Team-Driven Schedule Metrics

Peter Kulik
Kulik & Lazarus Consulting, Inc.
March, 1996

Software development projects are increasingly managed by teams, either explicit cross-functional teams or de-facto implementation teams driven by product complexity. Teaming introduces additional levels of complexity and risk – and potentially significantly greater effectiveness. In particular, there is a growing recognition of the need to aggressively manage development risk, and a consequent need for effective, easy-to-use team-oriented risk management tools. The Team-Driven Metric for Schedule ($TDM_{Schedule}$) tool was developed to improve development risk management in a team-based environment, and has proven highly effective on a wide variety of projects.

Software development is inherently complex, with a large number of factors interacting in ways that are difficult to predict and in some cases counter-intuitive [1]. Cross-functional teams at once hold great promise and an additional level of complexity for management of software development projects.

In particular, cross-functional teams can be an excellent forum for risk management activities. Challenges of software risk management include [2]:

- Understanding of software development management
- Disciplined implementation of good development management techniques
- Availability of tools to perform Software Risk Management
- Broad view of Software Risk Management.

A carefully-constructed cross-functional team can contain an excellent base of experience to successfully execute software development projects. Also, with disciplined team leadership and operation, the implementation of good development management techniques can be consistently applied throughout a project.

The Team-Driven Metric for Schedule ($TDM_{Schedule}$) tool addresses the latter two challenges.

Copyright © 1996, **Kulik & Lazarus Consulting, Inc.**
http://www.klci.com
513-291-1851

Input	First Customer Installation Estimate
Team member 1	Dec-94
Team member 2	Oct-94
Team member 3	Oct-94
Team member 4	Sep-94
Team member 5	Nov-94
Team member 6	Dec-94
Team member 7	Sep-94
Detailed Project Schedule	Sep-94
Mean	Oct-94
Std. Dev. (days)	38.93

Table 1: $TDM_{Schedule}$ input for the project profile shown in Figure 1.

$TDM_{Schedule}$ is a diagnostic tool developed specifically for team-based risk management in execution of product development projects. It enables teams to take a holistic, top-down view of a project in terms of risk, and reflects team member expectations of project results. $TDM_{Schedule}$ has been implemented as one of **Kulik & Lazarus Consulting, Inc.'s** Team-Driven Metrics in the Simplified Risk Assessment℠ service, and has proven effective on a wide variety of projects.

Motivation

$TDM_{Schedule}$ enables a team to define project schedules in terms of risk. The mapping of schedule duration to project risk is implemented by capturing and quantifying team member expectations.

$TDM_{Schedule}$ is based on the following principles:

- Team member expectations are seldom unfulfilled.
- Cross-functional input provides strength in multiple perspectives.
- Anonymous polling minimizes inevitable political influences and team dynamics.

Team member expectations have proven a powerful leading indicator in project execution. For example, if team members expect a project completion date significantly later than shown in the project schedule, experience has proven the project will be late – it is extremely unlikely that the team will surprise itself by finishing early.

Different perspectives are a core strength of teams, and when leveraged successfully, enable teams to grow in effectiveness. Even for projects not being driven by explicit cross-functional teams, we have found that a de-facto implementation team will almost always exist. Team members inevitably have quite different perspectives and expectations, which may not always be evident for all team members in team interaction.

Third, political influences and team dynamics inevitably constrain team discussion and interaction. However, when anonymity is preserved and team member inputs are considered individually, these influences are minimized.

The $TDM_{Schedule}$ tool treats predictions, particularly schedule predictions, as inherently statistical in nature. Statistical analysis is most useful when the following conditions hold: [3]

- High level of variation
- Large amounts of data to analyze
- Existing data represents a subset of the entire population
- Need to draw practical conclusions

These conditions are commonly found in software development. [4]

Benefits

Benefits of implementing the $TDM_{Schedule}$ tool include:

- Helps ensure plan integrity
- Quantifies project risk
- Measures the impact of risk management action
- Builds team consensus
- Serves as a leading indicator of expected schedule slippage
- Increases the impact of risk management actions

Copyright © 1996, **Kulik & Lazarus Consulting, Inc.**
http://www.klci.com
513-291-1851

Example

An example $TDM_{Schedule}$ profile is shown in Figure 1. This profile is from an actual project which the author led. The implementation team was cross-functional, and included seven members representing engineering, product management, finance, purchasing, and quality assurance. Team Member inputs are shown in Table 1 (note that the project schedule was used as an eighth input).

Several points to note about this $TDM_{Schedule}$ profile:

- The schedule is defined in terms of risk by mapping the First Customer Installation date on the x-axis into confidence levels on the y-axis.
- The average of team member responses is 9-Oct – this is the point of *50%* confidence!
- Potential project variation is shown by the steepness or shallowness of the curve and the time scale of the x-axis.
- The discrepancy between the project schedule and team expectations is one to two months at 80% to 90% confidence.

In this example, the $TDM_{Schedule}$ profile was used as an input to project and risk management planning. By aggressively managing risks along the project's critical path, the team met customer commitments and delivered on-schedule.

Implementation

$TDM_{Schedule}$ is implemented using the following three steps:

1. Poll team members:
   "What do you believe is a *realistic* date for first customer installation?"
2. Calculate the mean and standard deviation of team member responses.
3. Use the mean and standard deviation to describe a statistical normal distribution.

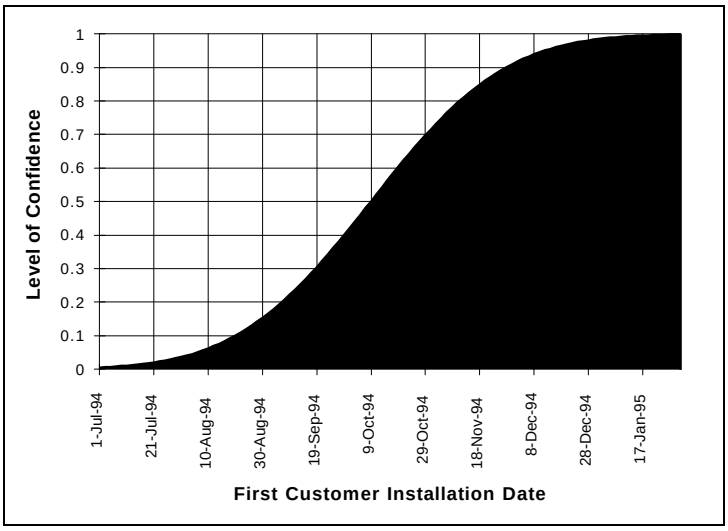

Figure 1: Example $TDM_{Schedule}$ profile based on the data contained in Table 1.

Polling team member s should be done anonymously and confidentially, preferably capturing input from all team members at the same time (e.g. at the end of a team meeting). A $TDM_{Schedule}$ profile will provide relevant data as soon as enough planning has been done for team members to have informed opinions about the project schedule.

The Mean and Standard Deviation can be calculated straightforwardly using spreadsheet functions.

Given the Mean and Standard Deviation, the cumulative probability of achieving any date can be calculated. For example, to create the normal distribution in Figure 1, the Microsoft Excel NORMDIST function was used.

Planning and Execution

As a top-down estimating tool, $TDM_{Schedule}$ complements bottom-up project schedule development. Early in a project, missed tasks, missed dependencies, and underscoped tasks are common in project schedules. These early schedules represent works in process, and need to be considered within the top-down context provided by the $TDM_{Schedule}$.

Early in project plan ning, $TDM_{Schedule}$ profiles will typically reflect conservative outlooks. It is at this point

Copyright © 1996, **Kulik & Lazarus Consulting, Inc.**
http://www.klci.com
513-291-1851

in time that project risk is greatest, since there are the most unknowns that have not yet been fully planned. The slope of the $TDM_{Schedule}$ curve will be relatively shallow, and the 90% certain date will be extended relatively far out in time.

The action of effective planning inherently reduces project risk. As progress is made in planning a project, the detailed schedule will generally shift to the *right* on a time-line, while the $TDM_{Schedule}$ profile will shift *left*. When these two methods converge – when completion date shown by the detailed project schedule corresponds to a $TDM_{Schedule}$ confidence level the team is comfortable with (generally 80% for mature organizations, and 90% for all others) – the team has a strong schedule.

At any point during project execution, the difference between the completion dates given by the project schedule and the $TDM_{Schedule}$ profile is a measure of plan integrity. If the detail schedule and $TDM_{Schedule}$ profile do not agree, the current plan does not have integrity, and the team needs to continue planning.

Risk Management

$TDM_{Schedule}$ is an excellent way to quantify project risk over time. It measures team member expectations, which are based on project progress and risk factors encountered by team members on a day-to-day basis.

$TDM_{Schedule}$ does not, however, determine the underlying reasons for project risk and the most effective ways to manage this risk. **Kulik & Lazarus Consulting, Inc.** has developed the Detailed Risk Assessment℠ service to accomplish this in-depth project risk analysis, identifying those factors with the greatest impact on a project's critical path, and recommending specific actions which can be taken to reduce risk and accelerate project completion.

Conclusion

$TDM_{Schedule}$ is a powerful risk management diagnostic tool for project teams. It can be easily implemented, and provides important insight into:

- Project risk level
- Project plan integrity
- Team member expectations
- Impact of risk management actions

This tool can be implemented throughout project planning and execution, and used as a top-down complement to detailed project schedules.

References

1. Putnam, Lawrence H., and Myers, Ware, Measures for Excellence, Yourdon Press, Englewood-Cliffs NJ, 1992.
2. Karolok, Dale, Software Engineering Risk Management, IEEE Computer Society Press, Los Alamitos CA, 1996.
3. Devore, Jay L., Probability and Statistics for Engineers, Brooks/Cole Publishing Company, Pacific Grove, California, 1982 (Third Edition).
4. Kulik, Peter J., "Team-Based Risk Management for Software Development", AT&T GIS Journal, December 1994.

***Peter Kulik** is Managing Partner of **Kulik & Lazarus Consulting, Inc.** With more than 10 years experience in all aspects of software development, he holds an MS in Engineering Management with the thesis "Practical Quantitative Methods for Software Development Process Management", a Certificate in Economics and Finance, and a BS in Electrical Engineering. He can be reached via e-mail at pkulik@klci.com.*

***Kulik & Lazarus Consulting, Inc.** focuses on enabling software development organizations to accelerate completion of their projects. Leveraging more than 25 years practical experience, we use innovative tools to apply leading-edge critical path and risk management methodologies in an action-oriented framework. Our services enable clients to identify, quantify, and proactively address opportunities to improve their project completion dates on projects of five to fifty people. Contact us at 513-291-1851, or via the World Wide Web at http://www.klci.com.*

Copyright © 1996, **Kulik & Lazarus Consulting, Inc.**
http://www.klci.com
513-291-1851