

Software Maintenance Technique

SE 323 Software Construction,
Testing and Maintenance



Agenda

- Software Maintenance People
- Software Maintenance Technique



2

Software Maintenance: Expectation

- End user
 - Ease of use issue
 - UI requires too much clicks before enter function
 - Product misbehavior
 - Malfunction
 - Product instability
 - Product that not stable
 - Performance issues
 - Performance
 - Response time of the software



3

Software Maintenance: Expectation

- Developer
 - Assistance in customization of products
 - Resolving the techniques in set-up product
 - Getting quick turn-around for a report
 - Participating in the technology conferences of vendors
 - Utilizing opportunities to provide feedback on feature



4

Software Maintenance: Expectation



- Administrator
 - Requirement of maintainability
 - Easy install for large number of installation
 - Assistance in installation of product
 - Manageability features
 - Area of enterprise software

5

Software Maintenance: Expectation



- IT Management
 - A more participatory role with the vendor in planning and execution of application update
 - Service Level Agreement
 - Quick turn-around for business impact problems
 - Uptime commitment
 - Receive proactive communication from vendor
 - Opportunity to provide input into future release

6

Software Maintenance: Expectation



- Senior management and executives
 - Strategic alignment such as technologies, feature planning, etc.
 - Long-term commitments

7

Maintenance Teams



- | | |
|--|---|
| • Development Projects | • Maintenance Projects |
| • Duration <ul style="list-style-type: none">• Relatively longer timeframes | • Duration <ul style="list-style-type: none">• Shorter timeframes |
| • Continuity <ul style="list-style-type: none">• Continuous | • Continuity <ul style="list-style-type: none">• Discrete/ discontinuous |
| • Incremental nature <ul style="list-style-type: none">• Can build on past work and is usually incremental | • Incremental nature <ul style="list-style-type: none">• Seldom incremental |

8

Maintenance Teams



- Development Projects
 - Ownership
 - Usually greater sense of ownership is easily achieved
 - Predictability
 - Amenable to planning
 - Nature of work
 - Can be proactive
- Maintenance Projects
 - Ownership
 - More difficult to achieve the sense of ownership
 - Predictability
 - Difficult to plan
 - Nature of work
 - Usually reactive

9

Maintenance Team: Skill Set Required for Maintenance Engineer



- Good knowledge of product usage
- Problem solving ability to enable quick progress from symptoms to root causes
- Good technical and debugging skill
- Good attitude
 - Appreciating the value of maintenance for the organization
- Good communication skill

10

Maintenance Team: Skill Set Required for Maintenance Engineer



- Ability to react quickly
 - Maintenance projects are time critical
 - Need immediately response
- Ability to handle multiple tasks simultaneously
- Process knowledge

11

Maintenance Team: Skill Set Required for Support Analyst and Product-in-charge



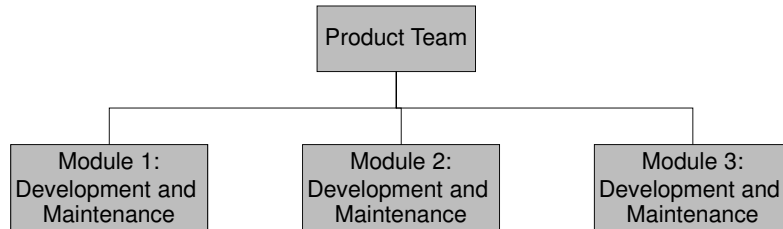
- Ability to evaluate the impact of a product and a problem on customer's business
- Understand the product's external behavior to ascertain the nature of the problem
- Communicate effectively within the organization
- Communicate effectively with the other side

12

Maintenance Team: Organization Base on Function



- Single Team Model



13

Maintenance Team: Organization Base on Function



- Single Team Model

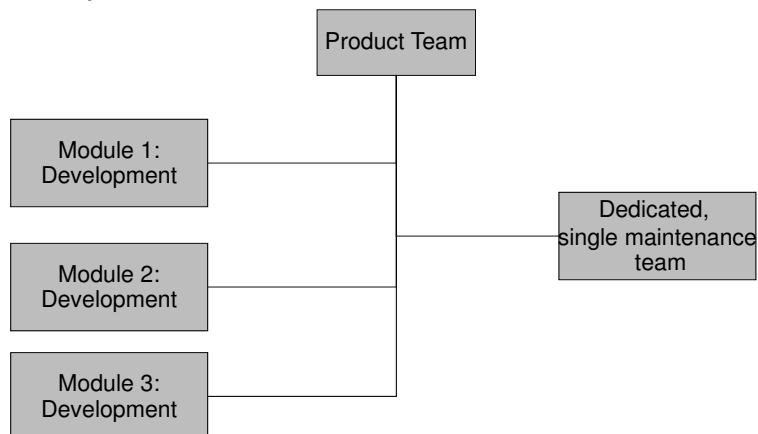
- Advantage
 - Single point for the quality of module
 - Bring better sense of ownership
 - Team work on new technologies and version
- Disadvantage
 - Planning and scheduling difficult

14

Maintenance Team: Organization Base on Function



- Separate team model



15

Maintenance Team: Organization Base on Function



- Separate team model

- Advantage
 - Clear demarcation of development and maintenance responsibilities
 - Easier for developer team to predict and commit deadline
 - Engineer can focus on specific skill sets
- Disadvantage
 - Learning curve
 - Price of recruitment

16

Maintenance Team: When?



- Single team Model when
 - Product is evolving with dynamically changing requirement
 - Underlying technology is an emerging technology
 - Version of product have a short time span
 - New release coming out more frequently
 - Skill set for understanding and maintaining the product are difficult to find
- Otherwise separate teams model

17

Maintenance Team: Organization Base on Location



- Divide by product model
 - For multiple software product, at multiple location
 - Responsible for maintain for their own product
- Divide by feature model
 - Point using this model
 - Feature that are closely dependent should be allocate to single location
 - Interface between modules should be de defined reasonably clearly
 - Having too many location will be counter-productive

18

Maintenance Team: Organization Base on Location



- Divide by version model
 - Product version becomes more stable, maintenance workload on the version decrease
 - Maintenance of older release move to a less expensive location

19

Maintenance Team: Organization Base on Location



- Follow the Sun model
 - Maintenance teams allocate around the world
 - The team which is in office hour handle change request

20

Maintenance Team: Organization Base on Location



- Follow the Sun
 - Key factors
 - Equally talented engineer at all location
 - Excellent team work, mutual respect and trust amongst the multiple location
 - Well established communication norms
 - Proper and foolproof work hand-over across multiple location
 - A fully integrated problem repository
 - Comparing with multiple shifts
 - Do not utilize global talent
 - No diversify risks sufficiently

21

Maintenance Team: Compensation and Reward System



- Support Analyst
 - Measurement
 - Number of calls taken and responded to
 - Resolve issues correctly, reduce the overall cost of maintenance
 - Resolving may be
 - Addressing the customer question
 - Analyzing the problem and routing it to the correct development/maintenance group

22

Maintenance Team: Compensation and Reward System



- Support Analyst
 - Rewards for
 - Helping other when appropriate in resolving issues
 - Mentoring and training offered to new analyst
 - Skill upgrade

23

Maintenance Team: Compensation and Reward System



- Maintenance engineer
 - Measurement
 - Number of problem fixed (and not fixed)
 - Response to critical problem
 - Can have customer satisfaction or not
 - Keeping up the adaptive maintenance and other assign activities
 - Skill upgrades acquired
 - Learning ability

24

Maintenance Team: Compensation and Reward System



- Maintenance engineer
 - Reward for
 - Developing and enhancing maintenance tools that increase productivity
 - Contributing to process improvements
 - Monitoring new engineering by providing them with the necessary training and coaching

25

Maintenance Team: Best Practice



- Job Rotation between various job function
 - Not boring?
 - Change responsibility to the correct people
 - Developer -> maintainer for that part
 - Enable people in to get a fully round perspective of the other job functions
 - Build personal rapport across the various team

26

Maintenance Team: Best Practice



- Internal deputations across location
 - Rotate maintenance leader a few months to another center
 - Learn difference experience to share with each other
- Planning preventive maintenance
 - Aggregate review and analysis when doing a corrective maintenance

27

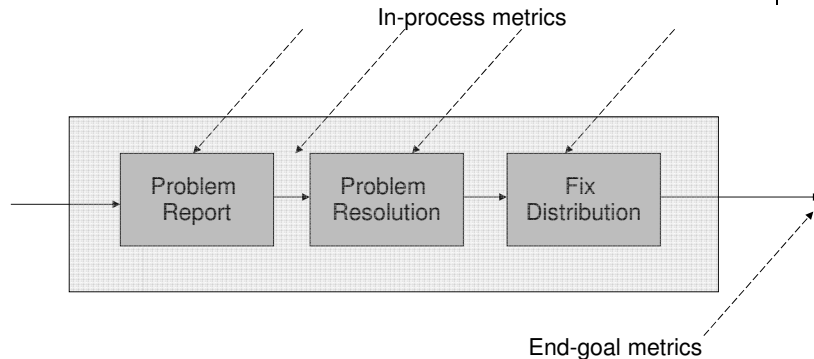
Metrics and Measurements



- In-process metric
 - Measurements to estimate progress
- End-goal metrics
 - Measurement of how well to achieved the goals

28

Metrics and Measurements



29

Metrics and Measurements



- End-goal metrics
 - Dictated by customer, market or senior management
 - Not directly under the control of the internal organization
 - Indicate customer expectation or market success
- In-process metrics
 - Identified by internal groups
 - Internal organizations should be able to control these metrics
 - Indicates which parts of the internal process are bottlenecks to achieving the end-goal metrics

30

Metrics and Measurements: Questions in a Metric Program



- What is our goal or where do we want to go?
- What is our current position
- Knowing where we are and where we want to go, what steps should we take?
- How do we measure our process?
- What corrective actions do we take when we see our problem deviating from the expected progress?

31

Metrics and Measurements: Common Metrics for Maintenance



- Mean time between failure (MTBF)
 - Average time between the arrival of bugs;
 - Indicate inherent quality
- Mean time to repair (MTBR)
 - Average time to fix a bug
 - Indicate responsiveness to bugs and effectiveness in fixing a problem

32

Metrics and Measurements: Common Metrics for Maintenance



- Number of problems responded to by “first level support”
 - Number of problem solved by call center or first contacted team
 - Indicate the effectiveness of first level support
- Classification of defects by severity
 - Indicate the nature of incoming problems and problem and hence measures

33

Metrics and Measurements: Common Metrics for Maintenance



- Classification of defects by product/component
 - Indicate the problem prone parts of a product
 - Points to be looked up into more carefully

34

Metrics and Measurements: Setting Target: SMART



- Specific
 - Target should be specific in term on number
- Measurable
 - Target should be measurable unambiguously and interpreted consistently
- Aggressive yet achievable
 - Should be possible to achieve
- Results-oriented
- Time-bound

35

Metrics and Measurements: People Issues



- Management commitment is essential for the success of metrics
- Not shooting the messenger
- Focus on analysis of aggregate result, not on individual performance
- Make sure the operational issues of metrics are understood consistently by everyone

36

Metrics and Measurements: Best Practice



- Automating the process through the effective use of repositories
- Integrating metrics in operational decision-making, not just as collection mechanism
- Over-dose of metrics
- Metrics as a “policing” activity

37

Reverse Engineering



- Forward Engineering
 - Traditional software engineering
- Reengineering
 - Process of reexamination and alteration
 - Altered by reverse engineering and then forward engineering
- Restructuring
 - Transformation of a system from one representational form to other

38

Reverse Engineering



- Reverse engineering
 - Identify the system's component and their interrelationships
 - Create representation of the system in another form or at higher level of abstraction

39

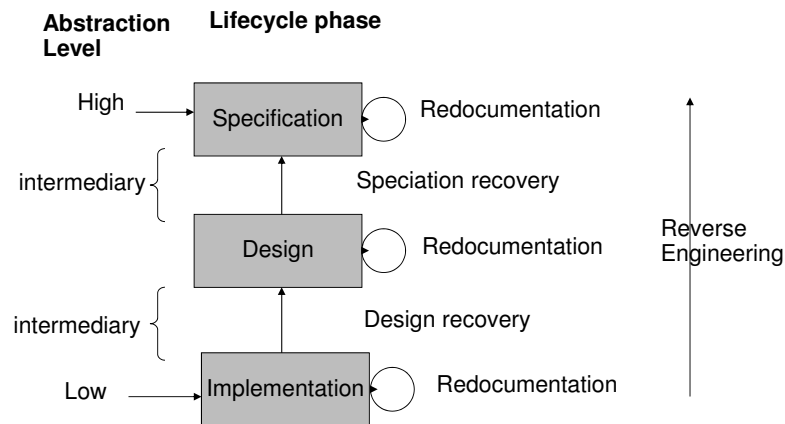
Reverse Engineering: Benefit in maintenance



- Enhances understanding
 - Assist identification of errors
- Facilitates identification and extraction of component affected by adaptive and perfective change
- Provide documentation or alternative view of the system

40

Reverse Engineering: Level of Reverse Engineering



41

Program Comprehension



- Significant proportion of maintenance effort and resources
 - Reading code at HP estimated cost 200 million dollar (1993)
- Understanding for maintenance involves
 - Knowledge of what the software system does
 - Identify where in the system changes are to be effected
 - Having an in-depth knowledge of how the parts to be corrected or modified work

42

Program Comprehension: Acquiring Information



- Acquiring information about certain aspects of the software system
 - Problem domain
 - Execution effect
 - Cause-effect relation
 - Product-environment relation
 - Decision-support features

43

Program Comprehension: Acquiring Information



- Problem Domain
 - To direct maintenance personnel
 - Selection of personnel with the appropriate level of expertise and skills
- Execution Effect
 - What results the program will produce for a given input
 - Result that individual program unit will produce
 - Assist maintenance personal to determine whether an implemented change achieved the desire effect

44

Program Comprehension: Acquiring Information



- Cause-Effect relation
 - Allow maintenance personnel to reason about how components of a software product interact during execution
 - Enable programmer to predict the scope of change
 - Any knock-on effect that may arise from the change
 - Trace the flow of information through the program

45

Program Comprehension: Acquiring Information



- Product-environment relation
 - Conditions and influences which act from outside upon the product
 - Help maintenance personnel to predict how change in environment will affect the product in general
- Decision-support features
 - Using some features for helping making decision
 - Eg. McCabe's metric

46

Program Comprehension: Comprehension Process Model



- Step 1 read about the program
 - System document
 - Data and control flow diagram
- Step 2 Read the source code
 - Global view
 - Scope on knock-on effect
 - Local view
 - Specific part of program

47

Program Comprehension: Comprehension Process Model



- Step 3 run the program
 - To study the dynamic behavior of program in action

48

Configuration Management



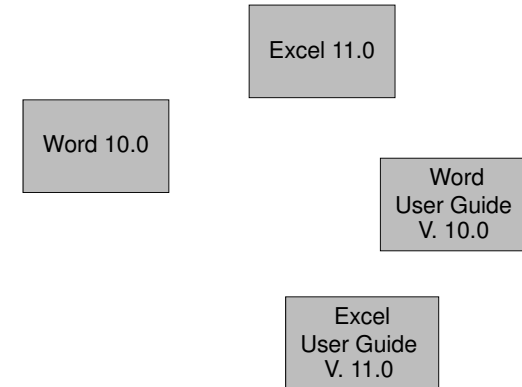
- Configuration is a set of related item
 - Uniquely identifiable
 - Consistent
 - Set of items is re-creatable as unit

49

Configuration Management: Customer and Producer Perspective



- Customer's view of configuration

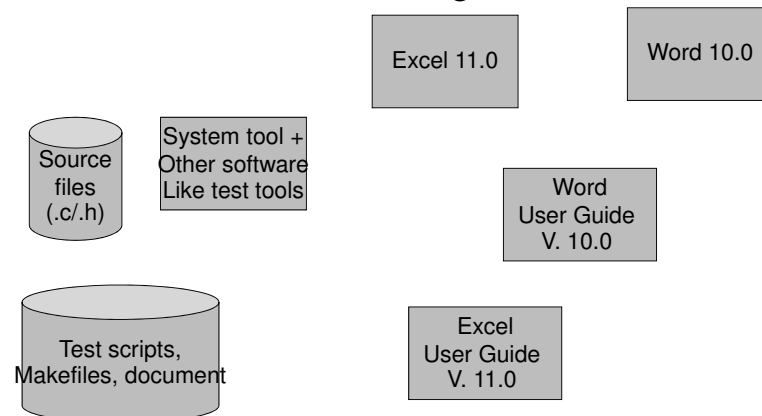


50

Configuration Management: Customer and Producer Perspective



- Producer's view of configuration



51

Configuration Management: Customer and Producer Perspective



- Customer
 - Quickly describe his or her environment to the software producer when reporting a problem
- Software Producer
 - Ascertain the legal validity of the customer possessing license to use this software
 - Try to recreate the customer environment of the software to simulate and resolve problems reported

52

Configuration Management: What is Configuration Item



- Identified or version
 - Each item has a version number
- Tracked
 - Any activity on the item is tracked
- Controlled
 - Any update to the item goes through a well-documented and controlled process

53

Configuration Management: Change Management



- Change control process is subdivided into:
 - Making a change request
 - Reviewing and authorizing the change
 - Carrying out the change
 - Reflecting the change in other related product

54

Configuration Management: Change Management



- Making a change request
 - Realize change would be implement
 - Sometime change occurred by business perspective but not aware of how impact technology
 - Decision should be made whether to implement or not

55

Configuration Management: Change Management



- Reviewing and authorizing the change
 - Considering from 3 dimensions
 - Necessity
 - Focus on 'what' and 'why'
 - Whether the change is required ?
 - What business or technology change warrants this change request?
 - A genuine requirement or 'cool things' without bussiness reason

56

Configuration Management: Change Management



- Appropriateness
 - Change is carried out the right way
 - Focus on 'how'
 - How the change is made?
 - Is this the best way to implement change ?
 - Is this minimal way to make a change
- Impact
 - Ensures that there are no hidden surprise or side effects in implementation this change
 - What impact would it have on other component ?
 - How much effort and resource will be needed

57

Configuration Management: Change Management



- Carrying out the change
 - Update must preserve "ACID" property
 - Atomicity
 - Change all files must take place as one unit
 - Consistency
 - Update start with a consistent state of the repository
 - Leave the repository in another consistent state
 - Isolation
 - Each change should not dependent on the sequence of other updates
 - Durability
 - Once change 'commit', they should permanent

58

Configuration Management: Best Practice



- Pragmatic, adaptable and practical processes
 - Process should be flexible enough to be responsive the change process does not get slow down
- Close collaboration between configuration management and software quality assurance groups
 - Change management and version management ensure the product satisfaction

59

Conclusion



- People issue in software maintenance
 - Understand customer point
 - Maintenance team point
 - How to manage people in maintenance team

60

Q&A

