Software Maintenance

2010. 04. 07

SW공학기술적용팀 김영수

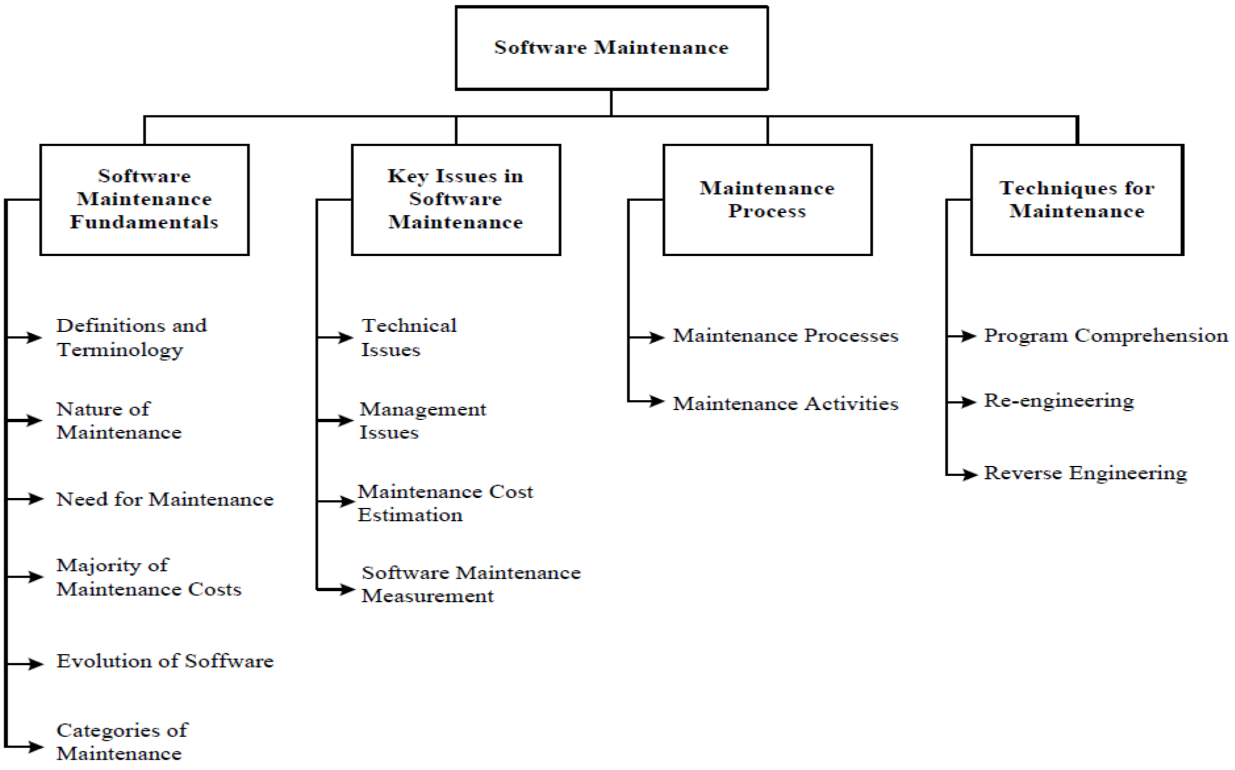
SWEBOK Seminar

Software Maintenance Introduction

- When ?
  - ✓ The maintenance phase of the life cycle begins following a warranty period or post-implementation support delivery, but maintenance activities occur much earlier.
- Importance
  - ✓ Software maintenance is an integral part of a software life
  - ✓ .. By keeping software operating as long as possible.
- In the Guide, software maintenance is defined as
  - ✓ the totality of **activities** required to provide **cost-effective support** to software.
- Activities
  - ✓ **Pre-delivery activities** include planning for post-delivery operations, for maintainability, and for logistics determination for transition activities.
  - ✓ **Post-delivery activities** include software modification, training, and operating or interfacing to a help desk.

KA(Knowledge Area)

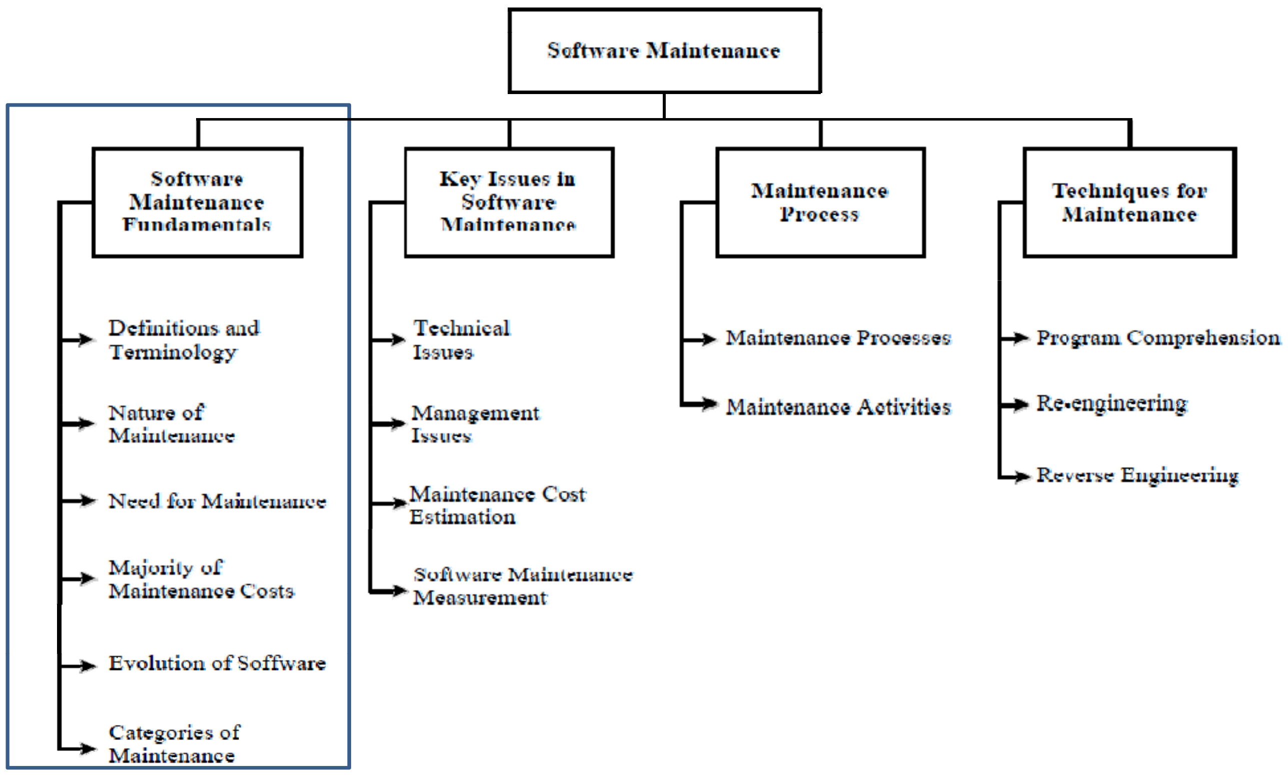
- The Software Maintenance KA is related to all other aspects of software engineering.



Software Maintenance
Software Maintenance Fundamentals
Key Issues in Software Maintenance
Maintenance Process
Techniques for Maintenance
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Processes
Maintenance Activities
Program Comprehension
Re-engineering
Reverse Engineering

Fundamentals

KA



Software Maintenance
Software Maintenance Fundamentals
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Key Issues in Software Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Process
Maintenance Processes
Maintenance Activities
Techniques for Maintenance
Program Comprehension
Re-engineering
Reverse Engineering

Definitions

- IEEE 1219,
  - ✓ the modification of a software product **after delivery**
  - ✓ maintenance **activities prior** to delivery of the software product,
- The IEEE/EIA 12207,
  - ✓ one of the primary life cycle processes
  - ✓ “modification to code and associated documentation due to a problem or the need for improvement. The objective is to modify the existing software product while **preserving its integrity**.”
- ISO/IEC 14764,
  - ✓ emphasizes the **pre-delivery aspects** of maintenance, planning
- SW공학 백서 (2009”, P20 chap2-5)
  - ✓ 유지보수란 개발이 종료된 SW가 사용자에 인수되고 설치되어진 후 일어나는 모든 SW공학적인 활동

Nature Of Maintenance

- Software maintenance sustains the software product throughout its operational life cycle.
  - ✓ Logged, Tracked, Determined(Code, Artifacts)
- Pfleeger [Pfl01] states that
  - ✓ “maintenance has a broader scope, with more to track and control” than development.
- A maintainer, its the primary activities (IEEE/EIA 12207) as
  - ✓ **process** implementation
  - ✓ problem and modification **analysis**;
  - ✓ modification **implementation**
  - ✓ maintenance **review/acceptance**
  - ✓ **migration**
  - ✓ and **retirement**.

Need for Maintenance

- Maintenance is needed to ensure that the software **continues to satisfy** user requirements .... using any software life cycle model (for example, spiral).
- Objectives,
  - ✓ **Correct** faults, **Improve** the design, Implement **enhancements, Interface** with other systems
  - ✓ **Adapt** programs so that different hardware, software, system features, and telecommunications facilities can be used
  - ✓ **Migrate** legacy software, **Retire** software
- The maintainer’s activities (Pfleeger [Pfl01]):
  - ✓ Maintaining control over the **software’s day-to-day** Functions
  - ✓ Maintaining control over software **modification**
  - ✓ **Perfecting** existing functions
  - ✓ **Preventing** software performance from **degrading to unacceptable levels**

Majority of Maintenance Costs

- A common perception of software maintenance is that it merely fixes faults.
  - ✓ over 80%, of the software maintenance effort is used for non-corrective actions. [Abr93, Pig97, Pre01]

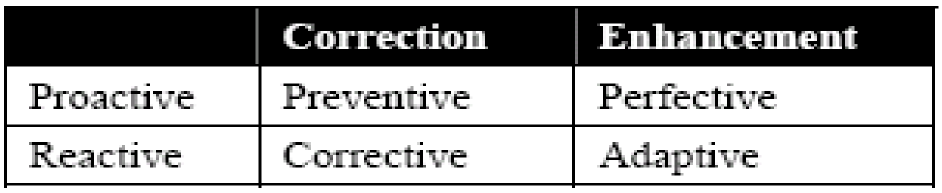
- Pfleeger [Pfl01] presents some of the technical and non-technical factors affecting software maintenance costs,
  - ✓ Application type
  - ✓ Software novelty(새로움의 정도)
  - ✓ Software maintenance staff availability
  - ✓ Software life span
  - ✓ Hardware characteristics
  - ✓ Quality of software design, construction,
  - ✓ documentation and testing

Software Evolution

- [Leh97] Key findings include
  - ✓ the fact that maintenance is **evolutionary developments**
  - ✓ maintenance decisions **aid**ed by understanding what happens to systems (and software) over time.
  - ✓ Others state that maintenance is continued development, except that there is an extra input (or constraint)–existing large software is **never complete** and **continues to evolve**.
  - ✓ As it evolves, it grows more complex
- predictive models to estimate maintenance effort have been made ...
  - ✓ Maintenance & Support (ISBSG, UKSMA)
    - www.isbsg.org, www.uksma.co.uk
  - ✓ COCOMO 81 : exclusion of, renewed(> 50%) development
  - ✓ COCOMO II : COCOMO81 + Scale Parameters
  - ✓ IFPUG : EFP = (ADD + CHG + CFP) × VAFa + (DEL × VAFb)
  - ✓ Nesma : Impact Factor based UFP
  - ✓ Perfective Maintenance Estimation based on FP : same FP value based approach to both of Dev-side, MA-side
  - ✓ etc

Categories of Maintenance

- ISO/IEC 14764
  - ✓ Preventive maintenance (예방적) ➔ Correction Category (순응적)
    - Modification of a software product **after delivery** to detect and correct latent faults in the software product before they become effective faults
  - ✓ Corrective maintenance (교정적) ➔ Correction Category (반응적)
    - Reactive modification of a software product performed **after delivery** to correct discovered problems
  - ✓ Perfective maintenance (완전적) ➔ Enhancement (순응적)
    - Modification of a software product after delivery to **improve** performance or Maintainability
  - ✓ Adaptive maintenance (적응적) ➔ Enhancement(반응적)
    - Modification of a software product performed after delivery to **keep** a software **product usable** in a changed or changing environment

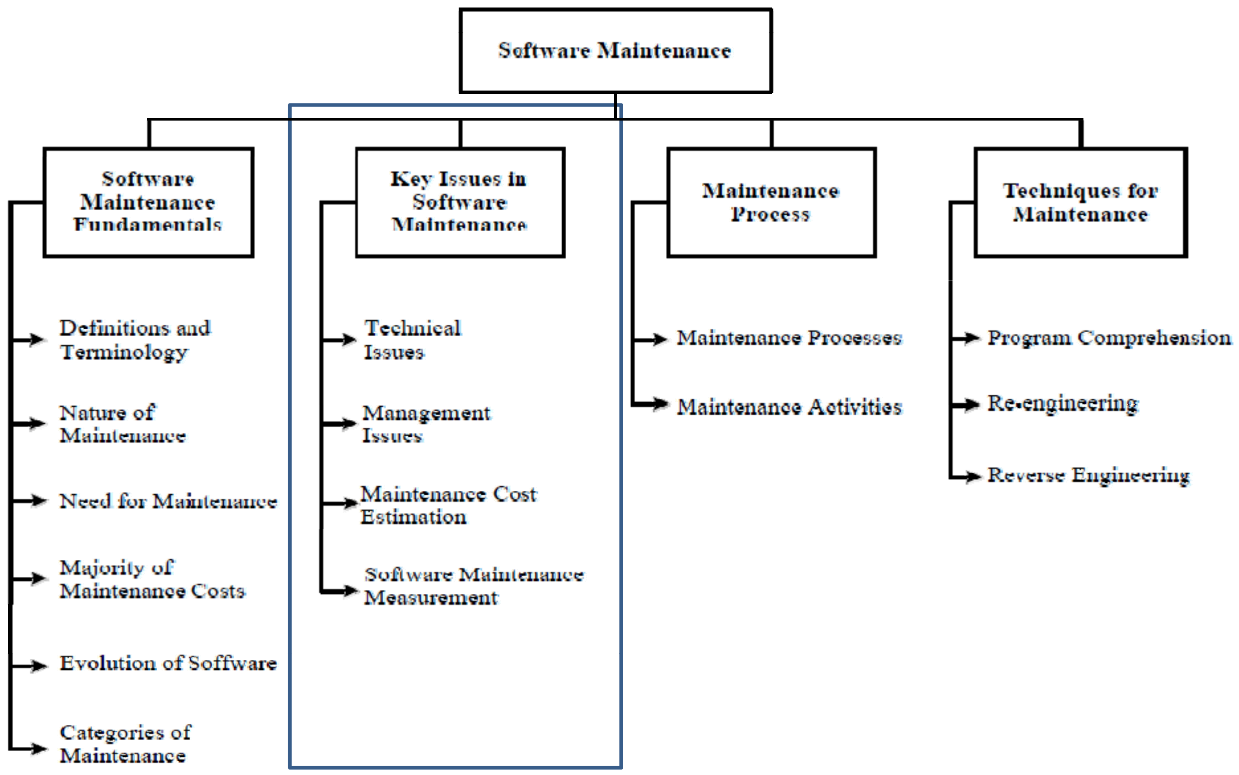


	Correction	Enhancement
Proactive	Preventive	Perfective
Reactive	Corrective	Adaptive

Table 1: Software maintenance categories

Issues

KA



Software Maintenance
Software Maintenance Fundamentals
Key Issues in Software Maintenance
Maintenance Process
Techniques for Maintenance
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Processes
Maintenance Activities
Program Comprehension
Re-engineering
Reverse Engineering

Group of Issues

- A number of key issues
  - ✓ competing with software developers for resources is a constant battle. Planning for a future release, while coding the next release and sending out emergency patches for the current release, also creates a challenge.
- Grouping
  - ✓ Technical issues
  - ✓ Management issues
  - ✓ Cost estimation
  - ✓ Measures

Technical Issues

- Limited understanding
  - ✓ DEF) how quickly a software engineer can understand where …
  - ✓ Research indicates that some **40% to 60% of the maintenance effort** is devoted to understanding
  - ✓ more difficult in text-oriented representation, in source code,
- Testing
  - ✓ Regression testing : is important to maintenance.
    - But unintended effects, ➔ no time)
  - ✓ the **challenge of coordinating** tests when different members of the maintenance team are working on different problems **at the same time**
  - ✓ it may be impossible to bring it **offline** to test.
- Impact analysis
- how to conduct, cost effectively,
- a complete analysis of the impact of a change in existing software
- Maintainability

Technical Issues

- Impact analysis
  - ✓ DEF) how to conduct, cost effectively, a complete analysis of the impact of a change in existing software
    - Impact analysis, Risk analysis by modification(MR, PR)
  - ✓ It is performed after a change request enters the software configuration management process. (Is it real ?)
  - ✓ [Art88] states that the objectives of impact analysis
    - **scope of a change** ,**estimates of resources**, **cost/benefits, Communication**
- Maintainability
  - ✓ (IEEE [IEEE610.12-90]) defines maintainability as the ease with which software can be maintained, enhanced, adapted, or corrected to satisfy specified requirements.
  - ✓ To reduce maintenance costs.
    - **Difficult** : not an important focus during the software development process.
    - Reviewed, controlled (be helped by systematic and mature processes, techniques, and tools)

Management Issues

- Alignment with organizational objectives
  - ✓ how to demonstrate the ROI of software maintenance activities.
  - ✓ But, Not Clear ← *issues 1*
    - In contrast(Development), software maintenance often has the objective of extending the life of a software for as long as possible.
- Staffing
  - ✓ "second-class citizens" ← issues2
- Process
  - ✓ software maintenance activities shares much in common with software development ← issue 3 (management cost)

Management Issues

- Organizational aspects of maintenance
  - ✓ DEF) how to identify which organization and/or function will be responsible
  - ✓ What is important is the delegation or assignment of the maintenance responsibility to a single group or person
- Outsourcing
  - ✓ less mission critical software, as companies are **unwilling to lose control** of the software used in their core business.
  - ✓ McCracken (McC02) states that 50% of outsourcers provide services without any clear service-level agreement.

Cost Estimation Issues

- Cost estimation
  - ✓ Maintenance cost estimates are affected by many technical and non-technical factors.
  - ✓ the use of parametric models , the use of experience[ISO14764-99:s7.4.1].
- Parametric models
  - ✓ [Boe81, Ben00] Of significance is that data from past projects are needed in order to use the models.
- Experience
  - ✓ Clearly, the best approach to maintenance estimation is to **combine** empirical data and experience.

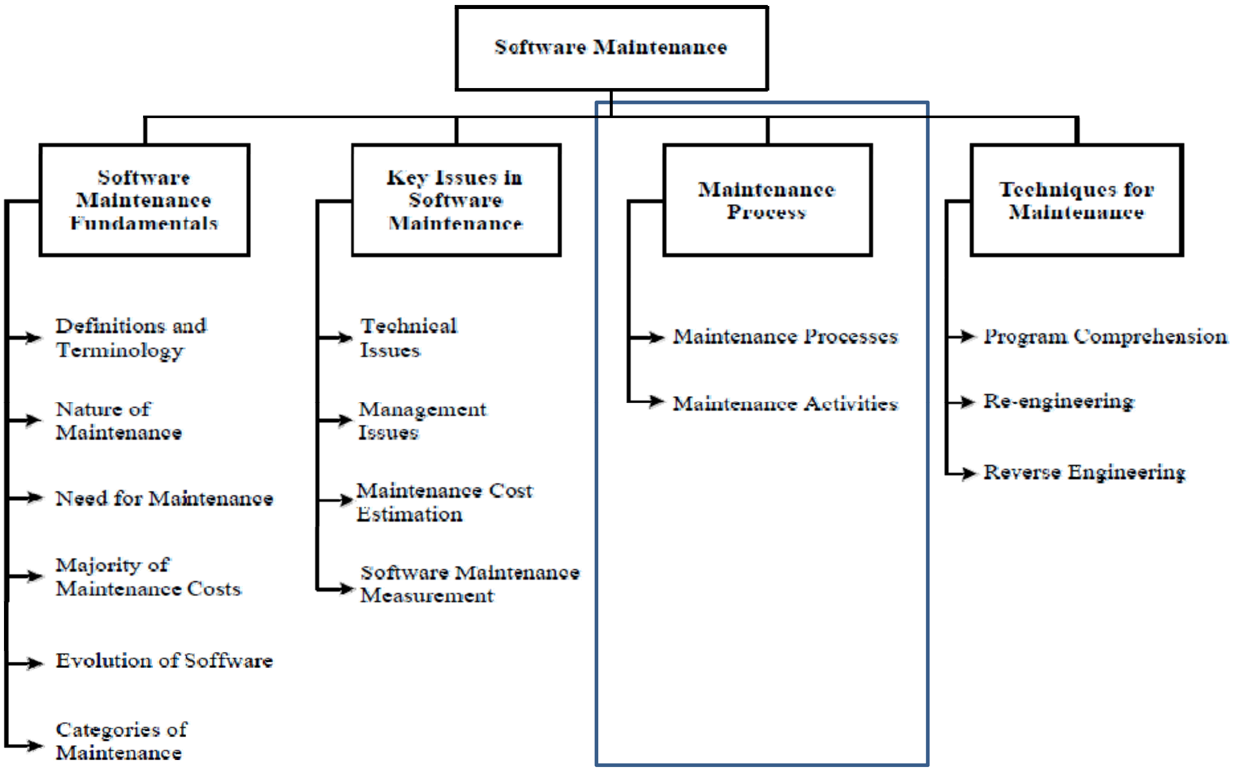
Measure Issues

- PSM project describes
  - ✓ an issue driven measurement process that is used by many organizations and is quite practical.
  - ✓ process and product measurement ➔ SWE Process KA.
  - ✓ The software measurement ➔ SWE Mgmt. KA
- Specific Measure
  - ✓ The maintainer must determine which measures are appropriate for the organization in question.
  - ✓ [IEEE1219- 98; ISO9126-01; Sta94] suggest
    - Analyzability (분석용이성)
    - Changeability (변경 용이성)
    - Stability (안정성)
    - Testability (시험 용이성)

Process

SWEBOK Seminar

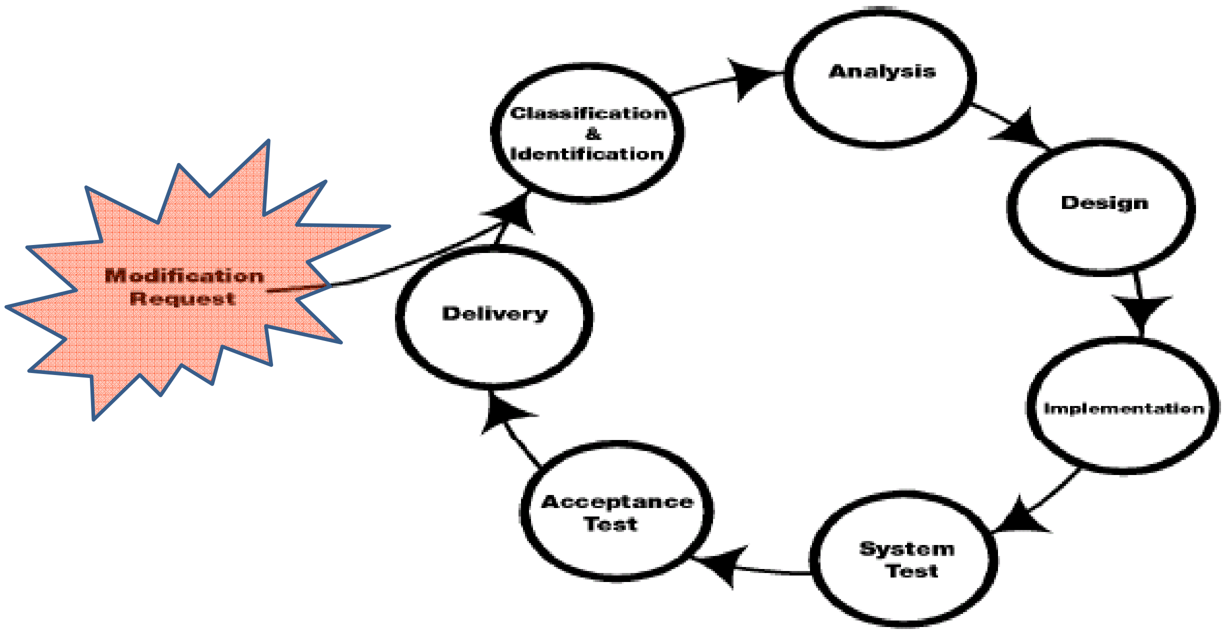
KA



Software Maintenance
Software Maintenance Fundamentals
Key Issues in Software Maintenance
Maintenance Process
Techniques for Maintenance
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Processes
Maintenance Activities
Program Comprehension
Re-engineering
Reverse Engineering

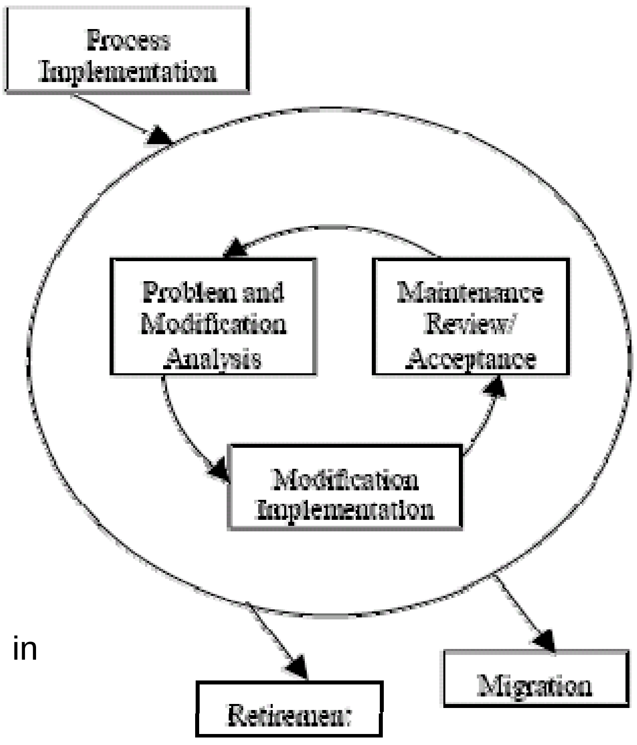
Maintenance Process

- IEEE 1219-98
  - ✓ starts with the software maintenance effort during the post-delivery stage and discusses items such as planning for maintenance.



Maintenance Process

- ISO/IEC 14764-00
  - ✓ Process Implementation
  - ✓ Problem and Modification Analysis
  - ✓ Modification Implementation
  - ✓ Maintenance Review/Acceptance
  - ✓ Migration
  - ✓ Software Retirement



Process Implementation
Problem and Modification Analysis
Maintenance Review/ Acceptance
Modification Implementation
Retirement
Migration

- Etc…
  - ✓ agile methodologies have been emerging which promote light processes
  - ✓ Xtreme maintenance are presented in (Poo01)

Maintenance Activities

- Unique activities
  - ✓ Transition : Developer ➔ Maintainer
  - ✓ Modification Request acceptance/rejection : can be rerouted to a developer
  - ✓ Modification Request and Problem Report Help Desk : end-user support function
  - ✓ Impact Analysis
  - ✓ Software Support : a request for information (Give me last month retrievals !~)
  - ✓ SLAs and specialized domain-specific) maintenance contracts

- Supporting activities
  - ✓ planning, software configuration management, verification and validation, software quality assurance, reviews, audits, and user training.

Maintenance Activities

- Maintenance planning activity
  - ✓ Business planning (organizational level) ➔ ISP
  - ✓ Maintenance planning (transition level) ➔ PM
    - Concept Document [ISO14764-99:s7.2]
      Scope, Adaptation(process), Identification(organization), estimation cost
  - ✓ Release/version planning (software level) ➔ SCM
    - Collect the dates of availability of individual requests : sizing
    - Agree with users on the content of subsequent releases/versions : Contract
    - Identify potential conflicts and develop alternatives : Risk Mgmt
    - Assess the risk of a given release : Assess,
    - Inform all the stakeholders : Coordination
  - ✓ Individual software change request planning (request level) ➔ SRS/SLA
    - Guideline : IEEE 1219, ISO/IEC 14764,

- Software configuration management
- Software quality

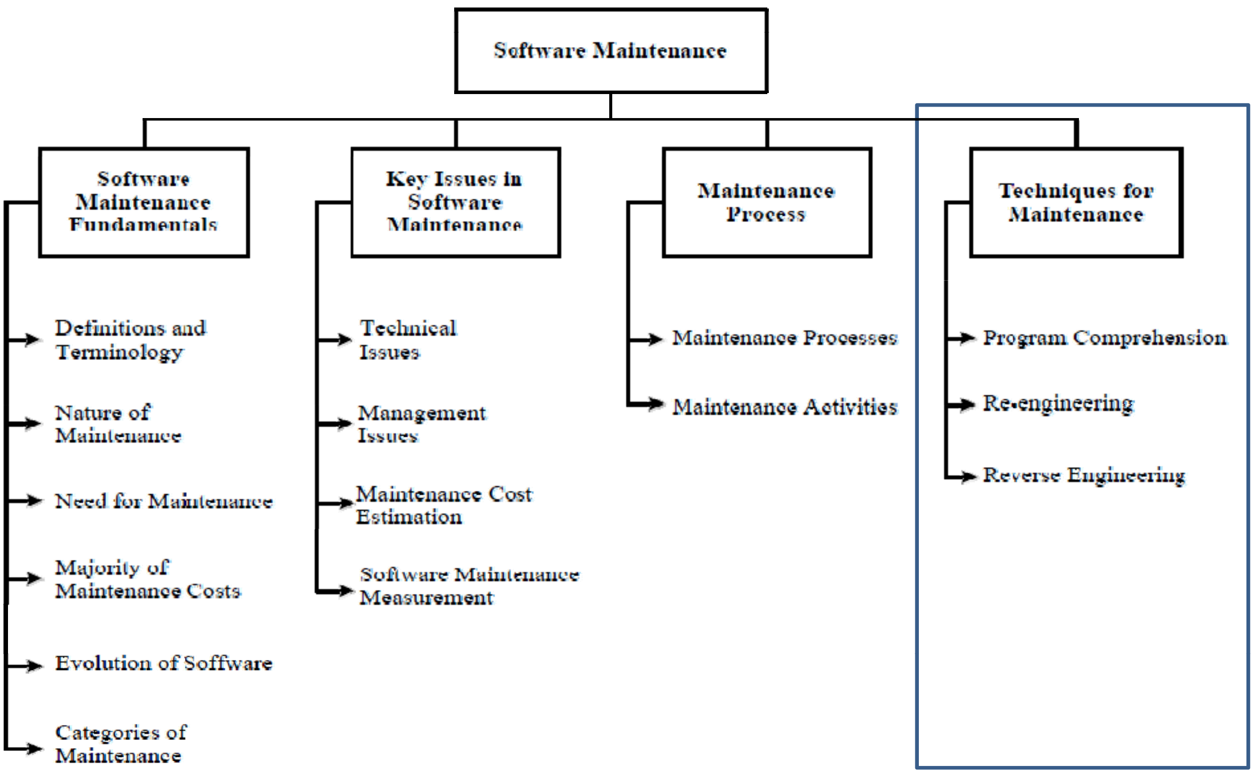
Maintenance Activities

- Software configuration management
  - ✓ software configuration management as a critical element(V&V, audit, authorize, implement, release..) of the maintenance process. [IEEE 1219]
  - ✓ SCM for software maintenance is different
    - the number of small changes that **must be controlled** on operational software.

- Software quality
  - ✓ The activities and techniques for Software Quality Assurance (SQA), V&V, reviews, and audits
  - ✓ Adapt SW Dev. Process, techniques and deliverables, is recommended [ISO 14764]

Techniques

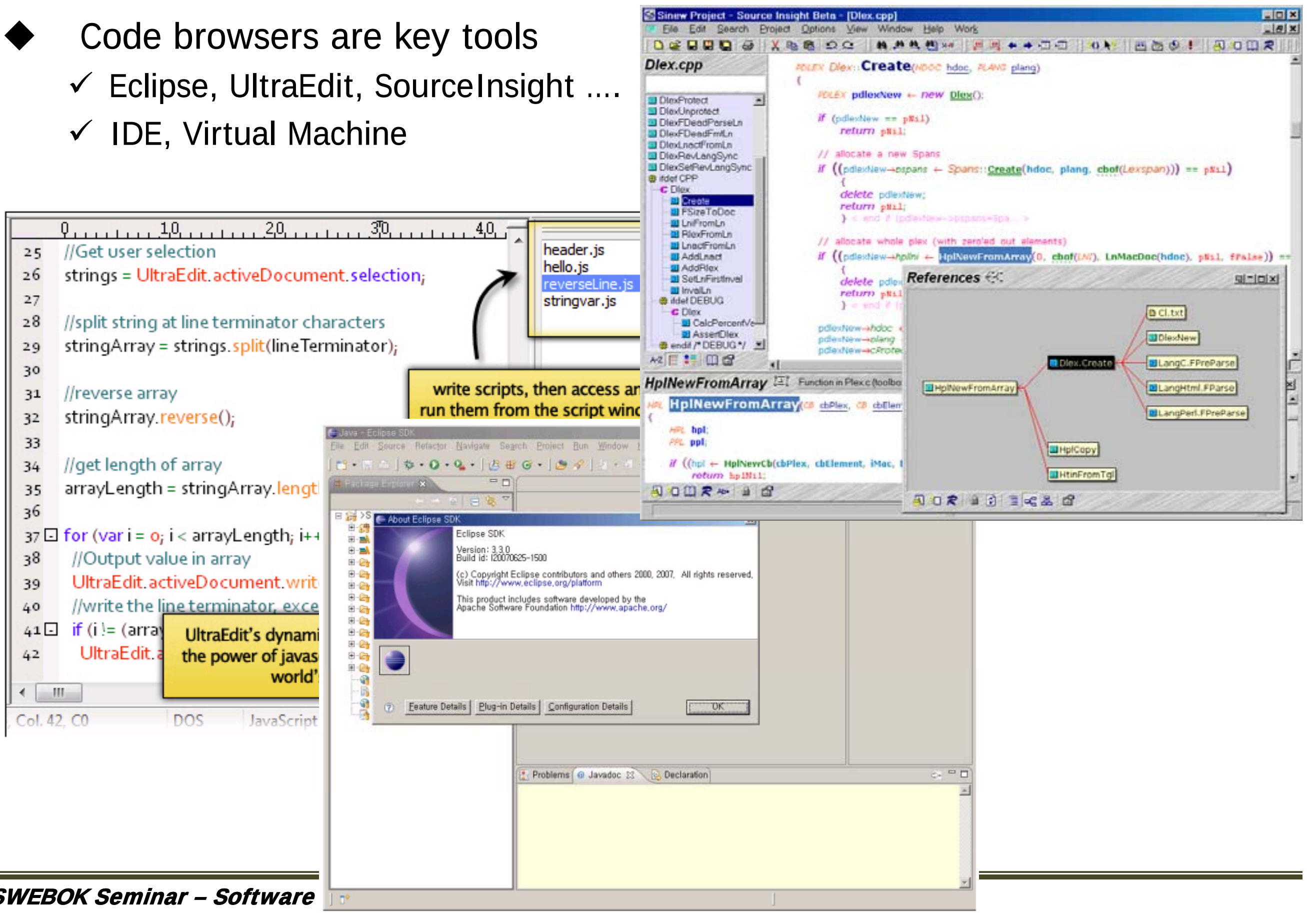
KA



Software Maintenance
Software Maintenance Fundamentals
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Key Issues in Software Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Process
Maintenance Processes
Maintenance Activities
Techniques for Maintenance
Program Comprehension
Re-engineering
Reverse Engineering

Program Comprehension

- Code browsers are key tools
  - ✓ Eclipse, UltraEdit, SourceInsight ....
  - ✓ IDE, Virtual Machine



```
//Get user selection
strings = UltraEdit.activeDocument.selection;

//split string at line terminator characters
stringArray = strings.split(lineTerminator);

//reverse array
stringArray.reverse();

//get length of array
arrayLength = stringArray.lengt

for (var i = 0; i < arrayLength; i++
  //Output value in array
  UltraEdit.activeDocument.writ
  //write the line terminator, exce
  if (i != (array
    UltraEdit.a
```

Eclipse SDK

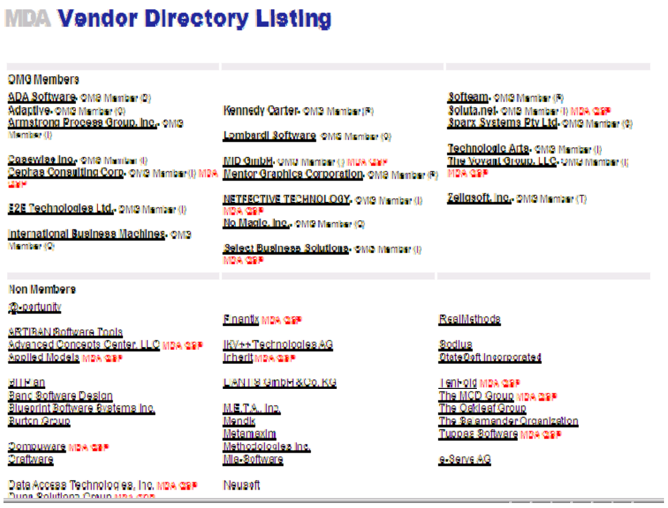
Version: 3.3.0
Build id: I20070625-1500

(c) Copyright Eclipse contributors and others 2000, 2007. All rights reserved.
Visit http://www.eclipse.org/platform

This product includes software developed by the
Apache Software Foundation http://www.apache.org/

Reengineering

- DEF) the examination and alteration of software to reconstitute it in a new form, and includes the subsequent implementation of the new form.
  - ✓ Expensive form of alteration
  - ✓ It is often not undertaken to improve maintainability, but to replace aging legacy software.

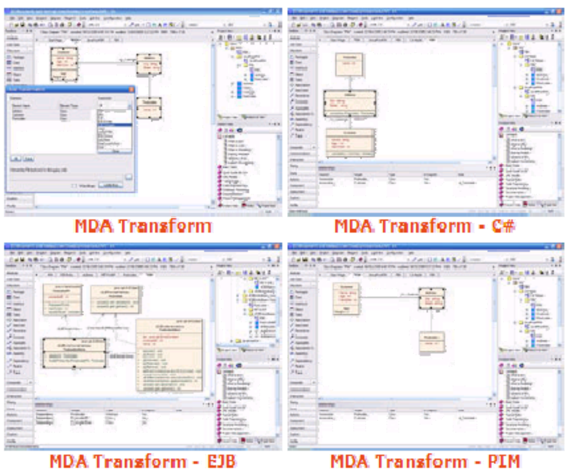


MDA Vendor Directory Listing

OMG Members		
ADA Software- OMG Member (C)		Softeam- OMG Member (P)
Adaptive- OMG Member (C)	Kennedy Carter- OMG Member (P)	Soluta.net- OMG Member (I) MDA QSP
Armstrong Process Group, Inc.- OMG Member (I)		Sparx Systems Pty Ltd- OMG Member (C)
	Lombardi Software- OMG Member (C)	
		Technologic Arts- OMG Member (I)
Casewise Inc.- OMG Member (I)	MID GmbH- OMG Member (I) MDA QSP	The Voyant Group, LLC- OMG Member (I) MDA QSP
Cephas Consulting Corp- OMG Member (I) MDA QSP	Mentor Graphics Corporation- OMG Member (P)	
	NETFECTIVE TECHNOLOGY- OMG Member (I) MDA QSP	Zellasoft, Inc.- OMG Member (T)
E2E Technologies Ltd.- OMG Member (I)		
	No Magic, Inc.- OMG Member (C)	
International Business Machines- OMG Member (C)		
	Select Business Solutions- OMG Member (I) MDA QSP	
Non Members		
@-portunity		
	Enantix MDA QSP	RealMethods
ARTiSAN Software Tools		
Advanced Concepts Center, LLC MDA QSP	IKV++ Technologies AG	Sodius
Applied Models MDA QSP	Interlt MDA QSP	StateSoft Incorporated
BITPlan	LANTIS GmbH & Co. KG	Tenfold MDA QSP
Banc Software Design		The MCD Group MDA QSP
Blueprint Software Systems Inc.	M.E.T.A., Inc.	The Oakleaf Group
Burton Group	Mendix	The Salamander Organization
	Metamaxim	Tuppas Software MDA QSP
Compuware MDA QSP	Methodologies Inc.	
Craftware	Mia-Software	e-Serve AG
Data Access Technologies, Inc. MDA QSP	Neusoft	
Dune Solutions Group MDA QSP		

http://www.omg.org/mda/presentations.htm

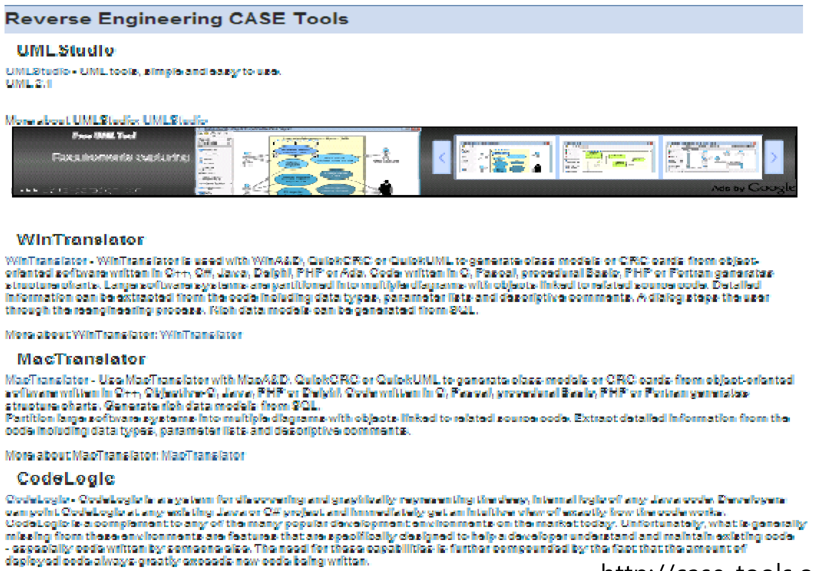
Get started with UML and MDA now.



http://www.sparxsystems.com/platforms/mda_tool.html

Reverse engineering

- DEF) the process of analyzing software to **identify** the software's components and their interrelationships and to create representations of the software in another form or at higher levels of abstraction.
  - ✓ it does **not change** the software, or result in new software.
  - ✓ produce call graphs and control flow graphs from source code.



Reverse Engineering CASE Tools

UMLStudio

UMLStudio - UML tools, simple and easy to use.
UML 2.1

More about UMLStudio: UMLStudio

WinTranslator

WinTranslator - WinTranslator is used with WinA&D, QuickCRC or QuickUML to generate class models or CRC cards from object-oriented software written in C++, C#, Java, Delphi, PHP or Ada. Code written in C, Pascal, procedural Basic, PHP or Fortran generates structure charts. Large software systems are partitioned into multiple diagrams with objects linked to related source code. Detailed information can be extracted from the code including data types, parameter lists and descriptive comments. A dialog steps the user through the reengineering process. Rich data models can be generated from SQL.

More about WinTranslator: WinTranslator

MacTranslator

MacTranslator - Use MacTranslator with MacA&D, QuickCRC or QuickUML to generate class models or CRC cards from object-oriented software written in C++, Objective-C, Java, PHP or Delphi. Code written in C, Pascal, procedural Basic, PHP or Fortran generates structure charts. Generate rich data models from SQL.
Partition large software systems into multiple diagrams with objects linked to related source code. Extract detailed information from the code including data types, parameter lists and descriptive comments.

More about MacTranslator: MacTranslator

CodeLogic

CodeLogic - CodeLogic is a system for discovering and graphically representing the deep, internal logic of any Java code. Developers can point CodeLogic at any existing Java or C# project and immediately get an intuitive view of exactly how the code works.
CodeLogic is a complement to any of the many popular development environments on the market today. Unfortunately, what is generally missing from these environments are features that are specifically designed to help a developer understand and maintain existing code - especially code written by someone else. The need for these capabilities is further compounded by the fact that the amount of deployed code always greatly exceeds new code being written.

http://case-tools.org/reverse_engineering.html

Thanks !!