

Effective Work Strategies under Schedule Pressure

Working as a project manager I found that during the last couple of years the gap between what can be done and what is demanded has been widening, resulting in extreme schedule pressure. It appears to me that a culture has developed that demands deadlines and dictates product requirements without compromises.

Extreme schedule pressure manifests itself in many ways. People experience extreme schedule pressure when they repeatedly fail to complete their assignments as expected in spite of working evenings and weekends. I have heard software professionals say that **"Things slow down as a whole" when the project is in crunch mode.** Project members also report that extreme schedule pressure does not bring out the best in people; interpersonal communication may get abrupt, tense or overbearing. On a personal level people experience stress, as they feel that they have to choose between family and work.



Looking for Solutions

In one of my recent projects I felt that my knowledge and experience failed to provide answers to the challenges created by an aggressive schedule. I realize that extreme schedule pressure is here to stay. We also cannot hope for a silver bullet. In addition, I strongly believe that more strategies exist than just working overtime.

Therefore, I started to look for effective work strategies that improve productivity under schedule pressure. I believe that the collective wisdom of experienced software professionals will help make progress on this tough issue.

INSIDE THIS ISSUE...

- 1 Effective Work Strategies under Schedule Pressure
- 4 P2E Calendar

Thus, a survey on excessive schedule pressure was born. I sent a survey via e-mail to members of professional software associations asking them to report their experiences working under extreme schedule pressure. The results of the survey led to a meeting and panel discussion exploring promising work strategies in more depth.

Survey Results

The survey asked software professionals to report on their personal actions when working under extreme schedule pressure.

Participants were also asked to list their boss's actions and the actions of their team as a whole. In all cases participants were asked to report short-term and long-term benefits as well as negative impacts of these actions. The survey also solicited suggestions for how to increase productivity. Finally people were asked whether they perceive any opportunities in schedule pressure.

Everybody praised the positive effect of some schedule pressure as long as it is within reason. The consensus was that it encourages decision-making and fosters creativity.

As can be expected, working overtime is still the most common work strategy in response to schedule pressure. Someone wrote: *"Circle the wagons and work gobs of overtime."*

Interesting though is the fact that personal actions and actions reported for the boss differ. The two personal actions with the highest number of responses are:

1. **Work overtime**
2. **Reduce scope and requirements**

Following are the two actions most frequently reported for the boss:

1. **Add or re-organize resources**
2. **Request overtime**

The respondents also reported far fewer actions for the boss or the team in comparison with personal action. Two interpretations may apply:

- ❖ People are re-trenching under pressure and take the role of single fighters asking themselves: *What can I do in the face of pressure and stress?* instead of looking at how the boss or other team members can help.
- ❖ People are not aware what the boss or other team members do

Reality is probably a combination of both.

Setting priorities for requirements and work was the work strategy most frequently mentioned for increasing productivity. Some suggested remedies addressing the symptom of “The whole slowing down” which may be caused by re-work and miscommunication.

1. More and better communication
2. Automation of routine tasks which reduces errors, but also speeds up the task completion and allows for more people to do the work.
3. Utilize architects early and keep them in the loop; this is an effective balance to the requirements volatility.

As a follow-up to the survey, a panel of experts discussed four innovative and promising work strategies in a public meeting. The following four work strategies were discussed:

1. Principles for maintaining perspective under pressure
2. Pair programming
3. Making consistent decisions throughout the project team
4. Testing under compressed schedules

Effective Work Strategies

Grace under Pressure

It has been mentioned many times that attitude is everything. A manager can instill a positive attitude by giving team members confidence in their abilities. Stress can cause fear of not being in control, fear from not knowing the roadmap, fear of not being able to handle the demands or not being respected for personal choices. Reminders of past accomplishments, a focus on what can be done and the results that need to be produced will diminish stress and increase confidence. Celebrating individual contributions and team success support a positive attitude throughout the project. It is important to take time out to acknowledge achievements. The celebration may be an afternoon at “Laser Storm” with the team, a day off or a dinner for two.

Following is a summary of principles that help managers make better decisions in the heat of extreme schedule pressure:

- ❖ Do it the right way the first time. Settling for less most likely will cause re-work delaying the schedule after all. Doing it right later is sometimes not an option.
- ❖ Use known processes and procedures. In the face of time pressure it is better to rely on what people know rather than introducing something new, even if it is better. If there is no process in place and no past experience to rely on a new process is better than none.
- ❖ Eliminate negative influences quickly. Don't let someone's negative attitude spoil others.
- ❖ Set the bar high. Appeal to people to stretch themselves outside their comfort zone; their reward will be professional growth.
- ❖ Have key technical people establish the process and serve as mentors. Relying heavily on the senior engineers to carry the torch most likely will cause them to burn out.
- ❖ Foster trust and encourage team members to help each other.

Pair Programming

Nobody would question that a positive attitude won't help getting more work done. Some people may question whether pair programming will improve the productivity of the engineering team especially when working under schedule pressure.

Pair programming does not mean that one engineer programs and the other watches. Rather, the engineer that does not program writes the unit tests, prepares the documentation or provides the in-process review preventing defects from getting into the source code.

Programming requires constant concentration, which most people cannot sustain for eight hours straight. By switching off and working in pairs people most likely will get more done. The following benefits were reported when practicing pair programming under schedule pressure:

- ❖ Fewer bugs at the unit level
- ❖ Reduced likelihood of integration bugs
- ❖ Reduced risks due to fewer single points of knowledge failure
- ❖ People write cleaner code when someone else is watching
- ❖ Mentoring and learning
- ❖ Uniformity of coding style develops
- ❖ Increased likelihood of reuse of existing components
- ❖ Culture of shared responsibility

Pair programming helps to mitigate problems that arise from extreme schedule pressure, such as miscommunication and re-work. As pair programming can prevent defects from being inserted into the product, it relieves schedule pressure from downstream activities, such as integration test and functional testing.

I have experienced excellent results when pairing engineers with different skill levels. This is especially true when the senior engineer does not only contribute technical skills, but also exhibits a positive attitude and has a Socrates style for transferring knowledge.

Achieving Consistent Decision Making

Under extreme schedule pressure more needs to get done in a day and as a result the number of decisions that each team member makes increases. Sometimes the only directive team members receive as guidance for their decision-making is “*Development speed is the primary goal*”. The particular interpretation is left up to each team member.

There are four suggested work strategies that help team members to make decisions that are consistent with the project goal, the scope and management priorities.

1. Guidelines for the development process
2. Getting the message out – top-down communication
3. Validating the message arrived – bottom-up communication
4. Assistance with issues and questions

If team members follow an official or defacto development process that defines deliverables, the sequence of activities and roles and responsibilities, guidelines exist against which to interpret management priorities and directives. Development guidelines, such as checklists and procedures for delivering source code to the source code base reduce the number of decisions that have to be made.

Getting the message out is not a one-time event, but rather a process, as the message needs to be tailored to each person’s perspective in order to reach all team members. For example, reaching a common understanding about the requirements involves many meetings where the requirements will be clarified, refined, and annotated and new requirements will surface. The information will be communicated in documents, project meetings, e-mail, on the project web site and one on one.

With increasing schedule pressure it is even more important to validate that the information that was shared has been interpreted as intended. An effective work strategy is to engage team members

one on one with open-ended questions, such as “How is the test environment working for you?”, “How is the Java conversion effort going for you?”, “How does this feature fit into the priorities?”.

In spite of top down and bottom up communication team members have specific questions in areas, such as design, work priorities and the interpretation of requirements. For example, individual decisions will be more consistent, if a chief architect is available to discuss and review design issues. A product manager or change control board guarantees consistency by clarifying and approving requirements. A project manager sets priorities for the work to be consistent with the project goals and the scope.

Testing under Compressed Schedules

If the schedule is aggressive for the developers, it will be even more aggressive for the testing organization. A delay in development does rarely result in an extension of the completion date for the project.

The role of testing is not only to test the product, but more importantly to report and communicate the status of the product. Software test is the first time when an accurate statement can be made about the completion status of the software system in terms of functions that are available for test and which functions work according to the requirements and which functions fail.

Under schedule pressure it is even more important that testers communicate every piece of information they know about the system. They describe what they did and what they saw and they communicate their plans for testing. After all, reported defects drive the work of the developers and the decisions of the managers.

Though testers often sacrifice the thoroughness of testing under schedule pressure – both coverage and depth – the importance of repeatable tests increases under schedule pressure. This means having written test procedures, applying configuration management to data files, configuration files and environment variables, and practicing configuration management of source code. Repeatable tests isolate the defects and make it possible to verify that the defects were fixed. They also make it possible to add more people to the testing effort.

Getting help from development increases the resources that test the product. Developers generally do not require any startup training and learn easily from examples.

P2E manages projects and we provide training.
Whatever you need! Learn more about P2E.
Call 303-499-7301
visit <http://www.ullamerz.com>
or E-mail: UllaMerz@aol.com

Conclusion

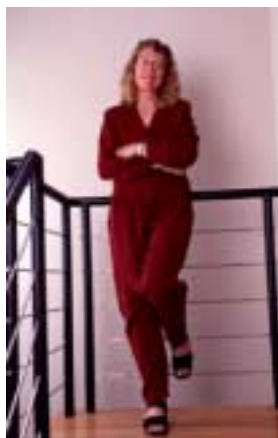
During the question and answer period of the meeting a member from the audience remarked that the focus has been on pressure caused by aggressive deadlines and he asked the question “*How about aggressive quality goals?*” **What would change, if meeting the deadline would not be the only success factor, but quality would be part of the measure of success?** What would happen, if companies would use their own software products in their day-to-day operations and would be their own beta customers for a new product release? I think the economic considerations to release the software product or not would change. The time and effort devoted to testing would not nearly get as compressed and development may have more time to review the design and source code. Holding those who dictate aggressive deadlines accountable for quality measures, such as defects found in the field, staff turnover rate and scheduled re-work incurred in the following release, creates the necessary checks and balances.

Unfortunately, walking away, re-negotiating the schedule or educating management are not always possible options in response to extreme schedule pressure. This report offers several promising and innovative work strategies that increase productivity under extreme schedule pressure. Readers examining these work strategies may observe that these strategies apply in many situations. A panel member agreed with this statement when he answered the question “*What would you do differently under schedule pressure?*” with a simple “*Nothing*”. Instead of doing something differently, I think, the discipline with which these work strategies are applied in a situation of extreme schedule pressure is very important. The work strategies described here cannot be compromised under pressure, but rather need to be adhered to faithfully.

I want to thank the following panel members for their contributions: Scott Allman, Melanie Dalby, Ron LeMaster and Martin Radley.

P2E Calendar of Activities

- ◇ **Thur., July 19th, 2001, 5:00 p.m. – 8:30 p.m.**—“Extreme Event: Continuous Integration & Planning Extreme Projects,” Speakers: Michael Two, Martin Fowler, XP Users Group. www.xpdenver.org/martin_fowler.html
- ◇ **Mon., December 12th – Fri. December 14th, 2001**—Hear Ulla speak, Project World San Jose, www.projectworld.com/2001/NorthAmerica/SanJose

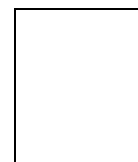


Ulla's Articles in Publication

- ◇ “Small Teams, Big Problems”, Software Development Magazine, March 2001, www.sdmagazine.com/articles/2001
- ◇ “Tough Customers: Toward Win-Win Solution”, Beyond Chaos: The Expert Edge in Managing Software Development, Larry Constantine editor, Addison-Wesley, ISBN 0-201-71960-6
- ◇ “Tactical Project Management at a Startup”. The Software Testing & Quality Engineering Magazine, accepted for publication

**YOU READ IT
HERE FIRST!**

Ulla Merz
P2E
5501 Aztec Court
Boulder, CO 80303



ADDRESS CORRECTION REQUESTED

**INSIDE...
HOW TO COPE WITH EXTREME SCHEDULE
PRESSURE**