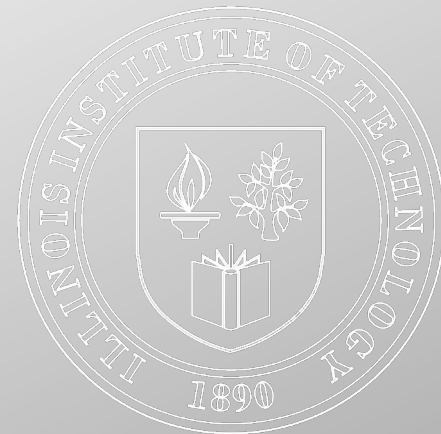
## Table 7.1 Differences between Development and Maintenance – (Maintenance Facilities)

- *Software Licenses:* Development and run-time licenses for all tools and platform software used by maintenance facilities (developmental licenses) and operational systems in the field (run-time licenses)
- *Facility Staff:* Services needed to keep facilities, equipment, and software operational; at a minimum includes network and security administrators, tool experts, technicians, and user representatives who actually work on-site performing software maintenance tasks



## 7.2 Integration Laboratories

- ◆ Both development and integration laboratories are consolidated into a single facility, but they are conceptually separated because each does a different job.



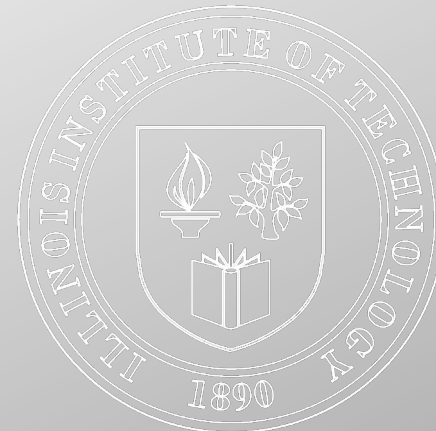
## 7.1 Software development/system integration laboratory capabilities

### Software Development

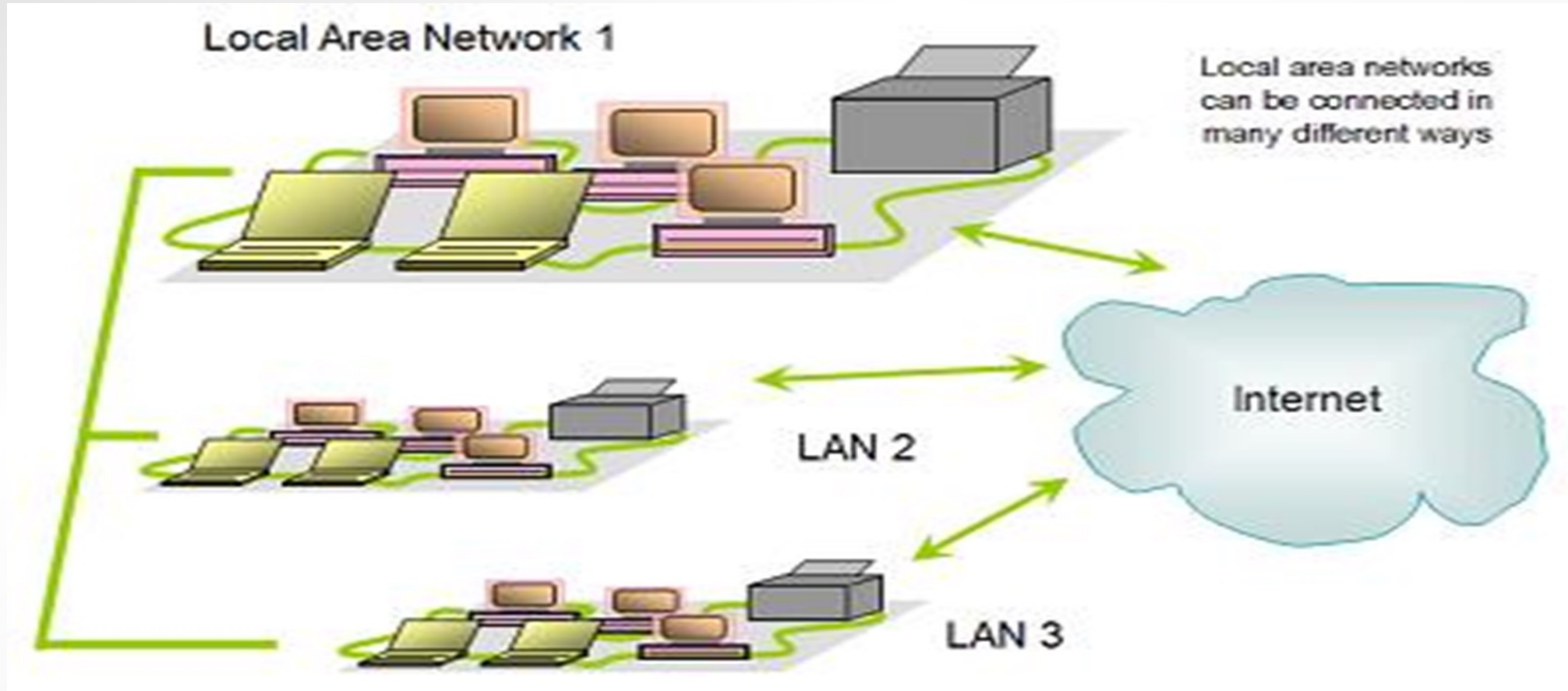
- ♦ Graphical workstation
- ♦ Development host
- ♦ Software development Environment (lots of tools)
- ♦ Central repository to house Builds and work-in-progress
- ♦ Internet and LAN access
- ♦ Access controls

### System Integration

- ♦ Graphical workstation
- ♦ Integration host
- ♦ Operational equipment
- ♦ Integration environment (test and performance measurement tools)
- ♦ Internet and LAN access
- ♦ Access control



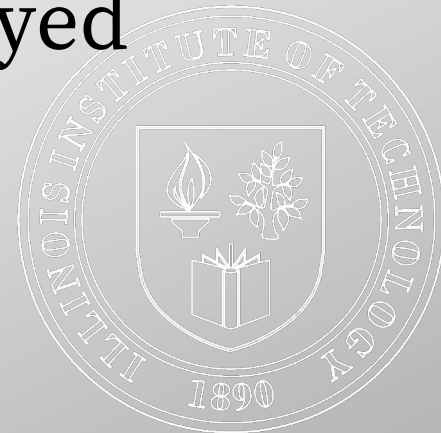
# LAN Connections





## 7.2 Integration Laboratories

- ♦ A separate system integration facility is essential when hardware and software are being developed in parallel for the project.
- ♦ Hardware testing is necessary to validate that things operate properly when real equipment is employed the system is hooked up to external interfaces.





## 7.3 Maintenance Facilities

- ♦ Maintenance facilities differ greatly in composition from those used for development and integration.
- ♦ For example, modern automobiles employ computers to perform functions like engine control, diagnostics, entertainment delivery, and suspension management.
- ♦ Diagnosis and repair of problems are facilitated by hooking up specialized test equipment to the car via the bus and running diagnostics.



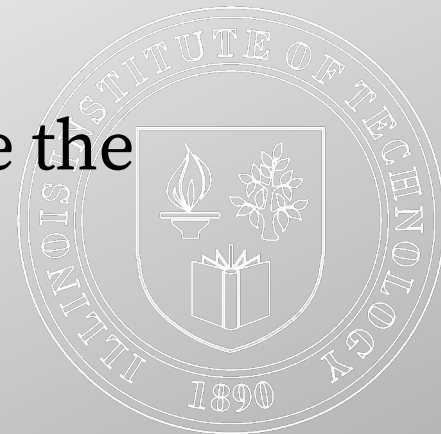
## 7.3 Maintenance Facilities

- ◆ For embedded systems specialized maintenance facilities need to put in place to update and repair tasks.
- ◆ Most organizations do not allow their operational sites to either install patches or perform emergency repairs.



## 7.4 Methods and Tools

- ♦ Methods and tools used during software maintenance are often different from those used during development.
- ♦ First depending on the age of the software, underlying language and technology used.
- ♦ Second, tools used during development may not be transferred to the maintenance group because they are proprietary, have license restrictions, or just too expensive.
- ♦ Third, tool needs may differ during maintenance because the work being performed differs.



## 7.4 Methods and Tools

- ♦ If the system is not broken they tend to use it because replacing the 20-30 years old systems may be too expensive.
- ♦ Most technology consists of pairings of methods, programming languages, and software tools.
- ♦ In the beginning there was assembly code, programming languages like COBOL, FORTON, Pascal, and PL/I, and basic software tools like compilers, debuggers, and editors to do the work.



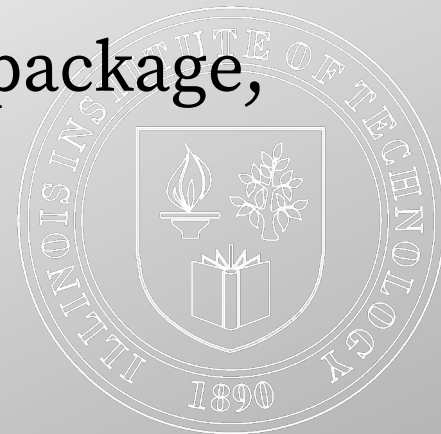
## 7.4 Methods and Tools

- ♦ In 1980s structured methods came in existence such as structured methods, data modeling, the Ada and C programming languages, SQL (system query languages), and toolsets, which most called software engineering environments (SEE).
- ♦ In 1990s, technology jumped to Object-oriented methods, object data modeling, the C++ and Java programming languages, scripting languages like Perl, the html (hypertext) and XML and Web-enabled tool chains like together using frameworks into the software environments.



## 7.4 Methods and Tools

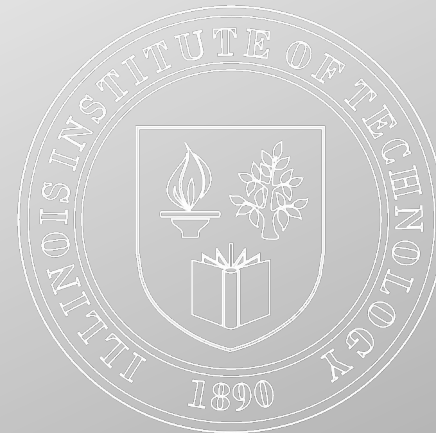
- ♦ Many maintenance shops hire retirees with the skills to do some of this work as special circumstance employees.
- ♦ Maintainer must make every effort possible to acquire underlying tools and technology as part of the transition agreement with developers.
- ♦ Maintainers have to deal with code written in different programming languages and with different underlying technologies, they have to modify, update, retest, tailor, package, and distribute their products to the field in a variety of configurations.





## 7.5 Where Investments are Needed

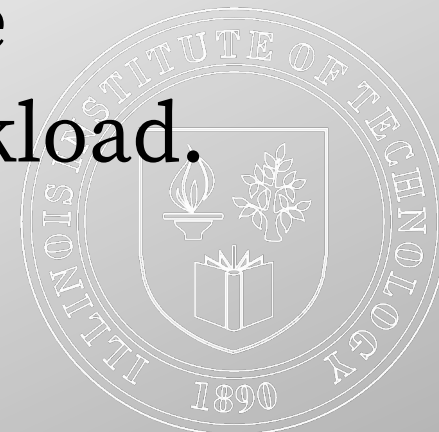
- ◆ Preplanning is needed for transition and turnover:
- ◆ *Facility location, layout including lighting, electrical and power distribution, and floor plans including entrance and egress of equipment:*
- ◆ Most projects may use existing laboratories





## 7.5 Where Investments are Needed

- ♦ *Office locations, layout including lighting and electrical, and floor plan:* Most projects use existing office space.
- ♦ *Equipment needs including specialized test and interface gear for both the laboratories and offices:*
- ♦ For new facilities, equipment resources must be selected and sized to accommodate project workload.



## 7.5 Where Investments are Needed

- ♦ *Specialized support for security:* Instituting network defenses and security may require specialized equipment and software (firewalls, intrusion prevention devices, virus software, etc.) to be purchased installed, operated, and kept up to date.
- ♦ *Software licenses including those required for platform updates and tools:* New versions of platform software must be installed along with released, installed and accepted.



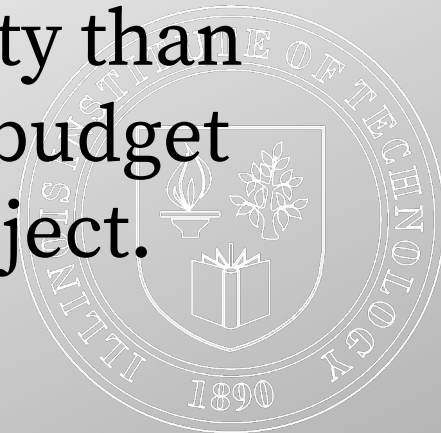
## 7.5 Where Investments are Needed

- ♦ *Software personnel need to be acquired, trained and put to work as the facilities equipment, and software are installed and readied for operations:* If you are using existing facility than it will fee-for-service. Always include costs in your budget for network and security administration for the project.
- ♦ *Specialized support for security:* Instituting network defenses and security may require specialized equipment and software (firewalls, intrusion prevention devices, virus software, etc.) to be purchased installed, operated, and kept up to date.



## 7.5 Where Investments are Needed

- ♦ *Software licenses including those required for platform updates and tools:* New versions of platform software must be installed along with released, installed and accepted.
- ♦ *Software personnel need to be acquired, trained and put to work as the facilities equipment, and software are installed and readied for operations:* If you are using existing facility than it will fee-for-service. Always include costs in your budget for network and security administration for the project.



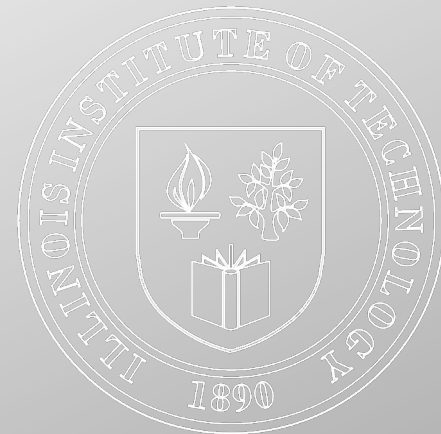
# Software Rejuvenation, Migration, and Retirement KPA

- ♦ A decision criterion is necessary to decide whether or not this activity is economically beneficial for software.
- ♦ Attempts to improve its quality, efficiency and general availability will be made.
- ♦ This also includes migration of software to another technical environment (platform).
- ♦ These are considered preventive maintenance activities.



# Software Rejuvenation, Migration, and Retirement KPA

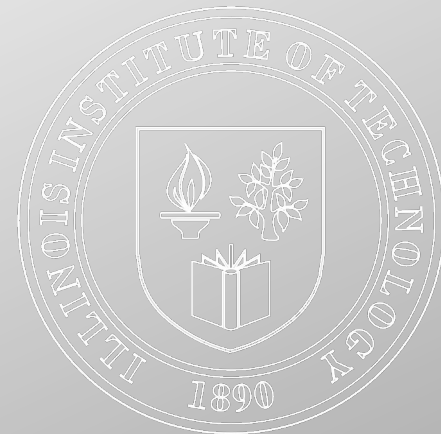
- ◆ This also includes migration of software to another technical environment (platform).
- ◆ These are considered preventive maintenance activities.
- ◆ When need is confirmed and approved by maintenance planning:
  - a. redocumenting the software
  - b. restructuring the software, or
  - c. performing software retro-engineering.





# Responsive User Support Structure

- ♦ **User Support:** The process of providing user support including training via mentoring, the staffing of a help desk, and user of a website.





## 8.1 Maintenance Releases

- ♦ **Maintenance:** Refers to the process of keeping a product current after delivery. This involves updating the product to address new functionality and repairs and sustaining the facilities and infrastructure required to accomplish this task.
- ♦ **Release:** A software version that is currently under configuration control and is made available to potential users for a specific purpose (replace previous version to fix problems and add functionality, beta testing, etc.)

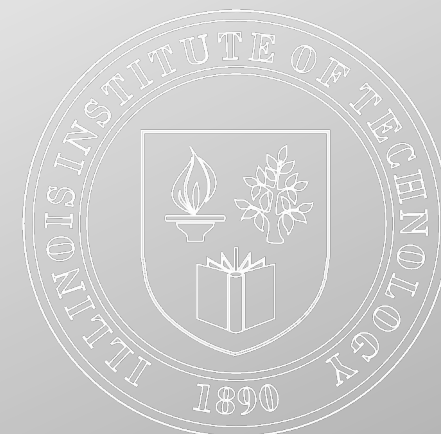
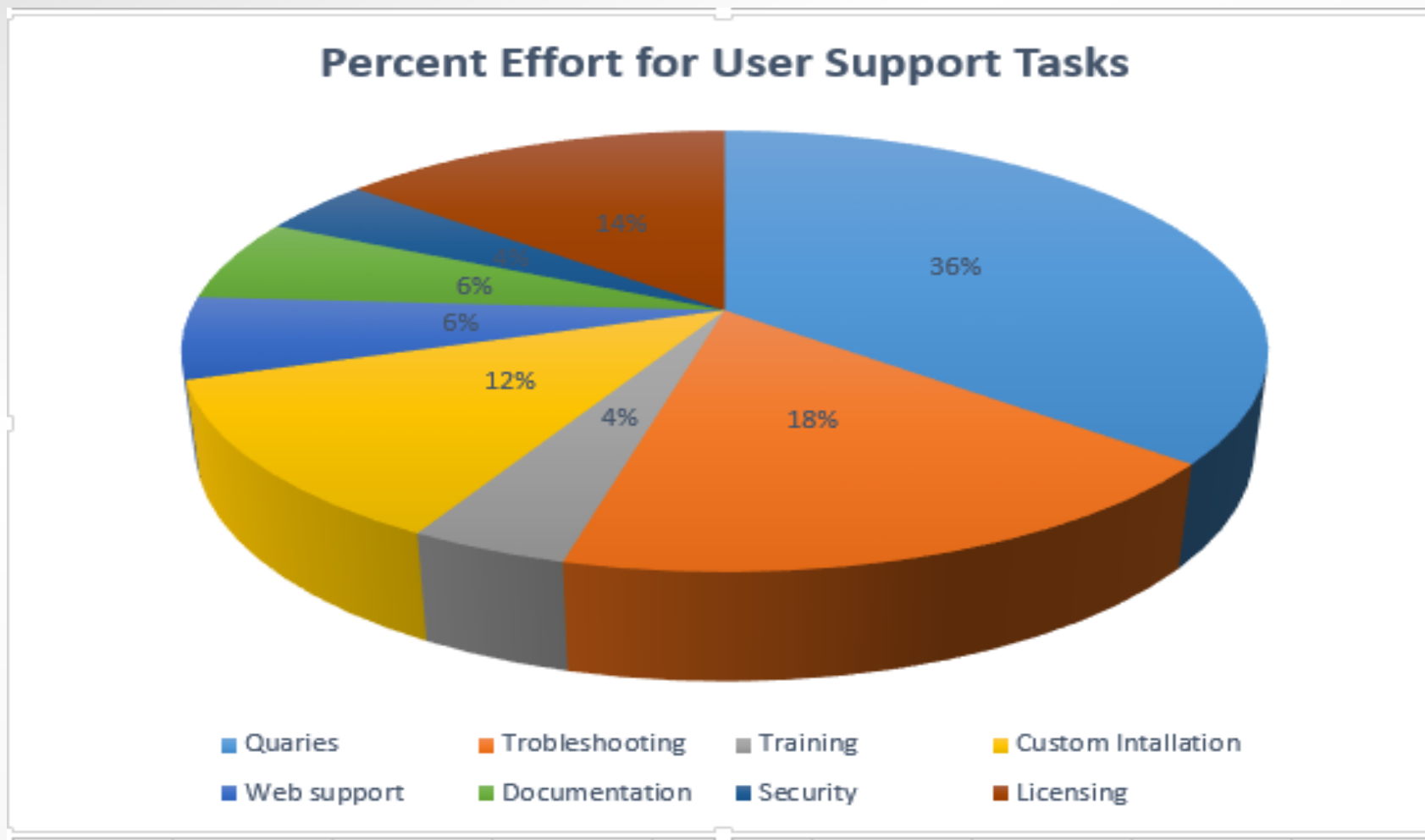


## 8.1 Maintenance Releases

- ◆ Support services can be offered during specific business hours or 24/7 depending on need.
- ◆ Technical support can be provided via telephone, onsite, via e-mail or a website.
- ◆ Common repairs virus, removals, driver issues, registry repairs, server issues, optimizations, and operating system security issues.



# User Support Tasks %



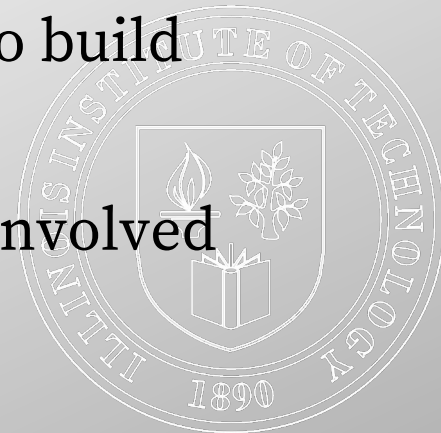
## 8.1 Maintenance Releases

- ♦ Answer user queries and frequently asked questions and provide quick services.
- ♦ Provide users with technical support.
- ♦ Conduct user training and build skills, knowledge, and experience with products.
- ♦ Provide custom installation, configuration, and software maintenance for existing application programs either remotely or on-site.
- ♦ Create a website to assist users and keep it up to date.
- ♦ Create documentation to help users with the frequently asked questions list and sets of “how to” instructions



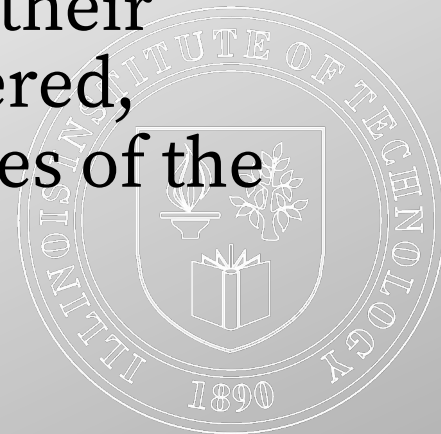
## 8.1 Maintenance Releases

- ♦ Administer security for the website and controlled products including software and documentation (security).
- ♦ Administer software licensing assuming that user packages have been acquired using some enterprise-wide license (licensing).
- ♦ CMMi-SVC model seem applicable to establishing user support and support centers.
- ♦ Customer oriented organization trains its staff to use information to build relationships.
- ♦ Measure the service quality and provide feedback about everyone involved including customers and workers.



## 8.1 Maintenance Releases

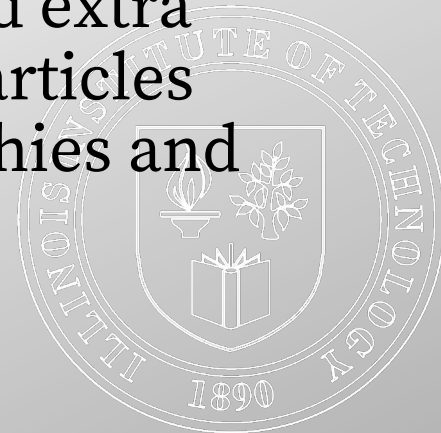
- ♦ Support organizations continuously assess reliability, responsiveness, assurance, empathy, and tangibles when rating service quality.
- ♦ Support organizations run themselves like a business. As such, they realize that customers need to feel that they are getting good value for their money.
- ♦ Support organizations establish service agreements with their customers to set expectations for the services to be delivered, service level requirements and targets, and responsibilities of the support organization, customer, and end user.





## 8.1 Maintenance Releases

- ♦ User will find a way to avoid using a product that is hard to understand and use. Your measurement program will highlight this when it provides current usage statics.
- ♦ Train rank-and-file employees in your user/customer support philosophy and practice.
- ♦ Changing behaviors is hard. Support organizations reward their staff for exhibiting customer-oriented behaviors. Cash rewards and extra time off. Other rewards include special recognition dinners, articles in the company newsletter, tickets to special events, and trophies and plaques.





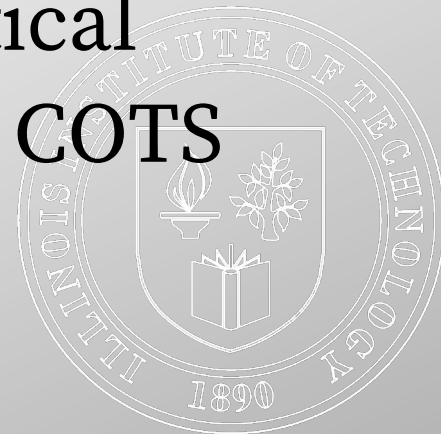
## 8.1 Maintenance Releases

- ◆ Support organizations communicate examples of excellent customer service whenever they occur throughout the organization.
- ◆ Social factors can make or break the perceptions relative to quality of service.
- ◆ Make sure that you put in place processes for emergency repairs, creating backups/recovery CDs and handling disasters.



## 8.2 Emergency Fixes

- ◆ User support is making emergency repairs during maintenance.
- ◆ These repairs may be needed to fix specific problems users are experiencing with the release in the field, address platform related problems or install critical vendor-supplied patches needed to keep legacy, COTS and open source packages.



## 8.2 Emergency Fixes

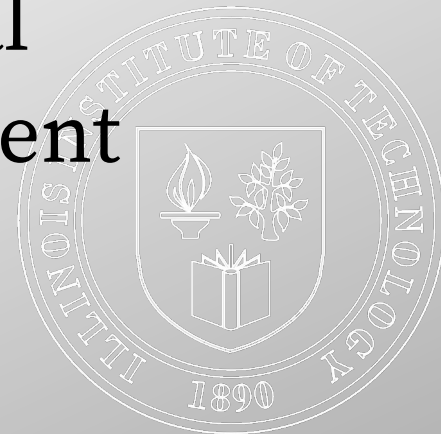
- ♦ The process assumes that your current release is baselined and managed under configuration control and that any changes that will be made to it will be documented using a software change request (SCR) before being approved for distribution:
- ♦ *Get authorization to make an emergency change.* Your process manual should tell you who is authorized to provide the emergency change approval.
- ♦ *Capture an image of the problem.* Capture the image and the operating environment present when it occurs.



## 8.2 Emergency Fixes

The image will allow you to work with the users remotely to resolve the issue.

- ♦ *Troubleshoot, isolate, and diagnose the root cause.* Find the root cause of the problem. It takes time and effort to find out the underlying cause. Your primary goal should be to minimize the potential impact on current operations.



## 8.2 Emergency Fixes

- ♦ *Identify packages and releases impacted.* Once the problem is isolated, you need to identify which packages and releases are impacted that needs repair.
- ♦ Configuration accounting system will help by providing the build configuration of all active releases and versions of the product.



## 8.2 Emergency Fixes

- ♦ *Review defect database.* Check your defect repair database, assuming that you have one, to determine whether this new problem is persistent error.
- ♦ You might consider replacing the component rather than patching or repairing it to cut down repeat errors. Fix old issue to avoid recurrence.





## 8.2 Emergency Fixes

- ♦ *Make the fix.* Working with a copy of the baselines affected, develop a fix.
- ♦ Test and qualify the fix prior to developing a patch for the installation in a unique version of the release that will be installed in the field either remotely or by technicians.
- ♦ *Document recommendation for future releases.* Those making the fix should write their ideas about what to do when making permanent repairs in both their release notes and in abbreviated form in the code.



## 8.2 Emergency Fixes

- ♦ *Create a patch release and distribute it to affected sites.*  
Next step a patch release should be prepared along with related release notes, installation guidelines, and tests/expected test results for shipment to the field. The patch release should be configured and distributed to field sites.



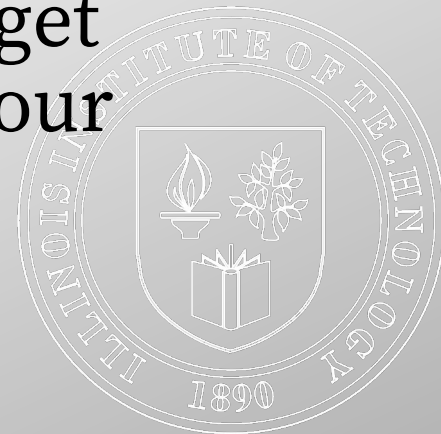
## 8.2 Emergency Fixes

- ♦ *Update defect database and patch baseline.* Final step is update defect database and prepare a patch baseline. The patch baseline should be uniquely numbered and placed under configuration control. This baseline should contain the release, an image of the platform it runs on, and related notes and installation guidelines, and tests/expected results.



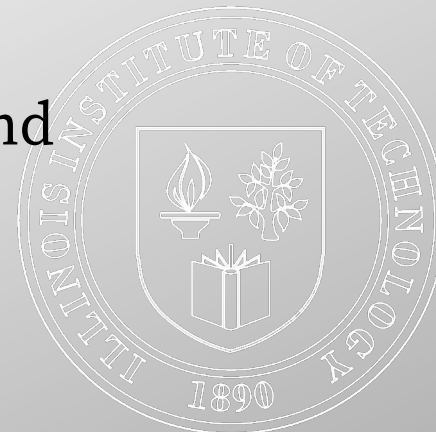
## 8.2 Emergency Fixes

- ♦ Another emergency repair essential is a bootable rescue/recovery CD. This CD contains software that enable you to repair, restore, or diagnose problems when your desktop crashes.
- ♦ Everyone who uses a desktop or is in charge of a server should build one of these CDs to recover should their system crash, hard disk get wiped out, or files/data get corrupted for whatever reason. Just put the CD in your drive and reboot.



## 8.2 Emergency Fixes

- ♦ Disaster planning helps you prepare for recovery. The plan should identify preventive, detective, and corrective measures. Preventive measures could include use of surge protectors or uninterrupted power supplies.
- ♦ Detective measures could include controls aimed at detecting unwanted events like fire or some detectors.
- ♦ Corrective measures could include controls aimed at recovering from system outages. (backups)
- ♦ The Resilience Management Model (RMM) can be used to define and improve processes for managing operational resilience in security management, business continuity management, and information technology (IT) operations management.



## 8.3 Help Desk

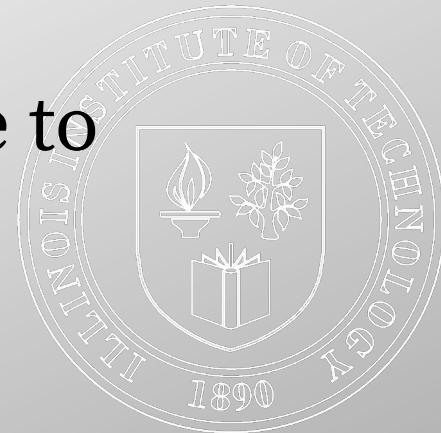
- ◆ Responsive support structure is providing users with needed assistance. This is by providing “help desk” or “call center” support.
- ◆ Prepare your help desk support according to call volume.
- ◆ Provide users/customers with access to bulletin boards, bug alerts, newsletters, shutdowns and work in progress.





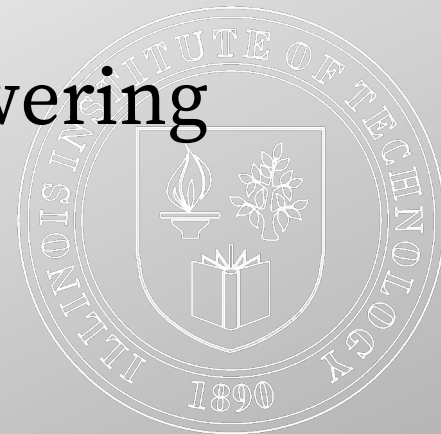
## 8.3 Help Desk

- ◆ Train your staff dealing with users/customers. Follow through and respond accordingly.
- ◆ Make sure that your staff speaks the customer's/user's spoken language like a native.
- ◆ Compile a list of answers for frequently asked questions and answers.
- ◆ Set priorities on calls and have a procedure in place to deal with emergencies.



## 8.3 Help Desk

- ♦ Have product specialists on-call should they need to be called upon to supply knowledge (of the platform, application, libraries, etc.) or work-specific problems.
- ♦ Have an escalation procedure in place should the user/customer want to elevate the problem to a supervisor for resolution or peace of mind.
- ♦ Provide a means to provide off-hours support (answering service, website, pagers, etc.).



## 8.3 Help Desk

- ♦ Supply a toll-free number if your competitors provide one.
- ♦ Keep track of call backlog, again, and time to resolve issues. Try to minimize the backlog and time to resolve by constantly improving your procedures, training, and work-related aids.
- ♦ If you charge for customer support using some fee-for-service arrangement, provide account management services such as account reviews.
- ♦ If you require logins, use strong passwords, and other authentication procedures to provide security and protect the privacy of your consumers.



## 8.4 Web Facilities

- ♦ Web facilities are created to enable user to both get answers to their questions and find information of interest quickly when working with the current release, websites have become the source for information because they can provide facts to the user 24/7.
- ♦ *Make it easy for users to find information of interest.* If website visitors cannot locate information of interest quickly, they may give up.
- ♦ *Make it easy to use for users to get help.* Organizations with the highest success are those that make visitors feel confident that they can do things on the website on their own.



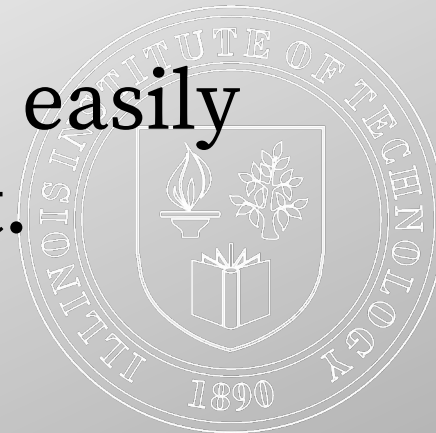
## 8.4 Web Facilities

- ◆ Make the information readily accessible.
- ◆ Design your site for “probabilities not possibilities.”
- ◆ Direct website visitors to FAQ list.
- ◆ Password is required for sensitive information.
- ◆ Visitors will remember email over user id for login access.



## 8.4 Web Facilities

- ♦ *Provide clear and readable content.* Ensure the content is clear, easy to read, and easy to understand. Avoid jargon and spell out acronyms whenever possible.
- ♦ *Offer users several choices of channels.* Consider following channels when building the website.
- ♦ *Tabs or pull-down menus.* This helps the visitor to easily navigate your site to find information of interest.





## 8.4 Web Facilities

- ♦ *Community.* Establish communities of interest to work common interests via blog or bulletin boards.
- ♦ *Online chat.* This provides visitor to interact in real time.
- ♦ *Email.* This channel provides the website visitor with a means to ask questions and get responses within a reasonable time period.
- ♦ *Click for a call back.* This channel enables the website visitor to set up a call in the future to talk about a topic and resolve issues.



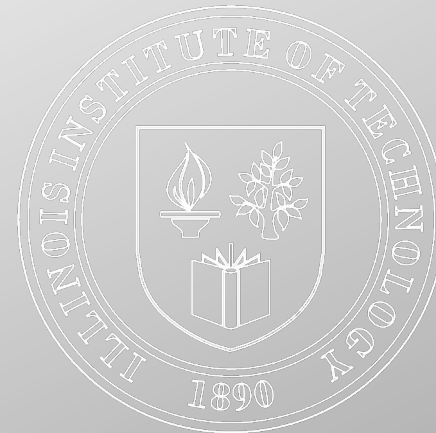
## 8.4 Web Facilities

- ◆ *Continuously improve website based on customer feedback.*  
Gather following information for improvements:
- ◆ Instant feedback
- ◆ User/customer surveys
- ◆ Web statistics and search logs
- ◆ Customer interviews and focus groups
- ◆ Independent usability testing



## 8.4 Web Facilities

- ♦ *Analyze site performance measures and data.* The most crucial performance issue is the load time, speed of search, wait times and responsiveness to queries.
- ♦ *Cumulative and reward good staff and customer behavior.* Provide reward financial or non-financial.



# Maintenance Process Focus

- ♦ A representative of the software maintenance organization is assigned to plan and coordinate improvement activities.
- ♦ Results of the software maintenance products/services customer survey is used to identify candidate improvements.
- ♦ The observations, comments, and complaints is from users/customers and interface groups ( developers, operations, help desk, subcontractors, etc.) are used to collect data for identifying candidate improvements.



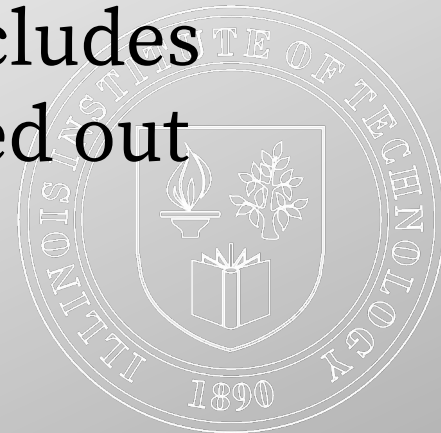
# Maintenance Process Focus

- ◆ The data on software failures is collected and used to identify candidate improvements to maintenance processes/products and also to the many interfaces with the other interface groups (developers, operations, help desk, subcontractors, etc.)
- ◆ The maintenance organization is subject to internal audits (from internal auditors (or other types of audits) and results are used to identify candidate improvements.



# Maintenance Process Focus

- ♦ A maturity assessment of some processes has been performed. At least one maintenance organization has conducted a process maturity assessment and the results are used to identify candidate improvements.
- ♦ Improvements to some processes have been initiated. The annual plan, of each maintenance organization, includes both the improvement activities planned and carried out during the year.





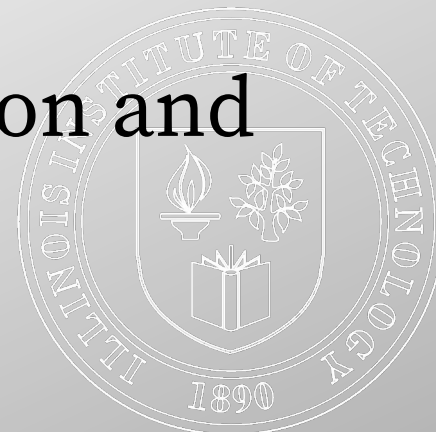
# Maintenance Training

- ◆ Training on communication with customers is offered to software maintenance engineers.
- ◆ Use of internal benchmarking data to guide the training of maintenance.
- ◆ Financial resources are available to maintenance managers for the education and training of each maintenance engineer.



# Maintenance Training

- ◆ There are plans and records that describe the education and training needed for each maintenance engineering position and application software. This training plan documents the training needs, the courses offered, and related material, credits, resources available, and the schedule of education and training activities.



# Maintenance Training

- ♦ The end-user training material is designed in accordance with a locally documented procedure.
- ♦ The user (and some other stakeholders) receives enough education and training to allow for autonomous use of the application software.

