12. Improving the test process

Next we cover the measurement and development of the testing process. The idea is to improve the quality of the products by improving the process that produces them.

Based on [Craig&Jaskiel 02]

- Testing organizations are affected by the same problem as other organizations: It is hard to make people change the way they work
  - When needs for change are presented, instead of thinking the best of the organization, people easily think what they are winning/losing personally
- The most important tasks when developing a testing process:
  - The identification of the areas that need improvement
  - Implementing improvements in these areas
The steps of improving the test process

- Find out the current level of the testing process (the whole process or some part of it)
- Set goals
- Find out the requirements for reaching the goals
  - The requirements should be realistic, exact and measurable
  - Prioritize the requirements
- Start the process improvement project in the same way as any other software development project
  - Ensure sufficient resources
- Make a plan that describes the steps for reaching the goal
  - Include a timetable, budget, risks, etc.

- Implement the changes gradually
  - So that not as many resources are needed at a time
  - The analysis of the results is easier
  - Use a pilot project
- Measure the results
  - Compare the results to the plan
- If necessary start again from the first step
- If a process is to be improved, it should be ensured that all the parties involved approve the changes needed
  - The acceptance can be gained for example with metrics, collecting feedback and using it, training and with internal "sponsors” inside the organization
12.1 ISO certification of quality system

- International Organization for Standardization
- The most significant of the ISO 9000 series standards concerning software engineering is 9001 and the guide for applying it 90003
- Reasons for getting certified:
  - The improvement of quality processes
  - Setting a point of comparison regarding improvements
  - Market reasons
    - In some areas of industry certification is a prerequisite for getting a deal, required by the client
    - Required for participating in invitations to tender in many fields of the public sector

- ISO 9001 certification is a heavy and time consuming process
  - Instead of the whole organization, only a part of it can be certified
  - Audits are a continuing expense
- For an improvement of the testing process ISO 9001 certification can act as a stepping stone
  - The most important things to be developed are repeatable and documented ways of working
There are conflicting experiences about ISO 9001 certification in the software industry:
- For some, it works as a marketing tactic that has little impact on the end products.
- On the other hand, if an organization is prepared to put effort into process improvement and there is enough motivation for certification, ISO 9001 might be just what is needed to get started with the work.

12.2 CMM

- **Capability Maturity Model**
- Organization-based evaluation model of a quality system
- Newest version is CMMI (CMM Integrated)
  - CMMI’s staged version corresponds to the old CMM
  - CMMI’s continuous version corresponds to the ISO/IEC 15504 standard (Spice)
- **Background**: Software Engineering Institute (SEI)
  - In the background of SEI is the US Department of Defense (DoD)
- CMM is a framework that consists of five maturity levels
  - Every level contains all the previous levels
- With the evaluation process, the organization can find out the level on which it is currently
  - Makes comparisons between different organizations possible
The idea of the maturity levels is that as the maturity increases, the risks of the product development should decrease and productivity and the quality of the products should increase.

As far as the testing process is concerned, it is unfortunate that the first two levels of CMM pay little attention to testing.

**Level 1 of CMM: Initial process**
- Everyone reaches this level
- The software processes are changed often when needed, without any discipline
- Individuals and their motivation, skills and knowledge affect the success
- In a crisis situation the leaders typically abandon the planned courses of action
  - Coding and testing are done as it seems best at a moment
• CMM level 2: Repeatable process
  – Emphasis on project management
  – Disciplined processes
  – The project schedules are realistic and are based on the requirements and experiences from previous projects
  – (Some) projects are based on repeatable processes

• CMM level 3: Defined process
  – The processes cover the whole organization, including management and lower level tasks
  – All projects are based on a repeatable process
  – Special effort put to documentation
• CMM level 4: Managed process
  - The metrics deployed at levels 3 and 4 are now used to understand and adjust the processes and products quantitatively
• CMM level 5: Optimizing process
  - The process improvement is continuous
  - Quantitative feedback from the projects and products

• CMM is very process centered and not all of its requirements are necessarily worthwhile to implement in all organizations
  - … Unless the goal is primarily to reach a certain level for marketing reasons, for instance
• Especially for smaller companies it might be better not to implement all the requirements of level 5
What could the levels of CMM mean for testing:

- At level 2 test managers should define the goals of testing and error detection and start a process for planning of testing within the organization.
- At level 3 a systematic testing process is started
  - Test managers should create a test organization
  - Testing plans should be integrated to the life cycle of the software development
  - The test processes should be observed and adjusted if necessary

- At level 4 quality/test managers should create organization-wide programs for inspections, technical training, coverage measurement and measuring the software quality
- At level 5 test managers should apply process control to avoid bugs and concentrate on the tasks that relate to quality control
• Organizations usually advance in CMM levels one step at a time
  – $1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 4$, $4 \rightarrow 5$
  – Some steps create more work than others
  – On the other hand, the obtained benefits also vary between the steps
• On the higher levels the stiffness of the processes might become a problem, if small changes have to be accepted with a great deal of bureaucracy

12.3 TPI

• Based on Tim Koomen, Martin Pol: Test Process Improvement, Addison-Wesley, 1999
• Test Process Improvement
• The model is industry-based and it is supposed to be a practical and tested way to improve a testing process
• The goal is ambitious, because one model should be able to cover all organizations, regardless of the size, skill level and the techniques used
• TPI and CMM
  – TPI is “CMM compatible” with its terminology
  – However, CMM is not a prerequisite for using TPI
  – Unlike CMM, TPI is only focused on testing
• Architecture of TPI:

Every key area defines 1-4 levels of maturity in the scale of A, B, C, or D in the increasing order of maturity
  – Every level contains the previous levels, just like in CMM
  – Level A is not reached automatically, a separate "starting level" describes this situation
The checkpoints define the maturity of the organization (A-D) within a key area

- For example the key area of reporting:
  - Error reporting
  - Progress reporting and prioritization of faults
  - Detection of the risks in the system and the organization, presenting of ideas of improvement, reinforced with metrics
  - Improvement suggestions relate to the improvement of the software process

The levels of key areas have dependencies with each other

- For example the A level of testing strategy requires that the specification techniques, commitment and motivation are also on the level A

- Although check points can be used to find out what should be developed inside each key area, there are also level based improvement suggestions for this very purpose
  - Suggestions are only suggestions, not prerequisites for reaching a certain level
• Key areas (20 pcs)
  – Testing strategy (A-D)
    • A strategy must concentrate on finding the most important flaws at the earliest possible moment and as cheaply as possible
    • Defines which tests cover the requirements and the quality risks
    • The whole strategy is affected by the strategies in different levels of testing and their compatibility
  – The life cycle model of the testing process (A-B)
    • Planning, preparation, specification, execution and finalization
    • Improves predictability
    • Makes it possible to adjust the testing process

– The early introduction of testing to the software development (A-D)
  • Although the tests are executed in the end phases of the software development, the testing process must begin a lot earlier
– Calculation and planning (A-B)
  • What to do, when and with what resources (personnel)
  • A basis for resource allocation
– The specification techniques for tests (A-B)
  • The evaluation of the quality and “depth” of test cases
  • The reusability of the test cases
– The techniques of static testing (A-B)
  • For example usage of checklists
Metrics (A-D)
- For the testing process, the most important metrics describe the advancement of the process and the quality of the test target
- When the process is improved, the metrics are used to evaluate the effects of the actions taken

Testing tools (A-C)
- For example, better motivation for testers vs. manual testing

Test environment (A-C)
- The office environment of testers (A)
  - Motivation, communication, the effectiveness of work
- Commitment and motivation (A-C)
  - Both management and workers (resource allocation etc.)

Knowledge and training (A-C)
- The testing team should consist of people whose knowledge and skills supplement each other, for example, knowledge of the domain and organization, programming and social skills
- Training helps with deficiencies

Comprehensiveness of methods (A-C)
- The methods used should be comprehensive enough to cover all needs and on the other hand be detailed enough, so that the same things need not be considered again every time the method is applied

Communication (A-C)
- Both inside the test group and external interest groups, such as developers, customers, users
- Informing about progress and quality
– Reporting (A-D)
  • Testing is about measuring quality and this information needs to be disseminated
– Error management (A-C)
  • Means have to be offered for the management to find out the life cycle of an error
  • Finding out the quality trends and their analysis that can help in providing justified advices to improve quality
– Testware management (A-D)
  • The securing of maintainability and reusability require management
  • Version control of testware
– Test process management (A-C)

– Evaluation (A-B)
  • All phase products, such as requirements and design are evaluated
  • The purpose is to find errors before the actual testing begins
– Low level testing (unit and integration testing) (A-C)
  • The purpose is to find errors as soon as possible
  • The error is made, found and fixed usually by the same person
  – Efficient, as not much communication is needed
• Maturity matrix of testing
  – Maturity matrix connects the key areas and levels to each other
  – The matrix shows that maturity in some key areas increases faster than in others
  – The key areas are inspected as a whole: if one key area is at the level A, the whole testing process is at best at the level A, although most of the key areas would already be at level B
  • One must remember that it is not possible to reach level D in all key areas
At the beginning, the matrix is used to evaluate the baseline of the organization by coloring those cells from left to right that define the current maturity in that key area.

After that the most immature key area is identified and the focus moves on improving that:
• Because there are dependencies between different key areas, one might have to improve some other key area first
  – In order for example to reach the level A in metrics, the commitment and motivation has to be at level B first

• TPI as a whole
  – The foundation seems reasonable as it is based on industrial experiences
  – Very process oriented, works better in a bigger organization than in a smaller one
  – Even if the method is not followed precisely, one can get lots of ideas from TPI
    • The model responds not only to the question "what to do", but also to the question "in which order to proceed"

• One can see from the model that in its birth country, the Netherlands, they do more service business than product oriented business in testing

• A consortium of German car manufacturers has made its own version of the model
  – This helps the improvement of testing processes in a situation where the manufacturers use common subcontractors
  – Tim Koomen told the following story in EuroSTAR 2004 conference: if one sees on the roadside a new car that has stopped working, there is about 50% chance that the problem is related to a software bug
13. Closing words of the course

- Would you like to have new features or get the old ones working properly?
  - Do you want to invest in creating features or testing them?
  - Or should one just change the whole software development process?
- Good testing cannot save a bad program
  - Bad testing can destroy a good program
    - Even in the good ones there are bugs, and only one bug is needed to cause an critical failure
- The best way to keep the amount of testing reasonable is to create fewer errors
- In any case, if errors are made they must be found as early as possible

Testing is a lot more than just running test cases
  - Testing is about obtaining quality-related information
- In contrast to a common belief, testing is creative, fun and interesting
  - Just like coding, but the goal is not to build but to destroy
- Testing can be made more effective in many different ways
  - Improve the communication between testers and developers and generate some of the test cases automatically using the UML models made by the designers
- The success of improving testing can be measured for example in the number of complaints made by the customers
Best practices?

- Unfortunately, it is not easy to find best practices
  - Like "use document inspections" or "avoid multiple inheritance"
- Testing, especially at higher levels (system, acceptance), is very context dependent
  - What works in one context might not work in another context

Books on software testing


