Creating an effective quality management method within software development

ERIK RAMBERG
Creating an effective quality management method within software development

Att skapa en effektiv kvalitetsäkringsmetod inom mjukvaruutveckling

Ramberg, Erik
E-postadress: erik.b.ramberg@gmail.com
Exjobb i: datalogi
Handledare: -
Examinator: Stefan Arnborg
Uppdragsgivare: svenskt dataspelsföretag
2015-04-09
Creating an effective quality management method within software development

Abstract
This thesis answers how to create an effective quality assurance method within software development, especially games development. The work has mainly consisted of interviews with employees and others within the same business as well as studies of literature.

The thesis also describes how the new quality management method got management acceptance, how it was implemented, what happened and recommendations.

The work was conducted in 2007 and the result in this report is still, eight years later, used by the company.

Att skapa en effektiv kvalitetsäkringsmetod inom mjukvaruutveckling

Sammanfattning
Denna rapport berättar hur man skapar en effektiv kvalitetssäkringsmetod inom mjukvaruutveckling, särskilt spelutveckling. Arbetet har i huvudsak bestått av intervjuer med företags anställda såväl som andra inom samma bransch samt litteraturstudier.

Rapporten beskriver även hur den nya kvalitetssäkringsmetoden förankrades i organisationen, hur den implementerades, vad som hände sen samt rekommendationer.

Arbetet utfördes 2007 och såsom framgår i denna rapport används resultatet, åtta år senare, fortfarande av företaget.
Preamble

This thesis work was conducted in 2007. The result was that I got offered an employment and the report of the thesis got lost between meetings and a potential new career.

The company, where I spent most of my time during the thesis, wants to be confidential. All persons I interviewed at other companies have also chosen to be confidential. Testing, QA processes or the lack of QA processes is obviously a sensitive topic in the games industry. I respect their decision.

Agile methods, such as Scrum, were at this point in time new to most software companies in Sweden. Although many companies had tried Scrum none of the companies I got in touch with really mastered the process or the method. The Swedish companies, whom I met, were in a phase where most focus was spent on product development with Scrum and almost no time was spent on testing with Scrum and Agile methods. Today many companies have mastered the Scrum process itself and there are literature, research and thoughts on testing within Scrum and Agile development. This is not covered in this thesis since it was known at the time.

I would like to thank the company and all its employees for making the thesis possible. Also special thanks to the three other companies who let me meet with their employees and in detail discuss their QA methods.

Furthermore, I take the opportunity to share my appreciation for Stefan Arnborg and his support with this thesis work, Ann Bengtson at KTH for guiding me towards my degree and of course my wife Veronica – without her the thesis would never have been finalized.

Finally, a great big thanks to my parents Laila and Bengt-Åke who made my studies at the Royal Institute of Technology possible. I promised them to graduate, but I never said when.
# Table of contents

Introduction ............................................................................................................... 1
  Quality as a concept .............................................................................................. 2
  Scrum ...................................................................................................................... 2
Method .................................................................................................................... 4
  Internal Interviews ............................................................................................... 4
  External Interviews .............................................................................................. 5
  Literature .............................................................................................................. 5
Internal interviews .................................................................................................. 6
  1. What is quality to you? ................................................................................... 6
      Conclusions ...................................................................................................... 6
  2. Is Acme good at quality? ................................................................................ 7
      Conclusions ...................................................................................................... 7
  3. What can Acme improve regarding quality? .................................................. 8
      Conclusions ...................................................................................................... 9
  4. How should QA work according to you? ....................................................... 9
      Conclusions .................................................................................................... 10
  5. What could QA do to make you more effective in your daily work? .......... 11
      Conclusions .................................................................................................... 11
  5. QA: The control mechanism or a supporting function? ............................. 11
      Conclusions .................................................................................................... 12
  6. How many resources do you think is needed? ............................................ 12
      Conclusions .................................................................................................... 12
  7. Other ................................................................................................................ 12
QA and testing in Literature .................................................................................. 14
Lessons Learned in Software Testing ................................................................. 14
  Why and for whom? ............................................................................................ 14
  Hard advice ........................................................................................................ 14
  Testing techniques ............................................................................................... 15
External interviews ................................................................................................ 17
  QA Lead (Company 1) ....................................................................................... 17
  UI Designer (Company 1) .................................................................................. 17
  Tester (Company 2) ............................................................................................. 18
  Conclusions ........................................................................................................ 18
Decisions and management acceptance ............................................................... 20
  Two sides ............................................................................................................. 20
  The middle ground ............................................................................................. 20
  The road to acceptance ..................................................................................... 21
The solution .......................................................................................................... 22
  QA Department .................................................................................................. 22
  Vision ................................................................................................................... 22
  Project checklist ................................................................................................. 22
  QA a supportive function .................................................................................. 22
Introduction

In 2007 I got in contact with a Swedish software company in the computer games industry. The company had been growing the past years, implemented a new development process and signed more prestigious clients than before. Despite the fact that eight years have passed, the company wants to remain confidential. Throughout the report the company will be referred to as Acme (A Company Manufacturing Everything).

Acme had been developing games for mobile phones and web browsers. Almost all games were delivered to a specific client who paid for the work and released the game under their brand on their web site. Acme was specialized in client to server solutions and turn based multi player games. At the time the company consisted of a management team of five people, a group of project managers, or producers as Acme themselves referred to them, between 30 and 40 software developers and four to five user interface experts / graphical designers.

Since the start of Acme, delivering quality and always improving had been high on the agenda. At this point in time, perhaps too much was happening at the same time and the company asked me to look into how to improve the quality of their projects and games. For me, this was a perfect task for a thesis work meanwhile it provided an opportunity to get my career going.

Before I started the thesis work I got the opportunity to sit down with the Acme CEO who described the task ahead:

- Quality has been on the Acme agenda since our very start. We have continuously tried to improve our games, our agreements, sales tools, development environments, time estimations and so forth, he says. And continues:
  - For a few years our biggest concern was technical quality, or perhaps, stability. Our products had many problems, like a nightly batch job not working causing my employees to stay awake all night and customers not being able to play multiplayer mode in our sports games without struggling. But Acme has been lucky, because we have many talented employees on all different levels. They have made sure we haven’t done any too big mistakes, yet more due to skill than due to good testing strategies.
  - Today, Acme is bigger than ever, working with more projects than ever and also with more demanding partners and clients than before. Furthermore we are also investing more of our own money into our games and basically we can’t afford serious mistakes. We can’t afford releasing games that don’t have 99.9% uptime, and even more importantly – we can’t release games that people won’t play… We are putting our faith into delivering such high quality products and services that players will subscribe to them or wager money in them, and choose our solutions before any of our competitors’… that’s hard to achieve!

The objective for the thesis project was clear – create an effective quality assurance method for software development. More specifically, a process and method for improving game’s quality and indirectly get more paying customers to Acme.
Quality as a concept

Quality is as a concept older than Christianity. Quality is stated as one of Aristotle’s categories\(^1\) and further developed during the middle ages when guilds made their members responsible for the quality of the goods or services they delivered through setting and maintaining standards for a guild membership.\(^2\)

During the First World War manufacturing processes became more complex and mass production was introduced. Product demand let the workmen earn more money through producing extra products, which led to occasional bad quality. Full time inspectors were introduced to identify and correct failures. This was improved even further during the Second World War with statistical quality control (SQC). SQC made the quality assurance process more effective through introducing control tools such as taking samples and using control charts rather than inspecting every product in a batch of hundreds.\(^3\)

In software development an often quoted fact is that Microsoft, despite having 3000 testers employed, released Windows 2000 computer operative system with more than 60,000 bugs.\(^4\) When I have heard this fact it has been used as an argument for releasing software products with bugs as a market reality. The product can be released as long as you are aware of all the bugs. My opinion is that it is not the entire truth - you do not only need to be aware of all the bugs, you also need to know how they affect your customers, their user experience and willingness to buy another product or recommend your products to a friend.

Scrum

When the thesis was started at Acme they company had just started implementing a new development process called Scrum. Scrum is on paper a real simple process that requires a Product Owner that can prioritize all requirements on a list (called backlog), a development team that is cross functional and can solve all requirements and a scrum master, often a team member, that will defend the team from any disturbance and make sure the team is only working on the most prioritized backlog items. The team will work in Sprints, which are time boxed efforts that starts with planning and must have a sprint goal. At the end of each sprint the software shall be integrated, tested and potentially shippable. A sprint is finalized with a sprint demo, showing what was achieved in the sprint to the product owner and other key stakeholders.\(^5\)

In the beginning the development teams at Acme really appreciated Scrum, they got to focus on developing and creating great games without being disturbed by others. The management team was satisfied as well, now they could finally get the games released on time. However, it did not take long until the product owners realized how hard it was to keep a backlog prioritized and to solve all requirements without having a fully cross functional team. Another problem was that it was impossible for the teams to deliver shippable products at the end of each sprint since they could not at the Sprint planning on beforehand estimate the time it would take to correct eventual bugs on software they had not produced yet.

\(^1\) [http://en.wikipedia.org/wiki/Categories_(Aristotle)]
\(^2\) [http://en.wikipedia.org/wiki/Quality_assurance]
\(^3\) [http://en.wikipedia.org/wiki/Quality_assurance]
\(^4\) [http://www.techrepublic.com/article/the-internet-and-asps-take-a-toll-on-software-testing/]
\(^5\) [http://en.wikipedia.org/wiki/Scrum_(software_development)]
Acme tried various lengths for the sprints, the different teams could choose themselves and most ended up in having 10 working days as the optimum length. Shorter sprints meant too much time spent on planning and sprint demo and longer sprints made it hard for the teams to keep control of everything they had planned and also made it slower for the product owner to get the team started on a new feature that did not get into the current sprint.

Bugs were not documented in any computerized system at the time and management had not gotten all the answers. Scrum was no answer to which date a game could be released with all promised features, it only promised to deliver the most prioritized features after each sprint.
**Method**

Quality Assurance, referred to as QA, is a very wide topic. I realized instantly that there were probably as many definitions of what a QA department should be doing as there were employees at Acme. For an outsider as myself it was quite obvious that Acme needed to do something about the quality, since they despite all the big clients and traffic to their web sites were lacking a major success.

Either they as a company had the talent to produce hits, and simply had not gotten everything in place yet, or they had not. With my thesis work in front of me I had to assume that Acme was not lacking talent, otherwise the work would be very short. I suspected that they were disrupting all talent with focusing on the wrong things satisfying the wrong needs. Hence, I felt that I needed to start with understanding what all Acme talents thought about quality and what was bothering them in their daily work.

Already after a few interviews I realized two important things. First, that I needed some external guidance and thoughts on QA to avoid my thesis work becoming too Acme biased. Secondly, there must be literature on testing and QA that at least could inspire what types of methods and processed that would suit Acme.

My method consisted of three activities:

- Internal interviews
- External interviews
- Studying from Literature

The findings from each activity will be detailed in the following chapters.

**Internal Interviews**

I really wanted to understand all the different opinions of QA in general and QA at Acme specifically. Instead of doing a web based survey or ask people to answer a questionnaire I took on the time consuming task to interview Acme employees for an hour each. In order to get some measurable data I decided on six to seven questions to use as guidance during the interview.

1. What does quality mean to you?
2. Is Acme good at Quality?
3. What can Acme improve regarding quality?
4. If you got to decide, how would QA work at Acme?
5. Should QA be a supportive function or more of a control mechanism?
6. What could QA do in order to help you (in your role) become more effective?

And if the interviewed person was a producer (project manager) or product owner I also asked:

7. How many resources from QA do you want to use during 2008?
External Interviews

The initial plan was to talk to different QA people among the Acme clients. But due to the limited time frame and the need from Acme that my thesis would deliver a solution, I choose to talk to external people with QA connections as close to the Acme premises as possible.

The external interviews were more or less open discussions about how quality work were conducted at their respective companies from their own personal perspective. What was bad, what was good and what could be improved.

Literature

I searched a lot on the internet for good advice on QA and good literature. It was suggested from several sources that the following books should be good guidance:


When writing the thesis I also studied:

Managing risk in software process improvement -An action research approach (Iversen, Mathiassen, Nielsen (MIS Quarterly Vol. 28 No 3, pp. 395-433/September 2004)
Internal interviews

Via email I asked everyone at Acme if they wanted to sit down alone with me and discuss QA for an hour. To my surprise 40 individuals choose to do this. I performed 34 individual interviews and two group interviews with three people at the time. An interesting fact was that when I walked around, asked people face to face I never got no as an answer. But when asking the last dozen of employees via email to choose from a few suggested times I managed to get only 3 interviews.

I have promised everyone involved that what was said during the interview would stay between us, and so it will. However, I have rewritten and listed some of the answers from the interviews and tried to remove anything that could reveal from whom a specific answer origins.

After I had listed the answers that were given on each question I have added a section with my conclusions, basically what implications it has for QA at Acme.

1. What is quality to you?

Finding a definition on “what is quality” is of course very hard, hence the question. But I was very surprised when I started to see a pattern and actually some sort of definition that probably all Acme employees could agree upon.

25% said something regarding expectations that could be summarized into:

*Quality is when your experience/impression of the product is on expectation and above.*

Another 25% of the interviewed people talked a lot about

*Quality is when there are no defects or bugs.*

Others spoke about:

*Stability, User friendliness, Good Experience, No negative experiences and Security.*

*A good first impression and Attention to detail. Polished and up to date*

A handful of persons spoke about quality regarding feelings and being proud. This I found extra interesting since it was not referred to in QA Literature and something everyone understands, but is hard to measure:

*Quality is when I am proud of what Acme has delivered and I want to show that to friends and family*

Luckily that person was proud and had showed the Acme games to friends and family.

Conclusions

Although I did not think Acme needs a definition of quality that is set in stone, I was convinced that defining the Acme view of quality would most likely help them when talking about quality and products in the future. Especially when they start talking about delivering on end user or player expectations on quality. Acme does not need to do make it a formal process and always talk about the quality definition, but a sentence or two describing a vision or goal would give all employees a common language and guidance.
Furthermore, talking about expectations raised the question about how to know what the users expect and that was excellent. That would basically force Acme to have an opinion (or even better to decide) on a target group and then ask that target group what they would expect from such a service or product that Acme is developing. This is logical and hard to argue against, but when asking Acme employees about the Acme games and target groups no one could mention a single game that had a defined target group.

User friendliness, stability, no defects and so forth were to me things that need to be considered to deliver on or above most users’ expectations. So in that sense those words actually fit the definition.

*Quality is when player experience/impression of the product is on expectation or above.*

2. Is Acme good at quality?

During the interviews I asked one straight question and noted the answer before elaborating on the topic. Is Acme good at quality? 80% answered No, 10% answered Yes and remaining 10% answered “both yes and no”.

All these people then followed their very (at least in some cases) binary answer with a sentence describing what they meant. I have rewritten and listed some of the discussions below.

- We lose continuous quality, but we are good in certain areas in the different products.
- We are guessing almost all the time, so No.
- It is often too stressful to deliver quality; there are always many things that “must” go into the release.
- We are lacking goals for what quality to deliver. We often speak about high quality, but what do we really mean? When are things done?
- We are not good, but we manage to get away with it since all individuals are good and don’t produce that many bugs.
- Things often feel bug free. But we never know, it is more of a gut feeling
- We deliver things with very high standards, especially on user interfaces.
- We have very high ambitions but we have forgotten the user.
- We don’t know today. But it feels like “no”.
- No, we don’t deliver on end user expectation but sometimes on our client’s / partner’s expectation.
- Yes! Because our things look so good!
- We don’t get time to deliver those last per cent of polish / functionality that is needed.
- We don’t test, we guess!
- Good at technology but poor in user experience, at least on some areas.

**Conclusions**

A majority of 80% claimed that Acme was bad at quality. My conclusion was that the high number comes from the fact that Acme always wants to improve its offerings and its development processes and also that they were lacking benchmarks how bad/good their competitors were. Furthermore, the answers were probably very well connected to people lacking goals and definitions on what the products should achieve. In other words, Acme employees did not know if they were good or bad on quality, and it felt safer to say no.
One could claim that it was obvious that Acme products lacked quality since Acme has not yet produced a hit, a game generating significant money for them. But so far Acme’s partners and clients liked the products that were delivered, so Acme could not be that bad after all. Or?

People probably answered “No” since they realized there were future room for improvement rather than measuring the actual quality of today.

3. What can Acme improve regarding quality?

Out of the definition on quality that people gave me and after having described why Acme isn’t good on quality people were asked to describe what can be improved.

What can Acme improve regarding quality?

- Prototyping. Learn to see when team members are just discussing out of their feelings as consumers/users. That is not worth much compared to the truth that a prototype will give you!
- Setup goals on quality.
- Have standardized directives on what is to be achieved before a release.
- Functional testing.
- Agreements! It is often there we “decide” upon functionality and quality without knowing what end users/players wants.
- The first 30 seconds in all our games.
- Better project management that understands the importance of correct prioritizations (I.e. not guesses on how things should work) and that sees testing as a normal part of development.
- Copy texts and communication towards our users.
- Find out what is important in the project/product/game and make sure you do that part very well!
- Find out when things are good enough!
- Test already on the concept/idea stage. Help with getting decisions on the target group, time estimations etc. (preferably through focus groups).
- Focus on the experience of the end user.
- Agree on dates when we stop developing new code and start to test and improve the existing code. Less stress and better prioritizations.
- The bug reports need to be written better. I.e. good descriptive steps on how to reproduce the bug etc.
- User tests and usability.
- More end users tests.
- Test products earlier and to a broader audience.
- Improve the teams. Make sure they work well together and have the correct resources. We also need good producers and they need rest from time to time.
- UI tests.
- Availability, browser support and down times.
- Our releases, make sure we have done the corrects things and that nothing is broken.
- Automated regression tests (so we never need to worry that something old is broken).
• Automated tests.
• Get rid of the “we need to test things before the release” which takes time from developers and “stops” the project for a while.
• Early prototypes.
• Give testers higher status!
• Start testing if something is fun or not.
• Stop putting ourselves in technical debt.
• Choose where we put our resources, more content and less “Back Office…”
• Let people focus on their jobs… too many with “two hats”.
• Too high ambitions to early… so we don’t reach our goals and we can’t even make the simplest things work. IE. We need to get good at what we are doing.
• Improve on the use of common game design in each project.
• Improve our processes, be proactive and define what we should do first etc.
• Make sure our sprint demos work… we can’t even deliver on a 3 week goal…
• User test products on potential users that have never seen the product (not our hard core users that obviously love the product already)
• Continuous builds!!! That will improve quality!
• A list with “quality conventions”.
• How we do things, why we do things and how the things we have done are experienced by the users.
• Product values, what should the team deliver to the end customer???(a community site, a site where you can win money, a play for fun site) What’s the target group?

Conclusions
Interesting and very promising that no one talked about Acme not having the ability or that Acme was not having the technical knowledge. It seemed to be a combination of designing by guesses, stress and almost never asking their players what they wanted that are the reasons to low quality.

Important conclusions to bring forward in this study can be summarized into:

• Help people stop guessing (when creating ideas or designing features) and try prototyping as soon as possible.
• Help people in understanding and creating goals and expectations for every project/product.
• Help people in performing various kinds of user tests, telling:
  o What is important in the product?
  o When are the important things good enough?

4. How should QA work according to you?
This was a very open question. Especially given that most people interviewed got 10 minutes to stipulate an answer. Once again I was surprised on how much people already had been thinking about this and with what precision everyone could describe their problem as well as a solution. Some actually described a full QA process, which I will not present in detail, but their suggestions have definitely guided me thoroughly in this thesis work.
How should QA work at Acme?

- A well working test team, one QA lead and testers that is put on loan to every team. Make sure that the testing is performed equally among all teams and that there are goals on what to achieve with the testing.
- Recommend things around testing and let everyone know what Acme thinks is ok/good enough.
- One tester per team, with no prestige. He/she must dare to say that things are broken, but also live with the fact that some things will not be fixed.
- Try things early (from a neutral perspective) so the developers get independent feedback and never make the wrong thing 100% finished…
- Take care of usability together with the fun-factor. Give people support in helping themselves.
- Pride is a much stronger driving force than threats and “we must do…”
- Tell the team if there are bugs, but do not correct them or decide what bugs are most important – that is up to the product owner
- Take care of the testing, projects need to be able to rent testers… and then take care of arranging users tests when Product owners/ project managers asks for it.
- Help out with the “definition of done” and support in how to reach there in the project plan.
- Have a pool of testers that can help the teams so we can do scrum fully through having one tester per team.
- Give it a go and decide. Become the QA evangelist and create a common terminology
- Focus on user experience and focus groups, don’t mess around with improving project development, programming etc
- QA should be selling internal services
- Get involved as early as possible… even at a concept phase and support people when making decisions.
- Product owner / producer are responsible for the quality, QA should help them through being proactive and giving support not demand things…
- Minimize risks through validating that the teams have made the correct conclusions/implementations
- Be proactive “It looks like you need help… where? Or wait, perhaps that game needs a helping hand?”
- Quick turnaround times
- Let some testers test the code a few weeks before release
- QA can verify the quality of the product before a release

Conclusions

It was a surprise to see that people, despite very different roles, were so aligned on what QA should do. I summarized all ideas into:

- Let all teams have one tester (not necessarily full time).
- Be independent and proactive – give teams support/recommendations on various types of testing.
- Leave the decisions and responsibility to the product owners.
5. What could QA do to make you more effective in your daily work?

This question tried to address the actual needs for each person I interviewed. Basically I wanted to know if there are other aspects of testing and quality that might be missed when only talking about the bigger picture.

- Help me through helping us all through getting a better understanding for testing, UI and game design in general
- I need help with proofing things… IE avoid discussions between two persons on whether we should do A or B
- Through introducing a QA process we all trust - some people would be able to focus on their tasks instead of running around worrying about quality.
- Put light on user experience, UI design and user tests – those areas needs more power within the organizations
- Help product owners and producers (actually everyone) when making decisions on Game Design graphical material, sound etc.
- Do small (clever) focus groups and increase the quality of the sprint demos.
- Write test cases.
- Help the teams in finding test resources.
- Information, so we know we are doing the correct things.
- Help teams with getting daily builds in place.
- Create directives on how to write bug reports and what does status OPEN mean, priority and severity etc.
- Make sure the sales process is followed and is working.
- Help us all through getting our common knowledge up to an Ok level (on availability, user experience etc)
- Continuous builds, regression tests (automated) and user tests on GUI already on the graphical sketch / idea.
- Deliver what the teams have ordered without your own QA opinion.

Conclusions

Once again interesting that people was aligned as well as that they saw many situations where they could get help. That was promising for a QA method and process, Acme employees have not only realized that they need help, they have also addressed where and how!

- Help teams providing better User experience.
- Help teams in getting daily/continuous builds in place.
- Provide information for business decision.

5. QA: The control mechanism or a supporting function?

This was a rather philosophical question. But I felt that it was important to get everyone’s opinion on where the responsibility and decision making power should live. Putting the final
Go/No Go decision at either QA or the product owner would have a great impact on how Acme run projects. People answered like this:

- Supporting function 85%
- Control mechanism 10%
- Both 5%

**Conclusions**

I was happy that so many as 85% wanted it to be a supportive function and I also felt that those 10% that wanted the control mechanism rather wanted to make sure that someone would start looking into quality and testing and making sure no project avoided it. In other words, supportive function was ok if all projects were told to receive support.

### 6. How many resources do you think is needed?

This question was asked to producers, product owners and members of the management team. I did not want correct estimations on resource needs, rather a feeling for how much help people thought QA could provide and at what cost.

- 1 tester per project at a minimum (not necessarily a full time)
- In the best of worlds 3 testers per live product.
- 2-3 persons at Acme in total.
- 1 to 1.5 Full time per product during 2008.
- 1 full time per project.

**Conclusions**

Being a person outside the organization writing and conducting a thesis study it felt overwhelming that the people at Acme with the money took QA and my work serious, that there was a support for introducing more resources than what Acme had at that point in time. Depending on how to interpret “1 full time per project” and the difference between product and project it would give Acme roughly two to eight full time testers.

### 7. Other

The interviews often started discussing topics connected to quality and Acme projects in general. There were thoughts and ideas that did not fit into any of my questions. However, these ideas and thoughts were so good that I wrote them down and would let them influence how the QA process would turn out.

- We should have a look at our pre studies, we often write a lot that no one reads. Why don’t we stop at 5 pages of documentation, and then implement prototypes so we know what we are talking about? Our client will know what to expect, and we can both test the idea on the potential players!
- Package the QA processes and work as a sellable toolkit! (At least to go into our sales presentations)
- Make the progress on quality in different projects visual. E.g. a matrix displaying the different teams/products.
- Try to classify the status of different projects… one common terminology for the whole company… Perhaps cars (Ferrari, Volvo, Skoda etc)… A Acme quality index. E.g.
What the status on Project number 3? The product owner claims 3, a Volvo and QA can verify and say yes or no, it is actually 2 a Skoda.

- It is important that Acme shows that quality is important! Today it feels like the agreements are more important. It is more important that we deliver everything in the agreement (that someone came up with in a stressful pre study) than that you do things good!
QA and testing in Literature

I have chosen to provide what found to be highlights for creating a new QA process at Acme from the two books; Lessons Learned in Software Testing\(^6\) and Managing the testing process.\(^7\)

**Lessons Learned in Software Testing**

The book goes straight to the point and says that the role of a tester needs to be defined for everyone in the project/company. Furthermore, that it is not the fault of the tester if things are not working. When having interviewed the Acme employees I developed the same thoughts although I couldn’t put my own words on it.

As the book moves forward it moves from talking just about the tester to a broader picture that really inspired me when creating the Acme QA process.

It beautifully states that testing is to find information as a basis for decisions. An angle on testing I had not thought about. It is simple to think of testing merely as a tester trying to find bugs that on a later stage a developer solves. I realized that a bug or an error actually does not need to be fixed, but there is a huge difference in having the information of a bug rather than releasing broken products.

**Why and for whom?**

The book further gives examples on what needs to be done on every test:

- What is the mission for the test this time? (needs to be defined in the beginning)
- What “clients” does the tester have? (Producer, programmer, support, marketing, player, top management)
- Find important bugs/issues fast (will define in what order you do further tests)
- Have as short feedback loops as possible

Again Lessons Learned in Software Testing provides, perhaps obvious facts, but for me with very little experience of testing, great advice that in order to be good at testing you actually need to know why you are testing and who will receive the information.

**Hard advice**

Being in the position of doing something for the first time for an organization with limited experience of QA and testing the book provided some advice that was straight forward and clear. It really helped me in the process to not be the mind behind these advices, rather be able to point to a great book on the topic.

- Just talking about beta tests is wrong and often leading to misunderstandings. There are many different types of beta tests that all serve different purposes.
- Bug reports must be selling themselves and be good. They are the reputation of every tester.

---


• Do never use your bug tracking tool to check how many bugs testers are reporting or how many bugs programmers are closing (as in determining who is lazy and not).

Testing techniques
There are many different testing techniques that can be grouped into:

People based testing:
• User testing
• Alpha testing
• Beta testing
• Bug bashes
• Subject matter expert testing
• Paired testing
• Etc.

Coverage based testing:
• Functional testing (white box or black box)
• Feature or function integration testing
• Menu tour
• Boundary testing
• Logic testing
• State based testing
• Etc.

Activity based testing:
• Regression testing
• Scripted testing
• Smoke testing
• Exploratory testing
• Scenario testing
• Etc.

However, there is a huge difference between trying to find out whether something breaks or not when you hit it with a sledgehammer and finding out whether a game is perceived as fun or not. I found that of these techniques the most suitable one for Acme would be User testing. I.e. trying to find users from the target group of the game and letting them play parts of the game and answer questions.

I also learned that the release processes at Acme required an extra thought. Even if the code was tested on a test environment, then moved to a stage environment where more tests were performed the company still needed tests once the code was in the production system and live for the end users. Many of the activity based testing techniques could address that.

Code testing techniques
There are also different types of techniques to test the code itself, such as:

Automated testing
• Design all your tests first! Then you check whether some of them are suitable for automation or not
• Try to let your automated tests do what a human being would find as boring, e.g. doing the same thing 1000 times…

15
Continuous builds / daily builds

- Allows automated smoke tests
- Allows Automated unit tests
- Allows test suites to be run every day

These code testing techniques seemed very helpful but I choose very early on to exclude them from my work. I knew that the developers wanted to implement these, but I also knew that I could not change the entire company with my thesis work. Given my task I prioritized helping Acme produce better games, or at least games that the intended target group appreciated. I was not going to battle the company CTO though trying to suggest new code testing techniques and indirectly suggesting he did not do his job properly.

However, I understood the importance of getting rid of reoccurring problems through automated tests and saw it as something that should be in the Acme QA Process at a later stage, once user tests were in place.
External interviews

Employees at Acme have a vast network of connections with other people in the computer games industry. Through their connections I managed to get meetings with one tester at a big company providing its customers with games played for cash (Company 2), one user interface designer and one QA Lead from one of the biggest gaming studios in the Nordic countries (Company 1).

Due to Non Disclosure Agreements and their respective work on games not yet being publically released I have left out certain parts of the interviews.

QA Lead (Company 1)

In his own words:
The QA Team is a natural part of the development team. They help the team developing the games through continuous testing and quality assurance. It is always the product owner that has the final say in discussions. The QA team makes sure that the game is possible to play (every day) and that a big Black Box testing can be conducted towards the end by a bigger test department.

Our company uses feature cards that describe roughly 30 very important features. It is the responsibility of QA to change these if any of the features have changed and send that to everyone involved, for instance if the animation for swimming or the functionality on a specific tool in the game has changed. This is how we spread knowledge about changes, otherwise other test teams or development teams might think that a tool or animation is bugged, if they don’t know it has changed.

We use “Button bots” – an internal name for scripts that tests series of weird button combinations and logs what happens in the game. This is some kind of randomized testing in order to find bugs like jumping 100 times in a row and then move right will cause a crash.

The company also has a compliance department that keep track on the demands from companies providing the gaming consoles in order to pass the game certification before the game is being released. E.g. demand on animation on loading screens, maximum 30 seconds intro with logos before the user ends up on start menu etc.

We do focus groups with friends and family from time to time. But only to put light on specific problems and hopefully to convince the game designer / product owner that some features might require a change.

We have recently started with one QA member in every scrum team and that seems to be a very good solution.

UI Designer (Company 1)

In her own words:
We perform focus groups on screenshots and the game story before even setting up a team and starting production. The main producer is deciding on everything, however he has help from many different teams and for instance one main game designer who is constantly playing games.
The biggest difference between our company and how I hear you describe Acme is that everyone at our company knows their games and has some sort of feeling for games. From management to tester they are all very interested in games and plays games on a daily basis for fun.

**Tester (Company 2)**

In his words:
Our company has put every tester into the different development teams so there is no real QA manager or QA organization. We use one tester for each of our products.

Our testing process could be described as follows:

1. There is a game idea and the developers start developing it according to specification.
2. The tester will in parallel with development write test cases in a test case tool called Test Link.
3. When developers are done the tester will execute the tests, then the project manager will decide when the testing is done and the game will be moved to stage environment.
4. On stage environment the site owner takes over responsibility for the game. Here the site owner and tester do some regression testing.
5. Site owner decides when to go live.

However, I do not like the process. Because as a tester I need very good and very detailed specifications of the game in order to write the correct test cases on beforehand. There are some benefits from not knowing the game too well before testing since you will discover things for yourself without being biased by the developers.

Furthermore, it is bad that we have a dedicated tester per team since that means we have lost communication between all testers. It was better when I a part of a test team / QA group where I had my home base and then would move around more between the different product teams.

Today there is a big risk if someone leaves the company since that tester is the only one who knows the tests for that specific.

As a tester I never give my opinion about a game whether it is fun or not, I simply just follow the test protocol.

**Conclusions**

It was obvious in Company 1 that they were putting a lot of the responsibility into the teams themselves. They were very assured that they would deliver quality since everyone at the company was in their own target group. They played their own games, some of them even all of the time. In some sense they were developing games they wanted to play themselves. The need for QA as in assuring the games being fun etc. when in development was not that high. That was a huge difference compared to Acme. But even so, before the company was putting anything into production they would perform focus groups based only on a story told orally and screenshots shown in power point. They then decided on the fundamentals of the game (environment, game idea etc.) and let the team have artistic freedom within those boundaries.

What Acme should learn from this:
- Test concepts and ideas before signing agreements
- Let QA members become team members
- Encourage staff to become gamers, especially of Acme’s own games. (Is it possible to
develop a state of the art online Fishing Game with team members who only are into chess?

What Acme can learn from Company 2 was:

- Let testers be part of a QA team (even though they also sit in different development teams from time to time)
- Do testing in iterations rather than on a test specification
Decisions and management acceptance

My respect for QA and testing increased tremendously throughout my work. I had gained a lot of knowledge in a field of software development that is probably the least talked about and least interesting to the majority of computer scientists. I realized that there were very little understanding for testing at Acme, I think it would be fair to say that I was on the same level as everyone else when I started. Kind of – we need testing. That’s it. And if that does not help, we need even more testing. No thoughts or analysis of what kind of testing was needed, for what purpose and who should act and decide upon the found information?

Perhaps one of the most important parts of creating the Acme QA process was the interviews with the Acme employees. I learned whom the most influential employees were and whom I needed on my side to get acceptance for the new process. Of course one could challenge that and claim that the process would not be the best possible process rather the most accepted or most wanted process. My opinion was that a process that would be used is a lot better than a process not being used, especially since I knew I could not be that off the target since I based my ideas on literature and interviews with experts in the field.

Two sides

At Acme everyone wanted the company to do something about testing. My work was most welcome, but sooner rather than later the view on my work and the Acme QA Process got two sides.

One side, to which I would say most of management belonged, wanted me to finish quickly and not do it too complicated. They simply wanted more testing and more employees dedicated to testing and thought that would do the trick. On the other side we had the employees, especially those working with user interfaces and game design. They were afraid that I would listen to management and now when Acme had the chance to do things right the process would be too thin and not considering all aspects of testing.

The middle ground

During my interviews I understood that the project managers, also called producers, had an important role at Acme. The role was important for making decisions about the games, planning the production and also the budget for the project. This group of people formed the perfect middle ground since they were a natural link between the development teams and management.

The producer wanted to produce best possible games and keep management happy meanwhile keeping the development team and designers happy as well. If the producer would approve and follow my suggested process I would most likely have a success. Whereas if the producers disapproved both my work and my soon to be suggested process, it would all be thrown out the window.
The road to acceptance

After I had identified the producers as being middle ground and important stakeholders I tried to get them involved in my work as soon as I could. They were all helpful and provided a lot of useful insights. The most useful one had nothing to do with the process itself, but money. Acme was organized in a way that gave a lot of responsibility to the producers once a client had bought a game. The producer would create a budget and at internal rate hire resources to the project. The producers taught me that if none of them would internally hire testers or QA resources there would be no QA team and no one to follow the process I was about to create.

When I learned this I started to attend weekly meetings where all producers met to decide on how to share resources. This was both a learning period but also my forum for getting acceptance for the Acme QA Process. I got the opportunity to present my thoughts to the meeting and listened carefully to their feedback. After a few weeks of more work I presented again, and through this process I got accepted in the producer group and instead of them having opinions on the process they started asking when they could get test resources for their projects. Without having any mandate for anything else than a thesis work I asked in return how many they wanted and got 2.5 as an answer. The thesis work was definitely on to something, but this also turned out to delay the thesis completion.

My next step was to talk to the CEO who gave me the task from the beginning. He was happy to hear that my work had proceeded in such a way that there was already a demand for QA resources fulfilling a process no one (except the producers) was aware of. He gave me ok and asked me kindly to hurry up.

I now knew I had the producers on my side as well as the CEO and the final piece of the puzzle was the development teams. They had been very supportive throughout my work and I didn’t see them as any threat as long as the QA process would help them do a better job rather then preventing them through needs of specifications and long meetings.

My work was at this point in time on everybody’s lips and I was afraid that things had gone over my head. Was this work really the solution to Acme’s problems? At least I had appeared very convincing, since I, most likely with the help of some friendly producer, got asked to take on a job as QA Manager, implementing the new Acme QA Process.

With a new job at hand my last step on the road to acceptance was to bring the whole company to a meeting and present myself as well as the new Acme QA Process.
The solution

The literature and interviews had given me a solution on how to implement a QA process at Acme. I was very confident with the solution since it was based on the ideas and thoughts from the employees themselves and also accepted by management.

Since a QA team was new to Acme I wanted to be very clear with what the QA team and process would do, and perhaps even more important, what it would not do.

QA Department
The QA department would act as a home base for everyone working with QA related tasks at Acme. As a beginning two system testers and one user tester was employed. In the QA department we would share knowledge and information and avoid inventing the wheel twice.

Vision
I stated the vision of the QA departments and reminded everyone at Acme as often as I found it necessary.

“QA are service providers with the mission to find information as a basis for decisions.”

Project checklist
In the QA team we jointly created a “How we do projects at Acme” checklist that must contain the following information:

- Describing roles and responsibilities
- Help on how to decide product goals and target audience
- Suggestions on project plan and when to do testing

QA a supportive function
QA would talk to the teams and ask proactively what kind of help they needed and try to solve problems as quickly as possible.

QA in Scrum at Acme
Acme’s interpretation and implementation of Scrum had resulted in a process showed in Figure 1.

Figure 1. Acme’s old development process.
In the old development process, before QA changed it, the teams made a plan for each sprint and started developing all items on the backlog. In Figure 1 the team took on the task to deliver item 1 to 10 from the backlog. When the sprint (usually two to three weeks) approached the end and the sprint demo the team would try to put the demo together and often ended up with something that was broken. Since the project had a testing period after the last sprint nor the team or the product owner was to worried since that was when all bug fixing would occur.

Sprint 2 followed with a sprint plan for item 11 – 20 and the team would have a hard time putting the demo together again.

The test period would start and from time to time some bugs were too hard to fix within the test period. The teams new that they needed two to three days to move the developed code from the local environment to live servers, this work was never planned for.

The product would be released and in worst case miss out the single most important feature prioritized in first place on the backlog since the team did not test it or correct any bugs until the test period. Hence, a new process was needed.

![Figure 2. Acme's new development process.](image)

The new process introduced a major change compared to the old process – the team would not move to develop item number two on the backlog until item number one (the most prioritized item) was tested and verified by the tester. This catered for actually having a working product at the end of each sprint and the team being able to put together a sprint demo. It of course meant that not as many items from the backlog would be started each sprint, but it made sure the ones that was started really worked as intended in the end.

To support this new process a system tester or test manager would write test cases after the sprint planning so tests could occur immediately after a feature had been developed. The system
tester or test manager would also participate on daily scrums in order to update test cases if new information appeared and to start testing features finalized by the developers.

The new process also required that automated continuous builds were implemented so that the development team did not have to spend time on deploying or building new versions for the testers on a daily basis.

The QA team also helped the development team in killing the “designing by guessing” phenomenon through providing quick and creative user tests, early prototypes and external tests. An important feature in a game could be verified by a user from the intended target group as being fun or not fun in just a few days.

In order to assist the product owners the QA team started providing an online survey tool for streamlined customer surveys on existing games. Actual users of the game were asked which features they were lacking the most and users that had left the game were asked why.
What happened

2007 to 2010
Acme formed a QA department consisting of four to five employees that followed the solution in previous chapter. A survey tool was implemented and regular user tests started and the development teams got support in getting smoke tests and continuous builds in place.

Everything was turning out better than I expected. Acme as a company, my QA department and the development teams were all doing great things. At this point in time all development work was focusing on new releases on already existing products.

Later on, one of Acme’s own game ideas was tested as an idea and concept. Acme as a company had listened to my findings and followed the suggestion on getting feedback before starting with the development. The conclusion from focus groups tests was that the game should not be built. The participants in the focus group (from the intended target group of the game) would not play the game. Unfortunately history repeated itself since the game had already been sold to one of Acme’s customers and hence needed to be developed nevertheless what the potential users said. Acme was not guessing anymore, they started developing knowing they game would fail. But hope amongst the management team was stronger than the truth I managed to put forward from a newly started QA department.

The new game became no success and I had been three years at Acme and it was about time for myself to move on.

2010 to 2015
Finalizing this thesis work I felt that the section above was not a complete answer what happened. I could only answer for the first three years since the thesis was started in 2007 and I left the company in 2010. Fortunately I have had the possibility to revisit Acme and find out what happened 2010 to 2015 through conducting one final interview with one of the Acme employees closest to QA, and he said:

Acme continued to change their processes to fit the thesis suggested process. A lot of games were tested with focus groups and code releases from the development teams contained less bug after each sprint. Processes were formed on how to handle releases in a more efficient way concerning all the different environments and even more automated tests were introduced. Acme later changed their direction and stopped developing games of their own. They bought games from others and integrated them with Acme’s existing partners. QA and testing moved towards more integration tests and less tests of the games themselves.

Studies in 2002 suggests that out 1,638 organizations conducting some kind of software process improvement, using a stepwise approach, only 34% will proceed to a second assessment of their process. Of those that made a second assessment 13% did not improve their quality and 3.1% moved to a lower level of quality. Even though this thesis work never measured any quality improvement at Acme, the Acme employee closest to QA concluded that:

The QA work changed the mindset of people and how Acme worked in general and it definitely pawed way for improved tests and release procedures. Things got better.

Recommendations

For anyone reading this thesis I have put together my recommendations on QA and when trying to change existing processes at a software company.

Testing is a natural part of development
Seeing testing as a natural part of development is key to success. A feature or user story is not done until it is tested. Verifying whether something is done or not should occur during development and not in a test phase afterwards.

Acceptance on all levels
Whenever implementing a new process it is highly recommended to get management acceptance as well as employee acceptance. A very natural, learning and inspiring way of getting acceptance is through interviewing parts of the staff. If done properly the interviewing process will provide a solution as if the staff came up with it themselves. I think this is also valid for other processes than QA.

QA on QA
The need for acceptance from a whole organisation and especially the management team can never be underestimated. When the QA department was formed everyone at Acme really listened to what was said and had respect for the new process. As time went by and QA became a natural part of the organisation it also became something that in everyone’s mind can be wrong in its findings or asking the wrong questions. I never revisited the new QA process or tried to get feedback on the QA department and its work. In other words, we never did QA on QA. If I would have revisited my thesis work and QA solution for Acme on a regular basis, with for instance interviews or surveys, I think management at Acme would have listened more on QA’s findings regarding the game that we knew would fail.

This is supported by the IDEAL-model presented in the Managing Risk report. In short the model suggests five phases that are iterated over and over again.

1. Initiating (set context, build sponsorship, charter infrastructure)
2. Diagnosing (characterize current and desired states, develop recommendations)
3. Establishing (set priorities, develop approach, plan actions)
4. Acting (create solution, pilot/test solution, refine solution, implement solution)
5. Learning (analyse and validate, propose future actions)

At the time for my thesis work I was not aware of the Managing Risk report, IDEAL-model nor Risk Management as a concept. Looking back I think it is fair to say that my thesis work followed the IDEAL-model really well in step 1 – 4. However, I never analysed what QA had achieved and what could be improved and no future actions were proposed for a second iteration. I.e. I missed the 5th step in the process, and I never started over again. If QA had continued to develop and improve the process, QA would probably have maintained the position as a department Acme employees and management would have listened to.

Finalizing the thesis I can not help smiling about an idiomatic expression also supported by the IDEAL-model – practice what you preach. In other words, it is most recommended to do QA on QA.
Literature


