

## Using Control Charts in Software Development

### On-line discussion with members of the W. Edwards Deming Institute

I belong to several groups on the Linked In web site and I find many of their on-line discussions to be both interesting and informative. Not long ago, a member of the W. Edwards Deming Institute Official Group wrote,

I'm having a hard time imagining how to use Control Charts in the Software Development industry. Software development is a new product development activity, one that requires innovation and creativity (not always, but still). It is rather difficult to have innovation while trying to reduce variation at the same time. Of course, Control Charting a process doesn't necessarily mean one reduces variation as identifying assignable causes is very useful.

Anyway, my question is: "What should one chart in this industry?"

I've seen people applying this visualization to Lead Time variation. But even for this measurement I cannot find that much value in visualizing its variation. Right now I'm working on a big project with tens of teams working on different parts of the product, with QA teams separated from development teams.

This post generated a few responses from other members; among them:

- "Dr. Don Wheeler's books or classes may be of help. He has SPC Press."
- "You can't or should generally not apply any data analysis method until you know what you wish to measure. A good way to understand what to measure is to start with your work processes – what are your key processes in delivering your ends? Once you know your processes, you can start to study them usually with flowcharts and ask what is the purpose of the process – this gives the key outcome measures and then further ask what aspects of the process or inputs to the process affects the outcomes. Now you have more measures to collect and analyze. Then your analysis should start with process behavior charts (control charts). This will likely be too much to do in one go and you need to develop confidence that it is worthwhile. Choose a key process and some useful measures of this as a start."

I started to feel that the exchange was missing the whole point. That is, are control charts even appropriate and applicable to new software development? I joined the conversation and posted:

One major challenge in software development (or any new product development setting) is reducing development lead times. Control charts analyze variation over the longer term. If a development group is taking so long to develop new software that a control chart is applicable, they've got bigger problems than trying to figure out how to use a control chart!

Better to concentrate on Design FMEA, DOE and other techniques that can help to accelerate the new product development. Schmidt and Launsby have a case study on software and operating system factors in their text, Understanding Industrial Designed Experiments, 4th Edition.

As it relates to the application of FMEA, you mentioned in your original post that you're "working on a big project with tens of teams working on different parts of the product, with QA teams separated from development teams." FMEA is best applied by cross-functional teams. In other words, a Design FMEA should not be performed by design engineers alone. The effort must also include Marketing, QA, Operations, Purchasing, Test and other groups. By the same token, later on a Process FMEA team's membership should not be limited to process or manufacturing engineers. Other members should be drawn from QA, Maintenance, Supply Chain and other groups.

Select and apply tools appropriate to new product development. Worry about control charts after releasing the new software into the process in which it's to be applied.

© James F. Leonard. All rights reserved.