Impact of Overtime and Stress on Software Quality

Conference Paper · January 2008

DOI: 10.13140/RG.2.2.12815.59041

2 authors:

Balaji Akula
CT Group of Institutions
2 PUBLICATIONS **12** CITATIONS
SEE PROFILE

James Cusick
IEEE Computer Society
140 PUBLICATIONS **360** CITATIONS
SEE PROFILE

Impact of Overtime and Stress on Software Quality

Balaji Akula & James Cusick
Wolters Kluwer Corporate Legal Services
New York, NY
{balaji.akula, james.cusick}@wolterskluwer.com

ABSTRACT

Producing software requires an environment conducive to concentration and measured work. In this paper several projects are presented with varying degrees of overtime where the resultant quality of the applications in terms of defect counts varies directly. It is proposed that stress in the work environment driven by overtime adversely affects software quality.

Introduction

Stress can be defined as any type of change that causes physical, emotional or psychological strain. (Scott, E., 2007). From time to time stress overwhelms us, in many cases this is indeed normal (hr.blr.com, Oct 2006). While it is virtually impossible for most people to eliminate stress we can certainly take steps to avoid stress and its potential to cause debilitating side effects where possible. According to Peggy Duncan (Scott, 2007), stress is caused due to a number of factors: disorganization, unclear goals, too many personal phone calls, disjointed processes, no routines, poor planning, procrastination, lack of focus, and lack of training are some of the reasons for stress. Statistical data from the National Institute for Occupational Health and Safety has disclosed that the number of American workers who consider stress to be a major problem in their lives has more than doubled during the past ten years (Missoncontrol.com, Sep 9 2007). Among professions, software engineering can produce stressful environments for its participants. In this paper the relationship of stressors to software quality is explored.

Background

As a starting point the question was asked, what is the one thing that can cause a high number of defects and do so repeatedly? There is no single answer to this question. Some of the reasons include but are not restricted to the lack of proper requirements change process, technical challenges, aggressive schedule, improper planning, stress, etc. This paper focuses on the impact of stress on the quality of product developed by software engineers over a period of 2 years. Any stress related health issues were not studied and are not reported here. The background for this effort stemmed from a high number of defects noticed during quality assurance of the software lifecycle.

We have made an attempt to establish a link between an aggressive schedule and its negative impact on software quality. The general pattern of this linkage is given below:

Aggressive Schedule ⟶ Overtime ⟶ Stress ⟶ Defects

As per the above pattern, Defects will increase when a team is working on an aggressive schedule. Putting teams under pressure leads to increases in overtime thus reducing concentration on quality and more focus on output. In general mental activity and clarity is required for any mind (Jeffries, E.R 2007). The more overtime the worker spends means that his mental activity and clarity decreases while stress increases. As corroborated by data presented in this paper it is proposed that defects are injected as a result of stress caused due to overtime. Moreover as per recent studies a tired/stressed programmer has double the chances of putting in a very serious bug but has probably also put in a much higher percentage of smaller ones (Jeffries, E.R 2007).

Defects are a very important metric in software engineering. When defects are taken out of context it gives a bad impression of the product in question. However, when taken as a whole, along with the project, the number of lines of code that is written, project duration, etc., it may not prove to be the demon that it is usually made out to be. Poor quality can act as a dampener in the following ways:

1) Can allow teams to deviate from core project activities

2) Can result in loss of trust between QA & Development teams

3) Increased stress during triage where defects are discussed, and in some cases can have a personal effect detrimental to project.

4) Worst case may result in delay of the project.

5) Finger pointing over defects among teams kills the team spirit.

6) Teams end up spending more time fixing defects than developing business critical features.

One of the most important reasons that stress is generated, according to our observations, is due to aggressive schedules especially being repeated project after project. While many project team members (the average age of the team member was 26 years) can absorb the health related issues of stress; their productivity suffers and manifests in the form of defects (read as poor quality). We have studied 4 projects over a period of 2 years and have observed the defect levels for each of the projects. All projects had a few similarities; all were web applications built using Microsoft .NET 2.0 technologies, 3 out of the 4 projects had an aggressive schedule, and all were completed as planned, 3 projects had a high defect rate. The results are provided below.

Results

Data on four projects were analyzed. The amount of scheduled work hours, actual overtime hours, and number of defects were recorded on each of these production releases. The overtime hours were seen as a reliable indicator of stress on the project team members and the level of defects was seen as an output of the stress on the project team members. In the diagrams below the actual data is reported from these commercial projects.

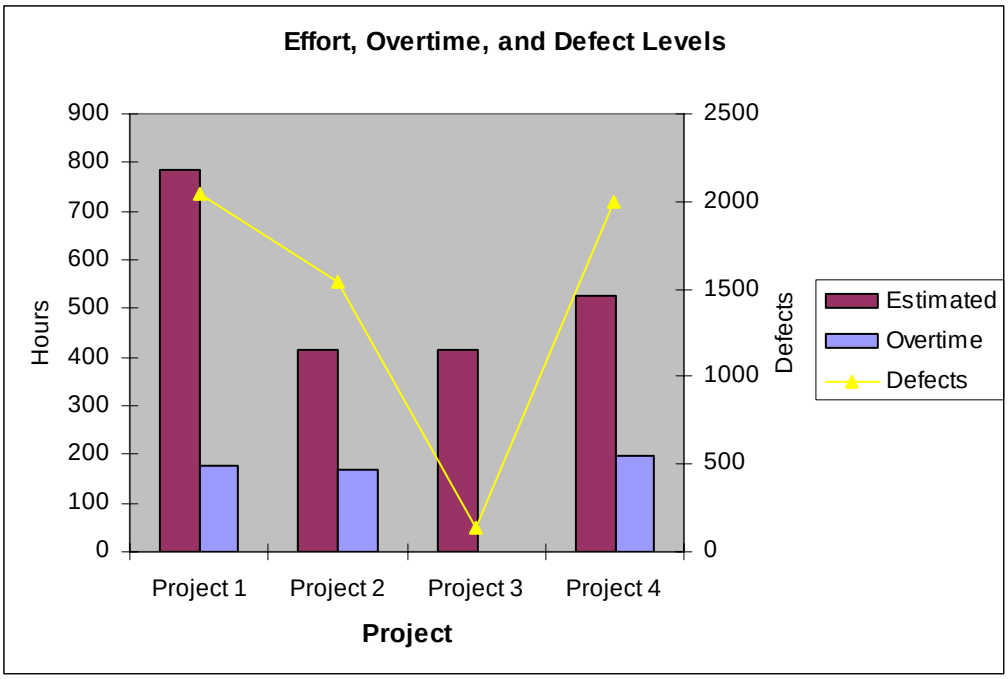

Fig 1: Project wise Effort, Overtime and Defects levels

It is proposed that high stress levels results in less quality and that high stress levels can been produced by high overtime. This observation is evident in Figure 1. As can be seen from the graph (actual data is provided in Table 1), the defect levels are low when overtime is lower or zero. Figure 1 plots the estimated hours for each project and the actual overtime hours for each project. Additionally, the line plot indicates the level of defects for each project. As can be seen, the projects with overtime had higher defect rates.

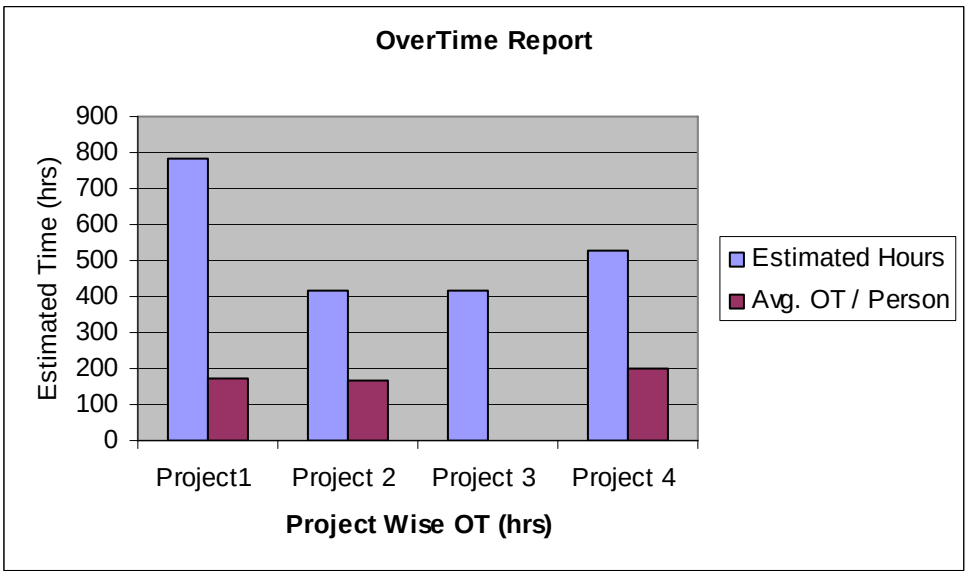

Fig2: Overtime hours spent on each project

The Graph in Figure 2 depicting the actual estimated time for construction and overtime effect on the team to complete project on time and as planned. Please also refer to the same in table format (Table 1). In most cases overtime runs at 10% of the estimated effort hours. Specifically, overtime ran at 150 to 200 hours per person on these projects. These projects are executed with a construction phase of usually 8 weeks plus an additional 8 weeks of QA. Thus, the projects are executed in a compressed time frame using overtime work to balance for the aggressive scheduling.

Project	**PROJECT 1**	**PROJECT 2**	**PROJECT 3**	**PROJECT 4**
Estimated person hours	784	416	416	528
Overtime person Hours	175	167	0	198
Defects	2038	1545	142	2000

Table1. Table data of Project wise defect count in relation to overtime.

Observations

From the above results, the following can be observed

- Defect counts increases when overtime is extensive.
- Stress on workers, and on Projects (via defects, delay etc) develops when projects are not planned and executed properly (that is, with excessive use of overtime).

In general overtime may be requested under the following conditions:

- To accommodate Changes/Requirements not in original estimates.
- Natural/unprepared calamities like viral outbreak, snowstorm etc.
- Technical challenges that cause delays in implementation.
- Under estimation

If a project has clocked overtime not due to any of the above reasons then the primary cause could well be due to incorrect planning. By incorrect planning it could mean that the estimates were aggressive or planning itself was way off the mark. From the

above data in table 1, it appears that the latter was the case, as barring ARWR2 all projects required overtime to complete as planned.

While it is comforting to note that the projects were completed on time, the stress factor to developers/project team is largely ignored as the team moves on to the next project or is consumed by other activities. Whatever the case the stress is continued from one project to another, as history repeats itself time after time.

A correlation of 0.98 (between overtime data and defect count) clearly indicates the effect of overtime driven stress on increased defect count. The defect count in PROJECT 3 is only 7% when compared to the average defect counts of other releases. The possible reason for PROJECT 3 having low defects and high quality can be adduced to the following:

- Zero overtime hours spent during project.
- Correct Planning and Estimation

There are alternative ways to interpret this data. One way of viewing this data is that the defects themselves lead to overtime which leads to stress. The question would be what leads to the defects in the first place. A second alternate view is around the volume of software being written. In this view since the more code that is written generates more defects then the high level of production of code leads to more defects which leads to overtime. These and other alternate views may have validity but they were not deeply considered, instead the primary view of this paper remains that aggressive scheduling drives defect levels which drives overtime.

Conclusion

More and more industries are looking at ways to reduce workplace stress; however the big question is how to accomplish this effectively and efficiently. By neglecting or not paying enough attention to workplace stress companies will end up nullifying their investment. The evidence that links stress to poor quality has been shown above. Critics

will argue that other factors also play a role in the quality of software products. There is no question about this premise. However, stress is one of the important factors that can lead to bad quality products as we have shown. According to Michael Beck (2005) productivity drops significantly when we are under stress and our ability to think clearly is impacted. A survey conducted in October 2006 indicates a loss of 1 hour or more *per day* in productivity due to stress (hr.blr.com, Oct 2006). As the saying goes it is better to strike when the iron is hot. So act now.

Some of the important lessons that can be learned are

- Avoid stress to improve productivity.
- Plan and do not under-plan or be too aggressive.
- Estimation approach – we can revisit the estimation practice that is followed and improve if required. We should try to follow what is in the estimate and try not to cut down on the same for whatever reasons.
- Defect prediction – Using our experience gained from previous projects it would make sense to predict the defect density based on estimation, component inventory, and previous data so that we can adjust our estimates accordingly.
- Quality development – we should strive to go for quality and avoid development during QA phase of the lifecycle. As mentioned below the cost of fixing defect is not only high but the probability of introducing new defects is high too.

Both the above are easy to avoid in most scenarios. The exception being projects which are designated as “Fast Track” and is aggressive from the beginning. The benefits are huge. I would imagine that the following would benefit software teams in the short and long term if implemented according to the nature of project:

1. Less stress to software engineer results in more productivity.

2. Less stress means low defects and indirectly less wrangling with the QA/ Project Management during triage. We are all aware of the battles between QA and Software Engineering teams. We can channel all the energy to good use instead of wrangling over defects.
3. Having a plan which has a buffer (say 2 weeks for example) is better than under-planning. Experienced Software developers are aware that the requirements keep changing, and/or new requirements keep cropping all the time. Hence having a buffer of a few weeks helps the project and in no way would delay the same.
4. It's well-established that it takes less time and costs much less to prevent defects than to fix them. A recent article in Crosstalk, the Journal of Defense Software Engineering, reports that "finding and fixing bugs is the most expensive cost element for large systems and takes more time than any other activity." (Jeffries, E.R, 2007)
5. Last but not the least by planning with the human factor in mind we will be appreciated for trying a humane approach to software development. There is sufficient stress in globally outsourced development without driving overtime hours skyward.

References

1. http://hr.blr.com/news.aspx?id=19241. October 13, 2006, Survey: Workplace Stress Has Substantial Impact on Productivity.

2. Elizabeth, Scott - http://stress.about.com/od/managementtools/a/peggyinterview.htm, 09/20/2007, An Interview with Productivity Expert Peggy Duncan.

3. Jeffries, E Ronald - http://www.xprogramming.com/xpmag/jatSustainablePace.htm, 2007. Viewed on Oct 16 2007.

4. http://www.missioncontrol.com/p_mcsys_stress.htm, 09/20/2007, How does stress Impact Productivity. – Last line, Para 1.

5. Beck, Michael - Manage Stress to Boost Productivity, 2005 http://www.xleaders.com/Articles/Beck/Manage%20Stress%20to%20Boost%20Productivity.pdf. Viewed on Oct 15 2007.

Further Reading

1. http://main.uab.edu/show.asp?durki=57296 - Ten Ways to Control Stress
2. http://www.webmd.com/anxiety-panic/guide/tips-to-control-stress - Mental Health: Tips to Control Stress
3. http://www.drake.co.nz/dbr/pdf/work_place_stress.pdf. - When Workplace Stress stifles productivity.