

CS605 Quiz 3 Fall 2022

1. BAC stands for

Budget at completion....confirm

2. Instead of going through the entire design in a meeting we narrow down the focus by conducting _____ for each component or small group of components.

Walkthrough....confirm

3. Let the value of Task set selector (TSS) for a new project is computed to be less than 1.2 ($TSS < 1.2$) then its degree of rigor will be

Casual....confirm

4. Let the value of Task set selector (TSS) for a new project is computed to be less than 1.2 ($TSS 1.0 - 3.0$) then its degree of rigor will be

Structured....confirm

5. Let the value of Task set selector (TSS) for a new project is computed to be less than 1.2 ($TSS < 2.4$) then its degree of rigor will be

Strict....confirm

6. BCWP refers to

Budgeted cost of work performed so far....confirm

7. The _____ is the amount of surplus time allowed in scheduling tasks so that the network critical path is maintained on schedule.

Total float....confirm

8. _____ reviews the process description for compliance with the organizational policies, standards and other parts of the software project plan

Software Quality Assurance (SQA) group.....confirm

9. If software testing teams defect removal efficiency is 80% then 20% bugs still exist in application.....confirm

10. _____ are sometimes also known as walkthroughs or inspections

Formal technical reviews.....confirm

11. _____ provides the management with the necessary data to be information about product quality through auditing and reporting functions.

Quality assurance.....confirm

12. Schedule Performance index (SPI) is equal to

$SPI = BCWP/BCWS$ confirm

13. Schedule variance (SV) is equal to

$SV = BCWP - BCWS$ confirm

14. For a project if value of TSS is 0.9 then the degree of rigor for this project will be

Casual.....confirm

15. The consequences or impact of risk will be _____ when the project will show failure to meet its requirements and hence will result in mission failure.

Catastrophic.....confirm

16. The consequences or impact of risk will be _____ when the project will show would degrade performance to a point where mission success is questionable.

Critical....confirm

17. The consequences or impact of risk will be _____ when the project will show result in degradation of secondary mission.

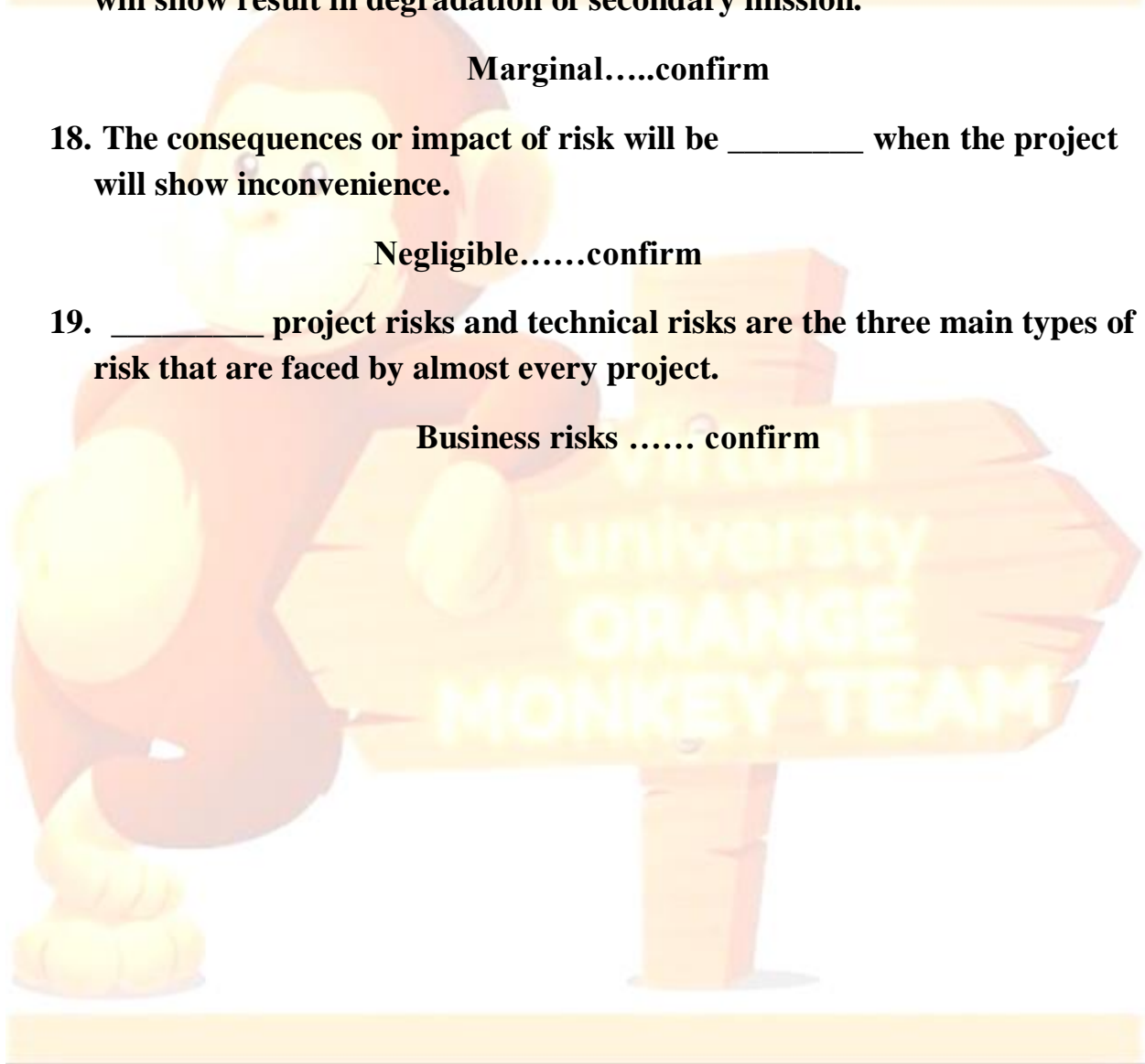
Marginal.....confirm

18. The consequences or impact of risk will be _____ when the project will show inconvenience.

Negligible.....confirm

19. _____ project risks and technical risks are the three main types of risk that are faced by almost every project.

Business risks confirm





CS605-Software
Engineering-II
(Solved Macq's)
LECTURE FROM
(23 to 45)



Junaidfzal08@gmail.com
Bc190202640@vu.edu.pk

FOR MORE VISIT
VULMSHELP.COME

JUNAID MALIK
0304-1659294

AL-JUNAID TECH INSTITUTE

www.vulmshelp.com



Language Courses Training Available

I'm providing paid courses in different languages within 3 Months, Certificate will be awarded after completion.

- HTML
- CSS
- JAVASCRIPT
- BOOTSTRAPS
- IQUERY
- PHP MYSQL
- NODES.JS
- REACT JS

LMS Handling Services

LMS Activities Paid Task

Assignments 95% Results

Quizes 95% Results

GDB 95% Results

For CS619 Project Feel Free To Contact With Me

Ph# 0304-1659294
Email: junaidfzal08@gmail.com

Question No: 1

_____ is defined as the probability of failure free operation of a computer program in a specified environment for a specified time.

▶ **Software Reliability**

- ▶ Software Usability
- ▶ Software Integrity
- ▶ Software Efficiency

Question No: 2

Schedule Performance Index (SPI) is equal to

▶ **BCWP / BCWS**

- ▶ BCWP * BCWS
- ▶ BCWP + BCWS
- ▶ BCWP - BCWS

Question No: 3

The two objectives of a _____ are to identify problem areas within the work product and a checklist of action items.

▶ **Review Issue List**

- ▶ Formal Technical Reviews
- ▶ Informal Technical Reviews
- ▶ Testing

Question No: 4

It is noted that a/an _____ review can be worse than no review at all

AL-JUNAID TECH INSTITUTE

▶ Uncontrolled

- ▶ Controlled
- ▶ Intermediate
- ▶ Normal

Question No: 5

The Review participants SHOULD NOT be focused on _____.

▶ Weaknesses of the developer

- ▶ Time Lines
- ▶ Review agenda
- ▶ Work product

Question No: 6

The progress on a project at any given point in time can be calculated by:

▶ Adding all the BCWS till that point for all tasks

- ▶ Multiplying all the BCWS till That point for all task
- ▶ Adding BCWS and EVA for all task still that point
- ▶ Dividing all the BCWS till That point for al tasks

Question No: 7

Milestone represents:

▶ The defined target which you need to achieve

- ▶ A predictable risk that may hit the project
- ▶ An unpredictable risk that may project
- ▶ The Loss that organization may suffer due to risk

AL-JUNAID TECH INSTITUTE

Question No: 8

If an error related to requirements is identified in testing phase, error will be considered an error of _____ phase.

▶ Requirement

- ▶ Design
- ▶ Code
- ▶ Testing

Question No: 9

_____ involves rework, repair, and failure mode analysis of the product.

▶ Internal Failure Cost

- ▶ External Failure Cost
- ▶ Appraisal Cost
- ▶ Prevention Cost

Question No: 10

If:

E = Errors found before shipment

D = Errors found during operation

Then Defect Removal Efficiency (DRE) is equal to:

▶ $DRE = E / (E+D)$

- ▶ $DRE = E * (E+D)$
- ▶ $DRE = E + (E+D)$
- ▶ $DRE = E - (E+D)$

Question No: 11

BCWS is term used in Eamed Value Analysis it stands for _____.

AL-JUNAID TECH INSTITUTE

▶ Budgeted Cost of Work Schedule

- ▶ Budgeted Cost of Whole System
- ▶ Beginning Cost of Whole System
- ▶ None of the given options

Question No: 12

_____ states that 80% of the defects can be traced to 20% of its causes.

▶ Pareto Principle

- ▶ Fred Brooks Principle
- ▶ Kraul Principle
- ▶ D.Streeter Principle

Question No: 13

The basis principle for a review is that the focus should be on the _____.

▶ Product and not the producer

- ▶ producer and not the Product
- ▶ Profit to be gained
- ▶ Presentation Slide

Question No: 14

Many defect identified in few components at design level testing/review, a software development team should_____.

▶ Allocate good and extra resources at coding level to review these components thoroughly

- ▶ Good achievement by designing team
- ▶ No need to allocated further resources at coding level to review these components

AL-JUNAID TECH INSTITUTE

- ▶ Mission accomplished once good number of error/flaws identified level

Question No: 15

Milestone represents:

- ▶ **The defined target which you need to achieve**
- ▶ A predicable risk that may hit the project
- ▶ A unpredictable risk that may hit the project
- ▶ The Loss that organization may suffer due to risk

Question No: 16

_____ a statistical technique is used to develop an overall indication of improvement in software quality.

- ▶ **Error index**
- ▶ Pareto Principle
- ▶ Quality Assurance
- ▶ Cost of Quality

Question No: 17

A model where Some errors are passed through to the next stem undebated such errors are said to have been_____.

- ▶ **Amplified**
- ▶ Reduced
- ▶ Neutralized
- ▶ Multiplexed

Question No: 18

Earned Value Analysis (EVA) is a _____ technique for assessing the progress of a project.

AL-JUNAID TECH INSTITUTE

▶ Quantitative

- ▶ Subjective
- ▶ Qualitative
- ▶ Descriptive

Question No: 19

It is noted that a/an _____ review can be worse than no review at all.

▶ Uncontrolled

- ▶ Controlled
- ▶ Intermediate
- ▶ Normal

Question No: 20

FTR is a quality control activity performed by software engineers.it stands for:

▶ Formal Technical Review

- ▶ Formatted Technical Review
- ▶ Feasibility of Technical Requirement
- ▶ Fetched Technical Requirement

Question No: 21

Earned Value analysis (EVA) is a _____ technique for assessing the progress of a project

▶ quantitate

- ▶ Subjective
- ▶ Descriptive
- ▶ Qualitatives

AL-JUNAID TECH INSTITUTE

Question No: 22

BCWS for a task i will be equal to:

- Risks appeared in task i
- **Effort (person-days etc) for task i**
- Resource deficiency for task i
- None of the given options

Question No: 23

SQA is an umbrella activity in which following activities are performed EXCEPT

- Review
- Testing
- Project Planning
- **Inspection**

Question No: 24

Schedule Performance Index (SPI) is equal to

- **BCWP / BCWS**
- BCWP * BCWS
- BCWP + BCWS
- BCWP – BCWS

Question No 25

Poka Yoke is the name of a

- **QA technique**
- Japanese dish
- Idiom
- None of the given options

Question No 26

_____ is one of the techniques to prepare project schedule once we have defined "task network"

- **Program evaluation and review techniques (PERT)**
- Critical Communication path method (CCPM)
- System resource negotiation method (SRNM)
- Strategic Schedule Development technique (SSDT)

AL-JUNAID TECH INSTITUTE

Question No 27

Quality cost may be divided into costs associated with:

- **Prevention, appraisal and failure**
- Customers, developers and maintenance
- Builds, releases and products
- Bugs, errors and Reports

Question No 28

The equation to find out the Availability of software is:

- $\text{Availability} = (\text{MTTF} \times \text{MTBF}) / 100$
- $\text{Availability} = (\text{MTTF} / \text{MTBF}) \times 100$
- $\text{Availability} = (\text{MTTF} + \text{MTBF}) \times 100$
- $\text{Availability} = (\text{MTTF} - \text{MTBF}) \times 100$

Question No 29

FTR stands for:

- Feasibility of technical requirement
- Fetched Technical Requirement
- Formal Technical Review
- None of the given options

Question No: 30

Earned Value Analysis (EVA) is a _____ technique for assessing the progress of a project.

- Qualitative
- Quantitative
- Subjective
- None of the given options

Question No: 31

FTR stands for:

- Feasibility of technical requirement
- Fetched Technical Requirement
- **Formal Technical Review**
- None

AL-JUNAID TECH INSTITUTE

Question No: 32

Configuration Auditing deals with:

- Ensuring that Resources have been allocated properly.
- Ensuring that changes are implemented parallel with SDLC framework activities.
- Ensuring that the changes have been implemented properly
- All of the given

Question No: 33

Poka-Yoke is the name of a:

- QA technique
- Japanese dish
- Idiom
- None

Question No: 34

Quality cost may be divided into costs associated with:

Question No: 35

If an error related to requirements is identified in the testing phase, this error will be considered as an error of _____

- Requirement
- Design
- Code
- Testing

Question No: 36

AL-JUNAID TECH INSTITUTE

In order to use the PERT and CPM, which one of the followings is NOT required?

- Decomposition of tasks-also known as the work breakdown structure (WBS).
- Teams communication mode details.
- Estimation of effort.
- Interdependencies.

Question No: 36

if:

E = Errors found before shipment

D = Errors found during shipment

Then Defect Removal Efficiency (DRE) is equal to:

- $DRE = E + (E+D)$
- $DRE = E - (E+D)$
- $DRE = E * (E+D)$
- $DRE = E / (E+D)$

Question No: 37

Following are effective guidelines for Review EXCEPT:

- We need to review the product, not the producer
- Be sensitive to personal egos
- Errors should be pointed out gently
- The tone should be high and strict

Question No: 38

The basic idea behind Software configuration management is:

- To manage and control the change
- To efficiently allocate the resources
- To properly manage status reporting

AL-JUNAID TECH INSTITUTE

- None

Question No: 39

Earned Value Analysis (EVA) is a ____ technique for assessing the progress of a project.

- Qualitative
- Quantitative
- Subjective
- None

Question No: 40

Quality cost may be delivered into costs associated with:

- Prevention, appraisal and failure
- Customers, developers and maintenance
- Builds, releases and products
- Bugs, errors and reports

Question No: 41

A Software Requirement Specification is traced if the ____ of its requirements is clear.

- Quality
- Slandered
- Origin ✓
- None

Question No: 42

Version Control discusses about:

- How does an organization control changes before and after software is released to a customer
- How the resources will be assigned to tasks in work breakdown structure (WBS)
- How the test cases will be executed before deploying each release.
- All of given

Question No: 43

The implemented requirement has been ____ through the selected approach, such as testing or inspection.

- Verified
- Deleted
- Approved

AL-JUNAID TECH INSTITUTE

- None

Question No: 44

The Software reengineering is a ____ activity.

- Code
- Out dated
- Document
- **Non-trivial**

Question No: 45

Which of the following is NOT an example of Software Configuration Item (SCI):

- Test Case
- **Integration Plan**
- Design specification
- None

Question No: 46

Mean Time Between Failure (MTBF) can be calculated by:

- **$MTBF = MTTF + MTTR$**
- $MTBF = MTTF - MTTR$
- $MTBF = (MTTF/MTTR)*100$
- $MTBF = (MTTR/MTTF)*100$

Question No: 47

____ software development is itself risky as changes to one part of the system inevitably involve further changes to other components.

- Forward
- Reverse
- Old
- **New**

Question No: 48

Program is restructured ____ the reverse engineering phase. In this case we modify source code and data in order to make it amenable to future changes.

- **After**
- None
- Along with
- Before

Question No: 49

AL-JUNAID TECH INSTITUTE

___Managers look at the system from the angle that does the system and associated business process make an effective contribution to the business goal?

- Senior
- IT
- None
- Line

Question No: 50

___user assess the system from the perspective of how effective do they find the system in supporting their business processes and how much of the system functionality is used.

- Senior
- IT
- Line
- End ✓

Question No: 51

A system is considered to be a legacy system if it has been in operation for ___years. A legacy system has many components.

- Less than seven
- Many ✓
- More than two
- Less than ten

Question No: 52

Legacy system migration, however, is an easy task and there are a number of risks involved that need to be mitigated

- Simple
- Crucial
- None
- Easy

Question No: 53

___engineering does not simply create a modern equivalent of an older program, rather new user and technology requirements are integrated into the reengineering effort

- Forward
- None
- Reverse
- BPR

AL-JUNAID TECH INSTITUTE

Question No: 54

Which of the following the NOT one of the attributes of requirements.

- Requirement ID
- Creation Date
- Created By
- **Received By**

Question No: 55

If the version of a release changes from 1.0.0 to 2.0.0 what will you conclude about new release (having version 2.0.0)

- There was no change in functionality
- **There was major change in functionality**
- There was partial change in functionality
- Bugs in previous functionality were removed

Question No: 56

Which of the following is NOT one of the tasks included in Software Configuration Management.

- Version Control
- Configuration Auditing
- **Test Case development**
- Change control

Question No: 57

_____ are sometimes also knowns as walkthroughs or inspects

- **Formal Technical review**
- Testing
- Debugging
- Informal Technical review

Question No: 58

In order to use the PERT and CPM, which one of the following is not required?

- Inerdependencies
- Estimation of effort
- **Teams communication mode details**
- Decomposition of tasks-also known as the work breakdown structure(WBS)

AL-JUNAID TECH INSTITUTE

Question No: 59

A requirement related detected in any (design, source code testing) will be called _____

- Requirements defect
- Design related defects
- **Source code defect**
- Zero level defect

Question No: 60

Cost of quality increase as we go from _____

- Detection to prevention to Failure
- **Prevention to Detection to Failure**
- Failure to Prevention to Detection
- Detection to Failure to Preservation

Question No: 61

Then Defect Removal Efficiency (DRE) is equal to:

ANS: - $DRE = E / (E+D)$

Question No: 62

_____ a statistical technique is used to develop an overall indication of improvement in software quality.

ANS: - Error Index

Question No: 63

The two objectives of a _____ are to identify problem areas within the work product and a checklist of action items.

ANS: - Formal Technical Reviews

Question No: 64

_____ states that 80% of the defects can be traced to 20% of its causes.

ANS: - Pareto Principle

Question No: 65

Work products used to track errors is/are _____ during software development.

ANS: - Requirements specification, design document and source code

AL-JUNAID TECH INSTITUTE

Question No: 66

BCWP refers to:

ANS: - Budgeted cost of work performed so far

Question No: 67

Milestone represents:

ANS: - The defined target which you need to achieve

Question No: 68

Earned Value Analysis (EVA) is a _____ technique for assessing the progress of a project.

ANS: - Quantitative

Question No: 69

FTR is a quality control activity performed by software engineers, it stands for:

ANS: - Formal Technical Review

Question No: 70

BCWS is a term used in Earned Value Analysis it stands for _____.

ANS: - Budgeted Cost of Work Schedule

Question No: 71

The degree of rigor in determining the set of tasks can be categorized as causal, Quick Reaction, strict and _____

- Uniform
- Unstructured
- Decomposed
- Structured

Question No: 72

if:

E = Errors found before shipment

D = Errors found during operation

Then Defect Removal Efficiency (DRE) is equal to:

- $DRE = E / (E + D)$
- $DRE = E + (E + D)$
- $DRE = E * (E + D)$
- $DRE = E - (E + D)$

Question No: 1

AL-JUNAID TECH INSTITUTE

A law affirming that to continue after a certain level of performance has been reached will result in a decline in effectiveness. This law is know as _____

- **Law of Diminishing returns**
- Law of effectiveness
- Law of Saturation
- Law of Marketing returns

Question No: 2

_____ is not the construction part of software development activities.

Requirement gathering

- Testing
- **Configuration Management (Page 8)**
- Coding

Question No: 3

Basic project management process are established at _____ level of CMM

- Chaotic
- Initial
- Level 1
- **Repeatable (Page 12)**

Question No: 4

The projects usually don't fail due to

- ambiguous/incomplete requirements
- changing customer requirements
- **the fact that company was not CMM certified (Page 28 - 29)**
- failure in project management

Question No: 5 _____ is a team organization where there is no permanent leader and task coordinators are appointed for short duration. Decisions on problems and approach are made by group consensus and communication among team is horizontal.

- **Democratic decentralized (DD) (Page 32)**
- Controlled decentralized (CD)
- Synchronous paradigm (SP)
- Controlled centralized (CC)

AL-JUNAID TECH INSTITUTE

Question No: 6

Which of these is not a valid reason for measuring software processes, products, and resources?

- **to price them**
- to improve them
- to characterize them
- to evaluate them

Question No: 7 Integrity can be measured by the following formula

- integrity= (1-threat) x (1-security)
- integrity= $\sum (1+threat) + (1-security)$
- **integrity= $\sum [(1-threat) \times (1-security)]$ (Page 69)**
- integrity= $\sum (1-threat) - (1-security)$

Question No: 8

The rapid application development model is

- Another name for component-based development.
- A useful approach when a customer cannot define requirements clearly
- **A high speed adaptation of the linear sequential model. (Page 19)**
- All of the given

Question No: 9 An _____ is a user identifiable group of logically related data or control information maintained within the boundary of the application

- **Internal logical file (ILF) (Page 42)**
- External Interface file (EIF)
- External input
- External Query

Question No: 10

When more than one user interpret the same requirement in different ways then we can say that the requirements are

- None of the given
- Incomplete
- **Ambiguous**
- Incorrect

Question No: 11

Testing and Software Quality Assurance activities are exactly similar activities with different names

AL-JUNAID TECH INSTITUTE

- True
- **False (Page 69)**

Question No: 12

If an error related to requirements is identified in testing phase, so this error will be considered as an error in _____ phase.

- Design
- Code
- Requirement
- **Testing**

Question No: 13

Usually the performance of the organization change over-night.

- True
- **False**

Question No: 14

Following are the implicit requirement(s) of software?

- Efficiency
- Ease of use
- Maintainability
- **All of the given (Page 106)**

Question No: 15

If an error related to requirements is identified in Coding, so this error will be considered as an error in _____ phase.

- Design
- **Code**
- Requirement
- Testing

Question No: 16

While developing the software if the bug is caught then it is termed as _____

- Error
- **Defect**
- Fault
- Mistake

AL-JUNAID TECH INSTITUTE

Question No: 17

Following are the types of Formal Technical Reviews except

- Inspection
- **JAD (Page 110)**
- Walkthrough

Question No: 18

The Review participants should not focus on

- Work product
- **Weaknesses of the developer (Page 113)**
- Time lines
- Review agenda

Question No: 19

Review meeting is attended by the following except

- Review leader
- All reviewers
- **CEO (Page 112)**
- Product Producer

Question No: 20

Which statement is true in the light of Pareto principle?

- **80% of defects are caused by 20 % of mistakes (Page 114)**
- 20% of defects are caused by 80 % of mistakes

Question No: 21

Reliability and Safety are same concepts with respect to software quality

- **True (Page 117)**
- False

Question No: 22

The prevention of defects being injected into the software can help in decreasing the cost of software.

- **True**
- False

Question No: 23

AL-JUNAID TECH INSTITUTE

Poka Yoke is the name of a

- Japanese dish
- **QA technique (Page 118)**
- Idiom
- None of the given

Question No: 24

If a new version of a product is released by fixing the bugs in the previous release then it is termed as

—

- **Product Update (Page 122)**
- Product Upgrade
- Defect Removal
- Product Performance

Question No: 25

The Evolution Graph signifies the

- **version relationship (SE-PRESSMAN, page 260)**
- baseline of a particular release
- change control activity
- all of the given

Question No: 26

In the software development life cycle, soon the defect is detected will results in

- increase in cost of software
- **decrease in software cost**
- both a and b are correct

Question No: 27

_____ testing verifies the correct implementation of internal units, structures and relationship among them.

- Black box
- **White box**
- Gray box

Question No: 28

The template for organizing SRS given by American Department of Defense and NASA should be used for

AL-JUNAID TECH INSTITUTE

- none of the given
- **large and complex projects**
- small project
- medium size projects

Question No: 29

If you know the relationship between the use cases, classes, GUIs, Test cases and other artifacts then we can say that requirements are traceable in that project.

- True
- **False (Page 133)**

Question No: 30

There may be one or more GUIs that are part of any good software but may not satisfy any requirement

- **True**
- Fals

Question No: 31

Following are the components of a Legacy system except:

- business processes
- system hardware
- **marketing (Page 134)**
- application software

Question No: 32

There are many reasons for which the legacy system become difficult to maintain EXCEPT

- Language in which system was developed become obsolete
- lack of consistency as different parts of the system have been developed by different teams
- None of these
- **No documentation is available (Page 134)**

Question No: 33 In _____, new user and technology requirement can also be integrated into the re-engineering effort.

- Backward re-engineering
- **Forward re-engineering (Page 140)**
- Business Engineering
- Business Process Engineering

Question No: 34

AL-JUNAID TECH INSTITUTE

Large classes reduces

- **Cohesion (Page 143)**
- Coupling
- Consistency
- Chains

Question No: 35

Level _____ is the lowest capability level of CMMI in Continuous representation

- 1
- **(Page 169)**
- 5
- 6

Question No: 36

In _____ representation of CMMI model, consideration is given to improve selected processes areas.

- **Staged (Page 169)**
- Continuous

Question No: 37

Empirical models are statistical models and are based upon historic data

- **True (Page 81)**
- False

Question No: 38

Error tracking provides a quantitative means of assessing the quality of the individuals implementing a software product.

- True
- **False**

Question No: 39

There are tools available in the market for project tracking. These tools can automatically track and manage the project replacing the need of any human being as a Project Manager.

- **True**
- False

Question No: 40

AL-JUNAID TECH INSTITUTE

Two tools for computing critical path and project completion times from activity networks are

- **CPM and PERT (Page 100)**
- DRE and SQA
- FP and LOC
- ASD and BS

Question No: 41

In order to measure the design quality, if the frequency of ripple defects is too large, then it means that there is tight coupling and hence the

- design is maintainable
- **design is not maintainable (Page 79)**
- design has completed
- none of the given

Question No:42 _____, there is both vertical and horizontal communication.

- **Controlled decentralized (CD) Page 32**
- Democratic decentralized (DD)
- Controlled centralized (CC)
- Synchronous paradigm (SP)

Question No: 43

The only reason for an estimate to be unreliable is lack of experience related to the application on the part of the estimator.

- true
- **false**

Question No: 44

A consideration of software scope must include an evaluation of all external interfaces.

- **true (Page 80)**
- false

Question No:45

Software quality is conformance to _____

- **Explicitly stated functional and non functional requirements (Page 106)**
- Explicitly documented development standrads

AL-JUNAID TECH INSTITUTE

- Implicit characteristics that all the professional software are expected to have
- All of the given options

Question No: 46

_____ Subgroups are those that the user has the option of using one or none of the subgroups during an elementary process.

- **Optional (Page 46)**
- Mandatory
- None of the Given
- RET

Question No: 47

Process discipline is unlikely to be rigorous, but where it exists it may help to ensure that existing processes are maintained during times of stress and this is done at CMM level ----- --.

- 1
- 4
- 3
- **2 (Page 172)**

Question No: 48

Which one of the following activities does not belong to Risk Mitigation, Monitoring, and Management Plan?

- **Risk Projection (Page 89)**
- Risk Mitigation
- Risk avoidance
- Risk Management and Contingency Planning

Question No: 49

Which one of the following principles/techniques is NOT software projects scheduling?

Interdependency identification

- Time allocation
- Effort validation
- **Function point analysis (Page 93)**
- Milestone definition

Question No: 50

AL-JUNAID TECH INSTITUTE

With a complex class you have to move data and methods around in small pieces to avoid errors, it seems slow but it is the _____ because you avoid debugging

- **Quickest (Page 163)**
- Problem
- None of the given
- Slowest

Question No: 51

The purpose of earned value analysis is to

- determine how to compensate developers based on their productivity
- **provide a quantitative means of assessing software project progress (Page 102)**
- provide a qualitative means of assessing software project progress
- set the price point for a software product based on development effort

Question No: 52

The RMMM plan assists the project team in developing strategy for dealing with risk. In this context, an effective strategy must consider:

- Risk avoidance
- Risk monitoring
- Risk management and contingency plan
- **All of the given choices (Page 89)**

Question No: 53

The component based development model is

- only appropriate for computer hardware design.
- not able to support the development of reusable components.
- **dependent on object technologies for support. (SE-PRESSMAN, 70)**
- not cost effective by known quantifiable software metrics.

Question No: 54

Which is used to determine the most viable option for cost estimation when the information in the "Decision tree" is complete.

- $E = 3.2 (KLOC)^{1.05}$
- **Expected cost = (path probability), x (estimated path cost) (Page 83)**
- Expected cost = (path probability), x (estimated path cost)

AL-JUNAID TECH INSTITUTE

Question No: 55

After building the Decision Tree, following formula is used to find the expected cost for an option.
Choose the correct formula:

- **Expected Cost= (path probability)ⁱ * (estimated path cost) (Page 83)**
- Expected Cost= (path probability)ⁱ / (estimated path cost)
- Expected Cost= (path probability)ⁱ + (estimated path cost)
- Expected Cost= (path probability)ⁱ - (estimated path cost)

Question No: 56

Degree of uncertainty that the product will meet its requirements and be fit for its intended use is the

- Cost risks
- Schedule risks
- **Performance risks (Page 87)**
- None of the given choices

Question No: 57

Every task or group of tasks should be associated with a project -----.

- Schedule
- Member
- Manager
- **Milestone (Page 93)**

Question No: 58

Every task should be assigned to a specific team-----.

- **Member (Page 93)**
- Manager
- Organizer
- None of the given

Question No: 59

Which one is not the Software project planning activity carried out by the project manager for estimation?

- Software scope estimation
- Resources requirements
- Time requirements
- **Product Quality (Page 80)**

AL-JUNAID TECH INSTITUTE

Question No: 60

Configuration Item identification involves:

- Identifying the structure of the s/w system
- **Uniquely identify individual components**
- Uniquely identifying various revisions
- All of the given

Question No: 61

Incomplete Configuration identification documents may result in:

- Schedule Product
- **all of the given choices are correct** [Click here for detail](#)
- Defective Product
- Higher Maintenance Costs

Question No: 62

Degree of uncertainty that the product will meet its requirements and be fit for its intended use is the

- Cost risks
- Schedule risks
- **Performance risks (Page 87)**
- None of the given choices

Question No: 63

Proactive risk management philosophy is also some times termed as Indiana Jones school of risk management

- true
- **false (Page 84)**

Question No: 64

Software Availability can be calculated by the following equation

- Availability = $(MTTR/MTTF) \times 100$
- Availability = $(MTBF/MTTR) \times 100$
- **Availability = $(MTTF/MTBF) \times 100$ (Page 116)**
- Availability = $(MTBF/MTTF) \times 100$

Question No: 65

AL-JUNAID TECH INSTITUTE

Phase Index can be calculated by the help of the following formula, where

E_i – the total number of errors uncovered during the i th step in the SE process

S_i – number of serious errors

M_i – number of moderate errors

T_i – number of minor errors

P_{Si} – product size at the i th step

w_s, w_m, w_t – weighting factors for serious, moderate, and minor errors

- **$P_{li} = w_s(S_i/E_i) + w_m(M_i/E_i) + w_t(T_i/E_i)$ (Page 115)**
- $P_{li} = w_s(S_i/E_i) + w_m(M_i/M_i) + w_t(T_i/T)$
- $P_{li} = w_s(S_i/E_i) + w_m(M_i/M_i) + w_t(T_i/E_i)$
- $P_{li} = w_s(S_i/S) + w_m(M_i/M) + w_t(T_i/E)$

Question No: 66

The higher the Error Index, the higher will be the Defect Removal Efficiency

- **True (Page 115)**
- False

Question No: 67

The goal of quality assurance is to provide management with the data needed to determine which software engineers are producing the most defects.

- true
- **False**

Question No: 68

From the following listed software development Model, which one is an object oriented model

- Classical life cycle model
- **Fountain model (Page 23)**
- Spiral model
- Waterfall model

Question No: 69

The prototyping model of software development is

- a reasonable approach when requirements are well defined.

AL-JUNAID TECH INSTITUTE

- **A useful approach when a customer cannot define requirements clearly.**
- the best approach to use for projects with large development teams.
- a risky model that rarely produces a meaningful product.

Question No: 70

Which one of the following does not belong to a strategy for dealing with risk?

- Risk avoidance
- **Security risk assessment (Page 89)**
- Risk monitoring
- Risk management and Contingency planning

Question No:71

A _____ entity is the one which have any -----in the problem domain without some other entity.

- Strong, Role*
- All of the given
- Weak, Function

None of the Given (Page 47)

A weak entity is the one which does not have any role in the problem domain without some other entity

Question No: 72

ILF is a ----- identifiable group of logically control informations -----the boundary of the application.

- **user, within (Page 42)**
- user, without
- All of the given
- user, along

Question No: 73

Several entities are always created ----- and deleted together then this is a strong indication that they should be grouped into ----- logical file/files.

Together, Single (Page 48)

- Together, Multiple
- Together, Double
- All of the given

Question No: 74

AL-JUNAID TECH INSTITUTE

Which of these software characteristics are used to determine the scope of a software project?

- context, lines of code, function
- context, function, communication requirements
- **information objectives, function, performance (Page 34)**
- communications requirements, performance, information objectives

Question No: 75

In _____ a team is structured along a traditional hierarchy of authority

- **closed paradigm (Page 32)**
- synchronous paradigm
- random paradigm
- open paradigm

Question No: 76

Integrity means that the software should

- help the users to enjoy usability
- none of the given
- **Withstand the attack from a hacker (Page 69)**
- help the hacker to hack the system

Question No: 77

Ambiguous requirements can be measured quantitatively

- False
- **True (Page 71)**

Question No: 78

The Spiral model of software development is of Iterative nature

- **True**
- false

Question No: 79

While performing risk analysis, the impact of risk **can not** be measured quantitatively

- True
- **False**

Question No: 80

AL-JUNAID TECH INSTITUTE

_____ is the measure of how many defects are removed by the quality assurance processes before the product is shipped for operatio

- **Defect removal efficiency (Page 69)**
- defect density
- None of These

Question No: 81

For software the total numbers of requirements are equal to _____.

- Functional requirements
- non-functional requirements
- **Functional requirements and non-functional requirements (Page 71)**
- None of these

Question No: 82

The _____ model is used to overcome issues related to understanding and capturing of user requirements.

Rapid Prototyping (Page 17)

- Water fall
- Build and Fix
- None of the above

Question No: 83

_____ is not the part of software development loop.

- Status Quo
- Problem definition
- Technical development
- **Task set (Page 10)**

Question No: 84

The level 1 of CMM is known as

- Managed
- Defined
- **Initial (Page 12)**
- Repeatable

Question No: 85

AL-JUNAID TECH INSTITUTE

According to DeMarco a good Project Manger

- should be very nervous
- should be very strict in official matters
- should not ignore any mistake from his team members.
- **should have a big heart and should ignore small mistakes (Page 31)**

Question No: 86

The best project team organizational model to use when tackling extremely difficult problems is the

- chief programmer team model
- **democratic decentralized model (Page 32)**
- controlled decentralized model
- controlled centralized model

Question No: 87

Which factor is the least important when choosing the organizational structure for a software team?

- degree of communication desired
- predicted size of the resulting program
- rigidity of the delivery date
- **size of the project budget (Page 32)**

Question No: 88

Product and process decomposition occurs simultaneously as the project plan evolves

- **True**
- False

Question No: 89

Measuring the quality and performance of a software through any mean is much better than just making a guess about it.

- **True**
- false

Question No: 90

_____ is data that influences an elementary process of the application being counted.

- Elementary Process
- External Query
- External Output

AL-JUNAID TECH INSTITUTE

➤ **Control Information (Page 43)**

Question No: 91

An entity which defines many-to-many relationship between two or more entities is called

Associative Entity Type (Page 47)

- **Associative Entity Type**
- Entity Subtype
- None of these

Question No: 92

Software risk impact assessment should focus on consequences affecting

- **performance, support, cost, schedule**
- planning, resources, cost, schedule
- marketability, cost, personnel
- business, technology, process

Question No: 93

The number of people required for a software project is determined

- **after an estimate of the development effort is made.**
- from an assessment of the technical complexity of the system.
- by the size of the project
- all of the given

Question No: 94

Defect Removal Efficiency (DRE) can be measured by where E is **Errors found delivery** and D is **error found after delivery** (typically within the first year of operation)

- **$DRE = E / (E + D)$ (Page 69)**
- $DRE = E - (E + D)$
- $DRE = E * (E + D)$
- None of the given

Question No: 95

In function point analysis technique EO stands for

- Export operation
- Export output
- **External output (Page 49)**

AL-JUNAID TECH INSTITUTE

- None of these

Question No: 96

In _____ a team is structured loosely and depends on individual initiative of the team members

- **Random paradigm (Page 32)**
- closed paradigm
- synchronous paradigm
- open paradigm

Question No:97

_____ is the first stage of waterfall lifecycle model

- **Requirement definition (Page 15)**
- Operation
- Unit testing
- Implementation

Question No: 98

_____ is not the management part of software development activities.

- **Coding pg#8**
- Configuration Management
- Quality Assurance
- Project planning **Question No: 99**

Software process and product quality are controlled at _____.

- level 1
- initial level
- **level-4 (Page 12)**
- level-3

Question No: 100

QA is an activity performed _____ of the software development.

- at the initial stage
- at the final stage
- **throughout (Page 25)**
- at the middle

Question No: 101

AL-JUNAID TECH INSTITUTE

_____ the ability to encourage people to create and feel creative.

- Organization
- Motivation
- **Innovation (Page 30)**
- Managerial Identity

Question No: 102

Caper Jones divided software related activities into _____ different categories.

- 35
- 40
- 20
- **25 (Page 7)**

Question No: 103

By default every organization is working at _____

- **level 1 (Page 12)**
- level 0
- level 3
- level 4

Question No: 104

Software Engineering is the set of _____ and _____ to develop software.

- languages , processes
- classes ,objects
- **processes, tools (Page 4)**
- tools, objects

Question No:105

Software has very close relationship with _____

- Biology
- Physics
- chemistry
- **Economics (Page 5)**

Question No: 106

AL-JUNAID TECH INSTITUTE

COCOMO is the classic LOC cost-estimation formula and used thousand delivered source instructions (KDSI) as his _____ of size.

- LOC
- milestone
- **unit**
- none of given

Question No: 107

_____ deadline is one of the reasons of project failure.

- achievable
- feasible
- **unrealistic (Page 29)**
- realistic

Question No: 108

_____ model is very sensitive to the risk.

- **Spiral (Page 21)**
- waterfall
- incremental
- RAD

Question No: 109

_____ is the ability to encourage technical people to produce to their best.

- improvement
- **motivation (Page 30)**
- innovation
- creation

Question No: 110

Caper Jones is famous researcher in the field of _____ who made a company named Software Productivity Research

- Biology
- Chemistry
- Mathematics
- **Software Engineering (Page 7)**

AL-JUNAID TECH INSTITUTE

Question No: 111

In _____ model user feedback is received very quickly because product is delivered in small versions.

- waterfall
- spiral
- **incremental (Page 18)**
- object oriented

Question No: 112

_____ approach is better for the team spirits point of view.

- Controlled Decentralized
- Informal, interpersonal procedures
- Controlled Centralized
- **Democratic Decentralized (Page 32)**

Question No: 113

Construction activities are directly related to the _____ of the software.

- Installation
- risk analysis
- **development (Page 8)**
- debugging

Question No: 114

Continuous process improvement is enabled by _____ feedback from the _____

- analytical , user
- mathematical, user
- logical, process
- **qualitative , process (Page 12)**

Question No: 115

The major activities related to software construction are

- **requirement gathering, design development, coding and testing (Page 8)**
- installation and training

AL-JUNAID TECH INSTITUTE

- quality assurance, configuration and planning
- implementation and management

Question No: 116

If requirements are uncertain, _____ will be a suitable.

- Incremental
- RAD
- **Waterfall (Page 34)**
- Prototype

Question No: 117

Capability maturity model (CMM) is used to judge the _____ model of an organization.

- Efficiency
- Performance
- Productivity
- **Maturity (Page 12)**

Question No: 118

After passing through all the stages of the software development when we deploy the new system at the user side that stage is called _____

Integration

Development

Installation

Status quo (Page 11)

Question No: 119

Defects per function points are a _____ metric.

- **One (Page 65)**
- Two
- Three
- Four

Question No: 120

In Capability Maturity Model (CMM), _____ performs optimization.

- level1

AL-JUNAID TECH INSTITUTE

- level 2
- level3
- **level5 (Page 12)**

Question No: 121

Defining a |Task Network| helps in defining

- The resources required for project
- Third party tool required for project
- **Sequence in which activities will be performed (Page 99)**
- None of these

Question No: 122

The first step in project planning is to

- Determine the budget.
- Select a team organizational model.
- Determine the project constraints.
- **Establish the objectives and scope. (Page 80)**

Question No: 123

Function Point analysis is helpful in calculating the size of the software for_____

- Both client and software organization
- Software organization
- Client
- **User (Page 38)**

Question No: 124

The extent to which a program satisfies its specification and fulfills the customer's mission objectives is said to be achieving the

- Usability
- Efficiency
- Reliability
- **Correctness**

Question No: 125

What activity/activities , we can perform to identify the needs and benefits of a task?

- Establish project statement
- **Identify needs and project constraints**

AL-JUNAID TECH INSTITUTE

- All of these

Question No:126

BCWS stands for:

- Budgeted cost of whole System
- **Budgeted cost of work schedule (Page 102)**
- None of the given options

Question No: 127

BCWP refers to:

- Normalized value of budgeted cost
- Overall expected cost of the project (anticipated cost)
- **Budgeted cost of work performed so far (Page 102)**
- Overall cost of the project (calculated after completion of project)

Question No: 128

Which statement is correct?

- **The greater the dependency between the components the greater is coupling**
- The lesser the dependency between the components the greater is coupling
- The greater the dependency between the components the lesser is coupling
- None of the given

Question No: 129

In context of moving range and individual control charts, UNPL stands for:

Universal Natural Process Line

Universal Natural Process Limit

Upper Natural Process Limit (Page 77)

Upper Natural Process Line

Question No: 130

Effort required to test a program to ensure that it performs its intended function _____

- **Testability page (Page 68)**
 - Bug fixing
 - Debugging
 - Security

Question No: 131

AL-JUNAID TECH INSTITUTE

The extent to which a program can be expected to perform its intended function with required precision is called _____

Usability

Reliability (Page 67)

Portability

Maintainability

Question No: 132

Review is a type of _____ that helps preventing the bugs to move in the next stage of software development Bug seeding tool

➤ **Filter (Page 109)**

➤ Alarm

Question No: 133

_____ help in finding the matrix to be stable or unstable

Control chart (Page 75)

Directed Graph

Cyclic chart

Base line graph

Question No: 134

Defect Removal Efficiency can be increased by

- By increasing the team member in a team
- **Identifying the defect in early stages of development so that it may not be amplified**
- Identifying the defect in the later stages of project
- By performing the testing activity only

Question No: 135

_____ give you a better insight into the state of the process or product

Metrics (Page 66)

Efficiency

Reliability

Usability

Question No: 136

Determination of the is a pre-requisite of all sorts of estimates, including, resources, time, and budget.

AL-JUNAID TECH INSTITUTE

software scope (Page 80)

software Risk
software Quality
software Management

Question No: 137

Following are effective guidelines for Review **except**

- We need to review the product not the producer
- Be sensitive to personal egos
- Errors should be pointed out gently
- **Tone should be high and strict (Page 113)**

Question No: 138

Function/Test matrix is a type of
Interim Test report
Final test report

Project status report

Management report

Question No: 139

Poka Yoke is Japanese term and it means

- **Mistake Proofing (Page 118)**
- Mistake Handling
- Mistake identification
- Mistake assurance

Question No: 140

we need to employ some statistical techniques and plot the result ----- . This is known as statistical control techniques.

Graphically (Page 74)

automatically
manually
personally

Question No: 141

The extent to which a program satisfies its specifications and fulfills the customer's mission objectives is

—

AL-JUNAID TECH INSTITUTE

Integrity

Reliability

Correctness (Page 67)

none of given

Question No: 142

Which is not correct in the context of Poka-Yoke technique?

- Simple and cheap
- **Sophisticated and expensive (Page 118)**
- Part of Process
- Indication point should be near to the place where the problem occurred

Question No: 143

Metrics to assess the quality of the analysis models and the corresponding software specification were proposed _____

in 1993- 1999.

Ricardo in 1993

Davis in 1990

Davis in 1993 (Page 71)

Question No: 144

If Configuration item identification is not identified, it is possible to control changes and establish records.

- **False (Page 121)**
- True

Question No: 145

Incomplete Configuration identification documents may result in:

Defective Product

Higher Maintenance Costs

Schedule Product

Meet Software Quality

A,B,C

B,C,D

A,B,D

AL-JUNAID TECH INSTITUTE

A,B,C,D

Question No: 146

When an Item is baselined, it becomes frozen, here frozen means, that the item can be changed only by creating an old version

- **True (Page 72)**
- False

Question No: 147

The most important objective of any engineering activity is to produce high quality product with limited resources and

Time (Page 66)

- Persons
- Cost
- Metrics

Question No: 148

Extent to which access to software or data by unauthorized persons can be controlled and called

- None of given
- Efficiency
- Reliability

Integrity (Page 67)

Question No: 149

Object un-lock is done by

- **check in (Page 125)**
- check out
- both check-in and check-out
- none of the given

Question No: 150

Requirements engineering is the basis of the contract between the developer and the client.

- True
- **False Written as a contract between client and contractor**

Question No: 151

AL-JUNAID TECH INSTITUTE

The amount of computing resources required by a program to perform its function is _____

Efficiency (Page 67)

Integrity

Reliability

None of given

Question No: 142

Legacy system migration is usually hard. One important reason is that no proper documentation of the system is available

- True
- **False (Page 134)**

Question No: 153

While assessing a legacy system for further decision, one should decide the following about the supplier

- Is supplier still in existence?
- Is supplier still in business?
- Is supplier can provide support?
- **All of the given (Page 137)**

Question No: 154

The application software is assessed on the basis of following factors except

- Documentation
- Understandability
- **Code Refactoring (Page 137)**
- Test data

Question No: 155

We modify the internal _____ in code re-structuring

- interface
- **design (Page 139)**
- data structure
- functionality

Question No: 156

_____ technique was initially developed for manufacturing processes in the 1920's by Walter Shewart.

Upper Control Line

Control chart (Page 75)

AL-JUNAID TECH INSTITUTE

CMP

Quality

Question No: 157

“A method requires lot of information from some other class” is a symptom of _____ bad smell

- Data clumps
- Lazy class
- **Feature envy (Page 143)**
- Long Method

Question No: 158

_____ integrates system and software disciplines into single process improvement framework for introducing new disciplines as needs arise.

- SEI
- **CMMI**
- CMM
- ISO

Question No: 159

FAST is the abbreviation of

- Facilitated Application Specification Technology
- **Facilitated Application Specification Technique (Page 80)**
- Facilitated Application Specialization Technique
- None of the above option

Question No: 160

In measuring Software Process Quality by using control charts, if the gap between the defects reported and defects fixed is increasing, then it means

The product is in unstable condition. (Page 78)

the product is ready for shipment

the product is in stable condition.

None of the above

Question No: 161

Although there are many different models developed by different researchers for estimation, all of them share which one of the following basic structure

AL-JUNAID TECH INSTITUTE

- $E = 3.2 (KLOC)^{1.05}$
- **$E = A + B * (ev)C$ (Page 81)**
- $E = [LOC \times B0.333/P]^3 \times (1/t4)$
- none of the given

Question No: 162

A _____ is a user recognizable subgroup of data elements within an ILF or EIF

- **Record element type (RET) (Page 46)**
- Data Element Type
- External Input
- External Query

Question No: 163

This chart is then used to develop the individual control chart is called statistical control techniques.

Yes (Page 77)

No

Question No: 164

Any delay in _____ path makes the whole project delayed.

- Logical
- Neural
- Physical
- **Critical**

Question No: 165

The responsibilities of a Project Manager does not include

- Make a schedule of project
- Allocate tasks to the resources
- Monitoring the tasks
- **All these options (Page 28)**

Question No: 166

The boundary time defines the following parameter(s) for a project

- the latest time for task initiation before the minimum project completion time is delayed
- The earliest finish
- The latest finish

AL-JUNAID TECH INSTITUTE

- **All of the given options (Page 100)**

Question No: 167

Quality is the compliance of software to implicit and explicit quality factors. Identify the explicit requirement from the following:

- Maintainability
- **Compliance with CMMI (Page 106)**
- Ease of use
- Efficiency

Question No: 168

Following are the benefits of conducting review except

- **Help in finding the size of the project**
- Helps in verifying the quality of product
- Help in identifying where improvement is required.
- Helps in identifying the bugs in the work product

Question No: 169

Mean Time To Repair (MTTR) is the

- Time when system remained unable for usage
- Time taken to fix the error
- **Time taken to repair the software (Page 116)**
- All of the given options are correct.

Question No:170

In order to use the Pert and CPM, which one of the following is not required?

- Decomposition of tasks - also known as the work breakdown structure (WBS)
- **Teams communication mode detail (Page 100)**
- Estimation of effort
- inter dependencies

Question No: 171

SCM is the requirement of the _____ level(s) of CMM

- First Second and later
- Third and later

AL-JUNAID TECH INSTITUTE

- **Only Second (Page 119)**

Question No:172

We can include following items during configuration item identification

- User Manuals and Documentations
- Source Code
- Software Requirement Specifications
- **All of the given choices are correct (Page 120)**

Question No: 173

CPM stands for

- critical project Measure
- **Critical Path Method (page 100)**
- Common Project Measure
- Critical Planning Model

Question No: 174

Software re-factoring is a process in which

- External behavior of the system does not change
- Internal behavior of the system does not change
- Design of the software changes
- **Architecture of the software changes**

Question No: 175

The Change Control Authority does not have the rights to permit to bring the change in the software

- True
- **False (Page 124)**

Question No: 176

Check in and check out is actually one and the same process with two different names

- True
- **False (Page 124)**

Question No: 177

There could be multiple GUIs to satisfy one requirement

- **True (Page 37)**

AL-JUNAID TECH INSTITUTE

- False

Question No: 178

_____ is the process of Design recovery. At this stage the documentation of the overall functionality is created.

- Database Engineering
- **Reverse Engineering (Page 139)**
- Refactoring
- Forward Engineering

Question No: 179

In code restructuring, we do not change the functionality of the code

- **True (Page 139)**
- False

Question No: 180

Earned Value Analysis (EVA) is a _____ technique for assessing the progress of a project.

Select correct option:

- Qualitative
- **Quantitative (page 102)**
- Subjective
- None of the given options

Question No: 181

A _____ process is "a set of logically related tasks performed to achieve a defined business outcome".

- **Business (Page 141)**
- Software
- CMM
- ISO

Question No: 182

Software Refactoring is a process in which

- **External behavior of the system does not change (Page 142)**
- Internal behavior of the system does not change
- Design of the software changes

AL-JUNAID TECH INSTITUTE

- Architecture of the software changes

Question No: 183

The formal methods model of software development makes use of mathematical methods to

- define the specification for computer-based systems
- **all of the given**
- develop defect free computer-based systems
- verify the correctness of computer-based systems

Question No: 184

Which one of the following describes the data and control to be processed, function, performance, constraints, interfaces, and reliability?

- Product Quality
- **Software scope estimation (Page 80)**
- Resources requirements
- Time requirements

Question No: 185

A significant SQA plan, among others, should include:

- **Resources required for project (Page 177)**
- Error tracking procedures
- Phases required in application development process
- None of the given option

Question No: 186

The software plan is not a static document, it is frequently adjusted to make the project appear on track to meet all deadlines and quality targets.

- **True**
- False

Question No: 187

The projects are classified into following categories except

- New application development
- Reengineering projects
- Concept development projects
- **Marketing Development Projects (Page 96)**

AL-JUNAID TECH INSTITUTE

Question No: 188

The processes at ——— is focus on continually improving process performance through both incremental and innovative technological changes/improvements.

- **Level 5**
- Level 1
- Level 4
- Level 3

Question No: 189

Quantitative process-improvement objectives for the organization are established, continually revised to reflect changing business objectives at ———.

- Level 3
- Level 2
- **Level 5**
- Level 1

Question No: 190

Reliability of a software is a

- functional requirement
- **non-functional requirement**
- Design Requirement
- None of the given

Question No: 191

Which of the following is NOT one of the degrees of rigor?

- Casual
- Structured
- Strict
- **TSS (Page 66)**

Question No: 192

Build and Fix model is a _____ type of software development activity.

- Mathematical
- Perfect
- **Haphazard**
- Planned

AL-JUNAID TECH INSTITUTE

Question No: 193

Three categories of risks are

- Business risks, personnel risks, budget risks
- **Project risks, technical risks, business risks (Page 86)**
- Planning risks, technical risks, personnel risks
- Management risks, technical risks, design risks

Question No: 194

MTTC is the abbreviation of

- measured time to change
- mean time to collaborate
- **mean time to change (Page 68)**
- measure time to cope

Question No: 195

While creating matrix for a set of projects, data should be collected that belongs to a similar domain

- **true (Page 72)**
- false

Question No: 196

_____ is one of the techniques to prepare project schedule once we have defined "task network"

- **Program evaluation and review techniques (PERT) (Page 100)**
- Critical Communication path method (CCPM)
- System resource negotiation method (SRNM)
- Strategic Schedule Development technique (SSDT)

Question No: 197

While managing the historical data, we should consider the following **except**

- **Data should be collected only from scientific projects and applications (Page 72)**
- Data should be collected from many projects
- Data should be consistent
- Correct data should be taken

Question No: 198

Quality costs may be divided into costs associated with

AL-JUNAID TECH INSTITUTE

- **prevention, appraisal, and failure**
- people, process, and product
- customers, developers, and maintenance
- all of the given

Question No:199

When a software is delivered to a client and then client reports the bug in it then that bug is termed as

- Error
- **Defect**
- Mistake
- Fault

Question No: 200

FTR stands for:

- Feasibility of technical requirement
- Fetched Technical Requirement
- **Formal Technical Review (Page 109)**
- None of the given options

Question No: 201

SQA is an umbrella activity in which following activities are performed EXCEPT

- Review
- Testing
- Project Planning
- **Inspection (Page 177)**

Question No: 202

Software configuration management is used to

- **Manage change in the software (Page 119)**
- Manage the financials accounts in an organization
- Manage the administration of company
- All of the given

Question No: 203

Which of the following is not TRUE about "Error Tracking" ?

AL-JUNAID TECH INSTITUTE

- Error tracking can be used to estimate the progress of the project.
- We track errors in work products to assess the status of a project.
- **Durint "Error Tracking", we may also need the historical data from similar projects. (Page 104)**
- All of the given options.

Question No: 204

SQA is an activity in which testing is performed only

- True
- **False (Page 177)**

Question No: 205

Schedule Performance Index (SPI) is equal to

- **BCWP / BCWS (Page 102)**
- BCWP * BCWS
- BCWP + BCWS
- BCWP - BCWS

Question No: 206

BCWS for a task i will be equal to:

- Risks appeared in task i
- **Effort (person-days etc) for task I (Page 102)**
- Resource deficiency for task i
- None of the given options

Question No: 207

if: E = Errors found before shipment D = Errors found during operation Then Defect Removal Efficiency (DRE) is equal to:

- $DRE = E + (E+D)$
- $DRE = E - (E+D)$
- $DRE = E * (E+D)$
- **$DRE = E / (E+D)$ (Page 104)**

Question No: 208

Quality assurance helps management in providing the necessary data about:

AL-JUNAID TECH INSTITUTE

- **Product quality (Page 106)**
- Errors/bugs left
- Errors/bugs highlighted
- Defect reports

Question No: 209

The progress on a project at any given point in time can be calculated by:

- **Adding all the BCWS till that point for all tasks (Page 102)**
- Multiplying all the BCWS till that point for all tasks
- Adding BCWS and EVA for all tasks till that point
- None of given options

Question No: 210

The process of rediscovering the software design is called

- **Reverse Engineering (Page 139)**
- Business Engineering
- Business Process Engineering
- Forward Engineering

Question No: 211

OCI stands for:

- **Output, Control and Input**
- Overflow of Certain Inputs
- Overflow of Certain Indexes
- None of given options

Question No: 212

A redesigned business process must be prototyped before it is fully integrated into the business.

- False
- **True (Page 141)**

Question No: 213

Critical path defines:

- The list of the resources that may be needed for future activities.
- **The chain of tasks that determines the duration of the project. (Page 100)**
- The list of the deficient resources, project is suffering from.
- None of the given options.

AL-JUNAID TECH INSTITUTE

Question No: 214

BAC stands for:

- Budgeted Analysis Cost
- Budget And Cost
- Budget at compilation
- **Budget at Completion (Page 102)**

Question No: 215

The first task in concept development project is to:

- Determine the cost of the concept
- Determine the resources required for the concept
- **Determine the scope of the concept (Page 99)**
- Determine the test cases for concept

Question No: 216

Milestone represents:

- A predictable risk that may hit the project
- An unpredictable risk that may hit the project
- The Loss that organization may suffer due to risk
- **The defined target which you need to achieve**

Question No: 217

Timeline chart is also known as the _____ chart

- **Gantt (Page 100)**
- PERT
- FREE LANCE
- None of the given

Question No: 218

The more you refine the tasks, the more you can:

- **Estimate task with accuracy**
- Conclude about scope with accuracy
- Schedule the project with accuracy
- All of the given options

Question No: 219

AL-JUNAID TECH INSTITUTE

In a complex system that will lead to much.....code to write and maintain.

- **More (Page 159)**
- None of the given
- Less
- Meaningless

Question No: 220

Which of the following is NOT one of the tasks in concept scoping:

- Identify needs and benefits
- Define desired output/control/input
- Define the function/behavior
- **Identify the ambiguous requirements (Page 100)**

Question No: 1

BCWS is term used in Eamed Value Analysis it stands for_____.

- ▶ **Budgeted Cost of Work Schedule**
- ▶ Budgeted Cost of Whole System
- ▶ Beginning Cost of Whole System
- ▶ None of the given options

Question No: 2

_____states that 80% of the defects can be traced to 20% of its causes.

- ▶ **Pareto Principle**
- ▶ Fred Brooks Principle
- ▶ Kraul Principle
- ▶ D.Streeter Principle

Question No: 3

AL-JUNAID TECH INSTITUTE

The basic principle for a review is that the focus should be on the _____.

▶ **Product and not the producer**

- ▶ producer and not the Product
- ▶ Profit to be gained
- ▶ Presentation Slide

Question No: 4

Many defect identified in few components at design level testing/review, a software development team should _____.

▶ **Allocate good and extra resources at coding level to review these components thoroughly**

- ▶ Good achievement by designing team
- ▶ No need to allocated further resources at coding level to review these components
- ▶ Mission accomplished once good number of error/flaws identified level

Question No: 5

Milestone represents:

▶ **The defined target which you need to achieve**

- ▶ A predicable risk that may hit the project
- ▶ A unpredictable risk that may hit the project
- ▶ The Loss that organization may suffer due to risk

Question No: 6

_____ a statistical technique is used to develop an overall indication of improvement in software quality.

▶ **Error index**

AL-JUNAID TECH INSTITUTE

- ▶ Pareto Principle
- ▶ Quality Assurance
- ▶ Cost of Quality

Question No: 7

A model where Some errors are passed through to the next stem undebated such errors are said to have been_____.

▶ Amplified

- ▶ Reduced
- ▶ Neutralized
- ▶ Multiplexed

Question No: 8

Earned Value Analysis (EVA) is a_____technique for assessing the progress of a project.

▶ Quantitative

- ▶ Subjective
- ▶ Qualitative
- ▶ Descriptive

Question No: 9

It is noted that a/an _____review can be worse than no review at all.

▶ Uncontrolled

- ▶ Controlled
- ▶ Intermediate
- ▶ Normal

AL-JUNAID TECH INSTITUTE

Question No: 10

FTR is a quality control activity performed by software engineers.it stands for:

▶ Formal Technical Review

- ▶ Formatted Technical Review
- ▶ Feasibility of Technical Requirement
- ▶ Fetched Technical Requirement

Question No: 11

Legacy system migration however is a _____ task and there are a number of risks involved that need to be mitigated.

▶ Crucial

- ▶ Simple
- ▶ Straight-forward
- ▶ Easy

Question No: 12

An Engineering Change Order (ECO) defines the change be made, its constraints and _____.

▶ Criteria for Review and Audit

- ▶ Removal of Change Request
- ▶ Approval by change control Board (CCB)
- ▶ Approval by Change Control Authority (CCA)

Question No: 13

A system is considered to be a legacy system if it has been in operation for _____ years. A legacy system has many components.

▶ Many

- ▶ Less then seven

AL-JUNAID TECH INSTITUTE

- ▶ More than two
- ▶ Less than ten

Question No: 14

A major issue in requirements engineering is the rate at which requirements change once the requirements phase has "officially" _____.

▶ Ended

- ▶ Approved
- ▶ Deleted
- ▶ Started

Question No: 15

The implemented requirement has been _____ through the selected approach, such as testing or inspection.

▶ Verified

- ▶ Deleted
- ▶ Approved
- ▶ Ignored

Question No: 16

According to a standish Group report, lack _____ is responsible for 13% of all project failures,

▶ User input

- ▶ Output
- ▶ Resources
- ▶ Development Plan

Question No: 17

AL-JUNAID TECH INSTITUTE

_____ is a quality assurance process used to reduce defects by avoiding or correcting mistakes in early design and development phases.

▶ Poka-Yoke

- ▶ Software Testing
- ▶ Software Change management
- ▶ Defect Removal

Question No: 18

According to Clemm, _____ allows the user to specify alternative forms of the system the selection of a appropriate versions.

▶ Configuration Management

- ▶ Process Management
- ▶ Quality Management
- ▶ Cost Management

Question No: 19

_____ is the probability that a program is operating according to requirements at a given point in time.

▶ Availability

- ▶ Reliability
- ▶ Usability
- ▶ Security

Question No: 20

_____ enhance the product feature set by replacing a product with a newer version of the same product.

▶ Upgrade

- ▶ Re-Engineering

AL-JUNAID TECH INSTITUTE

- ▶ Update
- ▶ Deployment

Question No: 21

The equation to find out the Availability of software is:

▶ **Availability = (MTTF / MTBF) x 100**

- ▶ Availability = (MTTF / MTBF) / 100
- ▶ Availability = (MTTF + MTBF) / 100
- ▶ Availability = (MTTF - MTBF) / 100

Question No: 23

The software reengineering is a _____ activity.

▶ **Non-trivial**

- ▶ Code
- ▶ Out dated
- ▶ Document

Question No: 24

A Software Requirement Specification (SRS) is traced if the _____ of requirements is clear.

▶ **Origin**

Page 133

- ▶ Quality
- ▶ Standard
- ▶ Grammar
- ▶ Started

Question No: 27

_____ are used by the development as a staging mechanism for functionality including regular builds.

▶ **Internal releases**

AL-JUNAID TECH INSTITUTE

- ▶ External releases
- ▶ Major releases
- ▶ Feature releases

Question No: 28

_____ are mechanisms that lead to prudential quality problem before it occur or its rapid detection it introduced.

▶ Mistake-Proofing

- ▶ Mistake-Profiling
- ▶ Mistake-Provoking
- ▶ Mismanagement-Proofing

Question No: 29

Version Control discusses about:

▶ How does an Organization control changes before and after installation released to a customer.

- ▶ How the resources will be assigned to tasks in work breakdown structure (WBS).
- ▶ How the test cases will be executed before deploying each release.
- ▶ How many feature will be included in software.

Question No: 30

Which of the following is NOT one the tasks includes in Software Configuration Management.

▶ Test Case development

- ▶ Version Control
- ▶ Configuration Auditing
- ▶ Change Control

Question No: 31

_____ error is most costly to fix.

AL-JUNAID TECH INSTITUTE

▶ Requirement

- ▶ Design
- ▶ Implementation
- ▶ Testing

Question No: 32

_____ analysis is the first step in the reengineering process.

▶ Inventory

- ▶ Code
- ▶ Database
- ▶ Document

Question No:32

Change Control process addresses the important question of:

who has the responsibility for approving and ranking changes (Page 121)

Question No: 33

_____ are mechanisms that lead to prevention of potential quality problem before it occurs or its rapid detection if introduced.

Mistake-Proofing (Page 117)

Question No 34

_____ is an analysis technique which can be used to predict the chain of events that can cause hazards.

Fault Tree Analysis (Page 117)

Question No 35

The goal of _____ is to manage and control change.

Software Configuration Management (Page 119)

Question No 36

The reliability of a software can be determined by “MTBF” which stands for _____.

Mean Time Between Failure (Page 116)

Question No 37

Legacy system migration however is a _____ task and there are a number of risks involved that need to be mitigated.

Easy (Page 134)

AL-JUNAID TECH INSTITUTE

Question No 38

_____ errors are most costly to fix.

Requirement (Page 127)

Question No 39

_____ is a quality assurance process used to reduce defects by avoiding or correcting mistakes in early design and development phases.

Defect Removal (Page 69)

Question No 40

Poka Yoke is a Japanese term and it means:

Mistake Proofing (Page 118)

Question No 41

A/An _____ is used to fine tune the product by fixing minor product defects.

Update

Question No 42

_____ is a quality assurance process used to reduce defects by avoiding or correcting mistakes in early design and development phases.

Defect Removal

Question No 43

_____ are mechanisms that lead to prevention of potential quality problem before it occurs or its rapid detection if introduced.

Mistake-Proofing (Page 117)

Question No 44

Which of the following is NOT one of the tasks included in Software Configuration Management:

Test Case development..... Page 121

Question No 45

‘CCB’ decides whether to allow or deny a change request, it stands for _____.

Change Control Board..... Page 124

Question No 46

Mean Time Between Failure (MTBF) can be calculated by _____.

$MTBF = MTTF + MTTR$ page 116

Question No 47

The goal of _____ is to manage and control change.

AL-JUNAID TECH INSTITUTE

Software Configuration ManagementPage 119

Question No 48

Software configuration management tasks include Identification, Version Control, Change Control, Configuration Auditing and _____.

Reporting 121

Question No 49

Poka Yoke is the name of a _____.

Quality Assurance(QA) technique Page 118

Question No 50

Normally the compound number, used for release version numbering, is of the format:

X.Y.Z

Question No 51

The goal of software development is to develop _____ software on time and on budget, that meets customers' real needs.

Quality Page 127

Question No 52

Configuration Status Reporting (CSR) reports on:

- A) What happened?
- B) Who did it?
- C) When did it happen?

D) All of the given options

Question No 53

The information items defined by Engineering Change Order (ECO) include:

- A) **Criteria for review and audit**
- B) Work breakdown structure
- C) The project Task Network
- D) The change control test case

Question No 54

_____ is one of the techniques to prepare project schedule once we have defined "task network".

AL-JUNAID TECH INSTITUTE

- A) **Program evaluation and review techniques (PERT)**
- B) Critical Communication path method (CCPM)
- C) System resource negotiation method (SRNM)
- D) Strategic Schedule Development Technique (SSDT)

Question No 55

CPM stands for:

- A) Critical Project Measure
- B) Critical Path Method**
- C) Common Project Measure
- D) Critical Planning Model

Question No 56

BAC stands for:

- A) Budgeted Analysis Cost
- B) Budget and Cost
- C) Budget at compilation
- D) Budget at Completion**

Question No 57

The third digit (i.e. "Z") of the release version number format is called:

- A) Major Release Number
- B) Feature Release Number
- C) Defect Repair Number**
- D) None of the given options

Question No 58

The equation to find out the Availability of software is:

- A) $\text{Availability} = (\text{MTTF} \times \text{MTTBF}) / 100$
- B) $\text{Availability} = (\text{MTTF} / \text{MTTBF}) / 100$**
- C) $\text{Availability} = (\text{MTTF} + \text{MTTBF}) / 100$
- D) $\text{Availability} = (\text{MTTF} - \text{MTTBF}) / 100$

AL-JUNAID TECH INSTITUTE

Question No 9

Schedule Performance Index (SPI) is equal to:

A) BCWP/BCWS

B) BCWP*BCWS

C) BCWP+BCWS

D) BCWP-BCWS

Question No 60

Following are effective guidelines for Review EXCEPT:

A) We need to review the product not the producer

B) Be sensitive to personal egos

C) Errors should be pointed out gently

D) Tone should be high and strict

Question No 61

The basic idea behind Software configuration management is:

A) To manage and control the change

B) To efficiently allocate the resources

C) To properly manage status reporting

D) None of the given options

Question No 61:

_____ enhances the product feature set by replacing a product with a newer version of the same product.

- Update
- Deployment
- Re-Enjoining
- **Upgrade**

Question No 62:

AL-JUNAID TECH INSTITUTE

Which of the following is NOT one of the possible values of “status” attributes of requirements?

- Verified
- **Received**
- Approved
- Proposed

Question No 63:

Which of the following the Not one of the attributes of requirements?

- **Received by**
- Created by
- Creation Date
- Requirements ID

Question No 64:

_____ examine the ways in which failure result in condition that can lead to mishap.

- Software Usability
- Software Availability
- Software Portability
- **Software Safety**

Question No 65:

Which of the following is not one of the responsibilities of configuration Audit team?

- To verify that a FTR has been conducted to assess technical correctness
- To verify that all related DCI's been properly updated.
- **To verify that all the change requests are deleted after implementation**
- To verify that DE standards have been properly applied.

Question No 66:

AL-JUNAID TECH INSTITUTE

In a review the _____ should be pointed out gently and the tone should be loose and constructive.

- Inputs
- Outputs
- Conditions
- **Errors**

Question No 67:

A Quantitative quality measuring techniques that helps to uncover root cause for an error and then work on to fix what was traced is called _____

- Non-Statistical Software Quality Assurance
- Formal Techniques review
- **Walkthrough** NOT SURE
- Statistical Software Quality Assurances

Question No 68:

A/An _____ is used to fine tune the product by fixing minor product defects.

- Deployment
- Upgrade
- Enhancement
- **Update**

Question No 69:

Which of the following is an example of Software Configuration Item (SCI):?

- Reusability
- Test Case and Integration Plan
- Business Plan
- Requirements Plan

Question No 70:

Which of the following is NOT an example of Software Configuration Item (SCI):?

- Test Case
- **Test Case and Integration Plan**

AL-JUNAID TECH INSTITUTE

- Design specification
- None

Question No 71:

If _____ is not handled properly may create certain problem like identification and tracking. Version and software delivery related problem.

- Process Implement
- Process Management
- **Changes Chaos**
- Change documentation

Question No 72:

If the version of a release changes from 1.0.0 to 2.0.0.what will you conclude about new release(having version 2.0.0)

- **There were major changes in functionality**
- Bugs in previous functionality was removed
- There were no changes in the functionality
- There were partially change in functionality

Question No 73:

Earned Value Analysis (EVA) is a _____ technique for assessing the progress of a project

- Qualitative
- **Quantitative**
- Subjective
- None

Question No 74:

The basic idea behind Software configuration management is:

- **To manage and control the change**
- To efficiently allocate the resources
- To properly manage status reporting
- None

Question No 75:

Following are effective guidelines for Review EXCEPT

- We need to review the product, not the producer
- Be sensitive to personal egos
- Errors should be pointed out gently
- **Tone should be high and strict**

AL-JUNAID TECH INSTITUTE

Question No 76:

if :

E = Errors found before shipment

D = Errors found during operation

Then Defect Removal Efficiency (DRE) is equal to:

- $DRE = E + (E+D)$
- $DRE = E - (E+D)$
- $DRE = E * (E+D)$
- $DRE = E / (E+D)$

Question No 77:

In order to use the PERT and CPM, which one of the following is NOT required?

- Decomposition of tasks - also known as the work breakdown structure (WBS)
- **Teams communication mode details**
- Estimation of effort
- Interdependencies

Question No 74:

If an error related to requirements is identified in testing phase, this error will be considered as an error of ___phase

- **Requirement**
- Design
- Code
- Testing

Question No 75:

Quality cost may be delivered into costs associated with:

- **Prevention, appraisal and failure**
- Customers, developers and maintenance
- Builds, releases and products
- Bugs, errors and reports

Question No 76:

Poka-Yoke is the name of a

- **QA technique**
- Japanese dish
- Idiom
- None

Question No 77:

Configuration Auditing deals with:

- Ensuring that Resources have been allocated properly
- Ensuring that changes are implemented parallel with SDLC framework activities
- **Ensuring that the changes have been implemented properly**
- All of the given

Question No 78:

FTR stands for:

- Feasibility of technical requirement
- Fetched Technical Requirement

AL-JUNAID TECH INSTITUTE

- **Formal Technical Review**
- None

Question No 79:

A Software Requirement Specification is traced if the ____ of its requirements is clear.

- Quality
- Slandered
- **Origin**
- None

Question No 80:

Version Control discusses about:

- **How does an organization control changes before and after software is released to a customer**
- How the resources will be assigned to tasks in work breakdown structure (WBS)
- How the test cases will be executed before deploying each release.
- All of given

Question No 81:

The implemented requirement has been ____ through the selected approach, such as testing or inspection.

- **Verified**
- Deleted
- Approved
- None

Question No 82:

The Software reengineering is a ____ activity.

- Code
- Out dated
- Document
- **Non-trivial**

Question No 83:

Which of the following is NOT an example of Software Configuration Item (SCI):

- Test Case
- **Integration Plan**
- Design specification
- None

Question No 84:

Mean Time Between Failure (MTBF) can be calculated by:

- **$MTBF = MTTF + MTTR$**
- $MTBF = MTTF - MTTR$
- $MTBF = (MTTF/MTTR)*100$

AL-JUNAID TECH INSTITUTE

➤ $MTBF = (MTTR/MTTF)*100$

Question No 85:

___ software development is itself risky as changes to one part of the system inevitably involve further changes to other components.

- Forward
- Reverse
- Old
- **New**

Question No 86:

Program is restructured ___ the reverse engineering phase. In this case we modify source code and data in order to make it amenable to future changes.

- **After**
- None
- Along with
- Before

Question No 87:

___ Managers look at the system from the angle that does the system and associated business process make an effective contribution to the business goal?

- **Senior**
- IT
- None
- Line

Question No 88:

___ user assess the system from the perspective of how effective do they find the system in supporting their business processes and how much of the system functionality is used.

- Senior
- IT
- Line
- **End**

Question No 89:

A system is considered to be a legacy system if it has been in operation for ___ years. A legacy system has many components.

- Less than seven
- **Many**
- More than two
- Less than ten

Question No 90:

Legacy system migration, however, is an easy task and there are a number of risks involved that need to be mitigated

- Simple

AL-JUNAID TECH INSTITUTE

- Crucial
- **None**
- Easy

Question No 91:

___engineering does not simply create a modern equivalent of an older program, rather new user and technology requirements are integrated into the reengineering effort

- **Forward**
- None
- Reverse
- BPR

Question No 92:

Which of the following the NOT one of the attributes of requirements.

- Requirement ID
- Creation Date
- Created By
- **Received By**

Question No 93:

If the version of a release changes from 1.0.0 to 2.0.0 what will you conclude about new release (having version 2.0.0)

- There was no change in functionality
- **There was major change in functionality**
- There was partial change in functionality
- Bugs in previous functionality were removed

Question No 95:

Which of the following is NOT one of the tasks included in Software Configuration Management.

- Version Control
- Configuration Auditing
- **Test Case development**
- Change control

AL-JUNAID TECH INSTITUTE





CS605- Software Engineering II
Solved MCQS
From Final term Papers

July 10,2013

MC100401285

Moaaz.pk@gmail.com

Mc100401285@gmail.com

PSMD01

FINALTERM EXAMINATION
Spring 2011
CS605- Software Engineering II (Session - 1)

Question No: 1 (Marks: 1) - Please choose one

A law affirming that to continue after a certain level of performance has been reached will result in a decline in effectiveness. This law is know as _____

- ▶ **Law of Diminishing returns** [click here for detail](#)
- ▶ Law of effectiveness
- ▶ Law of Saturation
- ▶ Law of Marketing returns

Question No: 2 (Marks: 1) - Please choose one

_____ is not the construction part of software development activities.

- ▶ Requirement gathering
- ▶ Testing
- ▶ **Configuration Management (Page 8)**
- ▶ Coding

Question No: 3 (Marks: 1) - Please choose one

Basic project management process are established at _____ level of CMM

- ▶ Chaotic
- ▶ Initial
- ▶ Level 1
- ▶ **Repeatable (Page 12)**

دنیا میں سب سے مشکل کام اپنی اصلاح اور سب سے آسان کام دوسروں پر نکتہ چینی کرنا ہے

Muhammad Moaaz Siddiq – MCS(4th)
Moaaz.pk@gmail.com
Campus:- Institute of E-Learning & Moderen Studies
(IEMS) Samundari

Question No: 4 (Marks: 1) - Please choose one

The projects usually don't fail due to

- ▶ ambiguous/incomplete requirements
- ▶ changing customer requirements
- ▶ **the fact that company was not CMM certified (Page 28 - 29)**
- ▶ failure in project management

Question No: 5 (Marks: 1) - Please choose one

_____ is a team organization where there is no permanent leader and task coordinators are appointed for short duration. Decisions on problems and approach are made by group consensus and communication among team is horizontal.

- ▶ **Democratic decentralized (DD) (Page 32)**
- ▶ Controlled decentralized (CD)
- ▶ Synchronous paradigm (SP)
- ▶ Controlled centralized (CC)

Question No: 6 (Marks: 1) - Please choose one

Which of these is not a valid reason for measuring software processes, products, and resources?

- ▶ **to price them** [Click here for detail](#)
- ▶ to improve them
- ▶ to characterize them
- ▶ to evaluate them

Question No: 7 (Marks: 1) - Please choose one

Integrity can be measured by the following formula

- ▶ integrity= (1-threat) x (1-security)
- ▶ integrity= $\sum (1+threat) + (1-security)$
- ▶ **integrity= $\sum [(1-threat) \times (1-security)]$ (Page 69)**
- ▶ integrity= $\sum (1-threat) - (1-security)$

Question No: 8 (Marks: 1) - Please choose one

The rapid application development model is

- ▶ Another name for component-based development.
- ▶ A useful approach when a customer cannot define requirements clearly
- ▶ **A high speed adaptation of the linear sequential model. (Page 19)**
- ▶ All of the given

اللہ کا خوف سب سے بڑی دانائی ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 9 (Marks: 1) - Please choose one

An _____ is a user identifiable group of logically related data or control information maintained within the boundary of the application

- ▶ **Internal logical file (ILF) (Page 42)**
- ▶ External Interface file (EIF)
- ▶ External input
- ▶ External Query

Question No: 10 (Marks: 1) - Please choose one

When more than one user interpret the same requirement in different ways then we can say that the requirements are

None of the given
Incomplete

Ambiguous [Click here for detail](#)

Incorrect

Question No: 11 (Marks: 1) - Please choose one

Testing and Software Quality Assurance activities are exactly similar activities with different names

True

False (Page 69)

Question No: 12 (Marks: 1) - Please choose one

If an error related to requirements is identified in testing phase, so this error will be considered as an error in _____ phase.

- ▶ Design
- ▶ Code
- ▶ Requirement
- ▶ **Testing**

Question No: 13 (Marks: 1) - Please choose one

Usually the performance of the organization change over-night.

▶ True

▶ **False** [Click here for detail](#)

بري صحبت سے تنہائی بہتر ہے اور تنہائی سے نيك صحبت بہتر ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 14 (Marks: 1) - Please choose one

Following are the implicit requirement(s) of software?

- ▶ Efficiency
- ▶ Ease of use
- ▶ Maintainability
- ▶ **All of the given (Page 106)**

Question No: 15 (Marks: 1) - Please choose one

If an error related to requirements is identified in Coding, so this error will be considered as an error in _____ phase.

- ▶ Design
- ▶ **Code**
- ▶ Requirement
- ▶ Testing

Question No: 16 (Marks: 1) - Please choose one

While developing the software if the bug is caught then it is termed as

- ▶ Error
- ▶ **Defect (not sure)**
- ▶ Fault
- ▶ Mistake

Question No: 17 (Marks: 1) - Please choose one

Following are the types of Formal Technical Reviews except

- ▶ Inspection
- ▶ **JAD (Page 110)**
- ▶ Walkthrough

Question No: 18 (Marks: 1) - Please choose one

The Review participants should not focus on

- ▶ Work product
- ▶ **Weaknesses of the developer (Page 113)**
- ▶ Time lines
- ▶ Review agenda

ایماندار کو غصہ دیر سے آتا ہے اور جلدی دور ہو جاتا ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 19 (Marks: 1) - Please choose one

Review meeting is attended by the following except

- ▶ Review leader
- ▶ All reviewers
- ▶ **CEO (Page 112)**
- ▶ Product Producer

Question No: 20 (Marks: 1) - Please choose one

Which statement is true in the light of Pareto principle?

- ▶ **80% of defects are caused by 20 % of mistakes (Page 114)**
- ▶ 20% of defects are caused by 80 % of mistakes

Question No: 21 (Marks: 1) - Please choose one

Reliability and Safety are same concepts with respect to software quality

- ▶ **True (Page 117)**
- ▶ False

Question No: 22 (Marks: 1) - Please choose one

The prevention of defects being injected into the software can help in decreasing the cost of software.

True

False

Question No: 23 (Marks: 1) - Please choose one

Poka Yoke is the name of a

- ▶ Japanese dish
- ▶ **QA technique (Page 118)**
- ▶ Idiom
- ▶ None of the given

Question No: 24 (Marks: 1) - Please choose one

If a new version of a product is released by fixing the bugs in the previous release then it is termed as ____

- ▶ **Product Update (Page 122)**
- ▶ Product Upgrade
- ▶ Defect Removal
- ▶ Product Performance

زندگی میں کامیابی کا یہی راز ہے کہ پریشانیوں سے پریشان مت بنو

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 25 (Marks: 1) - Please choose one

The Evolution Graph signifies the

- ▶ **version relationship (SE-PRESSMAN, page 260)**
- ▶ baseline of a particular release
- ▶ change control activity
- ▶ all of the given

Question No: 26 (Marks: 1) - Please choose one

In the software development life cycle, soon the defect is detected will results in

increase in cost of software

decrease in software cost

both a and b are correct

Question No: 27 (Marks: 1) - Please choose one

_____ testing verifies the correct implementation of internal units, structures and relationship among them.

Black box

White box [click here for details](#)

Gray box

Question No: 28 (Marks: 1) - Please choose one

The template for organizing SRS given by American Department of Defense and NASA should be used for

- ▶ none of the given
- ▶ **large and complex projects**
- ▶ small project
- ▶ medium size projects

Question No: 29 (Marks: 1) - Please choose one

If you know the relationship between the use cases, classes, GUIs, Test cases and other artifacts then we can say that requirements are traceable in that project.

- ▶ True
- ▶ **False (Page 133)**

دنیا کی سب سے بڑی فتح نفس پر قابو رکھنا ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 30 (Marks: 1) - Please choose one

There may be one or more GUIs that are part of any good software but may not satisfy any requirement

- ▶ True
- ▶ False

Question No: 31 (Marks: 1) - Please choose one

Following are the components of a Legacy system except:

- ▶ business processes
- ▶ system hardware
- ▶ **marketing (Page 134)**
- ▶ application software

Question No: 32 (Marks: 1) - Please choose one

There are many reasons for which the legacy system become difficult to maintain EXCEPT

- ▶ Language in which system was developed become obsolete
- ▶ lack of consistency as different parts of the system have been developed by different teams
- ▶ None of these
- ▶ **No documentation is available (Page 134)**

Question No: 33 (Marks: 1) - Please choose one

In _____, new user and technology requirement can also be integrated into the re-engineering effort.

- ▶ Backward re-engineering
- ▶ **Forward re-engineering (Page 140)**
- ▶ Business Engineering
- ▶ Business Process Engineering

Question No: 34 (Marks: 1) - Please choose one

Large classes reduces

- ▶ **Cohesion (Page 143)**
- ▶ Coupling
- ▶ Consistency
- ▶ Chains

Question No: 35 (Marks: 1) - Please choose one

Level _____ is the lowest capability level of CMMI in Continuous representation

- ▶ 1
- ▶ **0 (Page 169)**
- ▶ 5
- ▶ 6

Question No: 36 (Marks: 1) - Please choose one

In _____ representation of CMMI model, consideration is given to improve selected processes areas.

▶ **Staged (Page 169)**

▶ Continuous

Question No: 37 (Marks: 1) - Please choose one

Empirical models are statistical models and are based upon historic data

True (Page 81)

False

Question No: 38 (Marks: 1) - Please choose one

Error tracking provides a quantitative means of assessing the quality of the individuals implementing a software product.

▶ True

▶ **False**

Question No: 39 (Marks: 1) - Please choose one

There are tools available in the market for project tracking. These tools can automatically track and manage the project replacing the need of any human being as a Project Manager.

▶ **True**

▶ False

Question No: 40 (Marks: 1) - Please choose one

Two tools for computing critical path and project completion times from activity networks are

▶ **CPM and PERT (Page 100)**

▶ DRE and SQA

▶ FP and LOC

▶ ASD and BSD

جھوٹ انسان اور ایمان دونوں کا دشمن ہے

FINAL TERM EXAMINATION
Spring 2010
CS605- Software Engineering II (Session - 1)

Question No: 1 (Marks: 1) - Please choose one

In order to measure the design quality, if the frequency of ripple defects is too large, then it means that there is tight coupling and hence the

- ▶ design is maintainable
- ▶ **design is not maintainable (Page 79)**
- ▶ design has completed
- ▶ none of the given

Question No: 2 (Marks: 1) - Please choose one

In _____, there is both vertical and horizontal communication.

Controlled decentralized (CD) Page 32

Democratic decentralized (DD)

Controlled centralized (CC)

Synchronous paradigm (SP)

Question No: 3 (Marks: 1) - Please choose one

The only reason for an estimate to be unreliable is lack of experience related to the application on the part of the estimator.

- ▶ true
- ▶ **false** [click here for detail](#)

Question No: 4 (Marks: 1) - Please choose one

A consideration of software scope must include an evaluation of all external interfaces.

- ▶ **true (Page 80)**
- ▶ false

Question No: 5 (Marks: 1) - Please choose one

Software quality is conformance to _____

Explicitly stated functional and non functional requirements (Page 106)

Explicitly documented development standards

Implicit characteristics that all the professional software are expected to have

All of the given options

عقل مند کہتا ہے میں کچھ نہیں جانتا جبکہ بے وقوف کہتا ہے کہ میں سب کچھ جانتا ہوں

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 6 (Marks: 1) - Please choose one

The boundary time defines the following parameter(s) for a project

the latest time for task initiation before the minimum project completion time is delayed

The earliest finish

The latest finish

All of the given options (Page 100)

Question No: 7 (Marks: 1) - Please choose one

Quality is the compliance of software to implicit and explicit quality factors. Identify the explicit requirement from the following:

- ▶ Maintainability
- ▶ **Compliance with CMMI (Page 106)**
- ▶ Ease of use
- ▶ Efficiency

Question No: 8 (Marks: 1) - Please choose one

Following are the benefits of conducting review except

- ▶ **Help in finding the size of the project**
- ▶ Helps in verifying the quality of product
- ▶ Help in identifying where improvement is required.
 - ▶ Helps in identifying the bugs in the work product

Question No: 9 (Marks: 1) - Please choose one

Mean Time To Repair (MTTR) is the

- ▶ Time when system remained unable for usage
- ▶ Time taken to fix the error
 - ▶ **Time taken to repair the software (Page 116)**
- ▶ All of the given options are correct.

Question No: 10 (Marks: 1) - Please choose one

In order to use the Pert and CPM, which one of the following is not required?

Decomposition of tasks - also known as the work breakdown structure (WBS)

Teams communication mode detail (Page 100)

Estimation of effort

inter dependencies

خود کو تمہیں سے بڑھ کر کوئی اچھا مشورہ نہیں دے سکتا

Question No: 11 (Marks: 1) - Please choose one
SCM is the requirement of the _____ level(s) of CMM

- ▶ First
- ▶ Second and later
- ▶ Third and later
- ▶ **Only Second (Page 119)**

Question No: 12 (Marks: 1) - Please choose one
We can include following items during configuration item identification:

- ▶ User Manuals and Documentations
- ▶ Source Code
- ▶ Software Requirement Specifications
- ▶ **All of the given choices are correct (Page 120)**

Question No: 13 (Marks: 1) - Please choose one
CPM stands for

- critical project Measure
- Critical Path Method (page 100)**
- Common Project Measure
- Critical Planning Model

Question No: 14 (Marks: 1) - Please choose one
Software re-factoring is a process in which

- External behavior of the system does not change
- Internal behavior of the system does not change
- Design of the software changes
- Architecture of the software changes** [Click here for detail](#)

Question No: 15 (Marks: 1) - Please choose one
The Change Control Authority does not have the rights to permit to bring the change in the software

- ▶ True
- ▶ **False (Page 124)**

Question No: 16 (Marks: 1) - Please choose one
Check in and check out is actually one and the same process with two different names

- ▶ True
- ▶ **False (Page 124)**

Question No: 17 (Marks: 1) - Please choose one
There could be multiple GUIs to satisfy one requirement

- ▶ **True (Page 37)**
- ▶ False

Question No: 18 (Marks: 1) - Please choose one
_____ is the process of Design recovery. At this stage the documentation of the overall functionality is created.

- ▶ Database Engineering
- ▶ **Reverse Engineering (Page 139)**
- ▶ Refactoring
- ▶ Forward Engineering

Question No: 19 (Marks: 1) - Please choose one
In code restructuring, we do not change the functionality of the code

- ▶ **True (Page 139)**
- ▶ False

Question No: 20 (Marks: 1) - Please choose one
Earned Value Analysis (EVA) is a _____ technique for assessing the progress of a project.
Select correct option:

- Qualitative
- ▶ **Quantitative (page 102)**
- Subjective
- None of the given options

Question No: 21 (Marks: 1) - Please choose one
A _____ process is “a set of logically related tasks performed to achieve a defined business outcome”.

- ▶ **Business (Page 141)**
- ▶ Software
- ▶ CMM
- ▶ ISO

Question No: 22 (Marks: 1) - Please choose one
Software Refactoring is a process in which

- ▶ **External behavior of the system does not change (Page 142)**
- ▶ Internal behavior of the system does not change
- ▶ Design of the software changes
- ▶ Architecture of the software changes

Question No: 23 (Marks: 1) - Please choose one

The formal methods model of software development makes use of mathematical methods to

- ▶ define the specification for computer-based systems
 - ▶ **all of the given** [click here for detail](#)
- ▶ develop defect free computer-based systems
- ▶ verify the correctness of computer-based systems

Question No: 24 (Marks: 1) - Please choose one

Which one of the following describes the data and control to be processed, function, performance, constraints, interfaces, and reliability?

- ▶ Product Quality
- ▶ **Software scope estimation (Page 80)**
- ▶ Resources requirements
- ▶ Time requirements

Question No: 25 (Marks: 1) - Please choose one

A significant SQA plan, among others, should include:

Select correct option:

Resources required for project (Page 177)

- Error tracking procedures
- Phases required in application development process
- None of the given options

Question No: 26 (Marks: 1) - Please choose one

The software plan is not a static document, it is frequently adjusted to make the project appear on track to meet all deadlines and quality targets.

- ▶ **True** [Click here for detail](#)
- ▶ False

Question No: 27 (Marks: 1) - Please choose one

The projects are classified into following categories except

- ▶ New application development
- ▶ Reengineering projects
- ▶ Concept development projects
- ▶ **Marketing Development Projects (Page 96)**

Question No: 28 (Marks: 1) - Please choose one

The processes at _____ is focus on continually improving process performance through both incremental and innovative technological changes/improvements.

▶ **Level 5** [Click here for detail](#)

- ▶ Level 1
- ▶ Level 4
- ▶ Level 3

Question No: 29 (Marks: 1) - Please choose one

Quantitative process-improvement objectives for the organization are established, continually revised to reflect changing business objectives at _____.

- ▶ Level 3
- ▶ Level 2
- ▶ **Level 5** [Click here for detail](#)
- ▶ Level 1

Question No: 30 (Marks: 1) - Please choose one

Reliability of a software is a

- ▶ functional requirement
- ▶ **non-functional requirement** [Click here for detail](#)
- ▶ Design Requirement
- ▶ None of the given

FINAL TERM EXAMINATION
Spring 2010
CS605- Software Engineering II (Session - 1)

Question No: 1 (Marks: 1) - Please choose one

Which of the following is NOT one of the degrees of rigor?

- Casual
- Structured
- Strict

TSS (Page 66)

جو شخص ناکامیوں سے ڈر کر بھاگتا ہے کامیابی اُس سے ڈر کر بھاگتی ہے

Muhammad Moaaz Siddiq – MCS(4th)
Moaaz.pk@gmail.com
Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari

Question No: 2 (Marks: 1) - Please choose one

Build and Fix model is a _____ type of software development activity.

- ▶ Mathematical
- ▶ Perfect
- ▶ **Haphazard** [Click here for detail](#)
- ▶ Planned

Question No: 3 (Marks: 1) - Please choose one

Three categories of risks are

Business risks, personnel risks, budget risks

Project risks, technical risks, business risks (Page 86)

Planning risks, technical risks, personnel risks

Management risks, technical risks, design risks

Question No: 4 (Marks: 1) - Please choose one

MTTC is the abbreviation of

- ▶ measured time to change
- ▶ mean time to collaborate
- ▶ **mean time to change (Page 68)**
- ▶ measure time to cope

Question No: 5 (Marks: 1) - Please choose one

While creating matrix for a set of projects, data should be collected that belongs to a similar domain

- ▶ **true (Page 72)**
- ▶ false

Question No: 6 (Marks: 1) - Please choose one

_____ is one of the techniques to prepare project schedule once we have defined "task network"

Program evaluation and review techniques (PERT) (Page 100)

Critical Communication path method (CCPM)

System resource negotiation method (SRNM)

Strategic Schedule Development technique (SSDT)

Question No: 7 (Marks: 1) - Please choose one

While managing the historical data, we should consider the following **except**

- ▶ **Data should be collected only from scientific projects and applications (Page 72)**
- ▶ Data should be collected from many projects
- ▶ Data should be consistent
- ▶ Correct data should be taken

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 8 (Marks: 1) - Please choose one

Quality costs may be divided into costs associated with

- ▶ **prevention, appraisal, and failure** [click here for detail](#)
- ▶ people, process, and product
- ▶ customers, developers, and maintenance
- ▶ all of the given

Question No: 9 (Marks: 1) - Please choose one

When a software is delivered to a client and then client reports the bug in it then that bug is termed as

- ▶ Error
- ▶ **Defect** [click here for detail](#)
- ▶ Mistake
- ▶ Fault

Question No: 10 (Marks: 1) - Please choose one

FTR stands for:

- Feasibility of technical requirement
- Fetches Technical Requirement
- Formal Technical Review (Page 109)**
- None of the given options

Question No: 11 (Marks: 1) - Please choose one

SQA is an umbrella activity in which following activities are performed EXCEPT

- Review
- Testing
- Project Planning
- Inspection (Page 177)**

Question No: 12 (Marks: 1) - Please choose one

Software configuration management is used to

- ▶ **Manage change in the software (Page 119)**
- ▶ Manage the financials accounts in an organization
- ▶ Manage the administration of company
- ▶ All of the given

جو لوگوں کے سامنے فخر کرتا ہے وہ لوگوں کی نظروں سے گر جاتا ہے

Question No: 13 (Marks: 1) - Please choose one

Which of the following is not TRUE about "Error Tracking" ?

Error tracking can be used to estimate the progress of the project.

We track errors in work products to assess the status of a project.

Durint "Error Tracking", we may also need the historical data from similar projects. (Page 104)

All of the given options.

Question No: 14 (Marks: 1) - Please choose one

SQA is an activity in which testing is performed only

True

False (Page 177)

Question No: 15 (Marks: 1) - Please choose one

Schedule Performance Index (SPI) is equal to

BCWP / BCWS (Page 102)

BCWP * BCWS

BCWP + BCWS

BCWP - BCWS

Question No: 16 (Marks: 1) - Please choose one

BCWS for a task i will be equal to:

Risks appeared in task i

Effort (person-days etc) for task I (Page 102)

Resource deficiency for task i

None of the given options

Question No: 17 (Marks: 1) - Please choose one

if: E = Errors found before shipment D = Errors found during operation Then Defect Removal Efficiency (DRE) is equal to:

DRE = E + (E+D)

DRE = E - (E+D)

DRE = E * (E+D)

DRE = E / (E+D) (Page 104)

Question No: 18 (Marks: 1) - Please choose one

Quality assurance helps management in providing the necessary data about:

Product quality (Page 106)

Errors/bugs left

Errors/bugs highlighted

Defect reports

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 19 (Marks: 1) - Please choose one

The progress on a project at any given point in time can be calculated by:

Adding all the BCWS till that point for all tasks (Page 102)

Multiplying all the BCWS till that point for all tasks

Adding BCWS and EVA for all tasks till that point

None of given options

Question No: 20 (Marks: 1) - Please choose one

The process of rediscovering the software design is called

▶ **Reverse Engineering (Page 139)**

▶ Business Engineering

▶ Business Process Engineering

▶ Forward Engineering

Question No: 21 (Marks: 1) - Please choose one

OCI stands for:

Output, Control and Input [Click here for detail](#)

Overflow of Certain Inputs

Overflow of Certain Indexes

None of given options

Question No: 22 (Marks: 1) - Please choose one

A redesigned business process must be prototyped before it is fully integrated into the business.

▶ False

▶ **True (Page 141)**

Question No: 23 (Marks: 1) - Please choose one

Critical path defines:

The list of the resources that may be needed for future activities.

The chain of tasks that determines the duration of the project. (Page 100)

The list of the deficient resources, project is suffering from.

None of the given options.

Question No: 24 (Marks: 1) - Please choose one

BAC stands for:

Budgeted Analysis Cost

Budget And Cost

Budget at compilation

Budget at Completion (Page 102)

Question No: 25 (Marks: 1) - Please choose one

The first task in concept development project is to:

Determine the cost of the concept

Determine the resources required for the concept

Determine the scope of the concept (Page 99)

Determine the test cases for concept

Question No: 26 (Marks: 1) - Please choose one

Milestone represents:

A predictable risk that may hit the project

An unpredictable risk that may hit the project

The Loss that organization may suffer due to risk

The defined target which you need to achieve [Click here for detail](#)

Question No: 27 (Marks: 1) - Please choose one

Timeline chart is also known as the _____ chart

▶ **Gantt (Page 100)**

▶ PERT

▶ FREE LANCE

▶ None of the given

Question No: 28 (Marks: 1) - Please choose one

The more you refine the tasks, the more you can:

Estimate task with accuracy

Conclude about scope with accuracy

Schedule the project with accuracy

All of the given options

Question No: 29 (Marks: 1) - Please choose one

In a complex system that will lead to muchcode to write and maintain.

▶ **More (Page 159)**

▶ None of the given

▶ Less

▶ Meaningless

عقل مند اپنے عیب خود دیکھتا ہے اور بیوقوفوں کے عیب دنیا دیکھتی ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 30 (Marks: 1) - Please choose one

Which of the following is NOT one of the tasks in concept scoping:

- Identify needs and benefits
- Define desired output/control/input
- Define the function/behavior

Identify the ambiguous requirements (Page 100)

FINALTERM EXAMINATION

Fall 2009

CS605- Software EngineeringII (Session - 1)

Question No: 1 (Marks: 1) - Please choose one

Defining a |Task Network| helps in defining

- The resources required for project
- Third party tool required for project

Sequence in which activities will be performed (Page 99)

None of these

Question No: 2 (Marks: 1) - Please choose one

The first step in project planning is to

- Determine the budget.
- Select a team organizational model.
- Determine the project constraints.

Establish the objectives and scope. (Page 80)

Question No: 3 (Marks: 1) - Please choose one

Function Point analysis is helpful in calculating the size of the software for _____

- ▶ Both client and software organization
- ▶ Software organization
- ▶ Client
- ▶ **User (Page 38)**

بد صورت چہرہ بد صورت دماغ سے بہتر ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 4 (Marks: 1) - Please choose one

The extent to which a program satisfies its specification and fulfills the customer's mission objectives is said to be achieving the

- ▶ Usability
- ▶ Efficiency
- ▶ Reliability

Correctness [click here for detail](#)

Question No: 5 (Marks: 1) - Please choose one

What activity/activities , we can perform to identify the needs and benefits of a task?

Establish project statement

Identify needs and project constraints

All of these

Question No: 6 (Marks: 1) - Please choose one

BCWS stands for:

Budgeted cost of whole System

Budgeted cost of work schedule (Page 102)

None of the given options

Question No: 7 (Marks: 1) - Please choose one

BCWP refers to:

Normalized value of budgeted cost

Overall expected cost of the project (anticipated cost)

Budgeted cost of work performed so far (Page 102)

Overall cost of the project (calculated after completion of project)

Question No: 8 (Marks: 1) - Please choose one

Which statement is correct?

- ▶ **The greater the dependency between the components the greater is coupling** [Click here for detail](#)
- ▶ The lesser the dependency between the components the greater is coupling
- ▶ The greater the dependency between the components the lesser is coupling
- ▶ None of the given

عقل مند آدمی اس وقت تک نہیں بولتا جب تک خاموشی نہیں ہو جاتی

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 9 (Marks: 1) - Please choose one

In context of moving range and individual control charts, UNPL stands for:

Universal Natural Process Line

Universal Natural Process Limit

Upper Natural Process Limit (Page 77)

Upper Natural Process Line

Question No: 10 (Marks: 1) - Please choose one

Effort required to test a program to ensure that it performs its intended function _____

Testability page (Page 68)

Bug fixing

Debugging

Security

Question No: 11 (Marks: 1) - Please choose one

The extent to which a program can be expected to perform its intended function with required precision is called _____

Usability

Reliability (Page 67)

Portability

Maintainability

Question No: 12 (Marks: 1) - Please choose one

Review is a type of _____ that helps preventing the bugs to move in the next stage of software development

- ▶ Bug seeding tool
- ▶ **Filter (Page 109)**
- ▶ Alarm

Question No: 13 (Marks: 1) - Please choose one

_____ help in finding the matrix to be stable or unstable

Control chart (Page 75)

Directed Graph

Cyclic chart

Base line graph

انسان دکھ نہیں دیتے بلکہ انسانوں سے وابستہ امیدیں دکھ دیتی ہیں

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 14 (Marks: 1) - Please choose one

Defect Removal Efficiency can be increased by

- ▶ By increasing the team member in a team
- ▶ **Identifying the defect in early stages of development so that it may not be amplified**
- ▶ Identifying the defect in the later stages of project
- ▶ By performing the testing activity only

Question No: 15 (Marks: 1) - Please choose one

_____give you a better insight into the state of the process or product

Metrics (Page 66)

Efficiency
Reliability
Usability

Question No: 16 (Marks: 1) - Please choose one

Determination of theis a pre-requisite of all sorts of estimates, including, resources, time, and budget.

software scope (Page 80)

software Risk
software Quality
software Management

Question No: 17 (Marks: 1) - Please choose one

Following are effective guidelines for Review **except**

- ▶ We need to review the product not the producer
- ▶ Be sensitive to personal egos
- ▶ Errors should be pointed out gently
- ▶ **Tone should be high and strict (Page 113)**

Question No: 18 (Marks: 1) - Please choose one

Function/Test matrix is a type of

Interim Test report
Final test report

Project status report [click here for detail](#)

Management report

بہترین تجربہ وہ ہے جس سے نصیحت حاصل ہو

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 19 (Marks: 1) - Please choose one

Poka Yoke is Japanese term and it means

- ▶ **Mistake Proofing (Page 118)**
- ▶ Mistake Handling
- ▶ Mistake identification
- ▶ Mistake assurance

Question No: 20 (Marks: 1) - Please choose one

we need to employ some statistical techniques and plot the result ----- . This is known as statistical control techniques.

Graphically (Page 74)

automatically
manually
personally

Question No: 21 (Marks: 1) - Please choose one

The extent to which a program satisfies its specifications and fulfills the customer's mission objectives is ____

Integrity
Reliability

Correctness (Page 67)

none of given

Question No: 22 (Marks: 1) - Please choose one

Which is not correct in the context of Poka-Yoke technique?

- ▶ Simple and cheap
- ▶ **Sophisticated and expensive (Page 118)**
- ▶ Part of Process
- ▶ Indication point should be near to the place where the problem occurred

Question No: 23 (Marks: 1) - Please choose one

Metrics to assess the quality of the analysis models and the corresponding software specification were proposed_____

in 1993- 1999.

Ricardo in 1993

Davis in 1990

Davis in 1993 (Page 71)

خوبصورتی علم و ادب سے ہوتی ہے لباس و حسن سے نہیں

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 24 (Marks: 1) - Please choose one

If Configuration item identification is not identified, it is possible to control changes and establish records.

▶ **False (Page 121)**

▶ True

Question No: 25 (Marks: 1) - Please choose one

Incomplete Configuration identification documents may result in:

- Defective Product
- Higher Maintenance Costs
- Schedule Product
- Meet Software Quality

A,B,C [Click here for detail](#)

B,C,D

A,B,D

A,B,C,D

Question No: 26 (Marks: 1) - Please choose one

When an Item is baselined, it becomes frozen, here frozen means, that the item can be changed only by creating an old version

▶ **True (Page 72)**

▶ False

Question No: 27 (Marks: 1) - Please choose one

The most important objective of any engineering activity is to produce high quality product with limited resources and-----

Time (Page 66)

Persons

Cost

Metrics

Question No: 28 (Marks: 1) - Please choose one

Extent to which access to software or data by unauthorized persons can be controlled and called _____

None of given

Efficiency

Reliability

Integrity (Page 67)

Question No: 29 (Marks: 1) - Please choose one

Object un-lock is done by

- ▶ **check in (Page 125)**
- ▶ check out
- ▶ both check-in and check-out
- ▶ none of the given

Question No: 30 (Marks: 1) - Please choose one

Requirements engineering is the basis of the contract between the developer and the client.

- ▶ True
- ▶ **False Written as a contract between client and contractor**

Question No: 31 (Marks: 1) - Please choose one

The amount of computing resources required by a program to perform its function is _____

Efficiency (Page 67)

Integrity
Reliability
None of given

Question No: 32 (Marks: 1) - Please choose one

Legacy system migration is usually hard. One important reason is that no proper documentation of the system is available

- ▶ True
- ▶ **False (Page 134)**

Question No: 33 (Marks: 1) - Please choose one

While assessing a legacy system for further decision, one should decide the following about the supplier

- ▶ Is supplier still in existence?
- ▶ Is supplier still in business?
- ▶ Is supplier can provide support?
- ▶ **All of the given (Page 137)**

Question No: 34 (Marks: 1) - Please choose one

The application software is assessed on the basis of following factors except

- ▶ Documentation
- ▶ Understandability
- ▶ **Code Refactoring (Page 137)**
- ▶ Test data

Question No: 35 (Marks: 1) - Please choose one

We modify the internal _____ in code re-structuring

- ▶ interface
- ▶ **design (Page 139)**
- ▶ data structure
- ▶ functionality

Question No: 36 (Marks: 1) - Please choose one

_____ technique was initially developed for manufacturing processes in the 1920's by Walter Shewart.
Upper Control Line

Control chart (Page 75)

CMP

Quality

Question No: 37 (Marks: 1) - Please choose one

“A method requires lot of information from some other class” is a symptom of _____ bad smell

- ▶ Data clumps
- ▶ Lazy class
- ▶ **Feature envy (Page 143)**
- ▶ Long Method

Question No: 38 (Marks: 1) - Please choose one

_____ integrates system and software disciplines into single process improvement framework for introducing new disciplines as needs arise.

- ▶ SEI
- ▶ **CMMI** [Click here for detail](#)
- ▶ CMM
- ▶ ISO

Question No: 39 (Marks: 1) - Please choose one

FAST is the abbreviation of

Facilitated Application Specification Technology

Facilitated Application Specification Technique (Page 80)

Facilitated Application Specialization Technique

None of the above option

تم اچھا کرو زمانہ تم کو برا سمجھے یہ اس سے بہتر ہے کہ تم برا کرو اور زمانہ تم کو اچھا سمجھے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 40 (Marks: 1) - Please choose one

In measuring Software Process Quality by using control charts, if the gap between the defects reported and defects fixed is increasing, then it means

The product is in unstable condition. (Page 78)

the product is ready for shipment

the product is in stable condition.

None of the above

Question No: 41 (Marks: 1) - Please choose one

Although there are many different models developed by different researchers for estimation, all of them share which one of the following basic structure

- ▶ $E = 3.2 (KLOC)^{1.05}$
- ▶ $E = A + B * (ev)C$ (Page 81)
- ▶ $E = [LOC \times B^{0.333/P}]^3 \times (1/t^4)$
- ▶ none of the given

Question No: 42 (Marks: 1) - Please choose one

A _____ is a user recognizable subgroup of data elements within an ILF or EIF

- ▶ **Record element type (RET) (Page 46)**
- ▶ Data Element Type
- ▶ External Input
- ▶ External Query

Question No: 43 (Marks: 1) - Please choose one

This chart is then used to develop the individual control chart is called statistical control techniques.

Yes (Page 77)

No

Question No: 44 (Marks: 1) - Please choose one

Any delay in _____ path makes the whole project delayed.

- ▶ Logical
- ▶ Neural
- ▶ Physical
- ▶ **Critical** [Click here for detail](#)

انسان کے لئے بری صحبت سے بڑھ کر بری کوئی چیز نہیں

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 45 (Marks: 1) - Please choose one

The responsibilities of a Project Manager does not include

- ▶ Make a schedule of project
- ▶ Allocate tasks to the resources
- ▶ Monitoring the tasks
- ▶ **All these options (Page 28)**

Question No: 46 (Marks: 1) - Please choose one

 Subgroups are those that the user has the option of using one or none of the subgroups during an elementary process.

- ▶ **Optional (Page 46)**
- ▶ Mandatory
- ▶ None of the Given
- ▶ RET

Question No: 47 (Marks: 1) - Please choose one

Process discipline is unlikely to be rigorous, but where it exists it may help to ensure that existing processes are maintained during times of stress and this is done at CMM level -----.

- ▶ 1
- ▶ 4
- ▶ 3
- ▶ **2 (Page 172)**

Question No: 48 (Marks: 1) - Please choose one

Which one of the following activities does not belong to Risk Mitigation, Monitoring, and Management Plan?

Risk Projection (Page 89)

Risk Mitigation
Risk avoidance
Risk Management and Contingency Planning

Question No: 49 (Marks: 1) - Please choose one

Which one of the following principles/techniques is NOT software projects scheduling?

Interdependency identification
Time allocation
Effort validation

Function point analysis (Page 93)

Milestone definition

Question No: 50 (Marks: 1) - Please choose one

With a complex class you have to move data and methods around in small pieces to avoid errors, it seems slow but it is the _____ because you avoid debugging

- ▶ **Quickest (Page 163)**
- ▶ Problem
- ▶ None of the given
- ▶ Slowest

FINAL TERM EXAMINATION
Fall 2008
CS605- Software EngineeringII (Session - 1)

Question No: 1 (Marks: 1) - Please choose one

The purpose of earned value analysis is to

- ▶ determine how to compensate developers based on their productivity
- ▶ **provide a quantitative means of assessing software project progress (Page 102)**
- ▶ provide a qualitative means of assessing software project progress
- ▶ set the price point for a software product based on development effort

Question No: 2 (Marks: 1) - Please choose one

The RMMM plan assists the project team in developing strategy for dealing with risk. In this context, an effective strategy must consider:

- ▶ Risk avoidance
- ▶ Risk monitoring
- ▶ Risk management and contingency plan
- ▶ **All of the given choices (Page 89)**

Question No: 3 (Marks: 1) - Please choose one

The component-based development model is

- ▶ only appropriate for computer hardware design.
- ▶ not able to support the development of reusable components.
- ▶ **dependent on object technologies for support. (SE-PRESSMAN, 70)**
- ▶ not cost effective by known quantifiable software metrics.

جھوٹ رزق کو کھا جاتا ہے

Question No: 4 (Marks: 1) - Please choose one

Which is used to determine the most viable option for cost estimation when the information in the “Decision tree” is complete.

$$E = 3.2 (\text{KLOC})^{1.05}$$

Expected cost = (path probability)_i x (estimated path cost) (Page 83)

Expected cost = (path probability)_i x (estimated path cost)

Question No: 5 (Marks: 1) - Please choose one

After building the Decision Tree, following formula is used to find the expected cost for an option. Choose the correct formula:

Expected Cost= (path probability)_i * (estimated path cost) (Page 83)

Expected Cost= (path probability)_i / (estimated path cost)

Expected Cost= (path probability)_i + (estimated path cost)

Expected Cost= (path probability)_i - (estimated path cost)

Question No: 6 (Marks: 1) - Please choose one

Degree of uncertainty that the product will meet its requirements and be fit for its intended use is the

Cost risks

Schedule risks

Performance risks (Page 87)

None of the given choices

Question No: 7 (Marks: 1) - Please choose one

Every task or group of tasks should be associated with a project -----.

Schedule

Member

Manager

Milestone (Page 93)

Question No: 8 (Marks: 1) - Please choose one

Every task should be assigned to a specific team -----.

▶ **Member (Page 93)**

▶ Manager

▶ Organizer

▶ None of the given

خاموشی غصے کا بہترین علاج ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 9 (Marks: 1) - Please choose one

Which one is not the Software project planning activity carried out by the project manager for estimation?

- ▶ Software scope estimation
- ▶ Resources requirements
- ▶ Time requirements
- ▶ **Product Quality (Page 80)**

Question No: 10 (Marks: 1) - Please choose one

Configuration Item identification involves:

- ▶ Identifying the structure of the s/w system
- ▶ **Uniquely identify individual components**
- ▶ Uniquely identifying various revisions
- ▶ All of the given

Question No: 11 (Marks: 1) - Please choose one

Incomplete Configuration identification documents may result in:

- ▶ Schedule Product
- ▶ **all of the given choices are correct** [Click here for detail](#)
- ▶ Defective Product
- ▶ Higher Maintenance Costs

Question No: 12 (Marks: 1) - Please choose one

Degree of uncertainty that the product will meet its requirements and be fit for its intended use is the

- ▶ Cost risks
- ▶ Schedule risks
- ▶ **Performance risks (Page 87)**
- ▶ None of the given choices

Question No: 13 (Marks: 1) - Please choose one

Proactive risk management philosophy is also some times termed as Indiana Jones school of risk management

- ▶ true
- ▶ **false (Page 84)**

Question No: 14 (Marks: 1) - Please choose one

Software Availability can be calculated by the following equation:

- ▶ Availability = $(MTTR/MTTF) \times 100$
- ▶ Availability = $(MTBF/MTTR) \times 100$
- ▶ **Availability = $(MTTF/MTBF) \times 100$ (Page 116)**
- ▶ Availability = $(MTBF/MTTF) \times 100$

Question No: 15 (Marks: 1) - Please choose one

Phase Index can be calculated by the help of the following formula, where
 E_i – the total number of errors uncovered during the i th step in the SE process
 S_i – number of serious errors
 M_i – number of moderate errors
 T_i – number of minor errors
 PS_i – product size at the i th step
 ws, wm, wt – weighting factors for serious, moderate, and minor errors

- ▶ **$P_i = ws(S_i/E_i) + wm(M_i/E_i) + wt(T_i/E_i)$ (Page 115)**
- ▶ $P_i = ws(S_i/E_i) + wm(M_i/M_i) + wt(T_i/T)$
- ▶ $P_i = ws(S_i/E_i) + wm(M_i/M_i) + wt(T_i/E_i)$
- ▶ $P_i = ws(S_i/S) + wm(M_i/M) + wt(T_i/E)$

Question No: 16 (Marks: 1) - Please choose one

The higher the Error Index, the higher will be the Defect Removal Efficiency

- ▶ **True (Page 115)**
- ▶ False

Question No: 17 (Marks: 1) - Please choose one

The goal of quality assurance is to provide management with the data needed to determine which software engineers are producing the most defects.

- ▶ true
- ▶ **False** [Click here for detail](#)

Question No: 18 (Marks: 1) - Please choose one

From the following listed software development Model, which one is an object oriented model

- ▶ Classical life cycle model
- ▶ **Fountain model (Page 23)**
- ▶ Spiral model
- ▶ Waterfall model

Question No: 19 (Marks: 1) - Please choose one

The prototyping model of software development is

- ▶ a reasonable approach when requirements are well defined.
- ▶ **A useful approach when a customer cannot define requirements clearly.** [Click here for detail](#)
- ▶ the best approach to use for projects with large development teams.
- ▶ a risky model that rarely produces a meaningful product.

Question No: 20 (Marks: 1) - Please choose one

Which one of the following does not belong to a strategy for dealing with risk?

Risk avoidance

Security risk assessment (Page 89)

Risk monitoring

Risk management and Contingency planning

افضل انسان وہ ہے جو اپنی اصلاح کی کوشش کرتا ہے

CS605 – Solved Quizzes & MCQs

Question No: 1 of 10 (Marks: 1) - Please choose one

A _____ entity is the one which have any ----- in the problem domain without some other entity.

Strong, Role*

All of the given

Weak, Function

None of the Given (Page 47)

Ref: - A weak entity is the one which does not have any role in the problem domain without some other entity

Question No: 2 of 10 (Marks: 1) - Please choose one

ILF is a ----- identifiable group of logically control informations ----- the boundary of the application.

user, within (Page 42)

user, without

All of the given

user, along

Question No: 3 of 10 (Marks: 1) - Please choose one

Several entities are always created ----- and deleted together then this is a strong indication that they should be grouped into ----- logical file/files.

Together, Single (Page 48)

Together, Multiple

Together, Double

All of the given

Question No: 4 of 10 (Marks: 1) - Please choose one

Which of these software characteristics are used to determine the scope of a software project?

- ▶ context, lines of code, function
- ▶ context, function, communication requirements
- ▶ **information objectives, function, performance (Page 34)**
- ▶ communications requirements, performance, information objectives

Question No: 5 of 10 (Marks: 1) - Please choose one

In _____ a team is structured along a traditional hierarchy of authority

▶ **closed paradigm (Page 32)**

▶ synchronous paradigm

▶ random paradigm

▶ open paradigm

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 6 of 10 (Marks: 1) - Please choose one

Integrity means that the software should

help the users to enjoy usability

none of the given

Withstand the attack from a hacker (Page 69)

help the hacker to hack the system

Question No: 7 of 10 (Marks: 1) - Please choose one

Ambiguous requirements can be measured quantitatively

False

True (Page 71)

Question No: 8 of 10 (Marks: 1) - Please choose one

The Spiral model of software development is of Iterative nature

True Click here for detail

false

Question No: 9 of 10 (Marks: 1) - Please choose one

While performing risk analysis, the impact of risk **can not** be measured quantitatively

▶ True

▶ **False** [Click here for detail](#)

Question No: 10 of 10 (Marks: 1) - Please choose one

_____ is the measure of how many defects are removed by the quality assurance processes before the product is shipped for operation

Defect removal efficiency (Page 69)

defect density

None of These

Question No: 1 of 10 (Marks: 1) - Please choose one

For software the total numbers of requirements are equal to _____.

Functional requirements

non-functional requirements

Functional requirements and non-functional requirements (Page 71)

None of these

اطمینان قلب چاہتے ہو تو حسد سے دور رہو

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 2 of 10 (Marks: 1) - Please choose one

The _____ model is used to overcome issues related to understanding and capturing of user requirements.

Rapid Prototyping (Page 17)

Water fall

Build and Fix

None of the above

Question No: 3 of 10 (Marks: 1) - Please choose one

_____ is not the part of software development loop.

Status Quo

Problem definition

Technical development

Task set (Page 10)

Question No: 4 of 10 (Marks: 1) - Please choose one

The level 1 of CMM is known as

Managed

Defined

Initial (Page 12)

Repeatable

Question No: 5 of 10 (Marks: 1) - Please choose one

According to DeMarco a good Project Manger

should be very nervous

should be very strict in official matters

should not ignore any mistake from his team members.

should have a big heart and should ignore small mistakes (Page 31)

Question No: 6 of 10 (Marks: 1) - Please choose one

The best project team organizational model to use when tackling extremely difficult problems is the

chief programmer team model

democratic decentralized model (Page 32)

controlled decentralized model

controlled centralized model

اس سے پہلے کہ تمہیں شہوت فتنے میں ڈالے نکاح کر لو

Question No: 7 of 10 (Marks: 1) - Please choose one

Which factor is the least important when choosing the organizational structure for a software team?

degree of communication desired

predicted size of the resulting program

rigidity of the delivery date

size of the project budget (Page 32)

Question No: 8 of 10 (Marks: 1) - Please choose one

Product and process decomposition occurs simultaneously as the project plan evolves

True [click here for detail](#)

False

Question No: 9 of 10 (Marks: 1) - Please choose one

Measuring the quality and performance of a software through any mean is much better than just making a guess about it.

True

false

Question No: 10 of 10 (Marks: 1) - Please choose one

_____ is data that influences an elementary process of the application being counted.

Elementary Process

External Query

External Output

Control Information (Page 43)

Question No: 1 of 10 (Marks: 1) - Please choose one

An entity which defines many-to-many relationship between two or more entities is called

Associative Entity Type (Page 47)

Attributive Entity Type

Entity Subtype

None of these

Question No: 2 of 10 (Marks: 1) - Please choose one

Software risk impact assessment should focus on consequences affecting

performance, support, cost, schedule [click here for detail](#)

planning, resources, cost, schedule

marketability, cost, personnel

business, technology, process

Question No: 3 of 10 (Marks: 1) - Please choose one

The number of people required for a software project is determined

- ▶ **after an estimate of the development effort is made.** [Click here for detail](#)
- ▶ from an assessment of the technical complexity of the system.
- ▶ by the size of the project
- ▶ all of the given

Question No: 4 of 10 (Marks: 1) - Please choose one

Defect Removal Efficiency (DRE) can be measured by where **E** is **Errors found delivery** and **D** is **error found after delivery** (typically within the first year of operation)

- ▶ **DRE= E/(E+D)** (Page 69)
- ▶ DRE= E - (E+D)
- ▶ DRE= E * (E+D)
- ▶ None of the given

Question No: 5 of 10 (Marks: 1) - Please choose one

In function point analysis technique EO stands for

- ▶ Export operation
- ▶ Export output
- ▶ **External output** (Page 49)
- ▶ None of these

Question No: 6 of 10 (Marks: 1) - Please choose one

In _____ a team is structured loosely and depends on individual initiative of the team members

- ▶ **Random paradigm** (Page 32)
- ▶ closed paradigm
- ▶ synchronous paradigm
- ▶ open paradigm

Question No: 7 of 10 (Marks: 1) - Please choose one

_____ is the first stage of waterfall lifecycle model

- ▶ **Requirement definition** (Page 15)
- ▶ Operation
- ▶ Unit testing
- ▶ Implementation

ہر چیز کی ایک پہچان ہوتی ہے اور عقلمند کی پہچان غور و فکر کرنا ہے اور غور و فکر کی پہچان خاموشی ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 8 of 10 (Marks: 1) - Please choose one

_____ is not the management part of software development activities.

- ▶ Coding p7
- ▶ Configuration Management
- ▶ Quality Assurance
- ▶ **Project planning (Page 7) not sure**

Question No: 9 of 10 (Marks: 1) - Please choose one

Software process and product quality are controlled at _____.

level 1

initial level

level-4 (Page 12)

level-3

Question No: 10 of 10 (Marks: 1) - Please choose one

QA is an activity performed _____ of the software development.

at the initial stage

at the final stage

throughout (Page 25)

at the middle

Question No: 1 of 10 (Marks: 1) - Please choose one

_____ the ability to encourage people to create and feel creative.

Organization

Motivation

Innovation (Page 30)

Managerial Identity

Question No: 2 of 10 (Marks: 1) - Please choose one

Caper Jones divided software related activities into _____ different categories.

35

40

20

25 (Page 7)

اپنی مرضی اور اللہ کی مرضی میں فرق کا نام غم ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 3 of 10 (Marks: 1) - Please choose one

By default every organization is working at _____

level 1 (Page 12)

level 0

level 3

level 4

Question No: 4 of 10 (Marks: 1) - Please choose one

Software Engineering is the set of _____ and _____ to develop software.

languages , processes

classes ,objects

processes, tools (Page 4)

tools, objects

Question No: 5 of 10 (Marks: 1) - Please choose one

Software has very close relationship with _____

Biology

Physics

chemistry

Economics (Page 5)

Question No: 6 of 10 (Marks: 1) - Please choose one

COCOMO is the classic LOC cost-estimation formula and used thousand delivered source instructions (KDSI) as his _____ of size.

LOC

milestone

unit [Click here for detail](#)

none of given

Question No: 7 of 10 (Marks: 1) - Please choose one

_____ deadline is one of the reasons of project failure.

achievable

feasible

unrealistic (Page 29)

realistic

وہ لوگ مبارک ہیں جو الفاظ سے نصیحت نہیں کرتے بلکہ عمل سے کرتے ہیں

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 8 of 10 (Marks: 1) - Please choose one
_____ model is very sensitive to the risk.

Spiral (Page 21)

waterfall
incremental
RAD

Question No: 9 of 10 (Marks: 1) - Please choose one
_____ is the ability to encourage technical people to produce to their best.

improvement
motivation (Page 30)
innovation
creation

Question No: 10 of 10 (Marks: 1) - Please choose one
Caper Jones is famous researcher in the field of _____ who made a company named Software Productivity Research

Biology
Chemistry
Mathematics
Software Engineering (Page 7)

Question No: 1 of 10 (Marks: 1) - Please choose one
Reel has defined _____ steps process to improve the chances of success.

3
4
5 (Page 35)
8

Question No: 2 of 10 (Marks: 1) - Please choose one
In _____ model user feedback is received very quickly because product is delivered in small versions.

waterfall
spiral
incremental (Page 18)
object oriented

خدا کے سوا کسی سے امید مت رکھو

Question No:3 of 10 (Marks: 1) - Please choose one
_____ approach is better for the team spirits point of view.

Controlled Decentralized
Informal, interpersonal procedures
Controlled Centralized

Democratic Decentralized (Page 32)

Question No: 4 of 10 (Marks: 1) - Please choose one
Construction activities are directly related to the _____ of the software.

Installation
risk analysis
development (Page 8)
debugging

Question No: 5 of 10 (Marks: 1) - Please choose one
Continuous process improvement is enabled by _____ feedback from the _____

analytical , user
mathematical, user
logical, process
qualitative , process (Page 12)

Question No: 6 of 10 (Marks: 1) - Please choose one
The major activities related to software construction are

requirement gathering, design development, coding and testing (Page 8)
installation and training
quality assurance, configuration and planning
implementation and management

Question No: 7 of 10 (Marks: 1) - Please choose one
If requirements are uncertain, _____ will be a suitable.

Incremental
RAD
Waterfall (Page 34)
Prototype

ہر کسی کی روٹی نہ کھا بلکہ ہر شخص کو اپنی روٹی کھلا

Question No: 8 of 10 (Marks: 1) - Please choose one

Capability maturity model (CMM) is used to judge the _____ model of an organization.

- Efficiency
- Performance
- Productivity

Maturity (Page 12)

Question No: 9 of 10 (Marks: 1) - Please choose one

After passing through all the stages of the software development when we deploy the new system at the user side that stage is called _____

- Integration
- Development
- Installation

Status quo (Page 11)

Question No: 10 of 10 (Marks: 1) - Please choose one

Defects per function points are a ___ metric.

One (Page 65)

- Two
- Three
- Four

Question No: 1 of 10 (Marks: 1) - Please choose one

In Capability Maturity Model (CMM), _____ performs optimization.

- level1
- level 2
- level3

level5 (Page 12)

Question No: 2 of 10 (Marks: 1) - Please choose one

If a software developer is going to develop software for a nuclear reactor which factor(s) will be more important?

- cost effective
- performance
- reliability

both b and c (Page 6)

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 3 of 10 (Marks: 1) - Please choose one
Capability Maturity Model (CMM) has _____ levels.

4

5 (Page 12)

3

2

Question No: 4 of 10 (Marks: 1) - Please choose one

Fred Brook is a famous software engineer who wrote a great book related to software engineering named _____.

A Mythical Man Month (Page 7)

A Mathematical Man
Software Methodologies
None of the above

Question No: 5 of 10 (Marks: 1) - Please choose one

W5HH Principle consists of seven _____.

Answers

Questions (Page 35)

points

steps

Question No: 6 of 10 (Marks: 1) - Please choose one

In Rational Unified Process (RUP) horizontal dimension represents the _____ aspect of the process.

Dynamic (Page 24)

Static

Both

Question No: 7 of 10 (Marks: 1) - Please choose one

There are two very different approaches to cost estimation. The older approach is called LOC estimation, since it is on initially estimating the number of lines of code that will need to be developed for the project.

Based [Click here for detail](#)

Good

None of given

Bad

دنیا حقیر نظر آتی ہے، جب غم یا خوشی کی انتہاء ہو جائے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 8 of 10 (Marks: 1) - Please choose one

Task coordinators are appointed for short period of time in _____ organization team structure.

controlled centralized

democratic decentralized (Page 32)

controlled decentralized

formal impersonal approach

Question No: 9 of 10 (Marks: 1) - Please choose one

_____ model is opposite to waterfall model.

Spiral

Prototype (Click here for detail)

Synchronize and Stabilize

Incremental

Question No: 10 of 10 (Marks: 1) - Please choose one

MOI model of leadership was developed by _____

Weinberg (Page 30)

Newton

Don Carlos

Question No: 1 of 10 (Marks: 1) - Please choose one

Each process define certian deliverables known as the _____ .

work products (Page 12)

softwares

final products

items

Question No: 2 of 10 (Marks: 1) - Please choose one

_____ procedures include QA activities, design and code review, and status meetings.

Formal, interpersonal (Page 33)

Informal, interpersonal

Controlled centralized

Formal, impersonal

Question No: 3 of 10 (Marks: 1) - Please choose one

Miscommunication among project staff is one of the reasons of the failure of the software project

True (Page 28)

False

Question No: 4 of 10 (Marks: 1) - Please choose one

According to DeMarco a good team leader should have only _____ characteristics.

Four (Page 31)

- five
- three
- seven

Question No: 5 of 10 (Marks: 1) - Please choose one

Synchronize and Stabilize model is adopted by _____.

Apple Macintosh

Microsoft (Page 19)

Sun Java System

Netsol Technologies

Question No: 6 of 10 (Marks: 1) - Please choose one

Extreme programming model is very effective when requirement are very _____ and the project scope is very _____.

Changing, limited (Page 23)

- Changing, unlimited
- None of these

Question No: 7 of 10 (Marks: 1) - Please choose one

_____ model is a haphazard type of software development activity.

Water fall

Incremental

Prototyping

Build and Fix

[Click here for detail](#)

Question No: 8 of 10 (Marks: 1) - Please choose one

Water fall model is a document driven model.

True (Page 16)

False

Question No: 9 of 10 (Marks: 1) - Please choose one

_____ paradigm, structures a team loosely and depends on individual initiative of the team members

Closed

Random (Page 32)

Open

Synchronous

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 10 of 10 (Marks: 1) - Please choose one

Spiral model has _____ dimensions.

2 (Page 20)

3

4

None of the above

Question No: 1 of 10 (Marks: 1) - Please choose one

The _____ model of software development is a good approach when core product is required quickly.

Incremental [click here for detail](#)

Linear Sequential

Prototyping

None of the above

Question No: 2 of 10 (Marks: 1) - Please choose one

In _____ team organization, there is no permanent leader rather there are task coordinators.

Controlled Decentralized

Democratic Decentralized (Page 32)

Controlled Centralized

Synchronous paradigm

Question No: 3 of 10 (Marks: 1) - Please choose one

_____ is an Object Oriented model.

Spiral

Water fall

Incremental

Fountain (Page 23)

Question No: 4 of 10 (Marks: 1) - Please choose one

_____ model is developed keeping in mind the element of risk in the development of software

Spiral (Page 20)

RAD

Incremental

Synchronize and Stabilize

بہادر آدمی ایک مرتبہ جبکہ بزدل آدمی کئی مرتبہ مرتا ہے

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 5 of 10 (Marks: 1) - Please choose one

Vision phase in a software process focuses on _____.

What

Why (Page 14)

How

Change

Question No: 6 of 10 (Marks: 1) - Please choose one

Rapid application development is another form of _____.

Incremental model (Page 19)

Prototyping model

Linear Sequential model

None of the above

Question No: 7 of 10 (Marks: 1) - Please choose one

_____ phase in a software process focuses on change.

Vision

Maintenance (Page 14)

Definition

Development

Question No: 8 of 10 (Marks: 1) - Please choose one

_____ model is adopted by Microsoft

RAD

Build and Fix

Spiral

Synchronize and Stabilize (Page 19)

Question No: 9 of 10 (Marks: 1) - Please choose one

_____ teams generate more and better solutions than individuals and are most useful for complex problems

Decentralized (Page 32)

Centralized

Question No: 10 of 10 (Marks: 1) - Please choose one

The _____ model of software development is a good approach when core product is required quickly.

Incremental [click here for detail](#)

Linear Sequential

Prototyping

None of the above

Question No: 1 of 10 (Marks: 1) - Please choose one

The _____ model of software development is a useful approach when a customer cannot define requirements clearly.

Prototyping [click here for detail](#)

Water fall

RAD

Build and Fix

Question No: 2 of 10 (Marks: 1) - Please choose one

_____ model has a major drawback in that the delivered product may not fulfill the customer's requirements.

Water fall (Page 17)

Build and Fix

Prototyping

Integrated water fall and prototyping

Question No: 3 of 10 (Marks: 1) - Please choose one

MOI model of leadership stands for

Motivation , Operationalize ,Integration

Misunderstanding , Object, Ideas

Motivation , Organization , Innovation (Page 30)

Miscommunication , Organization, Invention

Question No: 4 of 10 (Marks: 1) - Please choose one

_____ activity is performed throughout software production

Maintenance

Development

Analysis

Quality Assurance (Page 25)

Question No: 5 of 10 (Marks: 1) - Please choose one

In _____ model the product is developed without any proper design and specifications.

Water fall

Build and Fix (Page 15)

Prototyping

None of the above

Question No: 6 of 10 (Marks: 1) - Please choose one
software project management focuses on the four P's: These are

People , Product , Process , Problem

People , Product , Process , Project (Page 29)

People , Passion , Process , Project

People , Passion , Planning , Project

Question No: 7 of 10 (Marks: 1) - Please choose one

The _____ dimension in rational unified process model represents the dynamic aspect of the process.

Horizontal (Page 24)

Angular

Vertical

Regular

Question No: 8 of 10 (Marks: 1) - Please choose one

_____ dimension of Spiral model represents the cumulative cost to date

Radial (Page 20)

Angular

Horizontal

circular

Question No: 9 of 10 (Marks: 1) - Please choose one

The important feature of extreme programming is the concept of _____.

Feedback

Risk assessment

Pair programming (Page 23)

Requirement elicitation

Question No: 10 of 10 (Marks: 1) - Please choose one

In software development _____ is performed at the end of each phase.

Verification (Page 26)

Validation

Question No: 1 of 10 (Marks: 1) - Please choose one

Unrealistic deadline is NOT one of the reasons of project failure.

True

False (Page 29)

Question No: 2 of 10 (Marks: 1) - Please choose one

Software _____ relates individual software measures to provide a normalized view.

Measure

Metric (Page 65)

Plan

Attribute

Question No: 3 of 10 (Marks: 1) - Please choose one

Quality can be measured if we measure the _____ of the product.

correctness

maintainability

integrity and usability

All of the given options (Page 68)

Question No: 4 of 10 (Marks: 1) - Please choose one

In case some software, it is extremely easy to operate with intuitive interface conforming to most excellent GUI practices of the industry. It needs very little training -----

Less than one hour [click here for detail](#)

Less than two hour

Less than five hour

Less than ten hour

Question No: 5 of 10 (Marks: 1) - Please choose one

Which one of the following is NOT a useful indicator of software quality?

Correctness

Code size (Page 67)

Maintainability

Integrity

Usability

Question No: 6 of 10 (Marks: 1) - Please choose one

The software reengineering process model includes restructuring activities for which of the following work items?

Code

Documentation

Data

All of the given options [click here for detail](#)

Question No: 7 of 10 (Marks: 1) - Please choose one

One graphical technique for determining whether a process exhibits out-of-control change behavior is a.

Control chart (Page 75)

Fishbone diagram

Pareto diagram

Process diagram

Question No: 8 of 10 (Marks: 1) - Please choose one

Defect per unit function point is a

Measure

Metric (Page 65)

Measurement

None of the above

Question No: 9 of 10 (Marks: 1) - Please choose one

Chart that is used to develop the individual control chart is called statistical control techniques.

Yes (Page 75)

No

Question No: 10 of 10 (Marks: 1) - Please choose one

_____ Is the set of tools and techniques used to develop software?

Engineering

Software Engineering (Page 4)

Question No: 1 of 10 (Marks: 1) - Please choose one

The characteristics of well developed software are_____

Reliability

User friendliness

Cost effective

All of the above (Page 5)

Question No: 2 of 10 (Marks: 1) - Please choose one

SE is a Balancing act

Yes (Page 5)

No

اچھائی کرنے کے لئے ہمیشہ کسی بہانے کی تلاش میں رہو

Question No: 3 of 10 (Marks: 1) - Please choose one

_____ Is the process of balancing among different characteristics of software?

Software Testing

Software Development (Page 6)

Software Management

Software Risk Analysis

Question No: 4 of 10 (Marks: 1) - Please choose one

_____ Is directly related to development of the software

Management

Construction (Page 8)

Coding

None of the above

Question No: 5 of 10 (Marks: 1) - Please choose one

_____ Basic project management processes established to track cost, schedule and functionality. Has the necessary process discipline in place to repeat earlier successes Related to projects

Level 1

Level 2 (Page 12)

Level 3

Level 4

Question No: 6 of 10 (Marks: 1) - Please choose one

Change control is not necessary if a development group is making use of an automated project database tool.

True

False [Click here for detail](#)

Question No: 7 of 10 (Marks: 1) - Please choose one

Which one of the following is the 4th level of the SEI Capability Maturity Model?

Initial or ad hoc

Optimizing

Managed (Page 12)

Defined

Repeatable

Question No: 8 of 10 (Marks: 1) - Please choose one

Doubling the size of your software project team is guaranteed to cut project completion time in half.

True

False [Click here for detail](#)

Question No: 9 of 10 (Marks: 1) - Please choose one

Which maturity level is characterised by:

"Basic project management processes established to track cost, schedule and functionality. Has the necessary process discipline in place to repeat earlier successes on projects with similar applications".

Level 1

Level 2 (Page 12)

Level 3

Level 4

Question No: 10 of 10 (Marks: 1) - Please choose one

Defect prevention is defined as:

Avoiding defect insertion

Avoiding defect insertion, but fixing errors when reported

Finding and fixing errors after insertion [click here for detail](#)

Finding and fixing errors after release

Not reporting errors

Question No: 1 of 10 (Marks: 1) - Please choose one

What activity does a software project manager need to perform to minimize the risk of software failure?

double the project team size

request a large budget

allow absolutely no schedule slippage

define milestones and track progress [Click here for detail](#)

Request 50% more time than estimated

Question No: 2 of 10 (Marks: 1) - Please choose one

The linear sequential model of software development is also known as the

Classical life cycle model [click here for detail](#)

Fountain model

Spiral model

Chaos model

Question No: 3 of 10 (Marks: 1) - Please choose one

Which of these benefits can be achieved when software is restructured?

higher quality programs

reduced maintenance effort

software easier to test

all of the given options [Click here for detail](#)

Question No: 4 of 10 (Marks: 1) - Please choose one

_____ QA technique involves typically execution of software and the observation of program behavior or outcome.

Inspection

Walkthroughs

Testing [Click here for detail](#)

Reviews

Question No: 5 of 10 (Marks: 1) - Please choose one

_____ testing verifies the correct handling of the external functions provided by the software

Black box [Click here for detail](#)

White box

Gray box

Question No: 6 of 10 (Marks: 1) - Please choose one

The quality expectations of a user are that a software system performs useful functions that

fit user needs

perform correctly over repeated period of time

both a and b [Click here for detail](#)

Question No: 7 of 10 (Marks: 1) - Please choose one

Incomplete requirements gathering can result in the _____ of the project

failure (Page 12)

reusability

success

easy maintenance

Question No: 8 of 10 (Marks: 1) - Please choose one

The root causes of project failure are

lack of user input

incomplete requirement and specification

creep requirement [Click here for detail](#)

all of the given options

کامیاب و کامران زندگی یہی ہے کہ جہاں رہو جس حال میں رہو خوش رہو

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus: - Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 9 of 10 (Marks: 1) - Please choose one

_____ is a bad smell when a change requires lots of little changes in a lot of different classes.

Short gun surgery [Click here for detail](#)

Duplicated code

Large class

Lazy class

Question No: 10 of 10 (Marks: 1) - Please choose one

If you find the code such that one type of change requires changing one subset of method, another type of change requires changing another subset then it is a symptom of _____

Divergent change [Click here for detail](#)

Duplicated code

Large class

Lazy class

Question No: 1 of 10 (Marks: 1) - Please choose one

Following are the reasons for project failure except

Requirements are not clear

The name of the software (Page 28)

Processes have not been established in the organization

Un-realistic deadline

Question No: 2 of 10 (Marks: 1) - Please choose one

Main characteristic of _____ model is risk analysis and management

Waterfall

RAD

Spiral [Click here for detail](#)

Incremental

اطمینان قلب چاہتے ہو تو حسد سے دور رہو

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Lecture No. 23

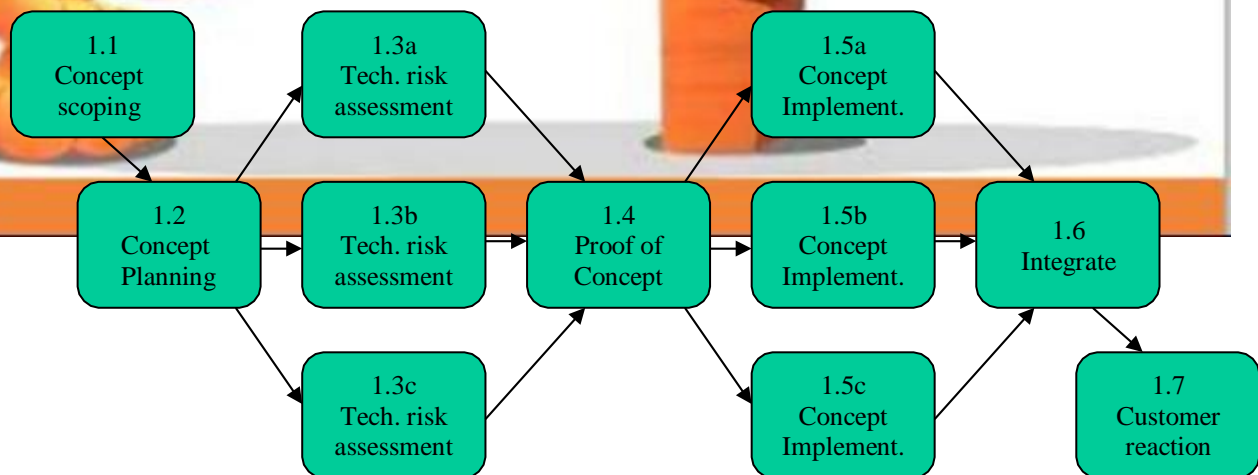
Task Network

Let us now apply the principles studied above to develop a plan for a concept development project. In this case the potential for new technology or new business idea is explored. Since it is a concept development project, the applicability is not certain but it appears to be useful and hence needs to be explored. Major tasks include:

- Concept scoping
- Preliminary concept planning
- Technology risk assessment
- Proof of concept
- Concept implementation
- Customer reaction to concept

Defining a Task Network

Once the tasks have been identified, we need to develop a task network to determine the sequence in which these activities need to be performed. This will ultimately lead to the time required to complete the project (to be discussed later). The following diagram shows the task network for the above project.



Lecture No. 24

Scheduling

Once we have the task network, we are now ready to prepare a schedule for the project. For this we use two techniques known as:

- Program evaluation and review techniques (PERT)
- Critical Path Method (CPM)

These are quantitative tools that allow the software planner to determine the critical path – the chain of tasks that determines the duration of the project and establish most likely time estimates for individual tasks by applying statistical models. They also help the planner to calculate boundary times that define a time window for a particular task.

The boundary time defines the following parameters for a project:

- The earliest time that a task can begin when all preceding tasks are completed in the shortest possible time
- The latest time for task initiation before the minimum project completion time is delayed
- The earliest finish
- The latest finish
- The total float – the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path is maintained on schedule

In order to use the PERT and CPM, the following is required:

- A decomposition of product function
- A selection of appropriate process model and task set
- Decomposition of tasks – also known as the work breakdown structure (WBS)
- Estimation of effort
- Interdependencies

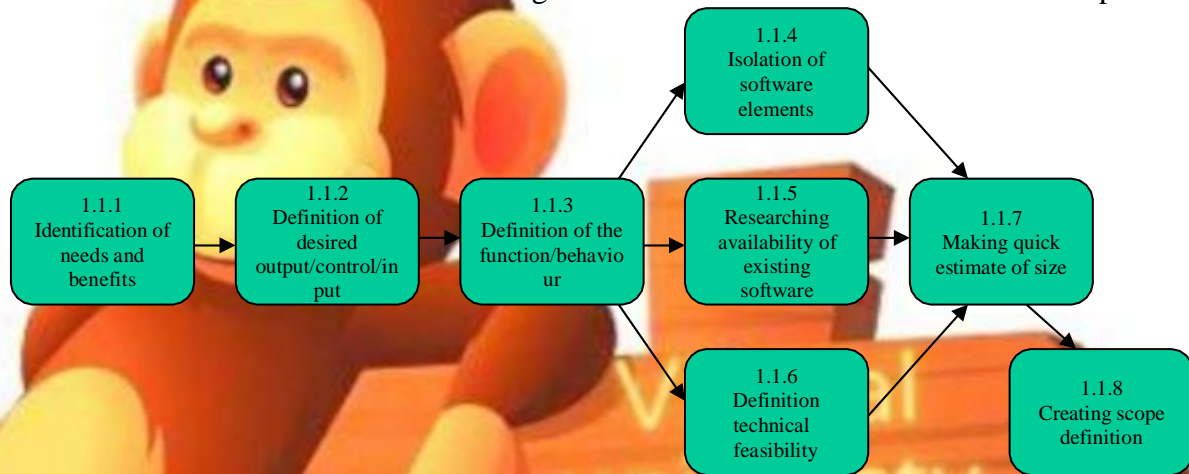
Timeline Chart

To develop the schedule for a project, time required for each activity in the Task Network is estimated. This analysis and decomposition leads to the development of a Timeline or Gantt Chart for the project which portrays the schedule for the project. As an example, let us assume that Concept Scoping (the first task in the above list) is further subdivided into the following sub-tasks with the associated estimated time requirements:

1. Identification of needs and benefits (3 days)
2. Definition of desired output/control/input (7 days)
3. Definition of the function/behaviour (6 days)
4. Isolation of software elements (1 day)
5. Researching availability of existing software (2 days)
6. Definition technical feasibility (4 days)
7. Making quick estimate of size (1 day)

8. Creating scope definition (2 days)

We also assume that the following task network for this was developed.



This is now converted in the following schedule in the form of a Gantt Chart. Note that, the concept of boundary time allows us to schedule Task Numbers 1.1.4 and 1.1.5 anywhere along Task Number 1.1.3. The actual time is determined by the project manager is based upon the availability of resources and other constraints. Each task is further subdivided in sub-tasks in the same manner until the schedule for the complete project is determined.

		Week1	Week2	Week3	Week4	Week5
1.1	Concept Scoping	[Solid black bar]				
1.1.1	Identify needs and benefits	[Solid black bar]				
1.1.2	Define desired output/control/input		[Solid black bar]			
1.1.3	Define the function/behaviour			[Solid black bar]		
1.1.4	Isolation software elements				[Solid black bar]	
1.1.5	Research availability of existing software				[Solid black bar]	
1.1.6	Define technical feasibility					[Solid black bar]
1.1.7	Make quick estimate of size					[Solid black bar]
1.1.8	Create scope definition					[Solid black bar]

Tracking a Schedule

Lecture No. 25

Project Tracking

A schedule is meaningless if it is not followed and tracked. Tasks and milestones defined in a project schedule must be tracked and controlled as project proceeds. Tracking methods include:

- Periodic project status meetings
- Evaluating the results of all reviews
- Determine whether project milestones have been accomplished by the scheduled date
- Comparing actual start date to planned start date
- Informal meetings with the practitioners
- Using earned value analysis
- Error tracking

The last two techniques are discussed in further detail in the following paragraphs:

Earned Value Analysis

Earned Value Analysis or EVA is a quantitative technique for assessing the progress of a project. The earned value system provides a common value scale for every software task, regardless of the type of work being performed. The total hours to do the whole project are estimated, and every task is given an earned value based on the estimated percentage of the total. In order to do the EVA, the budgeted cost of work schedule (BCWS) is determined as follows:

Let $BCWS_i = \text{effort (person-days etc) for task } i$

BCWS is then the Progress so far – add all $BCWS_i$ so far.

Now

$$BAC = \text{budget at completion} = \sum BCWS_i$$

Now if BCWP is the Budgeted Cost of Work Perform, then

Schedule performance index	$SPI = BCWP/BCWS$
Schedule variance	$SV = BCWP - BCWS$

SPI close to 1 indicates efficient execution.

Similarly

$$\text{Percent scheduled for completion} = BCWS/BAC$$

$$\text{Percent complete} = BCWP/BAC$$

$$\text{Actual cost work performed ACWP}$$

$$\text{Cost performance index CPI} = BCWP/ACWP$$

$$\text{Cost variance CV} = BCWP - ACWP$$

Now, value of CPI close to 1 means project is within its defined budget.

Therefore, by using SPI and CPI we estimate how the project is progressing. If we have these values close to 1, it means that we have had good estimates and the project is under control.



Lecture No. 26

Error Tracking

Error tracking can also be used to estimate the progress of the project. In this case we track errors in work products (requirement specifications, design documents, source code etc) to assess the status of a project. The process works as follows:

We collect error related metrics over many projects and determine our defect removal efficiency in the following manner:

Defect removal efficiency, $DRE = E / (E+D)$, where

- E – errors found before shipment
- D – errors found during operation

It provides a strong indication of the effectiveness of the quality assurance activities.

Now let us assume that we have collected the following errors and defect data over the last 24 months:

- Errors per requirement specification page – E_{req}
- Error per component – design level – E_{design}
- Errors per component – code level – E_{code}
- DRE – requirement analysis
- DRE – architectural design
- DRE – coding

We now record the number of errors found during each SE step and calculate current values for E_{req} , E_{design} , and E_{code} . These values are then compared to averages of past projects. If the current results vary more than 20% from average, there may be cause for concern and there is certainly cause for investigation.

Example

- E_{req} for the current project = 2.1
- Organizational average = 3.6
 - Two possibilities
 - The team has done an outstanding job
 - The team has been lax in its review approach
 - If the second scenario appears likely
 - Build additional design time

This can also be used to better target review and/or testing resources in the following manner:

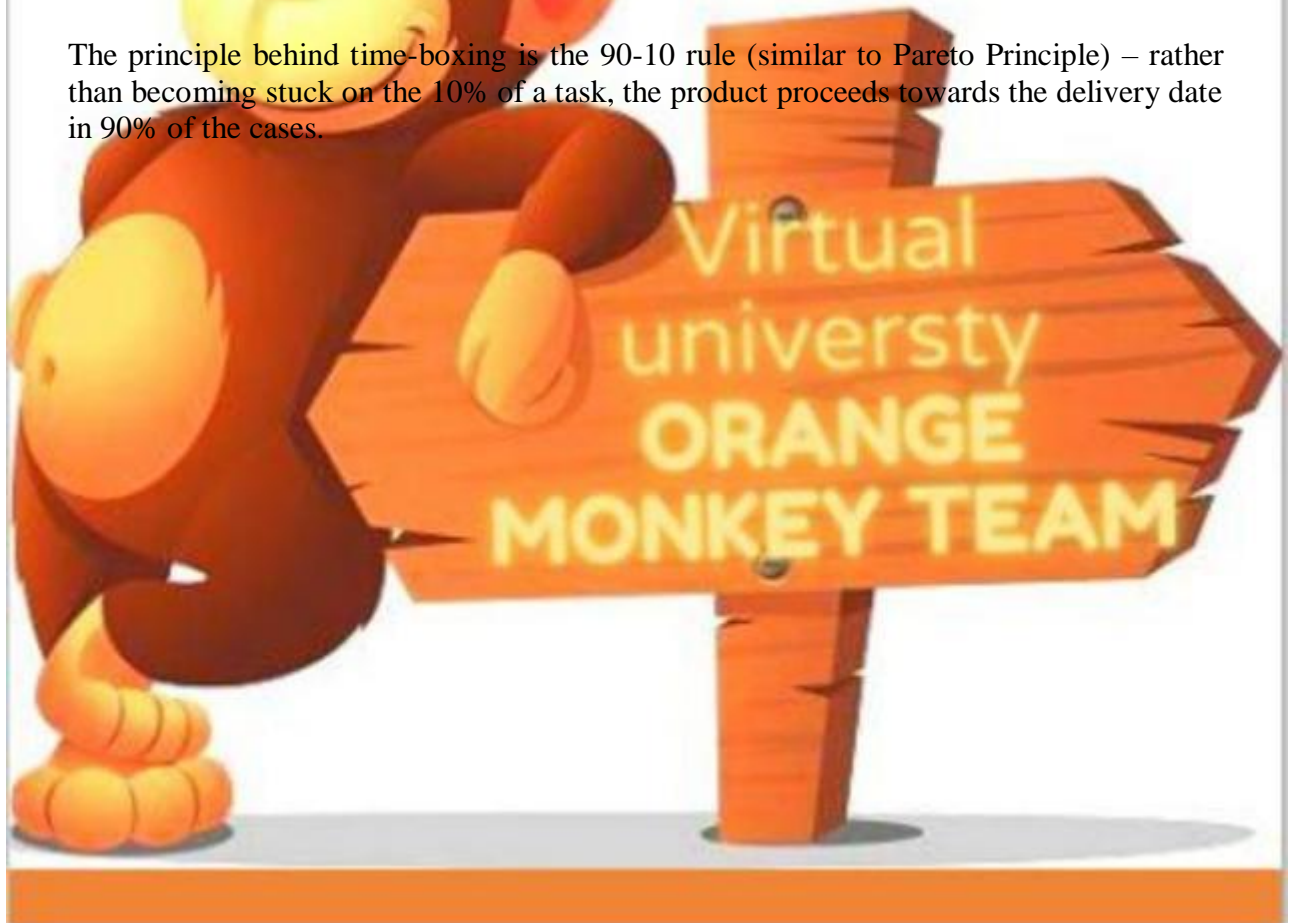
- 120 components
- 32 exhibit $E_{design} > 1.2$ average
- Adjust code review resources accordingly

Time Boxing

Time-boxing is used in severe deadline pressure. It is a use incremental strategy where tasks associated with each increment are time-boxed in the following manner:

- Schedule for each task is adjusted by working backward from the delivery date.
- A box is put around each task
- When a task hits the boundary of the box, work stops and next task begins

The principle behind time-boxing is the 90-10 rule (similar to Pareto Principle) – rather than becoming stuck on the 10% of a task, the product proceeds towards the delivery date in 90% of the cases.



Lecture No. 27

Software Quality Assurance

Quality cannot be assured without first understanding its nature and characteristics. So the first question one has to ask is: what is quality?

Software quality is defined as conformance to explicitly stated functional and non-functional requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

This definition emphasizes upon three important points:

- Software requirements are the foundation from which quality is measured. Lack of conformance is lack of quality
- Specified standards define a set of development criteria that guide the manner in which software is engineered. If the criteria are not followed, lack of quality will almost surely result.
- A set of implicit requirements often goes unmentioned (ease of use, good maintainability etc.)

Another very important question is: Do you need to worry about it after the code has been generated? In fact, SQA is an umbrella activity that is applied throughout the software process.

Also, do we care about internal quality or the external quality? And finally, is there a relationship between internal and external qualities? That is, does internal quality translate in external quality?

In the literature, quality has been defined through in many different manners. One group believes that the quality has measurable characteristic such as cyclomatic complexity, cohesion, and coupling.

We can then talk about quality from different aspects. Quality of design tries to determine the quality of design related documents including requirements, specifications, and design. Quality of conformance looks at the implementation and if it follows the design then the resulting system meets its goals then conformance quality is high.

Are there any other issues that need to be considered? Glass defines quality as a measure of user satisfaction which is defined by

compliant product + good quality + delivery within budget and schedule

DeMarco defines product quality as a function of how much it changes the world for the better.

So, there are many different way to look at the quality.

Quality Assurance

Goal of quality assurance is to provide the management with the necessary data to be informed about product quality. It consists of auditing and reporting functions of

management. If data provided through QA identifies problems, the management deploys the necessary resources to fix it and hence achieves desired quality control.

Cost of quality

A very significant question is: does quality assurance add any value. That is, is worth spending a lot of money in quality assurance practices? In order to understand the impact of quality assurance practices, we have to understand the cost of quality (or lack thereof) in a system.

Quality has a direct and indirect cost in the form of cost of prevention, appraisal, and failure.

If we try to prevent problems, obviously we will have to incur cost. This cost includes:

- Quality planning
- Formal technical reviews
- Test equipment
- Training

We will discuss these in more detail in the later sections.

The cost of appraisal includes activities to gain insight into the product condition. It involves in-process and inter-process inspection and testing.

And finally, failure cost. Failure cost has two components: internal failure cost and external failure cost. Internal failure cost requires rework, repair, and failure mode analysis. On the other hand, external failure cost involves cost for complaint resolution, product return and replacement, help-line support, warranty work, and law suits.

It is trivial to see that cost increases as we go from prevention to detection to internal failure to external failure. This is demonstrated with the help of the following example:

Let us assume that a total of 7053 hours were spent inspecting 200,000 lines of code with the result that 3112 potential defects were prevented. Assuming a programmer cost of \$40 per hour, the total cost of preventing 3112 defects was \$382,120, or roughly \$91 per defect.

Let us now compare these numbers to the cost of defect removal once the product has been shipped to the customer. Suppose that there had been no inspections, and the programmers had been extra careful and only one defect one 1000 lines escaped into the product shipment. That would mean that 200 defects would still have to be fixed in the field. As an estimated cost of \$25000 per fix, the cost would be \$5 Million or approximately 18 times more expensive than the total cost of defect prevention

That means, quality translates to cost savings and an improved bottom line.

SQA Activities

There are two different groups involved in SQA related activities:

- Software engineers who do the technical work
- SQA group who is responsible for QA planning, oversight, record keeping, analysis, and reporting

Software engineers address quality by applying solid technical methods and measures, conducting formal and technical reviews, and performing well planned software testing. The SQA group assists the software team in achieving a high quality product.

SQA Group Activities

An SQA plan is developed for the project during project planning and is reviewed by all stake holders. The plan includes the identification of:

- Evaluations to be performed
- Audits and reviewed to be performed
- Standards that are applicable to the project
- Procedures for error reporting and tracking
- Documents to be produced by the SQA group
- Amount of feedback provided to the software project team

The group participates in the development of the project's software process description. The software team selects the process and SQA group reviews the process description for compliance with the organizational policies, internal software standards, externally imposed standards, and other parts of the software project plan.

The SQA group also reviews software engineering activities to verify compliance with the defined software process. It identifies, documents, and tracks deviations from the process and verifies that the corrections have been made. In addition, it audits designated software work products to verify compliance with those defined as part of the software process. It, reviews selected work products, identifies, documents, and tracks deviations; verifies that corrections have been made; and reports the results of its work to the project manager.

The basis purpose is to ensure that deviations in software work and work products are documented and handled according to documented procedures. These deviations may be encountered in the project plan, process description, applicable standards, or technical work products. The group records any non-compliance and reports to senior management and non-compliant items are recorded and tracked until they are resolved.

Another very important role of the group is to coordinate the control and management of change and help to collect and analyze software metrics.

Quality Control

The next question that we need to ask is, once we have defined how to assess quality, how are we going to make sure that our processes deliver the product with the desired quality. That is, how are we going to control the quality of the product?

The basic principle of quality control is to control the variation as variation control is the heart of quality control. It includes resource and time estimation, test coverage, variation in number of bugs, and variation in support.

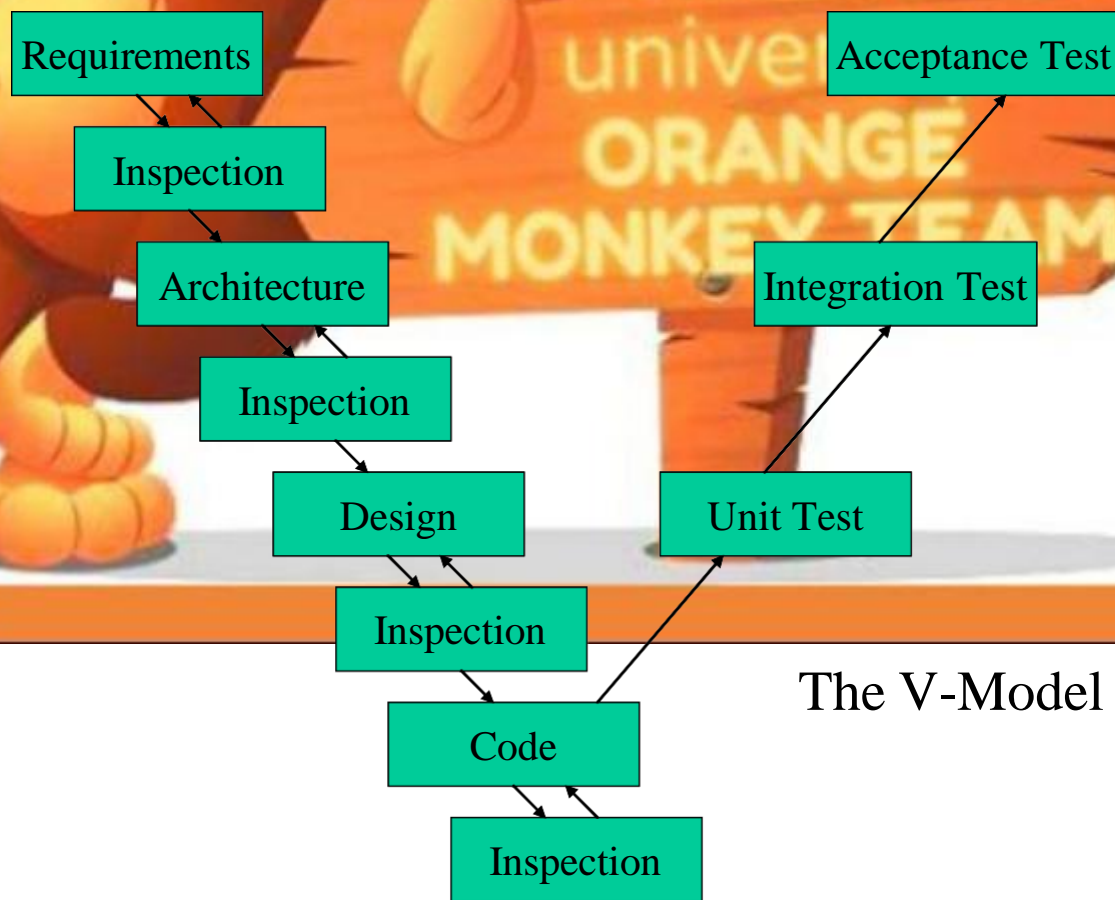
From one project to another we want to minimize the predicted resources needed to complete a project and calendar time. This involves a series of inspection, reviews, and tests and includes feedback loop. So quality control is a combination of measurement and feedback and combination of automated tools and manual interaction.

Lecture No. 28

Software Reviews

Software reviews are the filter for the software engineering process. They are applied at various different points and serve to uncover errors that can be removed and help to purify the software engineering activities.

In this context it is useful to look at the “V-model” of software development. This model emphasizes that SQA is a function performed at all stages of software development life cycle. At the initial stages (requirement, architecture, design, code), it is achieved through activities known as Formal Technical Reviews or FTR. At the later stages (integration and acceptance), testing comes into picture.



The V-Model

Importance of reviews

Technical work needs reviewing for the same reason that pencils need erasers: To err is human. The second reason that we need technical reviews is although that people are good at catching errors, large class of errors escape the originator more easily than they escape anyone else.

Freedman defines a review – any review – as a way of using the diversity of a group of people to:

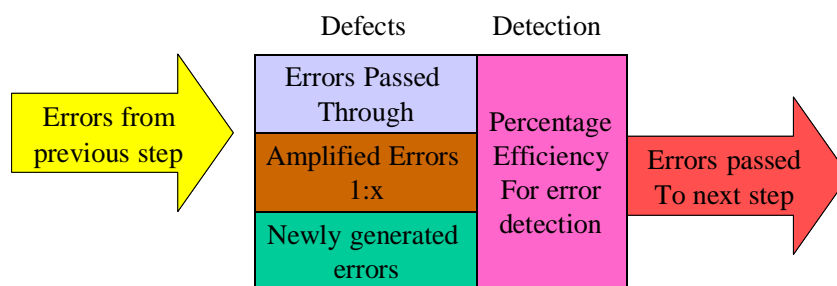
- Point out needed improvements in the product of a single person or team
- Confirm those parts of a product in which improvement is either not desired or not needed
- Achieve technical work of more uniform, or at least more predictable, quality than can be achieved without reviews, in order to make technical work more manageable.

Reviews help the development team in improving the defect removal efficiency and hence play an important role in the development of a high-quality product.

Types of Reviews

There are many types of reviews. In general they can be categorized into two main categories namely informal and formal technical reviews. Formal Technical reviews are sometimes called as walkthroughs or inspections. They are the most effective filter from QA standpoint. To understand the significance of these reviews, let us look at the defect amplification model shown below.

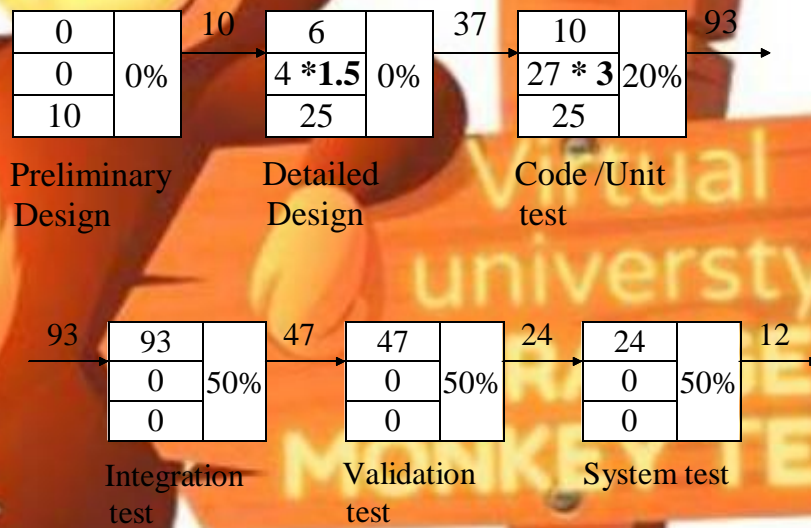
This model depicts that each development step inherits certain errors from the previous step. Some of these errors are just passed through to the next step while some are worked on and hence are amplified with a ratio of 1:x. In addition, each step may also generate some new errors. If each step has some mechanism for error detection, some of these errors may be detected and removed and the rest are passed on to the next step.



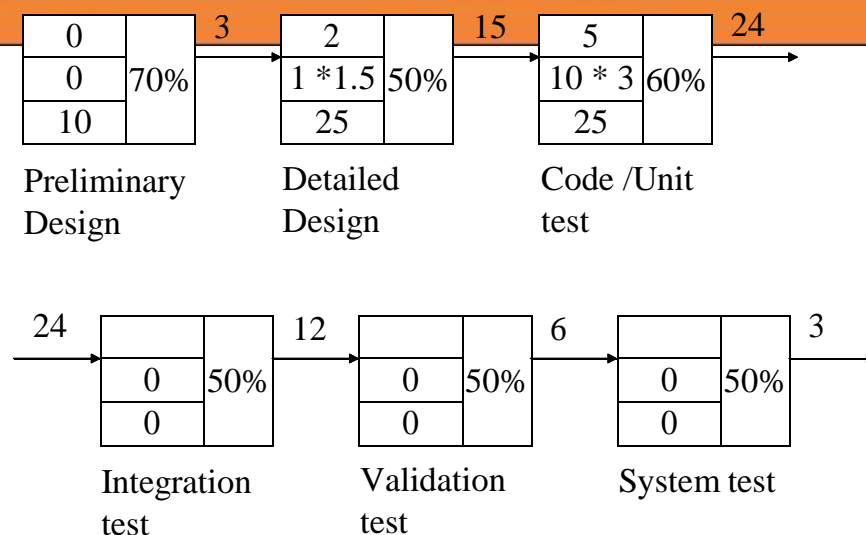
Development Step

Let us now assume that we do not have any SQA related activities for the first two stages and we are only using testing for detection of any defects. Let us assume that the

Preliminary design generated 10 defects which were passed on to Detailed design. At that phase, 6 defects were passed on to the next stages and 4 were amplified at a ration of 1:1.5. In addition, there were 25 new defects introduced at this stage. Therefore, a total of 37 defects were passed on to the next stage as shown in the diagram. In the Code and Unite test stage, we start to test our system and assuming 20% defect removal efficiency of this stage, 94 defects (80% of (10 + 27 * 3 + 25)) are passed on to the next stage. This process continues and the system is delivered with 12 defects remaining in the product.



If FTR are used in the earlier stages, the quality of the end-product is much better as shown in the following diagram. Note that in this case we have code inspection in addition to unit testing at the third stage and the defect removal efficiency of that stage is 60%.



Lecture No. 29

Formal Technical Reviews

Formal Technical Reviews are conducted by software engineers. The primary objective is to find errors during the process so that they do not become defects after release of software as they uncover errors in function, logic design, or implementation. The idea is to have early discovery of errors so they do not propagate to the next step in the process. They also ensure that the software has been represented according to predefined standards and it is developed in a uniform manner. They make projects more manageable and help groom new resources as well as provide backup and continuity.

FTRs include walkthroughs, inspections, and other small group technical assessments of software.

Guidelines for walkthroughs

FTRs are usually conducted in a meeting that is successful only if it is properly planned, controlled, and attended. The producer informs the PM that the WP is ready and the review is needed. The review meeting consists of 3-5 people and advanced preparation is required. It is important that this preparation should not require more than 2 hours of work per person. It should focus on specific (and small) part of the overall software. For example, instead of the entire design, walkthroughs are conducted for each component, or small group of components. By narrowing focus, FTR has a high probability of uncovering errors.

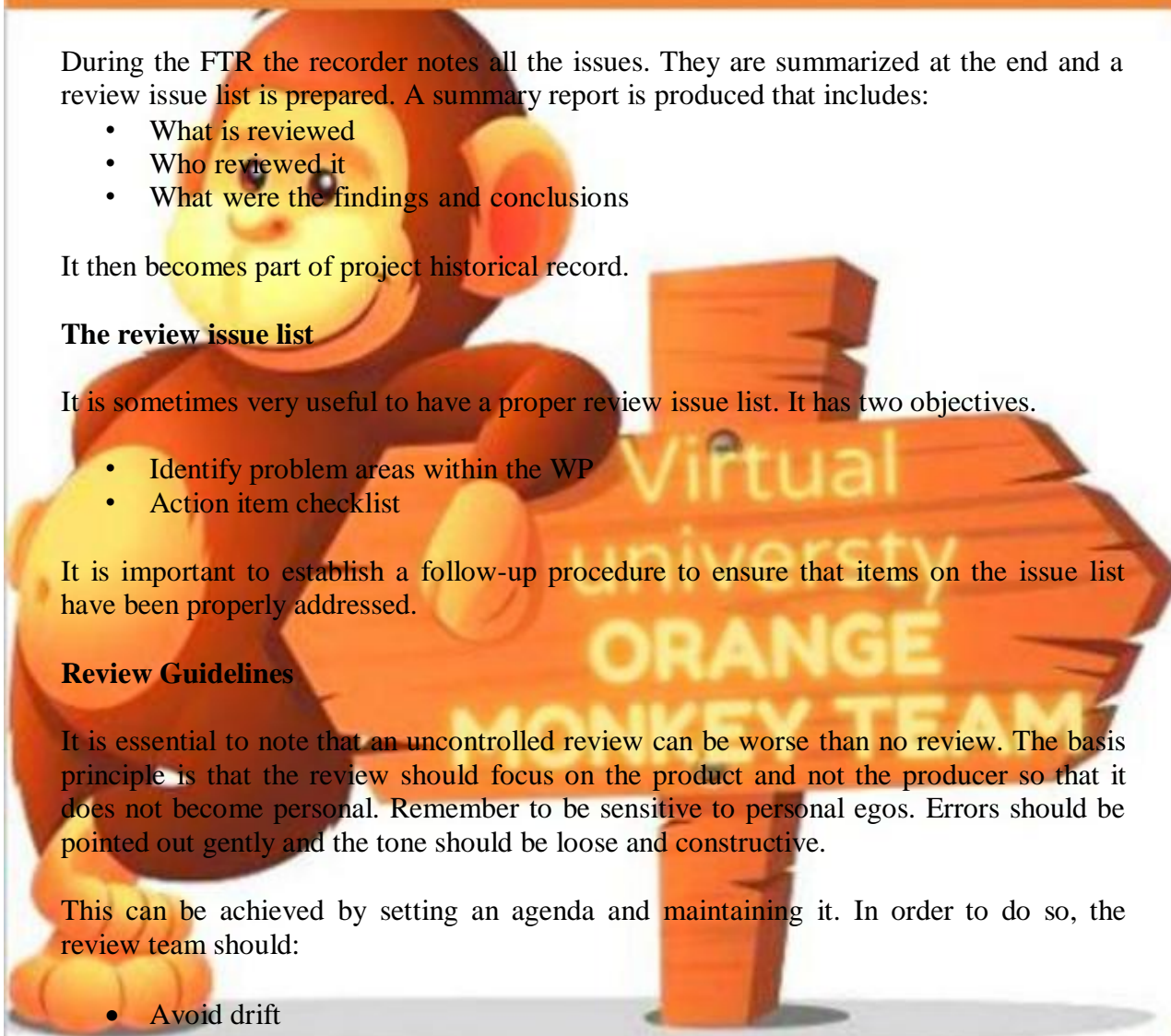
It is important to remember that the focus is on a work product for which the producer of the WP asks the project leader for review. Project leader informs the review leader. The review leader evaluates the WP for readiness and if satisfied generates copies of review material and distributes to reviewers for advanced preparation. The agenda is also prepared by the review leader.

Review Meetings

Review meeting is attended by the review leader, all reviewers, and the producer. One of the reviewer takes the roles of recorder. Producer walks through the product, explaining the material while other reviewers raise issues based upon their advanced preparation. When valid problems or errors are recorded, the recorder notes each one of them. At the end of the RM, all attendees of the meeting must decide whether to:

- Accept the product without further modification
- Reject the product due to severe errors
 - Major errors identified
 - Must review again after fixing
- Accept the product provisionally
 - Minor errors to be fixed
 - No further review

Review Reporting and Record keeping



During the FTR the recorder notes all the issues. They are summarized at the end and a review issue list is prepared. A summary report is produced that includes:

- What is reviewed
- Who reviewed it
- What were the findings and conclusions

It then becomes part of project historical record.

The review issue list

It is sometimes very useful to have a proper review issue list. It has two objectives.

- Identify problem areas within the WP
- Action item checklist

It is important to establish a follow-up procedure to ensure that items on the issue list have been properly addressed.

Review Guidelines

It is essential to note that an uncontrolled review can be worse than no review. The basis principle is that the review should focus on the product and not the producer so that it does not become personal. Remember to be sensitive to personal egos. Errors should be pointed out gently and the tone should be loose and constructive.

This can be achieved by setting an agenda and maintaining it. In order to do so, the review team should:

- Avoid drift
- Limit debate and rebuttal
- Enunciate problem areas but don't try to solve all problems
- Take written notes
- Limit the number of participants and insist upon advanced preparation
- Develop a checklist for each product that is likely to be reviewed
- Allocate resources and schedule time for FTRs
- Conduct meaningful training for all reviewers
- Review your early reviews
- Determine what approach works best for you

Lecture No. 30

Statistical Software Quality Assurance

Statistical SQA is a technique that measures the quality in a quantitative fashion. It implies that information about defects is collected and categorized and an attempt is made to trace each defect to underlying cause. It uses Pareto Principle to identify vital causes (80% of defects can be traced to 20% of causes) and moves to correct the problems that have caused the defects.

Example

Let us assume that information about defects is collected for one year and categorized as follows:

1. Incomplete or erroneous specifications (IES)
2. Misinterpretation of customer communication (MCC)
3. Intentional deviation from specifications (IDS)
4. Violation of programming standards (VPS)
5. Error in data representation (EDR)
6. Inconsistent component interface (ICI)
7. Error in digital logic (EDL)
8. Incomplete or erroneous testing (IET)
9. Inaccurate or incomplete documentation (IID)
10. Error in programming language translation of design (PLT)
11. Ambiguous or inconsistent HCI (HCI)
12. Miscellaneous (MIS)

The following data is collected for these categories

<i>Error Category</i>	Serious	Moderate	Minor	Sub Total
IES	34	68	103	205
MCC	12	68	76	156
IDS	1	24	23	48
VPS	0	15	10	25
EDR	26	68	36	130
ICI	9	18	31	58
EDL	14	12	19	45
IET	12	35	48	95
IID	2	20	14	36
PLT	15	19	26	60
HCI	3	17	8	28
MIS	0	15	41	56

Total	128	379	435	942
-------	-----	-----	-----	-----

We can easily see the following:

- IES, MCC, and EDR are the vital errors – cause 53% of all errors
- IES, EDR, PLT, and EDL are vital if only serious errors are considered

We now start corrective action focused on vital few. For example for EDR we review the data representation techniques to identify the possible improvement areas and adopt a use case tool for data modeling and perform stringent data design reviews.

Error Index (EI)

Another statistical technique known as Error Index (EI) is used to develop an overall indication of improvement in software quality. The EI is computed as follows:

Let

- E_i – the total number of errors uncovered during the i th step in the SE process
- S_i – number of serious errors
- M_i – number of moderate errors
- T_i – number of minor errors
- PS – product size at the i th step
- w_s, w_m, w_t – weighting factors for serious, moderate, and minor errors. Recommended values for these are 10, 3, 1 respectively.

At each step of the software process a *Phase Index* is computed as:

$$PI_i = w_s(S_i/E_i) + w_m(M_i/E_i) + w_t(T_i/E_i)$$

Now EI is computed as the cumulative effect on each $PI_i = \sum(i \times PI_i)/PS$

It is important to note that weighting errors encountered in the SE processes more heavily than those encountered earlier. As stated earlier, it can be used to develop an overall indication of improvement in software quality.

Software Reliability

Software reliability is another very important quality factor and is defined as probability of failure free operation of a computer program in a specified environment for a specified time. For example, a program X can be estimated to have a reliability of 0.96 over 8 elapsed hours.

Software reliability can be measured, directed, and estimated using historical and development data. The key to this measurement is the meaning of term failure. Failure is defined as non-conformance to software requirements. It can be graded in many different ways as shown below:

- From annoying to catastrophic
- Time to fix from minutes to months
- Ripples from fixing

It is also pertinent to understand the difference between hardware and software reliability. Hardware reliability is predicted on failure due to wear rather than failure due to design. In the case of software, there is no wear and tear. The reliability of software is determined by Mean time between failure (MTBF). MTBF is calculated as:

$$MTBF = MTTF + MTTR$$

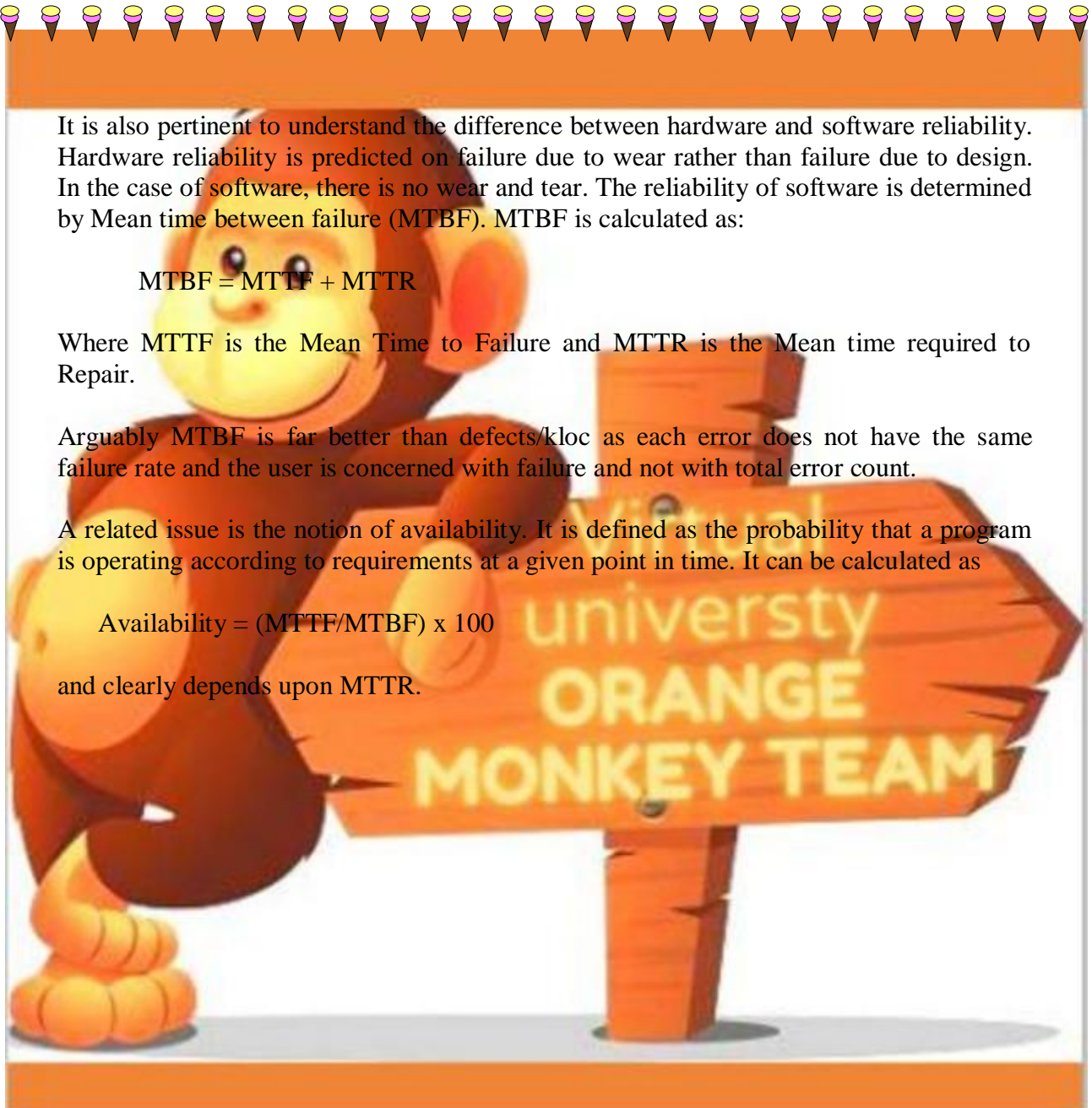
Where MTTF is the Mean Time to Failure and MTTR is the Mean time required to Repair.

Arguably MTBF is far better than defects/kloc as each error does not have the same failure rate and the user is concerned with failure and not with total error count.

A related issue is the notion of availability. It is defined as the probability that a program is operating according to requirements at a given point in time. It can be calculated as

$$\text{Availability} = (MTTF/MTBF) \times 100$$

and clearly depends upon MTTR.



Lecture No. 31

Software Safety

Software Safety is a software SQA activity that focuses on identification of potential hazards that may affect software negatively and cause an entire system to fail. Modeling and analysis process is conducted as part of software safety and hazards are identified and categorized by criticality and risk.

Example

Let us assume that the following hazards are associated with a computer-based cruise control for an automobile:

- Causes uncontrolled acceleration that cannot be stopped
- Does not respond to depression of brake pedal
- Does not engage when switch is activated
- Slowly loses or gains speed

Once system-level hazards are identified, analysis techniques are used to assign severity, and probability of occurrence. This technique is similar to risk analysis. To be effective, software must be analyzed in the context of the entire system. Analysis techniques such as fault tree analysis can be used to predict the chain of events that can cause hazards and the probability that each of these events will occur to create the chain. Once hazards are identified and analyzed, safety-related requirements can be specified for the software.

Reliability and safety are closely related. Software reliability uses statistical techniques to determine the likelihood that a software failure will occur. Occurrence of a software failure does not necessarily result in a hazard or mishap. On the other hand, software safety examines the ways in which failures result in conditions that can lead to a mishap.

Lecture No. 32

Poka-Yoke (Mistake-Proofing)

Poka-yoke is a QA technique developed by Shingo at Toyota in 1960's. Poka-yoke devices are mechanisms that lead to prevention of potential quality problem before it occurs or the rapid detection of quality problems if they are introduced.

- Examples:
 - Light on if the car door is not properly closed
 - Warning beep if the engine is turned-off when lights are on

Characteristic of a Poka-yoke device

- It is simple and cheap
- It is part of the process
- It is located near the process task where the mistake occurs

Example of Poka-yoke in software

Let us assume that a software products company sells application software to an international market. The pull-down menus and associated mnemonics provided with each application must reflect the local language. For example, the English language menu item for "Close" has the mnemonic "C" associated with it. When the application is sold to Urdu speaking users, the same menu item is "Band Karen" with mnemonic "ب". To implement the appropriate menu entry for each locale, a "localizer" translates the menus accordingly. The problem is to ensure that each menu entry conforms to appropriate standards and there are no conflicts, regardless of the language used.

We can consider a prevention device to write a program to generate mnemonics automatically, given a list of labels in each menu. It would prevent mistakes, but the problem of choosing a good mnemonic is difficult and the effort required would not be justified by the benefit gained. It is therefore not a good Poka-yoke device as it is not simple.

We now consider another prevention device. In this case we write a program to prevent the localizer from choosing mnemonics that did not meet the criteria. It would prevent mistakes but the benefit is minimal as incorrect mnemonics are easy enough to detect and correct after they occur. So once again this is not a good choice.

We now consider a detection device. In this case we write a program to verify that the chosen menu labels and mnemonics meet the criteria. We run this program to detect errors. Errors are sent back to localizer. Since this device requires a lot of back and forth movement, it violates the principle of co-location.

We now modify it to the following detection device. We write a program to verify that the chosen menu labels and mnemonics meet the criteria. The localizer would run this program to detect errors. It would give quick feedback and hence it is a good Poka-yoke device.

Lecture No. 33

Software Configuration Management (SCM)

You may recall that software configuration management (SCM) is one of the five KPA required for an organization to be at CMM level 2. That means, according to SEI, effective project management is not possible without having a proper SCM function in place.

The basic idea behind SCM is to manage and control change. As mentioned by Bersoff, no matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle. It is therefore essential that we manage and control it in a fashion that this continuous change does not convert into chaos.

This has become more important in the context of contemporary software development as we are getting more and more complex software projects in terms of size, sophistication, and technology. In addition, these software systems are used by millions of users all over the world. These systems need multilingual and multi-platform support (hardware, software) and have to operate in a distributed environment. That means that a software system may come in many configuration flavors including desktop, standard, professional, and enterprise versions. There is a brutal competition out there and any complacency may result in losing a big market share. The huge maintenance frequency – corrective and adaptive – makes life even more difficult.

More complex development environment with shorter reaction time results in confusion and chaos!

Change Chaos

This frequent change, if not managed properly, results in chaos. First of all there would be problems of identification and tracking which would result in questions like the following:

- “This program worked yesterday. What happened?”
- “I fixed this error last week. Why is it back?”
- “Where are all my changes from last week?”
- “This seems like an obvious fix. Has it been tried before?”
- “Who is responsible for this change?”

Then there are problems of version selection. The typical problems faced are:

- “Has everything been compiled? Tested?”
- “How do I configure for test, with my updates and no others?”
- “How do I exclude this incomplete/faulty change?”
- “I can’t reproduce the error in this copy!”
- “Exactly which fixes went into this configuration?”
- “Oh my God!. I need to merge 250 files!”
- Nobody knows which versions of programs are final
 - Where is the latest version?
 - Which version is the right one? I have so many
 - I have lost my latest changes
- Latest versions of code overwritten by old versions
- There was a minor problem, I fixed it but it is no longer working

- I can't figure-out the changes made to the previous version.
- I can't go back

Then there are software delivery problems.

- "Which configuration does this customer have?"
- "Did we deliver a consistent configuration?"
- "Did the customer modify the code?"
- "The customer skipped the previous two releases. What happens if we send him the new one?"
- Shipped the wrong version to the client.

This is not all. There may be more chaos in the following shapes and forms:

- The design document is out of sync with programs
- I don't know if all the changes that were suggested have been incorporated
- Which code was sent to testing?

SCM is a function that, if implemented, will reduce these problems to a minimal level.

Configuration management

As defined by CMM, the purpose of SCM is to establish and maintain the integrity or software products through the project's life cycle.

Configuration management is concerned with managing evolving software systems. It acknowledges that system change is a team activity and thus it aims to control the costs and effort involved in making changes to a system.

SCM involves the development and application of procedures and standards to manage an evolving software product and is part of a more general quality management process.

Baseline

When released to CM, software systems are called *baselines* and serve as the starting point for further development. A baseline is a software configuration management concept that helps us to control change without seriously impeding justifiable change. It is defined by IEEE as:

A specification or a product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

Software Configuration Item (SCI)

A Software Configuration Item is the information that is created as part of the software engineering process. Typical SCIs include requirement specifications, design specification, source code, test cases and recorded results, user guides and installation manuals, executable programs, and standards and procedures (for example C++ design guidelines).

Software Configuration Management Tasks

Software configuration management tasks include:

- 
- Identification
 - Version Control
 - Change Control
 - Configuration Auditing
 - Reporting

Identification addresses how does an organization identify and manage the many existing versions of a program in a manner that will enable changes to be accommodated efficiently?

Version Control talks about how does an organization control changes before and after software is released to a customer? It is actually a combination of procedures and tools to manage different versions of the software configuration.

Clemm states that

Configuration management allows the user to specify alternative configurations of the software system through the selection of the appropriate versions. This is supported by associating with each software version, and then allowing configuration to be specified and constructed by describing the set of desired attributes.

A version has many different attributes. In the simplest form a specific version number that is attached to each object and in the complex form it may have a string of Boolean variables (switches) that indicate specific types of functional changes that have been applied to the system.

The Change Control process addresses the important question of who has the responsibility for approving and ranking changes. Configuration Auditing deals with ensuring that the changes have been made properly and finally Reporting talks about the mechanism used to apprise others of changes that are made. Configuration Identification involves identification of a tool for SCM. Then a baseline is established and identified which is then used to identify configurable software items. At the minimum, all deliverables must be identified as configurable items. This includes design, software, test cases, tutorials, and user guides.

Lecture No. 34

Product Release Version Numbering System

Product release is the act of making a product available to its intended customers. After a product has had its first release, it enters a product release cycle. New versions of the product are made available that may fix defects or add features that were not in previous releases. These changes are categorized as updates or upgrades. An update fixes product defects. An upgrade enhances the product feature set and will include updates.

Release Numbering

Each individual product release is viewed as being in a unique state which is the total set of functionality possessed by the product release. Release numbering is a mechanism to identify the product's functionality state. Each release will have a different product state and hence will have a different release number. Although there is no industry standard, typically, a three field compound number of the format "X.Y.Z" is used. The different fields communicate functionality information about the product release.

The first digit, X, is used for the major release number which is used to identify a major increase in the product functionality. The major release number is usually incremented to indicate a significant change in the product functionality or a new product base-line.

The second digit, Y, stands for feature release number. The feature release number is iterated to identify when a set of product features have been added or significantly modified from their originally documented behaviour.

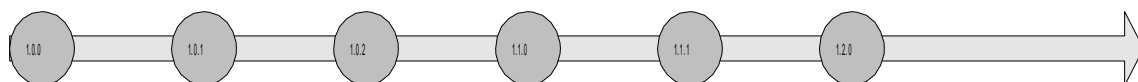
The third digit, Z, is called the defect repair number and is incremented when a set of defects is repaired. Defect repair/maintenance is considered to be any activity that supports the release functionality specification and it may a fix for some bugs or some maintenance to enhance the performance of the application.

Conventionally, a release number starts with a major number of one, followed by zero for its feature and maintenance numbers. This results in a release number 1.0.0. If the first new release that is needed is a defect repair release, the last digit would be iterated to one, resulting in 1.0.1. If two additional defect repair releases are needed, we would eventually have a release number of 1.0.3. Assume that an upgrade feature release is now needed. We will need to iterate the second field and will roll back the defect repair number to 0, resulting in a release number of 1.1.0. When we iterate the major release identifier, both the feature and defect numbers would be reset back to zero.

Internal Release Numbering

A special type of release is internal release. Internal releases are used by the development organization as a staging mechanism for functionality. The most common internal releases are the regular builds. A common way to number internal builds is to use the same release number that would be used for final release with some additional information added to it to identify the built. It is suggested that we add an extra (fourth) field to identify and keep track of internal builds.

The following diagram depicts the above mentioned version numbering system.



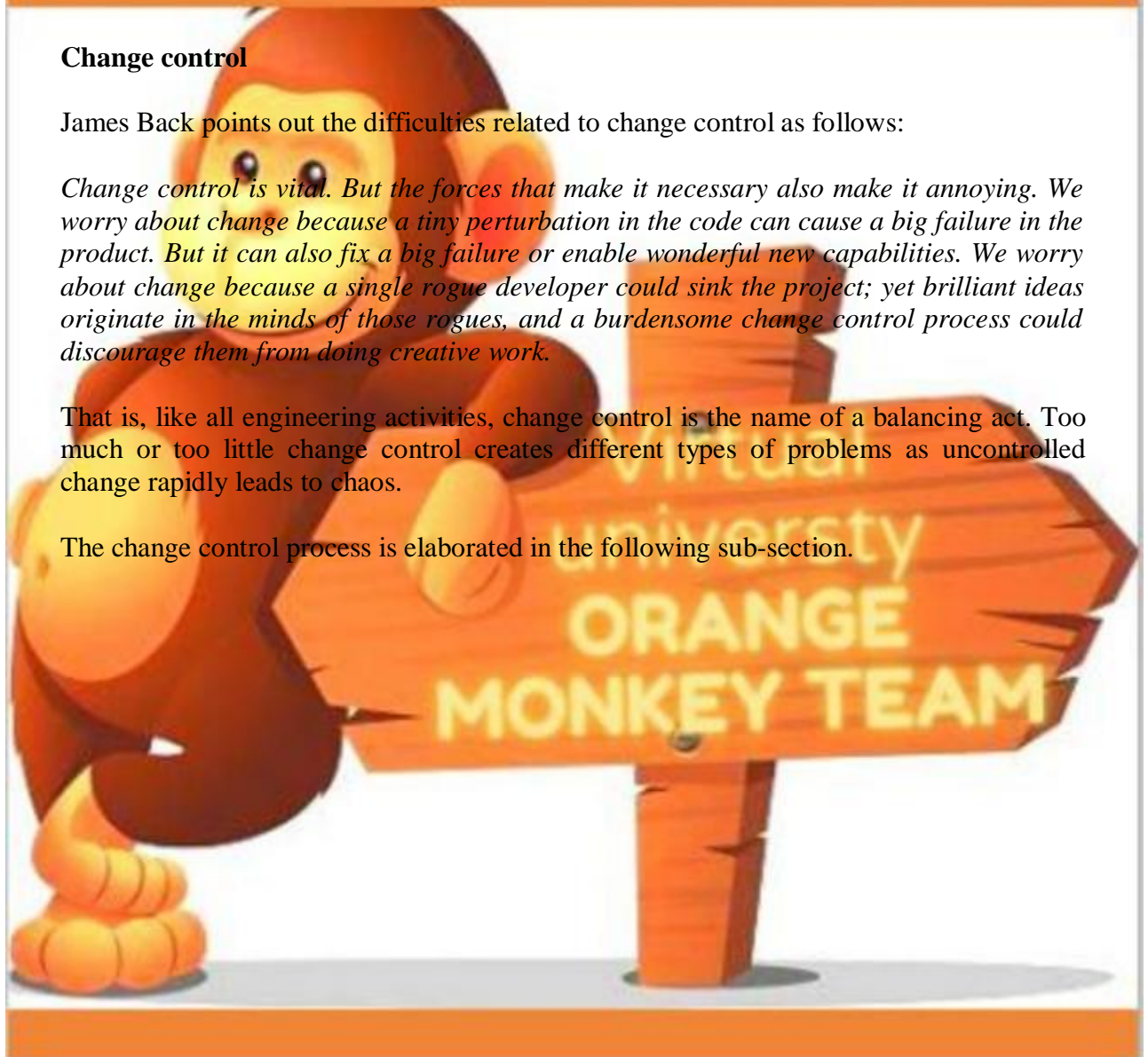
Change control

James Back points out the difficulties related to change control as follows:

Change control is vital. But the forces that make it necessary also make it annoying. We worry about change because a tiny perturbation in the code can cause a big failure in the product. But it can also fix a big failure or enable wonderful new capabilities. We worry about change because a single rogue developer could sink the project; yet brilliant ideas originate in the minds of those rogues, and a burdensome change control process could discourage them from doing creative work.

That is, like all engineering activities, change control is the name of a balancing act. Too much or too little change control creates different types of problems as uncontrolled change rapidly leads to chaos.

The change control process is elaborated in the following sub-section.



Lecture No. 35

Change Control Process

The first component of the change control process is the Change Control Authority (CCA) or a Change Control Board (CCB). A CCA or CCB includes people from both developer and client side.

Whenever a change is required, the CCB decides whether to allow this change to happen or deny it. If it is decided that a change is needed, an Engineering Change Order or ECO is generated. An ECO defines the change to be made, the constraints that must be respected, and the criteria for review and audit. The change control process thus involves the following steps.

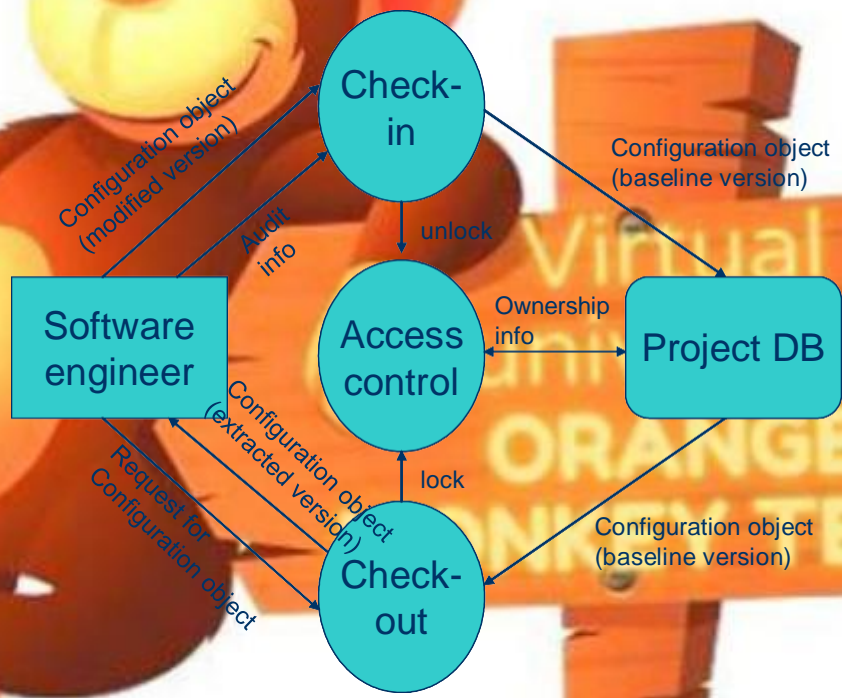
- 1) need for change is recognized
- 2) change request from user
- 3) developer evaluates
- 4) change report is generated
- 5) change control authority (CCA) decides
- 6) Either step 6a or 6b is performed. Steps numbers 7 to 17 are performed only if step 6b is performed.
 - a)
 - i) change request is denied
 - ii) user is informed
 - iii) no further action is taken.
 - b) assign people to SCIs
- 7) check-out SCIs
- 8) make the change
- 9) review/audit the change
- 10) check-in SCIs
- 11) establish a "baseline" for testing
- 12) perform SQA and testing activities
- 13) check-in the changed SCIs
- 14) promote SCI for inclusion in next release
- 15) rebuild appropriate version
- 16) review/audit the change
- 17) include all changes in release

Thus, a change is incorporated in a controlled and strict manner.

Check-in and check-out

In SCM, the processes of Check-in and Check-out take a central stage. These are two important elements of change control and provide access and synchronization control. Access control manages who has the authority to check-out the object and synchronization control ensures that parallel changes by two different people do not overwrite one another.

Synchronization control implements a control on updates. When a copy is checked-out, other copies can be checked out for use only but they cannot be modified. In essence, it implements a single-writer multiple-readers protocol. This process is depicted in the following diagram.



Configuration Audit

Configuration audit ensures that a change has been properly implemented. It involves formal technical reviews and software configuration audit. Configuration audit assesses a configuration object for characteristics that are generally not considered during audit. It is conducted by the SQA group. It looks into the following aspects of the change:

- Has the change specified in the ECO been made? Have any additional modifications been incorporated?
- Has a FTR been conducted to assess technical correctness?
- Has the software process been followed?
- Have the SE standards been properly applied?
- Has the change been highlighted in the SCI?
 - Change date and author
 - Have the attributes of the configuration object been updated?
- Have the SCM procedures for noting the change, recording it, and reporting it been followed?
- Have all related SCI's been properly updated?

An audit report is finally generated and any non-compliances are highlighted so that they may be corrected.

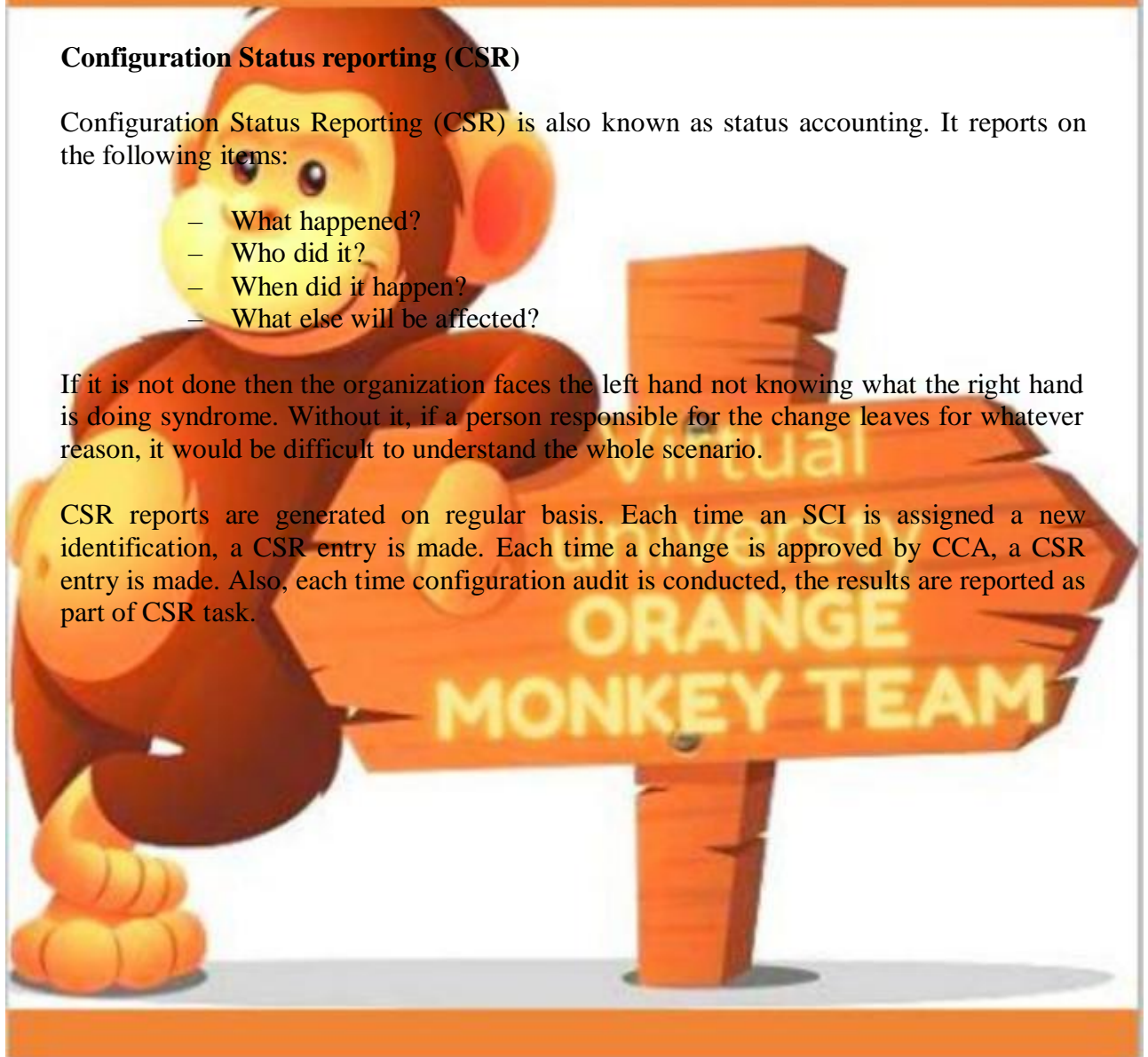
Configuration Status reporting (CSR)

Configuration Status Reporting (CSR) is also known as status accounting. It reports on the following items:

- What happened?
- Who did it?
- When did it happen?
- What else will be affected?

If it is not done then the organization faces the left hand not knowing what the right hand is doing syndrome. Without it, if a person responsible for the change leaves for whatever reason, it would be difficult to understand the whole scenario.

CSR reports are generated on regular basis. Each time an SCI is assigned a new identification, a CSR entry is made. Each time a change is approved by CCA, a CSR entry is made. Also, each time configuration audit is conducted, the results are reported as part of CSR task.



Lecture No. 36

Requirement Management and CMM

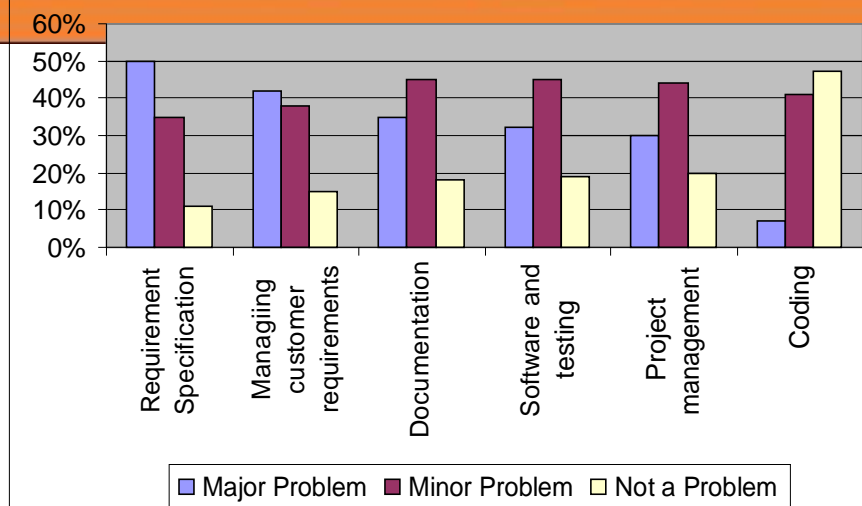
CM standards

CM should always be based on a set of standards which are applied within an organisation. Standards should define how items are identified, how changes are controlled and how new versions are managed. Standards may be based on external CM standards (e.g. IEEE standard for CM ANSI/IEEE Std. No. 828-1983, 1042-1987, 1028-1988). Existing standards are based on a waterfall process model - new standards are needed for evolutionary development.

The Requirement Problem

The goal of software development is to develop quality software – on time and on budget – that meets customers' real needs. Project success depends on good requirement management. It may be recalled that requirement errors are the most common type of software development errors and the most costly to fix. It may also be recalled that requirement errors are listed as one of the roots causes of software project failure. According to a Standish Group report, lack of user input is responsible for 13% of all project failures; incomplete requirements and specifications for 12% of all project failures; and changing requirements are responsible for 12% of all project failures.

Largest software problems by category



The above graph shows the result of an industry survey. It is clear to see that requirement specifications are considered to be the most significant cause of major software problems by a majority of practitioners.

Requirement Management

Requirement management is also one of the 5 KPA defined at CMM level 2. Without having a proper requirement management function, chances of a project's success are slim.

Requirements Management KPA Goals statement says that:

1. The software requirements are controlled to establish a baseline for software engineering and management use.
2. Software plans, products, and activities are kept consistent with the software requirements.

Requirement Management is defined as a systematic approach to eliciting, organizing, and documenting the requirements of the system, and a process that establishes and maintains agreement between the customer and the project team on the changing requirements of the system.

It includes establishing and maintaining an agreement with the customer on the requirement for the software project. It involves

- Defining the requirement baseline
- Reviewing proposed requirement changes and evaluating the likely impact of each proposed change before deciding whether to approve it
- Incorporating approved requirement changes in the project in a controlled manner
- Keeping project plans current with the requirements
- Negotiating new commitments based on estimated impact on changed requirements
- Tracing individual requirements to their corresponding design, source code, and test cases
- Tracking requirement status and change activity throughout the project

Requirement Attributes

We need to tag requirements with certain attributes in order to manage them in an orderly fashion. Attributes are used to establish a context and background for each requirement. They go beyond the description of intended functionality. They can be used to filter, sort, or query to view selected subset of the requirements. A list of possible attributes is enumerated as below:

1. Requirement ID
2. Creation date
3. Created by
4. Last modified on
5. Last modified by
6. Version number
7. Status
8. Origin
9. Subsystem

- 10. Product Release
- 11. Priority

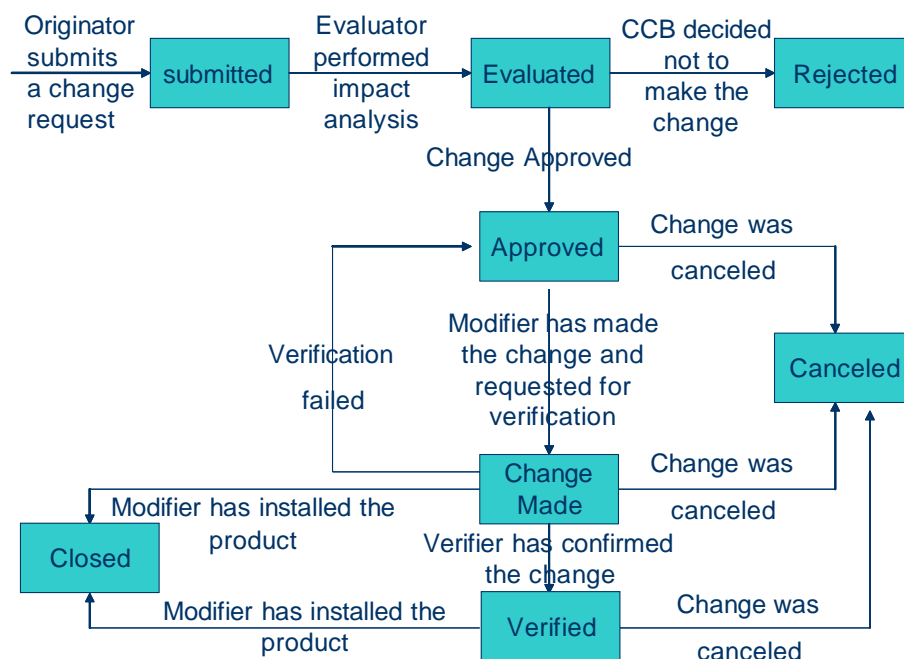
Requirement Status

The requirement status attribute is one of the most useful ones. It can be used to keep track of different requirements going through different phases. The possible status values are proposed, approved, implemented, verified, and deleted. These are elaborated in the following paragraphs.

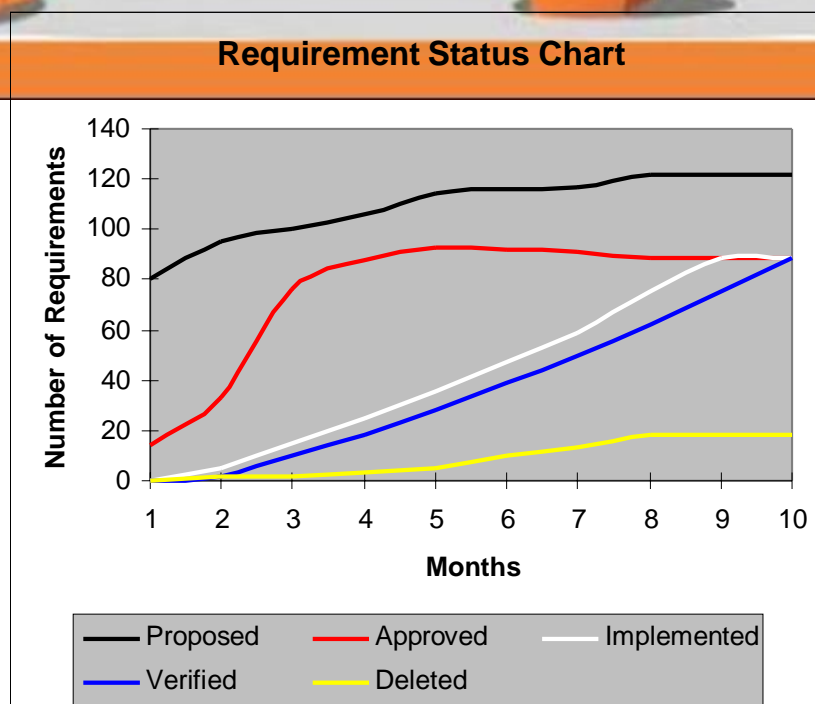
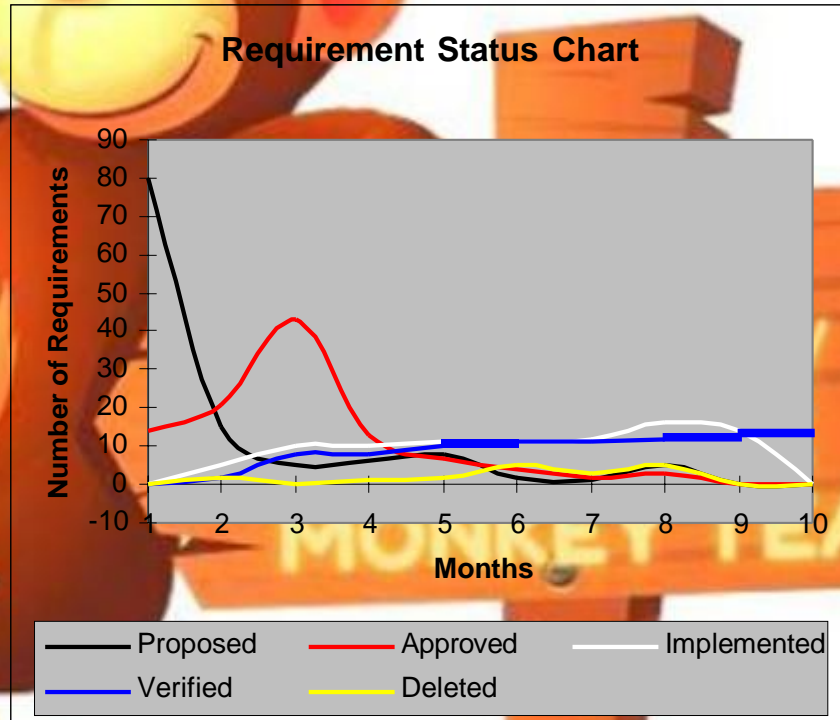
1. Proposed: The requirement has been requested by a source who has the authority to provide requirements.
2. Approved: The requirement has been analyzed, its impact on the rest of the project has been estimated, and it has been allocated to the baseline for a specific build number or product release. The software development group has committed to implement the requirement.
3. Implemented: The code that implements the requirement has been designed, written, and unit tested.
4. Verified: The implemented requirement has been verified through the selected approach, such as testing or inspection. The requirement has been traced to pertinent test cases. The requirement is now considered complete.
5. Deleted: A planned requirement has been deleted from the baseline. Include an explanation of why and by whom the decision was made to delete the requirement.

Change Request Status

As the requirement go through different phases, their status is updated accordingly. The following state transition diagram captures the sequence of requirement status changes.



The changes in the requirement status can be plotted as shown below to get an idea of the stability of the requirements and the progress of the project. It is easy to see that it is normal to have unstable requirements in the beginning but if they requirements stayed volatile till the end then the progress would be slow.



Managing Scope Creep

We must always remember that requirements will change, no matter what. That means we have to be able to manage changing requirements. Software organizations and professionals must learn to manage changing requirements. A major issue in requirements engineering is the rate at which requirements change once the requirements phase has “officially” ended. We therefore need to try to take it to a minimum level. For that we need to measure the change activity.



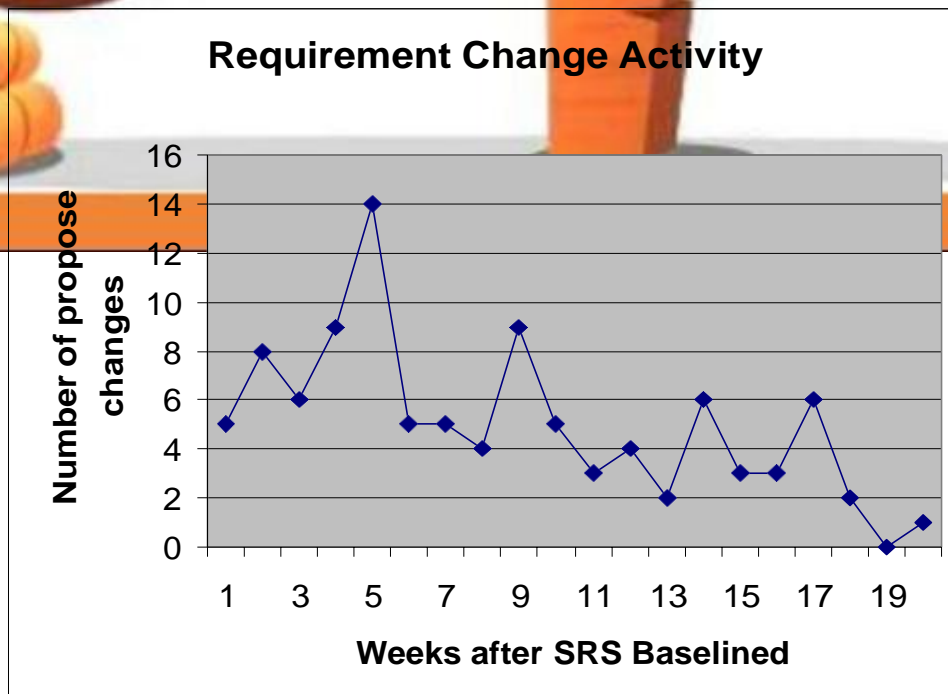
Lecture No. 37

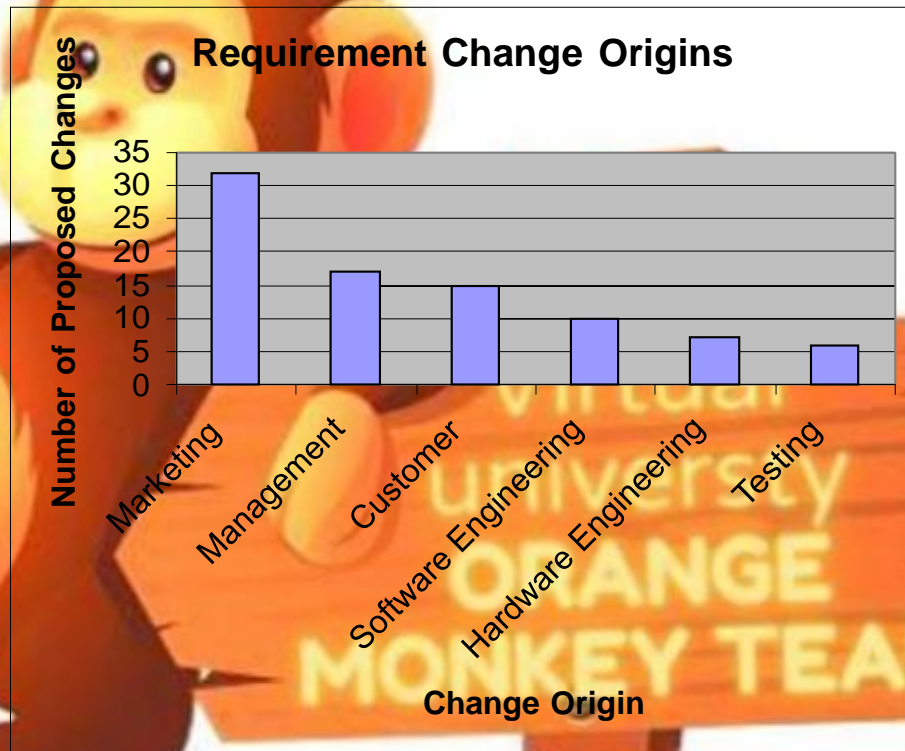
Measuring Change Activity

Measurement of change activity is a way to assess the stability of the requirements and to identify opportunities for process improvement. In this regards, the following could be measured

- The number of change requests received, open, and closed
- The cumulative number of changes made including added, deleted, and modified requirements
- The number of change requests that originated from each source
- The number of changes proposed and made in each requirement since it was baselined
- The total effort devoted to handling changes

These can then be plotted as shown in the graphs below to get an idea of the stability of the systems. It is important to note that the sooner the requirements come to a stable state after establishing the baseline the better it is for the project. It is also useful to track the source of the requirement changes so that processes governing those areas causing more frequent changes may be strengthened in future projects.





Requirement Traceability

Requirement traceability is a very important consideration for requirement management. It is really hard to manage requirements that are not traceable.

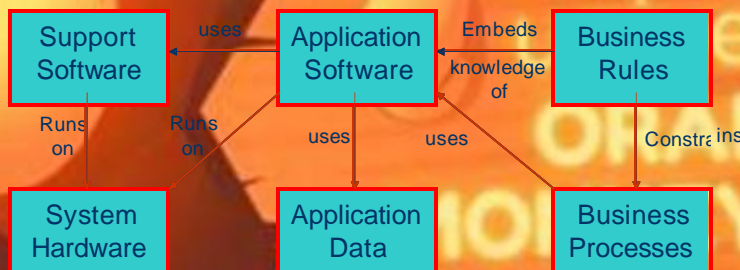
A Software Requirement Specification (SRS) is traced if the origin of its requirements is clear. That means that the SRS includes references to earlier supportive documents. An SRS is traceable if it is written in a manner that facilitates the referencing of each individual requirement stated therein.

It is important to trace requirements both ways. That is from origin of a requirement to how it is implemented. This is a continuous process. It is also important that the rationale of requirements must also be traced. Traceability is important for the purposes of certification, change impact analysis, maintenance, project tracking, reengineering, reuse, risk reduction, and testing. That is it plays an important role in almost every aspect of the project and its life cycle.

Lecture No. 38

Legacy systems

A system is considered to be a legacy system if it has been in operation for many years. A legacy system has many components. These include business processes, business rules, application software, application data, support software, and system hardware. The relationship among these components is shown in the following diagram.



Maintaining Legacy System

Maintaining legacy system is expensive. It is often the case that different parts of the system have been implemented by different teams, lacking consistency. Part or all of the system may be implemented using an obsolete language. In most cases system documentation is inadequate and out of date. In some cases the only documentation is the source code. In some cases even the source code is not available.

Many years of maintenance have usually corrupted the system structure, making it increasingly difficult to understand. The data processed by the system may be maintained in different files which have incompatible structures. There may be data duplication and the documentation of the data itself may be out of date, inaccurate, and incomplete.

As far as the system hardware is concerned, the hardware platform may be outdated and is hard to maintain. In many cases, the legacy systems have been written for mainframe hardware which is no longer available, expensive to maintain, and not be compatible with current organizational IT purchasing policies.

Support software includes OS, database, and compiler etc. Like hardware, it may be obsolete and no longer supported by the vendors.

A time therefore comes when an organization has to make this decision whether to keep the old legacy system or to move it to new platform and environment. Moving it to new environment is known as legacy system migration.

Legacy migration risks

Legacy system migration however is not an easy task and there are a number of risks involved that need to be mitigated. First of all, there is rarely a complete specification of the system available. Therefore, there is no straight forward way of specifying the services provided by a legacy system. Thus, important business rules are often embedded in the software and may not be documented elsewhere. Business processes and the way legacy systems operate are often intertwined. New software development may take several years.

New software development is itself risky as changes to one part of the system inevitably involve further changes to other components.

We therefore need to assess a legacy system before a decision for migration is made.

Legacy System Assessment

For each legacy system, there are four strategic options:

1. Scrap the system completely: This is the case when system is not making an effective contribution to business processes and business processes have changed significantly and the organization is no longer completely dependent upon the system.
2. Continue maintaining the system: This option is used when system is still required, it is stable, and requirements are not changing frequently
3. Transform the system in some way to improve its maintainability: this option is exercised when system quality has been degraded and regular changes to the system are required.
4. Replace the system with a new system: this path is taken when old system cannot continue in operation and off-the shelf alternative is available or system can be developed at a reasonable cost.

For these decisions, a legacy system can be assessed from two different perspectives – business value and quality. The following four quadrant assessment matrix can be used for this purpose.

Business Value	High	<ul style="list-style-type: none">•Important for business•Cannot be scrapped•Low quality means high operational cost•Candidates for system transformation or replacement	<ul style="list-style-type: none">•Must be kept in business•High quality means low cost of maintenance•Not necessary to transform or replace•Continue normal operation
	Low	<ul style="list-style-type: none">•Keeping these systems in operation will be expensive•Rate of return to the business is small•Candidates for scrapping	<ul style="list-style-type: none">•Low business value but not very expensive to maintain•Not worth the risk of replacing them•Should be normally maintained or scrapped
		Low	High



Business Value Assessment

It is important to note that this is a subjective judgment and requires different business viewpoints. These view points include end-users, customers, line managers, IT managers, and senior managers.

End Users assess the system from the perspective of how effective do they find the system in supporting their business processes and how much of the system functionality is used.

The customers look at the system and ask is the use of the system transparent to customer or are their interaction constrained by the system, are they kept waiting because of the system, and do system errors have a direct impact on the customer.

From an IT Manager's perspective the following questions need to be asked: Are there difficulties in finding people to work on the system? Does the system consume resources which could be deployed more effectively on other systems?

Line Managers ask: do managers think that the system is effective in contributing to success of their unit? Is the cost of keeping the system in use justified? Is the data managed by the system critical for the functioning of the manager's unit?

Senior Managers look at the system from the angle that does the system and associated business process make an effective contribution to the business goal?



Lecture No. 39

Environment Assessment

The legacy system also needs to be assessed from an environment's perspective. This involves looking at the supplier, failure rate, age, performance, support requirements, maintenance cost, and interoperability.

These angles are elaborated in the following paragraph:

Supplier stability: Is the supplier still in existence? Is the supplier financially stable and likely to continue in existence? If the supplier is no longer in business, is the system maintained by someone else?

Failure rate: Does the hardware have a high rate of reported failure? Does the support software crash often and force system restarts?

Age: How old is the hardware and software?

Performance: Is the performance of the system adequate? Do performance problems have a significant effect on system users?

Support requirements: What local support is required by hardware and software? If there are high costs associated with this support, it may be worth considering system replacement?

Maintenance Cost: What are the costs of hardware maintenance and software licenses?

Interoperability: Are there problems interfacing the system with other systems? Can compilers etc be used with current versions of the operating system? Is system emulation required?

Application software assessment

The application software is assessed on the following parameters:

Understandability: How difficult is it to understand the software code of the current system? How complex are the control structures that are used?

Documentation: What system documentation is available? Is the documentation complete, consistent, and up-to-date?

Data: Is there an explicit data model for the system? To what extent is data duplicated in different files?

Programming Language: Are modern compilers available for the programming language? Is the language still used for new system development?

Test Data: Does test data for the system exist? Is there a record of regression tests carried out when new features have been added to the system?

Personnel skills: Are there people available who have the skills to maintain the system?

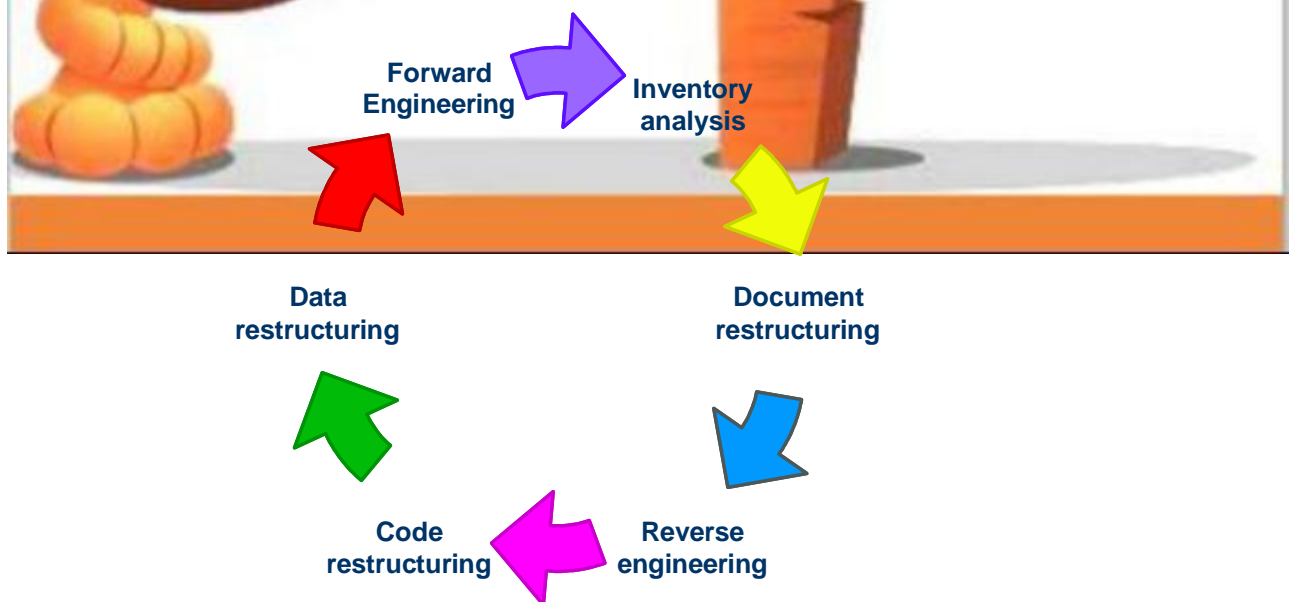
As migration is a very costly and risky business, the decision to migrate the system is made after assessing it from all these angles and it is determined that the time has come to migrate this system and it is worth spending the required amount of money and time for undertaking that effort.

Software Reengineering

Software solutions often automate the business by implementing business rules and business processes. In many cases, the software makes the business processes. As these rules and processes change, the software must also change. A time comes when these changes become very difficult to handle. So reengineering is re-implementing legacy systems to make them more maintainable. It is a long term activity.

Software Reengineering Process Model

The software reengineering is a non-trivial activity. Just like legacy migration, careful analysis must be carried out before a decision for reengineering is taken. The following process model can be used to reengineer a legacy system.



Inventory analysis

Inventory analysis is the first step in the reengineering process. At this stage, inventory of all applications is taken a note of their size, age, business criticality, and current maintainability is made. Inventory should be updated regularly as the status of the application can change as a function of time.



Document restructuring

The next step in the reengineering process is document restructuring. Weak documentation is a trademark of many legacy applications. Without proper documentation, the hidden rules, business processes, and data cannot be easily understood and reengineered.

In this regards, the following options are available:

1. **Create documentation:** Creating documentation from scratch is very time consuming. If program is relatively stable and is coming to the end of its useful life then just leave it as it is.
2. **Update documentation:** This option also needs a lot of resources. A good approach would be to update documentation when the code is modified.

Reverse engineering

Reverse engineering is the next step in the process. Reverse engineering for software is a process for analyzing a program in an effort to create a representation of the program at a higher level of abstraction than the source code. Reverse engineering is the process of design recovery. At this stage, documentation of the overall functionality of the system that is not there is created. The overall functionality of the entire system must be understood before more detailed analysis can be carried out.

Reverse engineering activities include:

- Reverse engineering to understand processing
- Reverse engineering to understand data
 - Internal data structures
 - Database structures
- Reverse engineering user interfaces

Program Restructuring

Program is restructured after the reverse engineering phase. In this case we modify source code and data in order to make it amenable to future changes. This includes code as well as data restructuring. Code restructuring requires redesign with same function with higher quality than original program and data restructuring involves restructuring the database or the database schema. It may also involve code restructuring.

Lecture No. 40

Forward Engineering

At the end forward engineering is carried out. It means incorporating the new business processes and rules in the system. Forward engineering requires application of SE principles, methods, and concepts to re-create an existing application. In most cases forward engineering does not simply create a modern equivalent of an older program, rather new user and technology requirements are integrated into the reengineering effort.

The Economics of Reengineering

As reengineering is a costly and risky undertaking, a cost benefit analysis for the reengineering effort must be carried out.

This analysis is carried out in the following manner.

Let

- P1 : current annual maintenance cost for an application
- P2 : current annual operation cost for an application
- P3 : current annual business value of an application
- P4 : predicted annual maintenance cost after reengineering
- P5 : predicted annual operations cost after reengineering
- P6 : predicted annual business value cost after reengineering
- P7 : estimated reengineering cost
- P8 : estimated reengineering calendar time
- P9 : reengineering risk factor (1.0 is nominal)
- L : expected life of the system

Now the cost of maintenance is calculated as:

$$C_{\text{maintenance}} = [P3 - (P1 + P2)] \times L$$

Cost of reengineering would then be given by the formula:

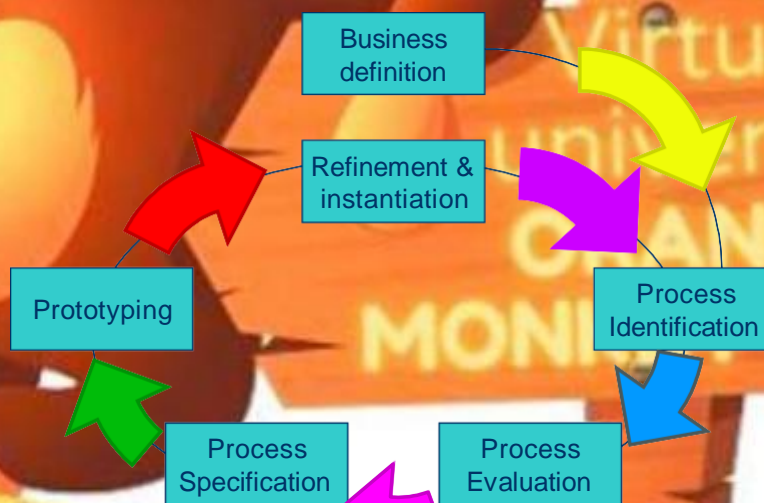
$$C_{\text{reengineering}} = [P6 - (P4 + P5) \times (L - P8) - (P7 \times P9)]$$

Lecture No. 41

Business Process Reengineering

A concept similar to software reengineering is of business process reengineering (BPR). A business process is “a set of logically related tasks performed to achieve a defined business outcome”. It is the way certain business is conducted. Purchasing services and supplies, hiring new employees, paying suppliers are examples of business processes.

For BPR the following process model may be used.



As obvious from the diagram, it starts with the business definition where business goals are identified. The key drivers could be cost reduction, time reduction, quality improvement, and personnel development and empowerment. It may be defined at the business level or for a specific component of the business.

The next step is process identification. At this time processes that are critical to achieving the goals are identified and are ranked by importance, and need for change.

The short listed processes are then evaluated. Existing process is analyzed and measured and process tasks are identified. The cost and time consumed is measured as well as the quality and performance problems are identified.

Then, process specification and design is carried out. Use cases are prepared for each process to be redesigned and a new set of tasks are designed for the processes and then they are prototyped.

A redesigned business process must be prototyped before it is fully integrated into the business.

Based on the feedback the business process is refined.

Lecture No. 42

Software Refactoring

Software refactoring is the process of changing a software system such that the external behavior of the system does not change while the internal structure of the system is improved. This is sometimes called “Improving the design after it has been written”.

Fowler defines refactoring as A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior. It is achieved by applying a series of refactorings without changing its observable behavior.

A (Very) Simple Example

Let us consider a very simple example. The refactoring involved in this case is known as “Consolidate Duplicate Conditional Fragments”. As the name suggests, this refactoring lets the programmer improve the quality of the code by grouping together the duplicate code, resulting in less maintenance afterwards.

```
if (isSpecialDeal()) {
    total = price * 0.95;
    send ();
}
else {
    total = price * 0.98;
    send ();
}
```

In this case, *send* is being called from different places. We can consolidate as follows:

```
if (isSpecialDeal())
    total = price * 0.95;
else
    total = price * 0.98;

send ();
```

It can further be improved if we calculate total outside the if statement as shown below.

```
if (isSpecialDeal())
    factor = 0.95;
else
    factor = 0.98;

total = price * factor;
send ();
```

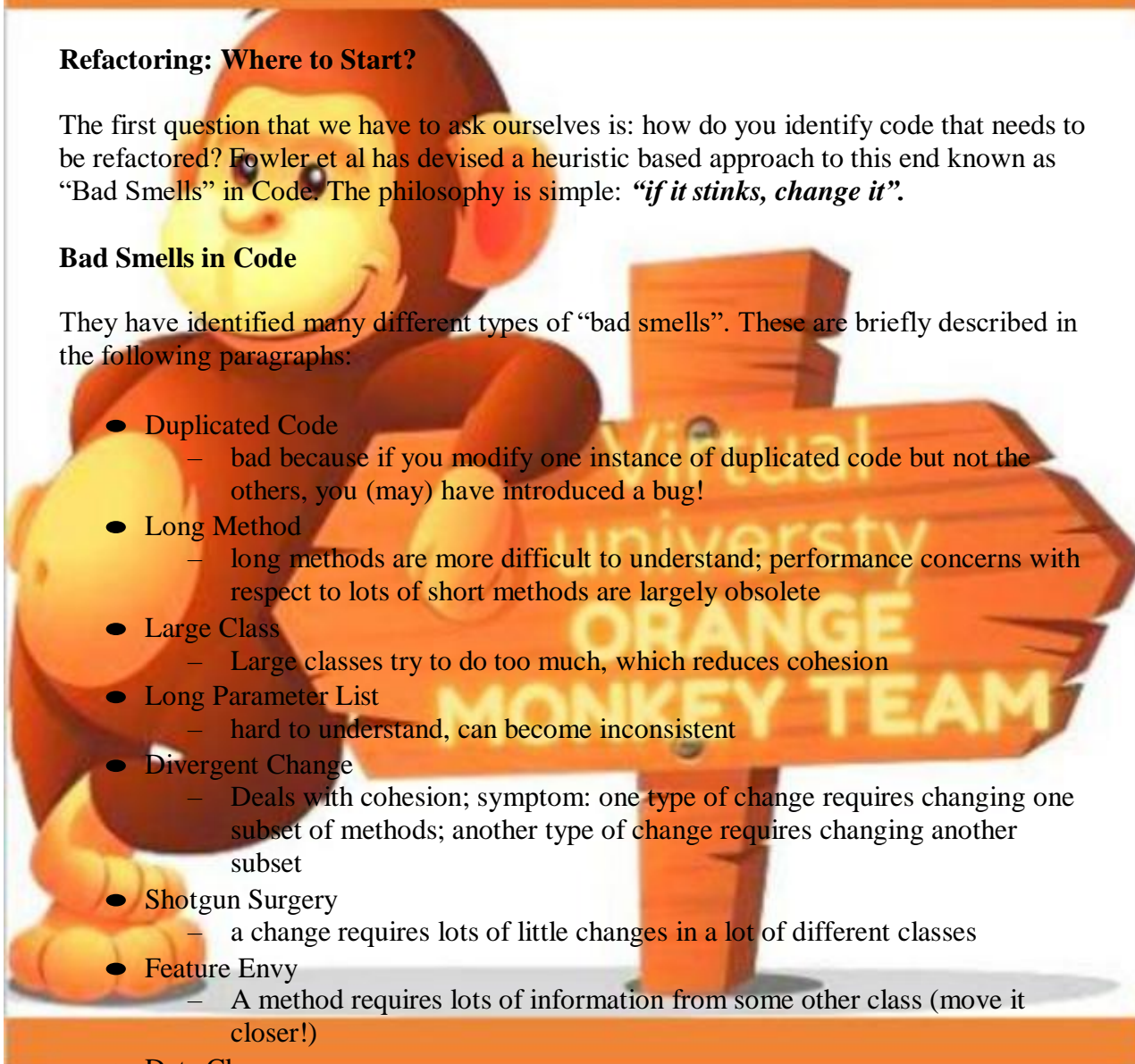
Although this is a trivial example, it nevertheless is useful and teaches us how can be consolidate code by grouping together pieces of code from different segments.

Refactoring: Where to Start?

The first question that we have to ask ourselves is: how do you identify code that needs to be refactored? Fowler et al has devised a heuristic based approach to this end known as “Bad Smells” in Code. The philosophy is simple: *“if it stinks, change it”*.

Bad Smells in Code

They have identified many different types of “bad smells”. These are briefly described in the following paragraphs:

- 
- Duplicated Code
 - bad because if you modify one instance of duplicated code but not the others, you (may) have introduced a bug!
 - Long Method
 - long methods are more difficult to understand; performance concerns with respect to lots of short methods are largely obsolete
 - Large Class
 - Large classes try to do too much, which reduces cohesion
 - Long Parameter List
 - hard to understand, can become inconsistent
 - Divergent Change
 - Deals with cohesion; symptom: one type of change requires changing one subset of methods; another type of change requires changing another subset
 - Shotgun Surgery
 - a change requires lots of little changes in a lot of different classes
 - Feature Envy
 - A method requires lots of information from some other class (move it closer!)
 - Data Clumps
 - attributes that clump together but are not part of the same class
 - Primitive Obsession
 - characterized by a reluctance to use classes instead of primitive data types
 - Switch Statements
 - Switch statements are often duplicated in code; they can typically be replaced by use of polymorphism (let OO do your selection for you!)
 - Parallel Inheritance Hierarchies
 - Similar to Shotgun Surgery; each time I add a subclass to one hierarchy, I need to do it for all related hierarchies
 - Lazy Class
 - A class that no longer “pays its way”. e.g. may be a class that was downsized by refactoring, or represented planned functionality that did not pan out
 - Speculative Generality
 - “Oh I think we need the ability to do this kind of thing someday”
 - Temporary Field

- An attribute of an object is only set in certain circumstances; but an object should need all of its attributes
- Message Chains
 - a client asks an object for another object and then asks that object for another object etc. Bad because client depends on the structure of the navigation
- Middle Man
 - If a class is delegating more than half of its responsibilities to another class, do you really need it?
- Inappropriate Intimacy
 - Pairs of classes that know too much about each other's private details
- Alternative Classes with Different Interfaces
 - Symptom: Two or more methods do the same thing but have different signature for what they do
- Incomplete Library Class
 - A framework class doesn't do everything you need
- Data Class
 - These are classes that have fields, getting and setting methods for the fields, and nothing else; they are data holders, but objects should be about data AND behavior
- Refused Bequest
 - A subclass ignores most of the functionality provided by its superclass
- Comments (!)
 - Comments are sometimes used to hide bad code
 - "...comments often are used as a deodorant" (!)

Breaking a Method

We have already seen example of duplicate code. We now look at another simple example of long method. Although, in this case, the code is not really long, it however demonstrates how longer segments of code can be broken into smaller and more manageable (may be more reusable as well) code segments.

The following code segment sorts an array of integers using "selection sort" algorithm.

```
for (i=0; i < N-1; i++) {
    min = i;
    for (j = i; j < N; j++)
        if (a[j] < a[min]) min = j;
    temp = a[i];
    a[i] = a[min];
    a[min] = temp;
}
```

We break it into smaller fragments by making smaller functions out of different steps in the algorithm as follows:

```

int minimum (int a[ ], int from, int to)
{
    int min = from;
    for (int i = from; i <= to; i++)
        if (a[i] < a[min]) min = i;
    return min;
}

```

```

void swap (int &x, int &y)
{
    int temp = x;
    x = y;
    y = temp;
}

```

The sort function now becomes simpler as shown below.

```

for (i=0; i < N-1; i++) {
    min = minimum (a, i, N-1);
    swap(a[i], a[min]);
}

```

It can be seen that it is now much easier to understand the code and hence is easier to maintain. At the same time we have got two separate reusable functions that can be used elsewhere in the code.

A slightly more involved example
(It has mostly been adapted from Fowler's introduction to refactoring which is freely available on the web)

Let us consider a simple program for a video store. It has three classes: Movie, Rental, and Customer. Program is told which movies a customer rented and for how long and it then calculates the charges and the Frequent renter points. Customer object can print a statement (in ASCII).

Here is the code for the classes. DomainObject is a general class that does a few standard things, such as hold a name.

```

public class DomainObject {

    public DomainObject (String name) {
        _name = name;
    };

    public DomainObject ()    {};

    public String name ()    {
        return _name;
    };
}

```

```

    public String toString() {
        return _name;
    };
    protected String _name = "no name";
}

```

Movie represents the notion of a film. A video store might have several tapes in stock of the same movie

```

public class Movie extends DomainObject {
    public static final int CHILDRENS = 2;
    public static final int REGULAR = 0;
    public static final int NEW_RELEASE = 1;

```

```

    private int _priceCode;

```

```

    public Movie(String name, int priceCode) {
        _name = name;
        _priceCode = priceCode;
    }

```

```

    public int priceCode() {
        return _priceCode;
    }

```

```

    public void persist() {
        Registrar.add ("Movies", this);
    }

```

```

    public static Movie get(String name) {
        return (Movie) Registrar.get ("Movies", name);
    }
}

```

The movie uses a class called a registrar (not shown) as a class to hold instances of movie. I also do this with other classes. I use the message persist to tell an object to save itself to the registrar. I can then retrieve the object, based on its name, with a get(String) method.

The tape class represents a physical tape.

```

class Tape extends DomainObject
{
    public Movie movie() {
        return _movie;
    }
    public Tape(String serialNumber, Movie movie) {

```

```

        _serialNumber = serialNumber;
        _movie = movie;
    }
    private String _serialNumber;
    private Movie _movie;
}

```

The rental class represents a customer renting a movie.

```

class Rental extends DomainObject
{
    public int daysRented() {
        return _daysRented;
    }
    public Tape tape() {
        return _tape;
    }
    private Tape _tape;
    public Rental(Tape tape, int daysRented) {
        _tape = tape;
        _daysRented = daysRented;
    }
    private int _daysRented;
}

```

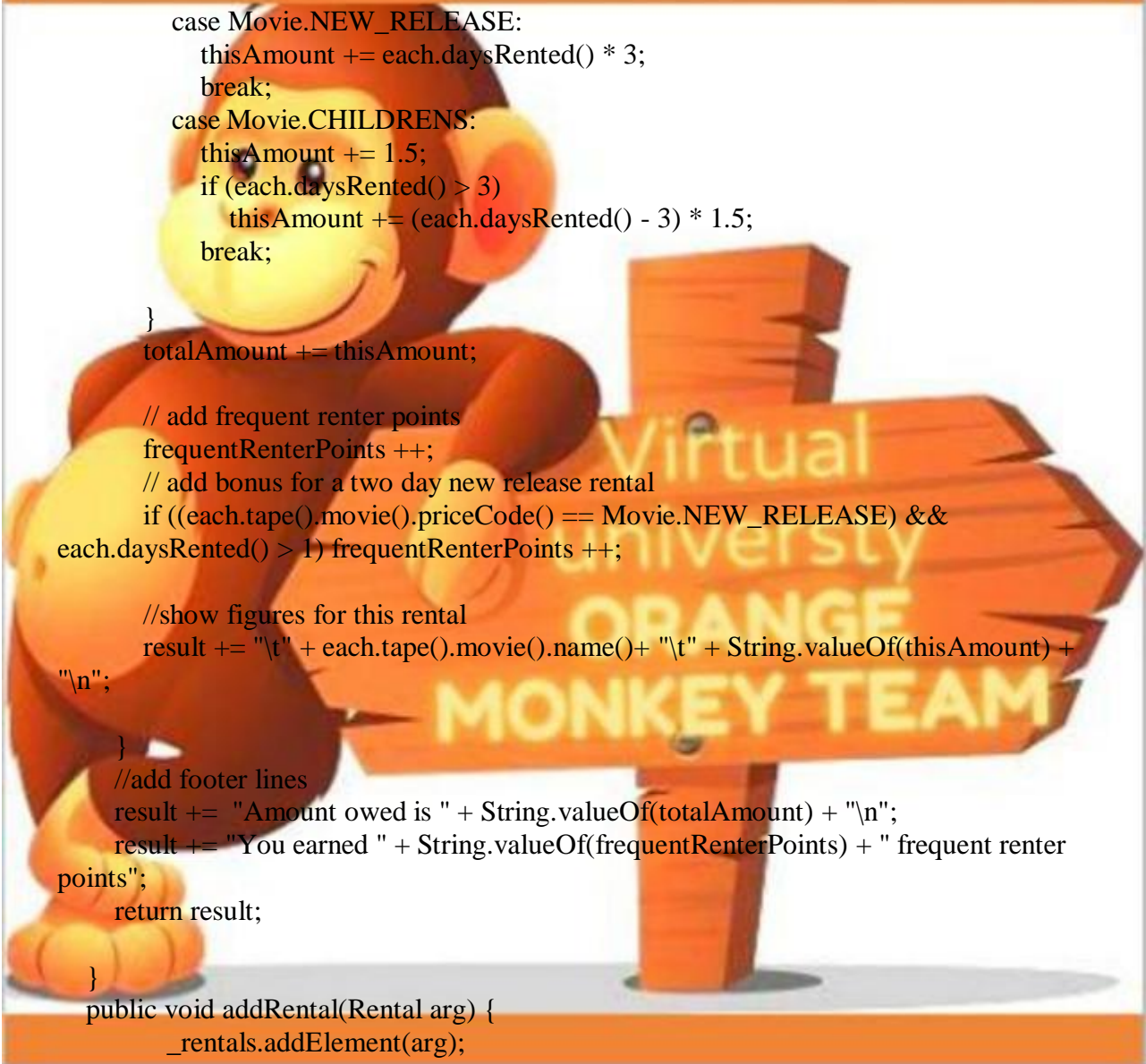
The customer class represents the customer. So far all the classes have been dumb encapsulated data. Customer holds all the behavior for producing a statement in its statement() method.

```

class Customer extends DomainObject
{
    public Customer(String name) {
        _name = name;
    }
    public String statement() {
        double totalAmount = 0;
        int frequentRenterPoints = 0;
        Enumeration rentals = _rentals.elements();
        String result = "Rental Record for " + name() + "\n";
        while (rentals.hasMoreElements()) {
            double thisAmount = 0;
            Rental each = (Rental) rentals.nextElement();

            //determine amounts for each line
            switch (each.tape().movie().priceCode()) {
                case Movie.REGULAR:
                    thisAmount += 2;
                    if (each.daysRented() > 2)
                        thisAmount += (each.daysRented() - 2) * 1.5;
                    break;

```



```

        case Movie.NEW_RELEASE:
            thisAmount += each.daysRented() * 3;
            break;
        case Movie.CHILDRENS:
            thisAmount += 1.5;
            if (each.daysRented() > 3)
                thisAmount += (each.daysRented() - 3) * 1.5;
            break;
    }
    totalAmount += thisAmount;

    // add frequent renter points
    frequentRenterPoints ++;
    // add bonus for a two day new release rental
    if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
        each.daysRented() > 1) frequentRenterPoints ++;

    //show figures for this rental
    result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(thisAmount) +
    "\n";
    }
    //add footer lines
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
    points";
    return result;
}

public void addRental(Rental arg) {
    _rentals.addElement(arg);
}

public static Customer get(String name) {
    return (Customer) Registrar.get("Customers", name);
}

public void persist() {
    Registrar.add("Customers", this);
}

private Vector _rentals = new Vector();
}

```

What are your impressions about the design of this program? I would describe as not well designed, and certainly not object-oriented. For a simple program is this, that does not really matter. Thereís nothing wrong with a quick and dirty *simple* program. But if we imagine this as a fragment of a more complex system, then I have some real problems with this program. That long statement routine in the Customer does far too much. Many of the things that it does should really be done by the other classes.

This is really brought out by a new requirement, just in from the users, they want a similar statement in html. As you look at the code you can see that it is impossible to reuse any of the behavior of the current statement() method for an htmlStatement(). Your only recourse is to write a whole new method that duplicates much of the behavior of statement(). Now of course this is not too onerous. You can just copy the statement() method and make whatever changes you need. So the lack of design does not do too much to hamper the writing of htmlStatement(), (although it might be tricky to figure out exactly where to do the changes). But what happens when the charging rules change? You have to fix both statement() and htmlStatement(), and ensure the fixes are consistent. The problem from cut and pasting code comes when you have to change it later. Thus if you are writing a program that you don't expect to change, then cut and paste is fine. If the program is long lived and likely to change, then cut and paste is a menace.

But you still have to write the htmlStatement() program. You may feel that you should not touch the existing statement() method, after all it works fine. Remember the old engineering adage: "if it ain't broke, don't fix it". statement() may not be broke, but it does hurt. It is making your life more difficult to write the htmlStatement() method.

So this is where refactoring comes in. When you find you have to add a feature to a program, and the program's code is not structured in a convenient way to add the feature; then first refactor the program to make it easy to add the feature, then add the feature.


Extracting the Amount Calculation

The obvious first target of my attention is the overly long statement() method. When I look at a long method like that, I am looking to take a chunk of the code an *extract a method* from it.

Extracting a method is taking the chunk of code and making a method out of it. An obvious piece here is the switch statement (which I'm highlighting below).

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        double thisAmount = 0;
        Rental each = (Rental) rentals.nextElement();

        //determine amounts for each line
        switch (each.tape().movie().priceCode()) {
            case Movie.REGULAR:
                thisAmount += 2;
                if (each.daysRented() > 2)
                    thisAmount += (each.daysRented() - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                thisAmount += each.daysRented() * 3;
                break;
```



```

case Movie.CHILDRENS:
    thisAmount += 1.5;
    if (each.daysRented() > 3)
        thisAmount += (each.daysRented() - 3) * 1.5;
    break;
}
totalAmount += thisAmount;

// add frequent renter points
frequentRenterPoints ++;
// add bonus for a two day new release rental
if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
each.daysRented() > 1) frequentRenterPoints ++;

//show figures for this rental
result += "\t" + each.tape().movie().name()+ "\t" + String.valueOf(thisAmount) +
"\n";
}
//add footer lines
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
return result;
}

```

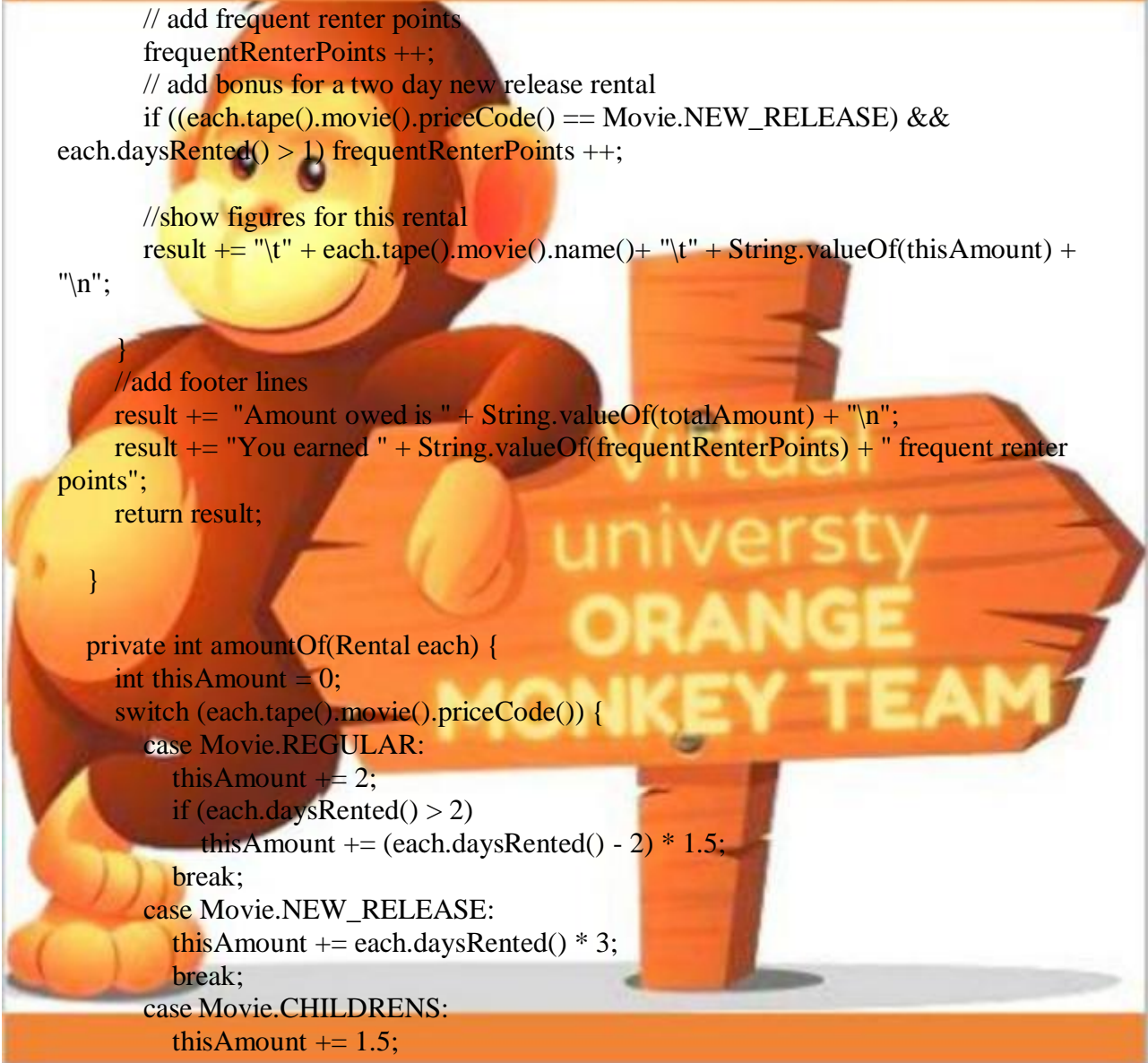
This looks like it would make a good chunk to extract into its own method. When we extract a method, we need to look in the fragment for any variables that are local in scope to the method we are looking at, that local variables and parameters. This segment of code uses two: each and thisAmount. Of these each is not modified by the code but thisAmount is modified. Any non-modified variable we can pass in as a parameter. Modified variables need more care. If there is only one we can return it. The temp is initialized to 0 each time round the loop, and not altered until the switch gets its hands on it. So we can just assign the result. The extraction looks like this.

```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        double thisAmount = 0;
        Rental each = (Rental) rentals.nextElement();

        //determine amounts for each line
        thisAmount = amountOf(each);
        totalAmount += thisAmount;
    }
}

```



```

// add frequent renter points
frequentRenterPoints ++;
// add bonus for a two day new release rental
if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
each.daysRented() > 1) frequentRenterPoints ++;

//show figures for this rental
result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(thisAmount) +
"\n";
}
//add footer lines
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
return result;
}

private int amountOf(Rental each) {
int thisAmount = 0;
switch (each.tape().movie().priceCode()) {
case Movie.REGULAR:
thisAmount += 2;
if (each.daysRented() > 2)
thisAmount += (each.daysRented() - 2) * 1.5;
break;
case Movie.NEW_RELEASE:
thisAmount += each.daysRented() * 3;
break;
case Movie.CHILDRENS:
thisAmount += 1.5;
if (each.daysRented() > 3)
thisAmount += (each.daysRented() - 3) * 1.5;
break;

}
return thisAmount;
}
}

```

When I did this the tests blew up. A couple of the test figures gave me the wrong answer. I was puzzled for a few seconds then realized what I had done. Foolishly I had made the return type of amountOf() int instead of double.

```

private double amountOf(Rental each) {
double thisAmount = 0;
switch (each.tape().movie().priceCode()) {
case Movie.REGULAR:
thisAmount += 2;
if (each.daysRented() > 2)

```

```

        thisAmount += (each.daysRented() - 2) * 1.5;
        break;
    case Movie.NEW_RELEASE:
        thisAmount += each.daysRented() * 3;
        break;
    case Movie.CHILDRENS:
        thisAmount += 1.5;
        if (each.daysRented() > 3)
            thisAmount += (each.daysRented() - 3) * 1.5;
        break;
    }
    return thisAmount;
}

```

It's the kind of silly mistake that I often make, and it can be a pain to track down as Java converts ints to doubles without complaining (but merrily rounding). Fortunately it was easy to find in this case, because the change was so small. Here is the essence of the refactoring process illustrated by accident. Because each change is so small, any errors are very easy to find. You don't spend long debugging, even if you are as careless as I am.

This refactoring has taken a large method and broken it down into two much more manageable chunks. We can now consider the chunks a bit better. I don't like some of the variables names in `amountOf()` and this is a good place to change them.

```

private double amountOf(Rental aRental) {
    double result = 0;
    switch (aRental.tape().movie().priceCode()) {
        case Movie.REGULAR:
            result += 2;
            if (aRental.daysRented() > 2)
                result += (aRental.daysRented() - 2) * 1.5;
            break;
        case Movie.NEW_RELEASE:
            result += aRental.daysRented() * 3;
            break;
        case Movie.CHILDRENS:
            result += 1.5;
            if (aRental.daysRented() > 3)
                result += (aRental.daysRented() - 3) * 1.5;
            break;
    }
    return result;
}

```

Is that renaming worth the effort? Absolutely. Good code should communicate what it is doing clearly, and variable names are key to clear code. Never be afraid to change the names to things to improve clarity. With good find and replace tools, it is usually not difficult. Strong typing and testing will highlight anything you miss. Remember any fool

can write code that a computer can understand, good programmers write code that humans can understand.

Lecture No. 43

Moving the amount calculation

As I look at `amountOf`, I can see that it uses information from the rental, but does not use information from the customer. This method is thus on the wrong object, it should be moved to the rental. To *move a method* you first copy the code over to rental, adjust it to fit in its new home and compile.

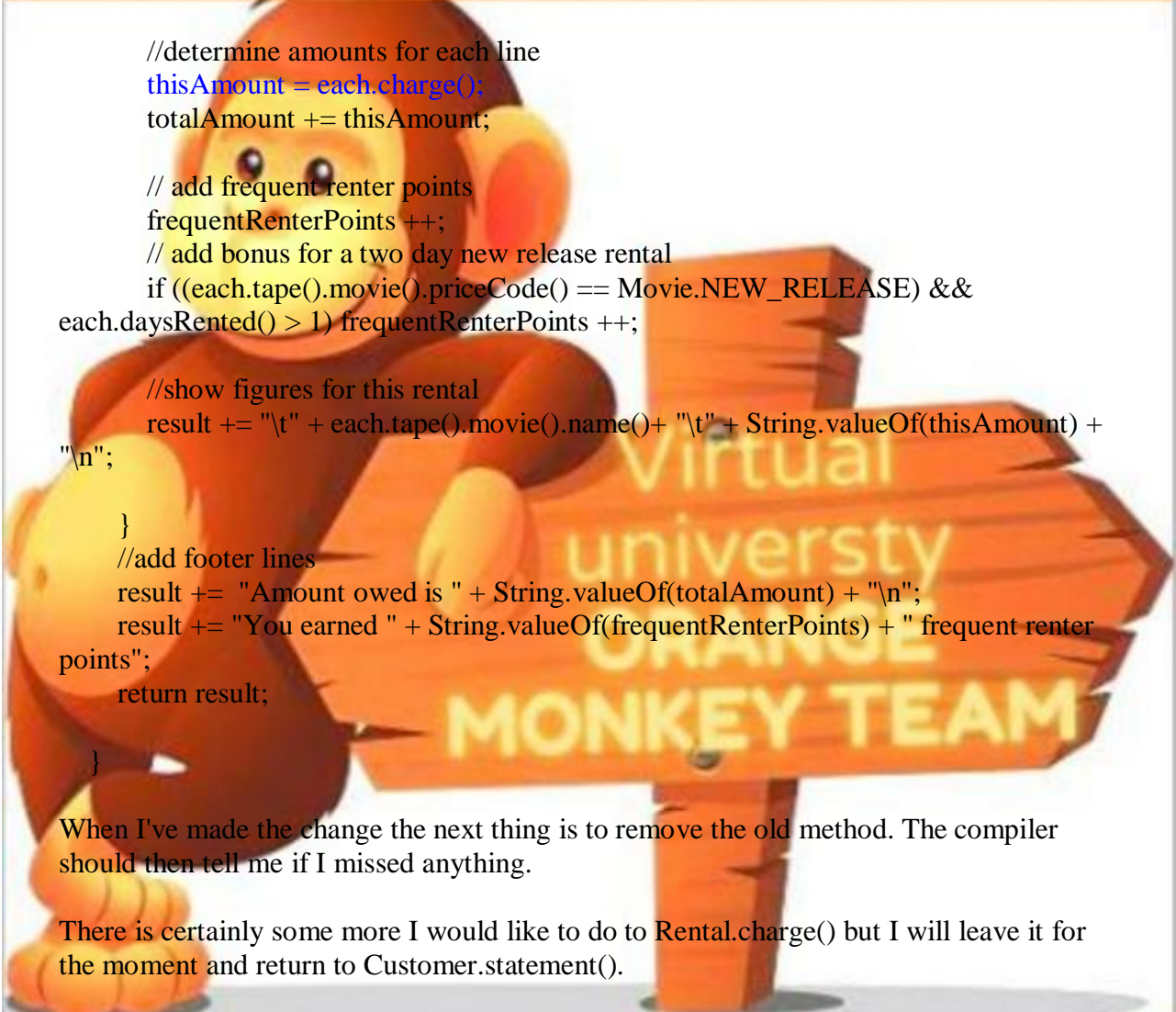
Class Rental

```
double charge() {
    double result = 0;
    switch (tape().movie().priceCode()) {
        case Movie.REGULAR:
            result += 2;
            if (daysRented() > 2)
                result += (daysRented() - 2) * 1.5;
            break;
        case Movie.NEW_RELEASE:
            result += daysRented() * 3;
            break;
        case Movie.CHILDRENS:
            result += 1.5;
            if (daysRented() > 3)
                result += (daysRented() - 3) * 1.5;
            break;
    }
    return result;
}
```

In this case fitting into its new home means removing the parameter.

The next step is to find every reference to the old method, and adjusting the reference to use the new method. In this case this step is easy as we just created the method and it is in only one place. In general, however, you need to do a find across all the classes that might be using that method.

```
public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        double thisAmount = 0;
        Rental each = (Rental) rentals.nextElement();
```



```

//determine amounts for each line
thisAmount = each.charge();
totalAmount += thisAmount;

// add frequent renter points
frequentRenterPoints ++;
// add bonus for a two day new release rental
if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
each.daysRented() > 1) frequentRenterPoints ++;

//show figures for this rental
result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(thisAmount) +
"\n";
}
//add footer lines
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
return result;
}

```

When I've made the change the next thing is to remove the old method. The compiler should then tell me if I missed anything.

There is certainly some more I would like to do to Rental.charge() but I will leave it for the moment and return to Customer.statement().

The next thing that strikes me is that thisAmount() is now pretty redundant. It is set to the result of each.charge() and not changed afterwards. Thus I can eliminate thisAmount by *replacing a temp with a query*.

```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        Rental each = (Rental) rentals.nextElement();

        //determine amounts for each line
        totalAmount += each.charge();

        // add frequent renter points
        frequentRenterPoints ++;
        // add bonus for a two day new release rental
        if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
each.daysRented() > 1) frequentRenterPoints ++;

```

```

        //show figures for this rental
        result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(each.charge())
+ "\n";

    }
    //add footer lines
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
    return result;
}

```

I like to get rid of temporary variables like thus as much as possible. Temps are often a problem in that they cause a lot of parameters to get passed around when they don't need to. You can easily lose track of what they are there for. They are particularly insidious in long methods. Of course there is a small performance price to pay, here the charge is now calculated twice. But it is easy to optimize that in the rental class, and you can optimize much more effectively when the code is properly refactored.

Extracting Frequent Renter Points

The next step is to do a similar thing for the frequent renter points. Again the rules vary with the tape, although there is less variation than with the charging. But it seems reasonable to put the responsibility on the rental. First we need to extract a method from the frequent renter points part of the code (highlighted below).

```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        Rental each = (Rental) rentals.nextElement();

        //determine amounts for each line
        totalAmount += each.charge();

        // add frequent renter points
        frequentRenterPoints ++;
        // add bonus for a two day new release rental
        if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
each.daysRented() > 1) frequentRenterPoints ++;

        //show figures for this rental
        result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(each.charge())
+ "\n";

    }
}

```

```

//add footer lines
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
return result;
}

```

Again we look at the use of locally scoped variables. Again it uses each, which can be passed in as a parameter. The other temp used is frequentRenterPoints. In this case frequentRenterPoints does have a value beforehand. The body of the extracted method doesn't read the value, however, so we don't need to pass it in as a parameter as long as we use an appending assignment.

```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        Rental each = (Rental) rentals.nextElement();

        //determine amounts for each line
        totalAmount += each.charge();

        // add frequent renter points
        frequentRenterPoints += frequentRenterPointOf(each);

        //show figures for this rental
        result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(each.charge())
+ "\n";
    }
}

```

```

//add footer lines
result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
return result;
}

```

```

int frequentRenterPointOf(Rental each) {
    if ((each.tape().movie().priceCode() == Movie.NEW_RELEASE) &&
each.daysRented() > 1) return 2;
    else return 1;
}

```

I did the extraction, compiled and tested, and then did a move. With refactoring small steps are the best, that way less tends to go wrong.



```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {
        Rental each = (Rental) rentals.nextElement();

        //determine amounts for each line
        totalAmount += each.charge();

        // add frequent renter points
        frequentRenterPoints += each.frequentRenterPoints();

        //show figures for this rental
        result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(each.charge())
        + "\n";
    }
    //add footer lines
    result += "Amount owed is " + String.valueOf(totalAmount) + "\n";
    result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
    points";
    return result;
}

int frequentRenterPoints() {
    if ((tape().movie().priceCode() == Movie.NEW_RELEASE) && daysRented() > 1)
    return 2;
    else return 1;
}

```

Removing Temps


As I suggested before, temporary variables can be a problem. They are only useful within their own routine, and thus they encourage long complex routines. In this case we have two temporary variables, both of which are being used to get a total from the rentals attached to the customer. Both the ascii and html versions will require these totals. I like to *replace temps with queries*. Queries are accessible to any method in the class, and thus encourage a cleaner design without long complex methods.

I began by replacing totalAmount with a charge() method on customer.

```

public String statement() {
    double totalAmount = 0;
    int frequentRenterPoints = 0;
    Enumeration rentals = _rentals.elements();
    String result = "Rental Record for " + name() + "\n";
    while (rentals.hasMoreElements()) {

```



```

Rental each = (Rental) rentals.nextElement();

// add frequent renter points
frequentRenterPoints += each.frequentRenterPoints();

//show figures for this rental
result += "\t" + each.tape().movie().name() + "\t" + String.valueOf(each.charge())
+ "\n";
}
//add footer lines
result += "Amount owed is " + String.valueOf(charge()) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints) + " frequent renter
points";
return result;
}

private double charge(){
double result = 0;
Enumeration rentals = _rentals.elements();
while (rentals.hasMoreElements()) {
Rental each = (Rental) rentals.nextElement();
result += each.charge();
}
return result;
}

```

After compiling and testing that refactoring, I then did the same for frequentRenterPoints.

```

public String statement() {
Enumeration rentals = _rentals.elements();
String result = "Rental Record for " + name() + "\n";
while (rentals.hasMoreElements()) {
Rental each = (Rental) rentals.nextElement();
//show figures for each rental
result += "\t" + each.tape().movie().name() + "\t" +
String.valueOf(each.charge()) + "\n";
}
//add footer lines
result += "Amount owed is " + String.valueOf(charge()) + "\n";
result += "You earned " + String.valueOf(frequentRenterPoints()) +
" frequent renter points";
return result;
}

private int frequentRenterPoints() {
int result = 0;
Enumeration rentals = _rentals.elements();
while (rentals.hasMoreElements()) {

```

```

        Rental each = (Rental) rentals.nextElement();
        result += each.frequentRenterPoints();
    }
    return result;
}

```

It is worth stopping and thinking a bit about this refactoring. Most refactorings reduce the amount of code, but this one increases it. That's because Java requires a lot of statements to set up a summing loop. Even a simple summing loop with one line of code per element needs six lines of support around it. It's an idiom that is obvious to any programmer but it is noise that hides what the intent of the loop is. As Java develops and builds up its ability to handle block closures in the style of Smalltalk, I expect that overhead to decrease, probably to the single line that such an expression would take in Smalltalk.

The other concern with this refactoring lies in performance. The old code executed the while loop once, the new code executes it three times. If the while loop takes time, this might significantly impair performance. Many programmers would not do this refactoring simply for this reason. But note the words "if" and "might". While some loops do cause performance issues, most do not. So while refactoring don't worry about this. When you optimize you will have to worry about it, but you will then be in a much better position to do something about it, and you will have more options to optimize effectively. (For a good discussion on why it is better to write clearly first and then optimize, see [McConnell, Code Complete].)

These queries are now available to any code written in the customer class. Indeed they can easily be added to the interface of the class should other parts of the system need this information. Without queries like these, other methods need to deal with knowing about the rentals and building the loops. In a complex system that will lead to much more code to write and maintain.

You can see the difference immediately with the `htmlStatement()`. I am now at the point where I take off my refactoring hat and put on my adding function hat. I can write `htmlStatement()` like this (and add an appropriate test).

```

public String htmlStatement() {
    Enumeration rentals = _rentals.elements();
    String result = "<H1>Rentals for <EM>" + name() + "</EM></H1><P>\n";
    while (rentals.hasMoreElements()) {
        Rental each = (Rental) rentals.nextElement();
        //show figures for each rental
        result += each.tape().movie().name() + ": " +
            String.valueOf(each.charge()) + "<BR>\n";
    }
    //add footer lines
    result += "<P>You owe <EM>" + String.valueOf(charge()) + "</EM><P>\n";
    result += "On this rental you earned <EM>" +
        String.valueOf(frequentRenterPoints()) + "</EM> frequent renter points<P>";
    return result;
}

```

There is still some code copied from the ascii version, but that is mainly due to setting up the loop. Further refactoring could clean that up further, extracting methods for header, footer, and detail line are one route I could take. But that isn't where I want to spend my time, I would like to move onto the methods I've moved onto rental. Back on with the refactoring hat.

Moving the Rental Calculations to Movie

Yes it's that switch statement that is bugging me. It is a bad idea to do a switch based on an attribute of another object. If you must use a switch statement, it should be on your own data, not on someone else's. This implies that the charge should move onto movie

```
Class movie {
    double charge(int daysRented) {
        double result = 0;
        switch (priceCode()) {
            case REGULAR:
                result += 2;
                if (daysRented > 2)
                    result += (daysRented - 2) * 1.5;
                break;
            case NEW_RELEASE:
                result += daysRented * 3;
                break;
            case CHILDRENS:
                result += 1.5;
                if (daysRented > 3)
                    result += (daysRented - 3) * 1.5;
                break;
        }
        return result;
    }
}
```

For this to work I have to pass in the length of the rental, which of course is data due of the rental. The method effectively uses two pieces of data, the length of the rental and the type of the movie. Why do I prefer to pass the length of rental rather than the movie's type? It's because type information tends to be more volatile. I can easily imagine new types of videos appearing. If I change the movie's type I want the least ripple effect, so I prefer to calculate the charge within the movie.

I compiled the method into movie and then adjusted the charge method on rental to use the new method.

```
Class rental {
    double charge() {
        return _tape.movie().charge(_daysRented);
    }
}
```

Some people would prefer to remove that chain of calls by having a charge(int) message on tape. This would lead to

```
Class rental
  double charge() {
    return _tape.charge(_daysRented);
  }
```

```
Class tape
  double charge() {
    return _movie.charge(_daysRented);
  }
```

You can make that change if you like, I don't tend to worry about message chains providing that they all lie in the same package. If they cross package boundaries, then I'm not so happy, and would add an insulating method.

Having done this with charge amounts, I'm inclined to do the same with frequent renter points. The need is less pressing, but I think it is more consistent to do them both the same way. And again if the movie classifications change it makes it easier to update the code.

```
Class rental
  int frequentRenterPoints() {
    return _tape.movie().frequentRenterPoints(_daysRented);
  }
```

```
class movie
  int frequentRenterPoints(int daysRented){
    if ((priceCode() == NEW_RELEASE) && daysRented > 1) return 2;
    else return 1;
  }
```

With these two changes I can hide those constants, which is generally a Good Thing. Even constant data should be private.

```
private static final int CHILDRENS = 2;
private static final int REGULAR = 0;
private static final int NEW_RELEASE = 1;
```

To really do this, however, I need to change a couple of other parts of the class. I need to change how we create a movie. I used to create a movie with a message like

```
new Movie ("Ran", Movie.REGULAR);
```

and the constructor

```
class Movie
  private Movie(String name, int priceCode) {
    _name = name;
```

```
        _priceCode = priceCode;
    }

```

To keep this type code hidden I need some creation methods.

```
public static Movie newNewRelease(String name){
    return new Movie (name, NEW_RELEASE);
}
public static Movie newRegular(String name){
    return new Movie (name, REGULAR);
}
public static Movie newChildrens(String name) {
    return new Movie (name, CHILDRENS);
}

```

Now I create a new movie with

```
Movie.newRegular("Monty Python and the Holy Grail");

```

Movies can change their classification. I change a movie's classification with

```
aMovie.setPriceCode(Movie.REGULAR);

```

I will need to add a bunch of methods to handle the changes of classification.

```
public void beRegular() {
    _priceCode = REGULAR;
}

public void beNewRelease() {
    _priceCode = NEW_RELEASE;
}

```

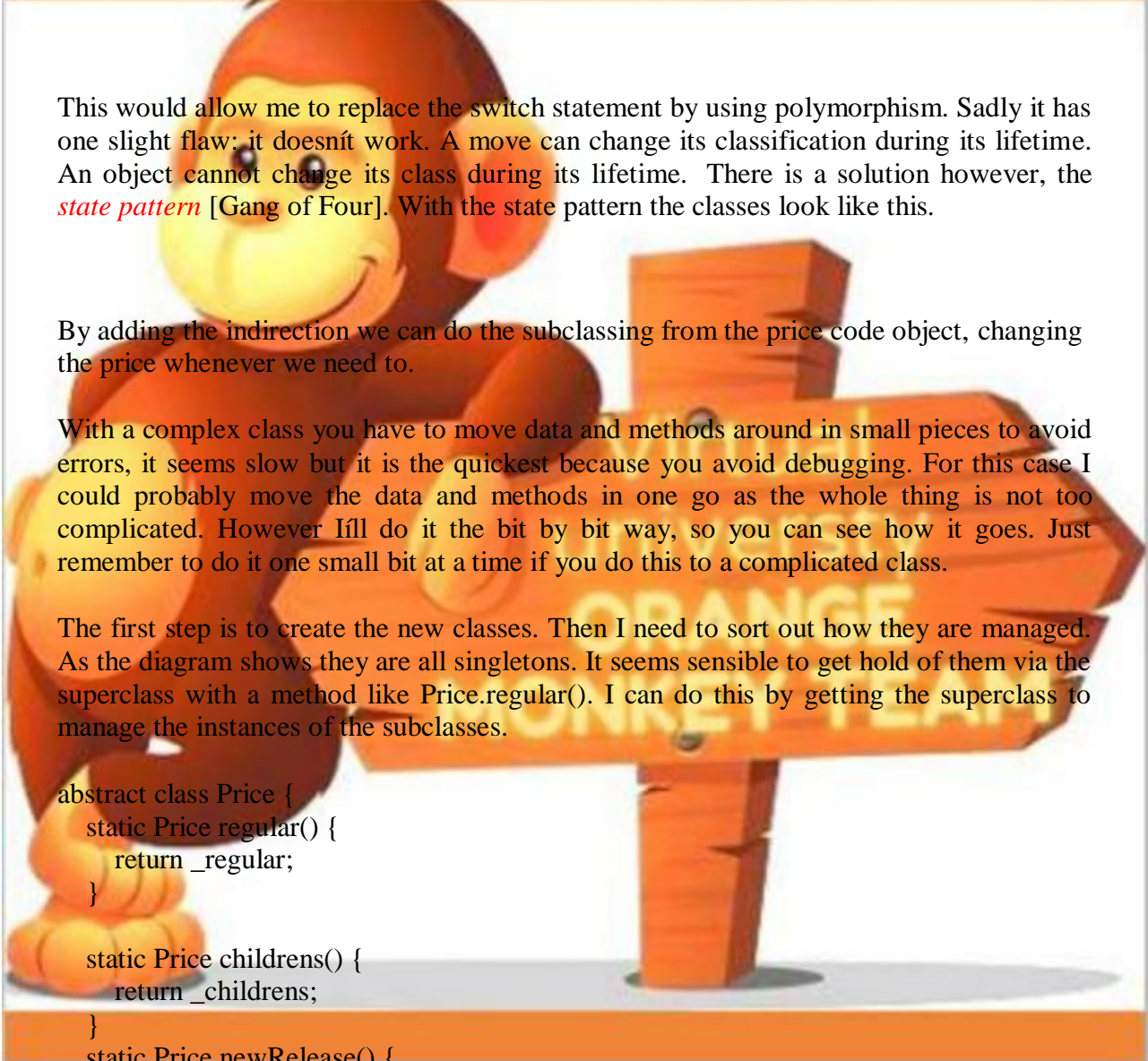
```
public void beChildrens() {
    _priceCode = CHILDRENS;
}

```

It's a bit of effort to set up these methods, but they are a much more explicit interface than the type codes. Just looking at the name of the method tells you what kind of movie you are getting. This makes the code more understandable. The trade off is that each time I add a price code I have to add a creation and update method. If I had lots of price codes this would hurt (so I wouldn't do it). If I have a few, however, then it's quite reasonable.

At last! inheritance

So we have several types of movie, which have different ways of answering the same question. This sounds like a job for subclasses. We could have three subclasses of movie, each of which can have its own version of charge.



This would allow me to replace the switch statement by using polymorphism. Sadly it has one slight flaw: it doesn't work. A movie can change its classification during its lifetime. An object cannot change its class during its lifetime. There is a solution however, the *state pattern* [Gang of Four]. With the state pattern the classes look like this.

By adding the indirection we can do the subclassing from the price code object, changing the price whenever we need to.

With a complex class you have to move data and methods around in small pieces to avoid errors, it seems slow but it is the quickest because you avoid debugging. For this case I could probably move the data and methods in one go as the whole thing is not too complicated. However I'll do it the bit by bit way, so you can see how it goes. Just remember to do it one small bit at a time if you do this to a complicated class.

The first step is to create the new classes. Then I need to sort out how they are managed. As the diagram shows they are all singletons. It seems sensible to get hold of them via the superclass with a method like `Price.regular()`. I can do this by getting the superclass to manage the instances of the subclasses.

```
abstract class Price {
    static Price regular() {
        return _regular;
    }

    static Price childrens() {
        return _childrens;
    }
    static Price newRelease() {
        return _newRelease;
    }

    private static Price _childrens = new ChildrensPrice();
    private static Price _newRelease = new NewReleasePrice();
    private static Price _regular = new RegularPrice();
}
```

Now I can begin to move the data over. The first piece of data to move over is the price code. Of course I'm not actually going to use the price code within the `Price` object, but I will give it the illusion of doing so. That way the old methods will still work. The key is to modify those methods that access and update the price code value within `Movie`. My first step is to *self-encapsulate the type code*, ensuring that all uses of the type code go through getting and setting methods. Since most of the code came from other classes, most methods already use the getting method. However the constructors do access the price code, I can use the setting methods instead.

```
public static Movie newNewRelease(String name){
```

```

Movie result = new Movie (name);
result.beNewRelease();
return result;
    }
    public static Movie newRegular(String name){
Movie result = new Movie (name);
result.beRegular();
return result;
    }
    public static Movie newChildrens(String name) {
Movie result = new Movie (name);
result.beChildrens();
return result;
    }

    private Movie(String name) {
        _name = name;
    }

```

After compiling and testing I now change getting and setting methods to use the new class.

```

public void beRegular() {
    _price = Price.regular();
}

public void beNewRelease() {
    _price = Price.newRelease();
}

public void beChildrens() {
    _price = Price.childrens();
}

public int priceCode() {
    return _price.priceCode();
}

```

And provide the priceCode methods on Price and its subclasses.

Class PriceÖ

```

abstract int priceCode();

```

Class RegularPriceÖ

```

int priceCode(){
    return Movie.REGULAR;
}

```

To do this I need to make the constants non-private again. This is fine, I don't mind them having a little fame before they bite the dust.

I can now compile and test and the more complex methods don't realize the world has changed.

After moving the data I can now start moving methods. My prime target is the charge() method. It is simple to move.

```
Class MovieÖ
    double charge(int daysRented) {
        return _price.charge(daysRented);
    }

Class PriceÖ
    double charge(int daysRented) {
        double result = 0;
        switch (priceCode()) {
            case Movie.REGULAR:
                result += 2;
                if (daysRented > 2)
                    result += (daysRented - 2) * 1.5;
                break;
            case Movie.NEW_RELEASE:
                result += daysRented * 3;
                break;
            case Movie.CHILDRENS:
                result += 1.5;
                if (daysRented > 3)
                    result += (daysRented - 3) * 1.5;
                break;
        }
        return result;
    }
}
```

Once it is moved I can start *replacing the case statement with inheritance*. I do this by taking one leg of the case statement at a time, and creating an overriding method. I start with RegularPrice.

```
Class RegularPriceÖ
    double charge(int daysRented){
        double result = 2;
        if (daysRented > 2)
            result += (daysRented - 2) * 1.5;
        return result;
    }
}
```

This will override the parent case statement, which I just leave as it is. I compile and test for this case, then take the next leg, compile and test. (To make sure I'm executing the subclass code, I like to throw in a deliberate bug and run it to ensure the tests blow up. Not that I'm paranoid or anything.)

```

Class ChildrensPrice
double charge(int daysRented){
    double result = 1.5;
    if (daysRented > 3)
        result += (daysRented - 3) * 1.5;
    return result;
}

```

```

Class NewReleasePriceÖ
double charge(int daysRented){
    return daysRented * 3;
}

```

When I've done that with all the legs, I declare the Price.charge() method abstract.

```

Class PriceÖ
abstract double charge(int daysRented);

```

I can now do the same procedure with frequentRenterPoints(). First I move the method over to Price.

```

Class MovieÖ
int frequentRenterPoints(int daysRented){
    return _price.frequentRenterPoints(daysRented);
}

```

```

Class PriceÖ
int frequentRenterPoints(int daysRented){
    if ((priceCode() == Movie.NEW_RELEASE) && daysRented > 1) return 2;
    else return 1;
}

```

In this case, however I won't make the superclass method abstract. Instead I will create an overriding method for new releases, and leave a defined method (as the default) on the superclass.

```

Class NewReleasePrice
int frequentRenterPoints(int daysRented){
    return (daysRented > 1) ?
        2:
        1;
}

```

```

Class PriceÖ
int frequentRenterPoints(int daysRented){
    return 1;
}

```

Now I have removed all the methods that needed a price code. So I can get rid of the price code methods and data on both Movie and Price.

Putting in the state pattern was quite an effort, was it worth it? The gain is now that should I change any of priceís behavior, add new prices, or add extra price dependent behavior; it will be much easier to change. The rest of the application does not know about the use of the state pattern. For the tiny amount of behavior I currently have it is not a big deal. But in a more complex system with a dozen or so price dependent methods this would make a big difference. All these changes were small steps, it seems slow to write it like this, but not once did I have to open the debugger. So the process actually flowed quite quickly.



Lecture No. 44

Capability Maturity Model Integration (CMMI)

Capability Maturity Model or CMM is a reference model of mature practices in a specified discipline, used to assess a group's capability to perform that discipline. In fact there are a number of CMMs. They differ by discipline (software, systems, acquisition, etc.), structure (staged versus continuous), how maturity is defined (process improvement path), and how capability is defined (institutionalization). Hence "Capability Maturity Model®" and CMM® are used by the Software Engineering Institute (SEI) to denote a particular class of maturity models.

Software CMM, the one we saw earlier on, is one of the many CMM models.

Name	Structure	Domain
Software CMM	staged	software development
System Engineering CMM	continuous	system engineering
Software Acquisition CMM	staged	software acquisition
System Security Engineering CMM	continuous	security engineering
FAA-iCMM	continuous	software engineering, systems engineering, and acquisition
IPD-CMM	hybrid	integrated product development
People CMM	staged	workforce
SPICE Model	continuous	software development

Since these models have different structure and application domains, an organization could potentially use many of these models for their different activities, at times it could become problematic for them.

CMMI Integrates systems and software disciplines into single process improvement framework and provides a framework for introducing new disciplines as needs arise. It can now be applied to just the software engineering projects in an organization or for the entire spectrum of activities outlined above.

CMMI Representations

A representation allows an organization to pursue different improvement objectives. There are two types of representations in the CMMI models: staged and continuous. The organization and presentation of the data are different in each representation. However, the content is the same.

Staged Representation

Staged representation is classical representation we have already seen previously. It:

- Provides a proven sequence of improvements, each serving as a foundation for the next.
- Permits comparisons across and among organizations by the use of maturity levels.

- Provides an easy migration from the SW-CMM to CMMI.
- Provides a single rating that summarizes appraisal results and allows comparisons among organizations

This representation indicates maturity of an organization's standard process -- to answer, "What is a good order for approaching improvement across the organization?"

You may recall that a maturity level is a well-defined evolutionary plateau of process improvement. There are five maturity levels and each level is a layer in the foundation for continuous process improvement using a proven sequence of improvements, beginning with basic management practices and progressing through a predefined and proven path of successive levels.

Each maturity level provides a necessary foundation for effective implementation of processes at the next level. Higher level processes have less chance of success without the discipline provided by lower levels. The effect of innovation can be obscured in a noisy process.

Higher maturity level processes may be performed by organizations at lower maturity levels, with the risk of not being consistently applied in a crisis.

Continuous Representation

Continuous representation allows you to select the order of improvement that best meets your organization's business objectives and mitigates your organization's areas of risk. It enables comparisons across and among organizations on a process-area-by-process-area basis and provides an easy migration from EIA 731 (and other models with a continuous representation) to CMMI.

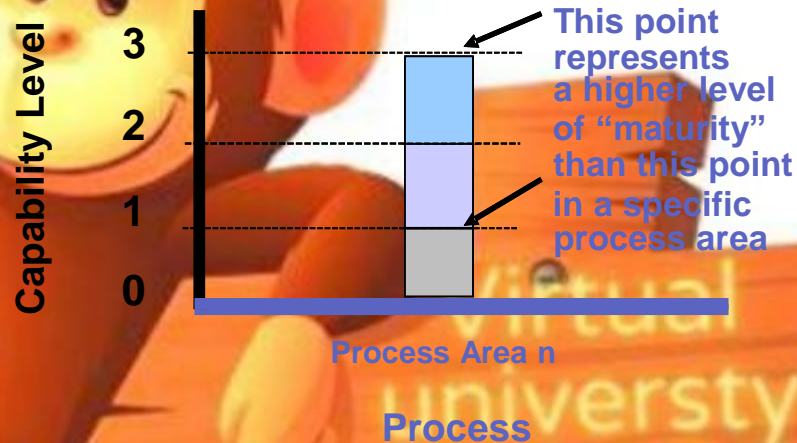
As opposed to the staged representation, it indicates improvement within a single process area -- to answer, "What is a good order for approaching improvement of this process area?"

Capability Levels

A capability level is a well-defined evolutionary plateau describing the organization's capability relative to a process area. There are six capability levels. For capability levels 1-5, there is an associated generic goal. Each level is a layer in the foundation for continuous process improvement. Thus, capability levels are cumulative, i.e., a higher capability level includes the attributes of the lower levels. The five (actually six) capability levels (starting from 0) are enumerated below in the reverse order, 5 being the highest and 0 being the lowest.

- 5 Optimizing
- 4 Quantitatively Managed
- 3 Defined
- 2 Managed
- 1 Performed
- 0 Incomplete

The process area capability of an implemented process can be represented by a bar as shown below.



Relating Process Area Capability and Organizational Maturity

Organizational maturity is the focus of the staged representation, whereas process area capability is the focus of the continuous representation.

Organizational maturity and process area capability are similar concepts. The difference between them is that organizational maturity pertains to a set of process areas across an organization, while process area capability deals with a set of processes relating to a single process area or specific practice.

Comparison of Representations

Staged

- Process improvement is measured using maturity levels.
- Maturity level is the degree of process improvement across a predefined set of process areas.
- Organizational maturity pertains to the "maturity" of a set of processes across an organization

Continuous

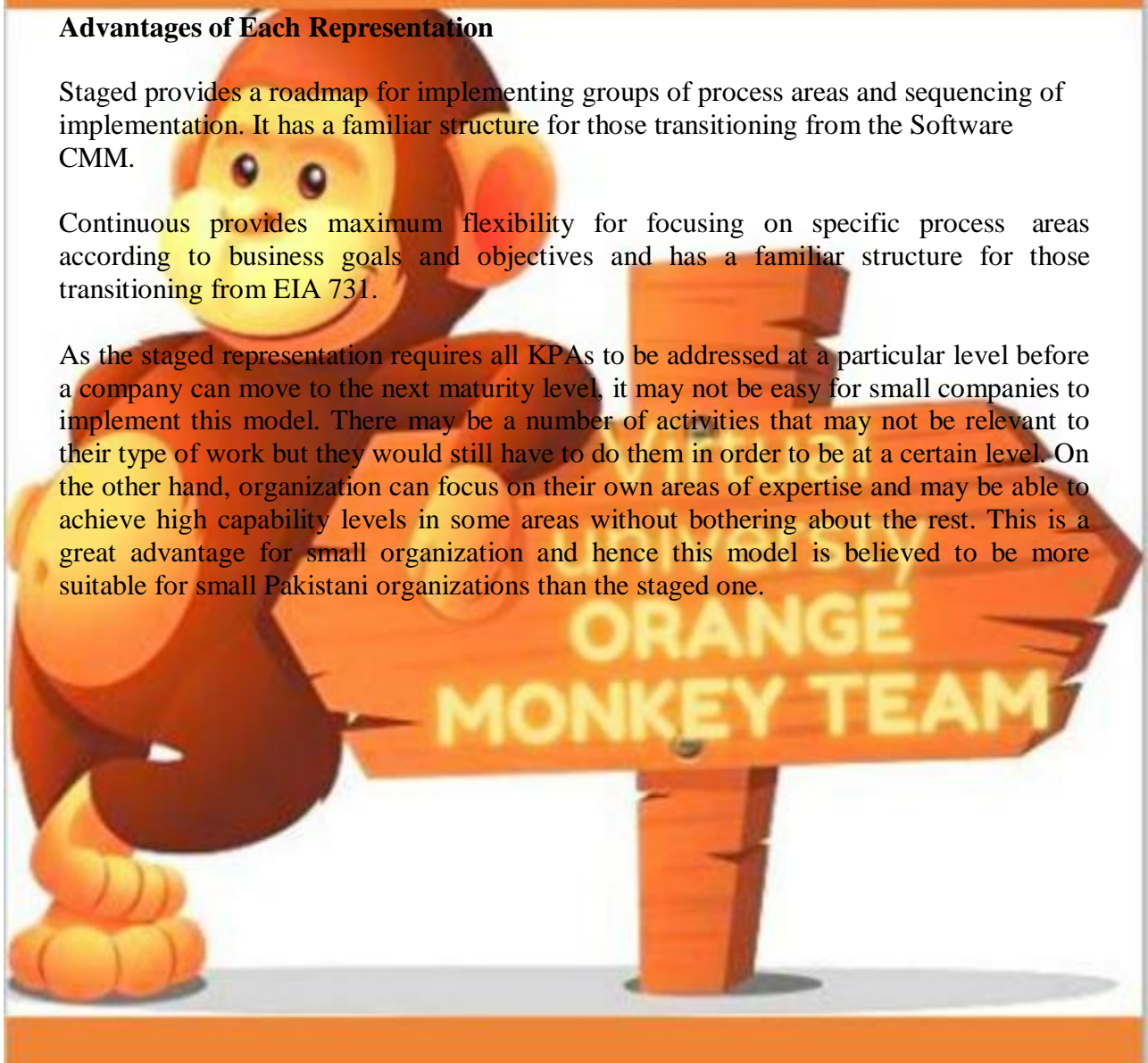
- Process improvement is measured using capability levels.
- Capability level is the achievement of process improvement within an individual process area.
- Process area capability pertains to the "maturity" of a particular process across an organization.

Advantages of Each Representation

Staged provides a roadmap for implementing groups of process areas and sequencing of implementation. It has a familiar structure for those transitioning from the Software CMM.

Continuous provides maximum flexibility for focusing on specific process areas according to business goals and objectives and has a familiar structure for those transitioning from EIA 731.

As the staged representation requires all KPAs to be addressed at a particular level before a company can move to the next maturity level, it may not be easy for small companies to implement this model. There may be a number of activities that may not be relevant to their type of work but they would still have to do them in order to be at a certain level. On the other hand, organization can focus on their own areas of expertise and may be able to achieve high capability levels in some areas without bothering about the rest. This is a great advantage for small organization and hence this model is believed to be more suitable for small Pakistani organizations than the staged one.





Lecture No. 45

CMM Overview

CMM Maturity Levels

There are five levels defined along the continuum of the CMM and, according to the SEI: "Predictability, effectiveness, and control of an organization's software processes are believed to improve as the organization moves up these five levels. While not rigorous, the empirical evidence to date supports this belief."

Level 1 - Ad hoc (Chaotic)

It is characteristic of processes at this level that they are (typically) undocumented and in a state of dynamic change, tending to be driven in an *ad hoc*, uncontrolled and reactive manner by users or events. This provides a chaotic or unstable environment for the processes.

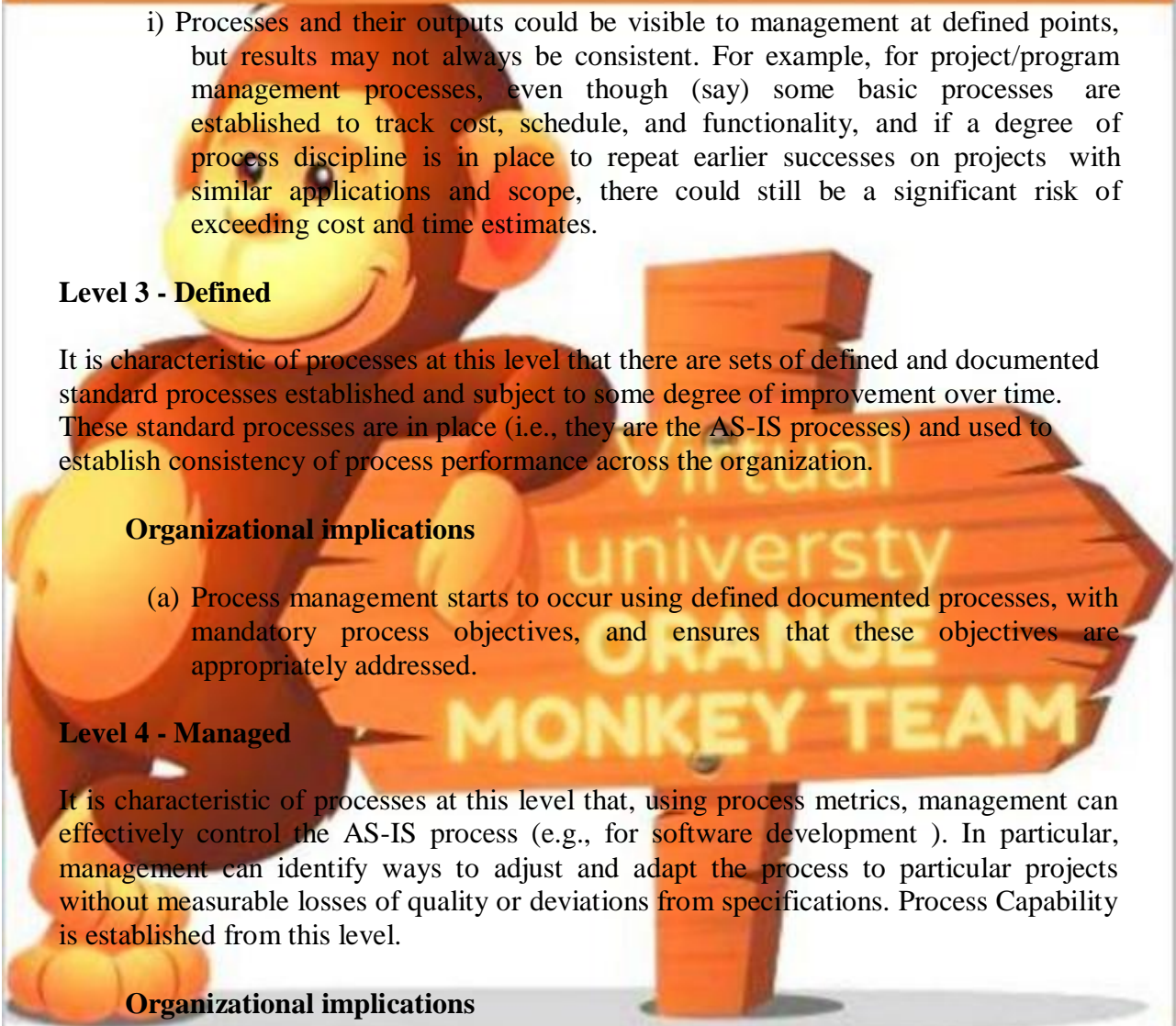
Organizational implications

- (a) Institutional knowledge tends to be scattered (there being limited structured approach to knowledge management) in such environments, not all of the stakeholders or participants in the processes may know or understand all of the components that make up the processes. As a result, process performance in such organizations is likely to be variable (inconsistent) and depend heavily on the institutional knowledge, or the competence, or the heroic efforts of relatively few people or small groups.
- (b) Despite the chaos, such organizations manage to produce products and services. However, in doing so, there is significant risk that they will tend to exceed any estimated budgets or schedules for their projects - it being difficult to estimate what a process will do when you do not fully understand the process (what it is that you do) in the first place and cannot therefore control it or manage it effectively.
- (c) Due to the lack of structure and formality, organizations at this level may over-commit, or abandon processes during a crisis, and it is unlikely that they will be able to repeat past successes. There tends to be limited planning, limited executive commitment or buy-in to projects, and limited acceptance of processes.

Level 2 - Repeatable

It is characteristic of processes at this level that some processes are repeatable, possibly with consistent results. Process discipline is unlikely to be rigorous, but where it exists it may help to ensure that existing processes are maintained during times of stress.

Organizational implications

- 
- i) Processes and their outputs could be visible to management at defined points, but results may not always be consistent. For example, for project/program management processes, even though (say) some basic processes are established to track cost, schedule, and functionality, and if a degree of process discipline is in place to repeat earlier successes on projects with similar applications and scope, there could still be a significant risk of exceeding cost and time estimates.

Level 3 - Defined

It is characteristic of processes at this level that there are sets of defined and documented standard processes established and subject to some degree of improvement over time. These standard processes are in place (i.e., they are the AS-IS processes) and used to establish consistency of process performance across the organization.

Organizational implications

- (a) Process management starts to occur using defined documented processes, with mandatory process objectives, and ensures that these objectives are appropriately addressed.

Level 4 - Managed

It is characteristic of processes at this level that, using process metrics, management can effectively control the AS-IS process (e.g., for software development). In particular, management can identify ways to adjust and adapt the process to particular projects without measurable losses of quality or deviations from specifications. Process Capability is established from this level.

Organizational implications

- a) Quantitative quality goals tend to be set for process output - e.g., software or software maintenance.
- b) Using quantitative/statistical techniques, process performance is measured and monitored and generally predictable and controllable also.

Level 5 - Optimizing

It is a characteristic of processes at this level that the focus is on continually improving process performance through both incremental and innovative technological changes/improvements.

Organizational implications

- (a) Quantitative process-improvement objectives for the organization are established, continually revised to reflect changing business objectives, and used as criteria in managing process improvement. Thus, process improvements to address common causes of process variation and measurably improve the organization's processes are identified, evaluated, and deployed.

- (b) The effects of deployed process improvements are measured and evaluated against the quantitative process-improvement objectives.
- (c) Both the defined processes and the organization's set of standard processes are targets for measurable improvement activities.
- (d) A critical distinction between maturity level 4 and maturity level 5 is the type of process variation addressed.

At maturity level 4, processes are concerned with addressing statistical *special causes* of process variation and providing statistical predictability of the results, and though processes may produce predictable results, the results may be insufficient to achieve the established objectives.

At maturity level 5, processes are concerned with addressing statistical *common causes* of process variation and changing the process (for example, shifting the mean of the process performance) to improve process performance. This would be done at the same time as maintaining the likelihood of achieving the established quantitative process-improvement objectives.

Extensions

Some versions of CMMI from SEI indicate a "level 0", characterized as "Incomplete". Some pundits leave this level out as redundant or unimportant, but Pressman and others make note of it.

Comparison of CMMI Representations Staged Continuous

What is a CMMI model representation? The answer requires an explanation of the structure of CMMI models. The basic building blocks in every CMMI model are called "process areas." A process area does not describe *how* an effective process is executed (e.g., entrance and exit criteria, roles of participants, resources). Instead, a process area describes *what* those using an effective process do (practices) and *why* they do those things (goals).

Selecting a Representation

When making the decision about which architectural representation to use for process improvement, comparative advantages of each approach as represented in the given Table.

Continuous Representation	Staged Representation
Grants explicit freedom to select the order of improvement that best meets the organization's business objectives.	Enables organizations to have a predefined and proven path.
Enables increased visibility into the capability achieved within each individual process area.	Builds on a relatively long history of use.
Supports a focus on risks specific to	Case studies and data exist that show return on investment.
	Permits comparisons across and among

individual process areas.

Affords a more direct comparison of process improvement to ISO 15504 because the organization of process areas is derived from 15504.

Allows the generic practices from higher capability levels to be more evenly and completely applied to all of the process areas.

organizations.

Introduces a sequence of improvements, beginning with basic management practices and progressing through successive levels, each serving as a foundation for the next.

Summarizes process-improvement results in a simple form—a single maturity-level number.

Project Management Concerns

Project management is the discipline of planning, organizing and managing resources to bring about the successful completion of specific project goals and objectives.

A project is a finite endeavor (having specific start and completion dates) undertaken to create a unique product or service which brings about beneficial change or added value. This finite characteristic of projects stands in sharp contrast to processes, or operations, which are permanent or semi-permanent functional work to repetitively produce the same product or service. In practice, the management of these two systems is often found to be quite different, and as such requires the development of distinct technical skills and the adoption of separate management.

There are several approaches that can be taken to managing project activities including agile, interactive, incremental, and phased approaches.

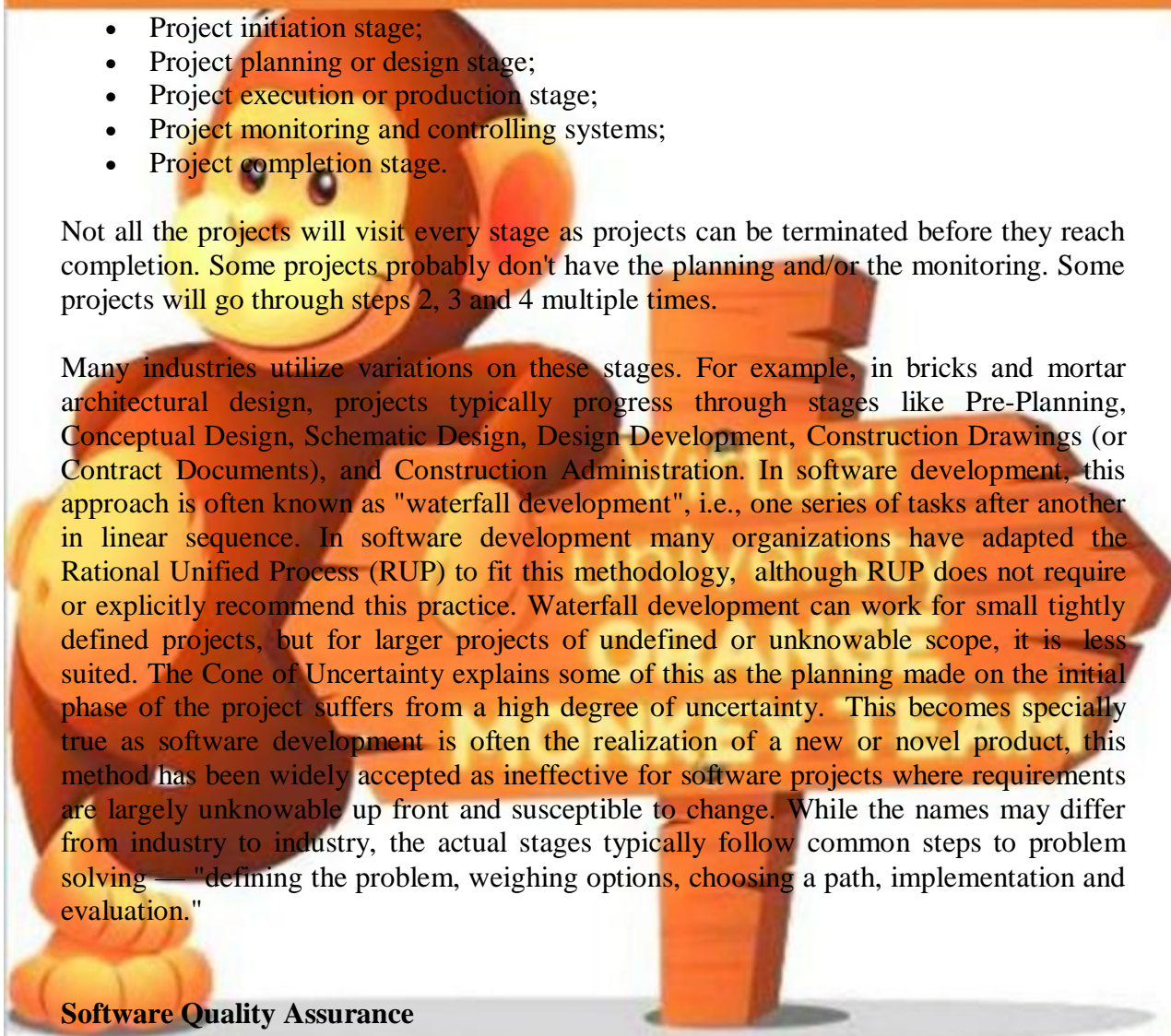
Regardless of the approach employed, careful consideration needs to be given to clarify surrounding project objectives, goals, and importantly, the roles and responsibilities of all participants and stakeholders.

The traditional approach

A traditional phased approach identifies a sequence of steps to be completed. In the "traditional approach", we can distinguish 5 components of a project (4 stages plus control) in the development of a project:



Typical development phases of a project

- 
- Project initiation stage;
 - Project planning or design stage;
 - Project execution or production stage;
 - Project monitoring and controlling systems;
 - Project completion stage.

Not all the projects will visit every stage as projects can be terminated before they reach completion. Some projects probably don't have the planning and/or the monitoring. Some projects will go through steps 2, 3 and 4 multiple times.

Many industries utilize variations on these stages. For example, in bricks and mortar architectural design, projects typically progress through stages like Pre-Planning, Conceptual Design, Schematic Design, Design Development, Construction Drawings (or Contract Documents), and Construction Administration. In software development, this approach is often known as "waterfall development", i.e., one series of tasks after another in linear sequence. In software development many organizations have adapted the Rational Unified Process (RUP) to fit this methodology, although RUP does not require or explicitly recommend this practice. Waterfall development can work for small tightly defined projects, but for larger projects of undefined or unknowable scope, it is less suited. The Cone of Uncertainty explains some of this as the planning made on the initial phase of the project suffers from a high degree of uncertainty. This becomes specially true as software development is often the realization of a new or novel product, this method has been widely accepted as ineffective for software projects where requirements are largely unknowable up front and susceptible to change. While the names may differ from industry to industry, the actual stages typically follow common steps to problem solving — "defining the problem, weighing options, choosing a path, implementation and evaluation."

Software Quality Assurance

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or CMMI.

This definition emphasizes upon three important points:

- Software requirements are the foundation from which quality is measured. Lack of conformance is lack of quality
- Specified standards define a set of development criteria that guide the manner in which software is engineered. If the criteria are not followed, lack of quality will almost surely result.
- A set of implicit requirements often goes unmentioned (ease of use, good maintainability etc.)

Another very important question is: Do you need to worry about it after the code has been generated? In fact, SQA is an umbrella activity that is applied throughout the software process.

Quality Assurance

Goal of quality assurance is to provide the management with the necessary data to be informed about product quality. It consists of auditing and reporting functions of management. If data provided through QA identifies problems, the management deploys the necessary resources to fix it and hence achieves desired quality control.

Cost of quality

A very significant question is: does quality assurance add any value. That is, is worth spending a lot of money in quality assurance practices? In order to understand the impact of quality assurance practices, we have to understand the cost of quality (or lack thereof) in a system.

Quality has a direct and indirect cost in the form of cost of prevention, appraisal, and failure.

If we try to prevent problems, obviously we will have to incur cost. This cost includes:

- Quality planning
- Formal technical reviews
- Test equipment
- Training

We will discuss these in more detail in the later sections.

The cost of appraisal includes activities to gain insight into the product condition. It involves in-process and inter-process inspection and testing.

And finally, failure cost. Failure cost has two components: internal failure cost and external failure cost. Internal failure cost requires rework, repair, and failure mode analysis. On the other hand, external failure cost involves cost for complaint resolution, product return and replacement, help-line support, warranty work, and law suits.

SQA Activities

There are two different groups involved in SQA related activities:

- Software engineers who do the technical work
- SQA group who is responsible for QA planning, oversight, record keeping, analysis, and reporting

Software engineers address quality by applying solid technical methods and measures, conducting formal and technical reviews, and performing well planned software testing. The SQA group assists the software team in achieving a high quality product.

SQA Group Activities

An SQA plan is developed for the project during project planning and is reviewed by all stake holders. The plan includes the identification of:

- Evaluations to be performed
- Audits and reviewed to be performed
- Standards that are applicable to the project
- Procedures for error reporting and tracking
- Documents to be produced by the SQA group

- Amount of feedback provided to the software project team

The group participates in the development of the project's software process description. The software team selects the process and SQA group reviews the process description for compliance with the organizational policies, internal software standards, externally imposed standards, and other parts of the software project plan.

Quality Control

The next question that we need to ask is, once we have defined how to assess quality, how are we going to make sure that our processes deliver the product with the desired quality. That is, how are we going to control the quality of the product?

The basic principle of quality control is to control the variation as variation control is the heart of quality control. It includes resource and time estimation, test coverage, variation in number of bugs, and variation in support.

From one project to another we want to minimize the predicted resources needed to complete a project and calendar time. This involves a series of inspection, reviews, and tests and includes feedback loop. So quality control is a combination of measurement and feedback and combination of automated tools and manual interaction.

The End