3.1

* **Enterprise Software + Software Quality discussion**

http://searchdatacenter.techtarget.com/feature/How-can-vendors-live-up-to-users-software-quality-standards

I would like to share an interesting article that I came across when I was reading more about how large enterprises maintain code quality on a regular basis given their volume of code is beyond comparison to what I have faced here at uni or during my job experience. More than knowing the technical aspects, I was interested in how quality is not an aspect but rather a mindset in these enterprises.

The aspect that I liked very much written by the author was how he mentioned that CIOs always seem to ask vendors for three things simultaneously: cheaper, faster and also better. He then further went to explain that 'better' is the best option and to only pick that. He argues by saying better quality software is cheaper to install and maintain. IT staff can then devote their time to worthwhile problems rather than spending time on the phone with vendor support. Better probably means longer release times, but higher code quality at the end of the day which will eventually set the company apart.

So after reading this, it brings me back to the discussion we all had in the lecture last week, about the pushing out the product out to the market now or later.

Before I end off, I would like to know on how you would prioritize these aspects - cheaper, better and faster?

1. Personally I would always pick "better". My reasoning is that consumers remember problems with software for much longer than they remember software missing a feature. For example, Apple Maps is frequently remembered for the bugs and poor quality, whereas Google Maps does not seem to be remembered for its original lack of Australian turn-by-turn navigation.

2. Depend on the situation we're in. For example, if we have a competitor which already launch their product and took over 30% of the market, I don't think we should stick with the 'better' option since it will take longer time. And longer time will resulted in higher cost and higher price for customers. By the time we finished our product, our competitor will have bigger market share, and thus it will be difficult to penetrate the market with high price (although our software has top-notch quality).

On the other hand, if we are in comfortable situation (no competitor, no time limit, etc.), "better" is the best option. Just like what the article said, it will be cheaper to install and maintain thus in the long run it will saves money for the company. It will bring good publicity for the company too, since everyone will remember it as good software (instead of bug-infested software) that comes from our company.

I agree that all depends. Don’t forget that sometimes there are contracts and due dates. I can tell you that in many times is more important end the development according to the date to avoid the payment of a penalty than the quality of the software. You always has the chance of a service pack or new release.

I agree that all depends of the situation, the project, the type of company, the type of development. All the variables are measured in money and you need to have that in mind all the time.

3. Quality is emerging as a facet of software development that is even bigger than the process itself. You have to juggle requirements, development, multiple releases, huge customer bases, and vendors that want everything perfect, cheap and faster.

The main reaction to this landscape is to take quality as an implicit part of the overall development process. It needs planning, execution and control

4. Although I would choose "better" as well, but I don't think there is always only one you can pick. There would be no end if the developers keep improving the product. By the end, all the products will be released still with some improvements that can be made. So if the developing team has a clear strategy and standard, with some successful experiences, I think they can still produce a product quick enough with a quality good enough (not perfect)

5. It depends on the cases, such as what type of software product they deliver and how the market of relevant field is. Nowadays, many company have mechanism for creating new product and new version faster, if the environment allows faults testing, they might more willing to do fast distribution, less concern "better", coz they think it would be better as product still being used.

However, in the case of business software system, it need to be very reliable with strict quality level. It cost a lot while fixing the bugs online, so they might put the "better" as the priority.

Also, they are trying to use all tools and techniques, achieving the "cheaper", coz use less resource to do more thing is another concern for all tech company.

3.2 Concept Map Software Quality Engineering(SQE).svg

3.3

1. What measures would you deploy and in which context related to "Quality Planning"?

Specific, Measurable/Testable, Attainable, Relevant, Time-bound, Realistic about where defects come from, Selects appropriate detection and prevention methods

Setting appropriate quality goals or standards with respect to Functionality, Reliability, Usability, Efficiency, Maintainability and Portability, to provide developers with more objective assessments of software performance during the execution of the project.

Plan software tests (unit, integration or system test) with complete list of test activities, or acceptance tests for external software, break down them into controllable time unit, use test management tools to monitor and control the tests, identified the source of defects.

Specify configuration management tools and procedures (include change-control procedures) to control and monitor the changes, ensure each change are traceable.

1. What measures would you deploy and in which context related to "Quality Assurance"?

Define appropriate support level for the application hosting in different environment.

Decide a certain level of performance of software would meet the requirement over the next years in different target environment, increase the efficiency of work

Plan designs, code inspections and reviews for changes.

Ensure the units or components are testable for themselves, and amount of codes for each are within an appropriate range, those are based on the good design of the system.

Plan the activities of auditing the performance for each period, continuous improve the efficiency of work.

1. What measures would you deploy and in which context related to "Quality Control"?

(Quality control is defined as a set of activities designed to evaluate the quality of a developed or manufactured product.

Code and design inspection before deploying into the staging test environment.

Quality control inspection and other relevant activities take place as the development or manufacturing of the product is completed yet before the product is shipped to the client.)

Host Daily regression test to ensure any changing taken place compatible with current production version.

Host system test during each milestone of the development processes and identify the results, ensure the system function properly within acceptable range of error.

Host performance test to ensure meet the required performance while increasing in users over the next years, use historical data for a comparison after each iteration of performance test.

Identify and improve the current issues in the system and management process, such as using six sigma, control the changes taking place.

1. Would you use Software Requirement Specifications? Where? What for?

Agree with customers and produce requirement documents before starting the project.

Define the goal and state what the system would look like after finishing the project.

Describe the software product perspective including functions and constraints,

Specify the functional and performance requirements.

State the system attributes: reliability, availability, maintainability, portability, usability, compatibility, security.

1. How would measure quality?

Software quality is the degree of conformance to specific functional requirements, specified software quality standards, and Good Software Engineering Practices.

a. Software Reliability: Mean time to failure, Probability of unavailability, Rate of failure occurrence

b. Availability: Robustness and Portability, Time to restart failures, Percentage of events causing failures, Probability of data corruption on failure, Percentage of target-dependent statements, Number of target systems

c. Working environments: (1) Effect on individual productivity and satisfaction: Comfort; Privacy; Facilities.

(2) Health and safety: Lighting; Heating; Furniture

1. What measures would you take to improve group working?
2. Team members should all be treated in a comparable way without favorites or discrimination.
3. Different team members have different skills and these differences should be respected.
4. Involve all team members and make sure that people’s views are considered.
5. Always be honest about what is going well and what is going badly in a project.
6. define the type of group composition(task-oriented, self-oriented, interaction-oriented)
7. Provide space for interaction and private working.
8. Exchange the information of works.
9. What is the cost of the measures you are proposing quality?

Reason based on the following categories: prevention, appraisal, internal failure management, external failure management, measurement. ---week1 side page16

1. Cost of executing a project so it is error-free or within an acceptable error range.
2. Cost of evaluating processes and their outputs to ensure quality, such as evaluate the SRS, identify the Reliability and Availability of software product.
3. Cost to correct identified defects before delivering to the customers.
4. Cost to correct detected errors before delivering to the customer.
5. Cost of equipment’s used to perform prevention cost and appraisal cost activities, such as regular code and design inspection, equipment to carry out the tests, improving working environment.

3.4 class sides and People-CMM-01mm001

1. Software Quality:

A product can be used as it was intended or within some range of errors. ---from class sides week 2

The degree to which a system, component, or process meets specified requirements. The degree to which a system, component, or process meets customer or user needs or expectations. --- From 24 page of Software-Quality-Assurance-From-Theory-to-Implementation DANIEL GALIN 2.4 Software quality – definition

1. What is the purpose, tasks, mechanisms, aims and means that are part of "Software Quality Engineering"?

The purpose is to ensure required level of quality of a software product is achieved.

Tasks are to define appropriate quality standards, procedures to ensure that those are followed.

And also aiming to build a quality culture in a working environment where everyone see quality as their responsibility.

We establish organizational procedures and standards as mechanisms for ensuring certain level of quality.

All those means the project to build a software product would satisfy the requirements. --- From class sides week 2, page 3

1. What is the difference between Quality Planning, Assurance, and Control in a software development environment?

Quality Planning starts at the beginning of each iteration of development phase, it identifies and selects the suitable quality standards for certain project and plan how to satisfy them, it also breaks down quality into deliverable items

During the development cycle, we periodically evaluate the whole performance of the project, ensure to satisfy standards for Assurance purpose. And also, we use certain mechanisms to control specific requirements and results of the project, ensure they meet and follow the standards and procedures. --- From class sides week 2, page 4

Short:

Quality Planning is performed before development is commenced, whereas QA & QC are performed during (or after) development.

Quality Planning is concerned with identifying initial standards and procedures, were as QA & QC are concerned with meeting standards.

Quality Assurance is concerned with preventing inadequate quality, were as Quality Control is concerned with testing if quality has been met.

Or:

|  |  |  |
| --- | --- | --- |
|  | When | Purpose |
| Quality Planning | An initial stage before Software Development Process | Setting up initial quality standards and procedures. |
| Quality Assurance | Periodically evaluate overall project performance during software Development Process | QA is focused on planning, documenting and agreeing on a set of guidelines that are necessary to assure quality. |
| Quality Control | after a product is developed and before it's released | QC is a reactive means by which quality is measured and monitored, including all operational techniques and activities used to fulfil requirements for quality. |

1. Describe what a “Software Requirement Specification” is, what is it used for? What is its purpose?

It describes the needs from the users and the constraints on the software in specification documents without implementation details, and what the software would do and not expected to do. It is used to produce an agreement between customers and suppliers and reduce the chances of later redesign. Its purpose is to allow relevant people could understand functions of the software and reliability, security, performance, usability, availability, portability constraints and external behavior of it from those documents. --- From class sides week 1, page 6-7

Short: (An SRS is a specification of functional and non-functional requirements for the project, agreed upon by stakeholders, users and developers. It is referred to throughout the project by developers to ensure the project is fulfilling the client and developer needs set out at the beginning.)

1. What is the People Capability Maturity Model? How many stages does it contain? Where would that be relevant in a software development environment?

It is a tool that helps you successfully address the critical people issues in your organization. It is a maturity framework that focuses on continuously improving the management and development of the human assets of an organization.

It employs the process maturity framework for managing and developing an organization’s workforce. --- From People Capability Maturity Model (P–CMM) Version 2.0(People-CMM-01mm001) chapter 1.1

It contains five maturity stage levels that establish successive foundations for continuously improving individual competencies, developing effective teams, motivating improved performance, and shaping the workforce the organization needs to accomplish its future business plans. --- From People Capability Maturity Model (P–CMM) Version 2.0(People-CMM-01mm001) chapter 1.1

At level 2, organizations establish a stable development environment in which to perform professional work, and enable people to repeat practices they have used successfully in the past that may benefit the development process.

At level 3, organizations identifies its best practices in its unique development environment and integrates them into a common process, then make them as standard process to lay the foundation for a professional culture, improve the effectiveness of development.

At level 4, organizations can manage the development through the data, which means that could apply profound knowledge and take correct development environment improving actions.

At level 5, organizations can improve software development continuously based on the data. And also enabling people to improve their work processes continuously.

Short:

Initial (level 1): the initial stage where processes are unpredictable, poorly controlled and reactive

Managed (level 2): where manages take responsibility for managing and developing their people

Defined (level 3): proactive - where the workforce and work-groups align with the business strategy and objectives

Predictable (level 4): where the workforce is empowered and integrated, and performance can be managed quantitatively

Optimizing (level 5): where the work group and organizational capability is continuously improved

1. Suppose you hear about the quality measures that are being used at a competing company. One of them is "Code inspections must be done at least at two different levels of the hierarchy". What do you think these means in the context of SQE?

One part of SQE is to ensure the software product comply with the relevant quality standards, such as delivering within range of error, code inspections would done by another reviewed teams, helping them address the design and code quality issues before deploying into production environment or shipping to the clients. And also reviewing low level functions would ensure basic functions of the system operate properly, at high level hierarchy, the interaction between the subsystems are often reviewed, ensure the business running properly in the systems.

1. You also hear that the rival company has different "channels" in which different versions of the product are manipulated. One of them is called the "release channel", which is the one in which the released version of the product is manipulated. They tell you that they use three more channels. Can you imagine what those are and what version of the product is manipulated?

Development channel: the stage of implementing the features

Testing channel: system is deploying into the specific test environment and codes are merged into the test branch

Beta release channel: products maybe in unstable version and taking the user accompany test involving invited person

Or:

Development channel: for developers working on an internal version, new features and functionality will be implemented and tested in this channel before its release.

Master channel: having only code that is entirely stable in their master branch – possibly only code that has been or will be released.

Topic channel: a short-lived branch that you create and use for a single particular feature or related work.

Or:

Stable channel: This channel has the full testing of project to avoid crashes and other issues.

Hotfix channel: Critical bug patches are made and tested here which are then merged to all of the above channels when ready.

Beta channel: If you are interested in seeing upcoming features and enhancements with minimal risk, Beta channel is the place to be.

Dev channel: We want the end-user to see what's new as soon as possible.