

## **Elaboration of Scrum Burndown Charts.**

Combining Control and Burndown Charts and Related Elements

## Introduction

When following the Scrum approach a tool frequently used is the Burndown Chart. A simple chart to show the rate at which a body of work is being delivered. One aspect of this chart is that it's easy to see at a glance if progress in completing the planned work is ahead of or behind the expected burndown rate.

The ease of recognising when a variation to the expected burndown rate has occurred is interesting but not useful of itself. It does not tell us anything directly about what should be done to remediate this situation, the level of remediation needed or if remediation is even warranted. It also tells us nothing of what effect that remediation will have at this point or during the remaining time the chart is tracking delivery.

The thesis of this paper is;

Simply observing variation on the Burndown chart is of no value unless it is responded to appropriately. The chosen response needs to be executed without the need to first conduct exploratory discussion and agreement, about the nature of the response, at the point of observing the variation. Given this an organisation must have its response to variation considered and defined before it is observed and so the organisation needs to understand the following key points:

- what level of standard deviation from the Mean is acceptable and whether this acceptability changes over time
- recognition of common-cause and special-cause events during tracking
- what response options are available to events causing non-standard deviation
- the effect of responses at the time of use and during the lifetime of the activity being charted
- ways these effects can be measured

There for it's evident the use of a Burndown chart alone is not sufficient to provide the level of understanding required and that additional techniques need to be used with the Burndown Chart.

This paper discusses the combined use of Burndown and Control Charts and a number of other approaches from the field of Quality, specifically the knowledge domain of Statistical Process Control (SPC). It should be noted that there are limits to the use of formal SPC techniques within software development. It is possible however to use similar approaches which are adjusted to fit the software development and testing context.

## Scrum Burndown Chart

The Burndown Chart provides an at a glance view of how progress is going with regards to a particular body of work. Charts usually apply to the progress of development work but can also be applied to other work that forms part of the project. Within the test domain Burndown Charts may apply to the rate of authoring of Test Cases after initial functional decomposition or perhaps the execution rate of the Test Cases during the execution phase.

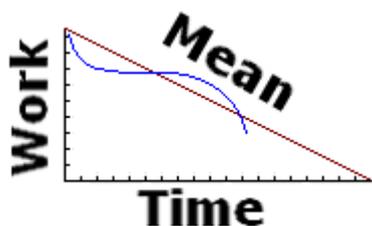


Diagram 1: Outline of a Burndown Chart showing the key axes, Mean line and rate of delivery being charted

Each Burndown Chart contains an axis that shows the work to be done, the time allocated and a mean line for planned progression. As the delivery of work commences the rate of progression is recorded on the chart, as shown by the blue line in diagram 1.

Where the blue line dips beneath the mean the rate of progression is less than that planned for. Where it rises above the Mean progression is greater than planned for.

## Process Control Charts

Within Quality Assurance there is a tool known as a Control Chart that allows the organisation to observe the behaviour of a process over time. A centre line is drawn to represent the Mean value the process can achieve, calculated from process data.

There is also an Upper and Lower Control limit, shown as UCL and LCL on the diagram. These limits define a zone within which process measures are expected to fall. Variance within these limits is called standard deviation and events causing this variation are called common-cause events.

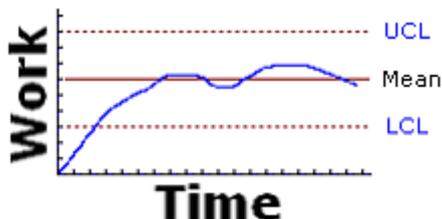


Diagram 2: An outline of a Control Chart showing the key axes, Mean line and rate of delivery being charted

Measures falling outside of these limits and so recorded as non-standard deviation are considered to be 'alarms' that indicate a Special-cause event has occurred. It's important to note that Special-cause events may occur that do not push the rate of delivery outside of the control limit. Similarly cumulative Common-cause events may push the rate of delivery outside of the control limits, manifest as non-standard deviation and trigger an alarm.

## Burndown Control Chart (BCC)

We've seen that the Scrum Chart provides us a view of the ongoing rate of delivery and the cumulative reduction of the work planned in the measurement period. In addition that the Control Chart provides a view of process performance during that period and allows us to quickly recognise when the process is not performing as expected.

Ideally we would be able to track both the rate of delivery and the process performance on a single chart. This would help ensure the team could quickly recognise when the process used to deliver the agreed work was likely to or had gone out of agreed control limits.

By overlaying the Control Limits around the mean line on the Burndown Chart we can draw on the benefits the Control Chart provides for process monitoring.

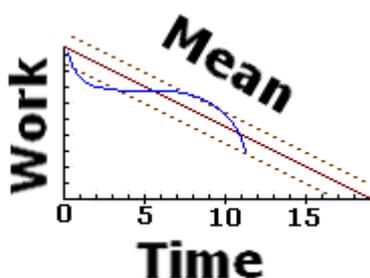


Diagram 3: An outline of a Burndown Control Chart showing how the Control and Burndown Charts combine

In diagram 3 we see the rate of progression is being tracked between two agreed control limits. This makes it much easier to know when a response to the process behaviour is needed and helps avoid the rate of delivery going further out of control. As an example, in diagram 3 we see the rate of delivery is below the LCL on days 2 and 3 triggering an alarm and a response that changes the rate of delivery going into days 4 and 5.

## Elements of the Burndown Control Chart

- **Measurement Definitions**

In order to set-up the Burndown Chart the organisation must have performed analysis of the effort and duration needed to deliver a certain set of work items. In addition they must be able to measure the progress of delivery in some consistent way. This analysis and related planning clearly needs to happen before delivery commences to set expectations around what will be delivered.

As part of using Scrum the organisation would typically analyse the total effort required for the set of activities that will deliver a proposed set of work items. These activities could include; analysis, design and development (or testing), then use the cumulative figure for all items as the measurement of total effort. They would then plan what can be delivered in the period available and move any excess to the next period. In Scrum these periods are referred to as Sprints and most Agile projects employing Scrum will have multiple Sprints.

→ *Further Discussion: Analysis and Estimation techniques*

- **Upper and Lower Control Limits**

To accurately set the UCL and LCL will require the organisation to have data about process performance from previous Sprints. At first reasonable estimations can be made by the project team based on what they think is achievable and it needs to be no more accurate as it's just a baseline against which to plot real data.

The team need to plot the Mean, UCL and LCL lines on the chart. To be clear of what each is:

- The Mean is the mathematical average of the duration needed for each work item within the Sprint.
- The UCL is extended timescales calculated against the longest time an item of work will take.
- The LCL is compressed timescales calculated against the least time an item of work will take.

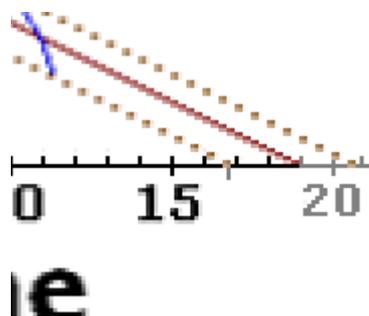


Diagram 4: BCC Showing the extended timeline if the UCL is the more realistic burndown rate.

In our BCC we can see each item of work, referred to as a Packet in Scrum, takes roughly two days to complete. Reviewing our BCC again we can see that if delivery is consistently at the Mean the Sprint will take around 19 days. If delivery is closer to the UCL it will be roughly 21 days and if closer to the LCL it will be more like 17 days.

→ *Further Discussion: Post Mortem Reviews, Calculating the Standard Deviation*

## Common-cause and Special-cause events

In a perfect world members of the project team would be able to analyse and plan to perfection, then deliver in an equally incredible way and keep the rate of delivery perfectly on the Mean. Clearly this isn't going to happen and there are many reasons why not, two broad definitions of which are Common-cause and Special-cause events.

Common-cause events are events that we expect to occur, that are predictable given our historic understanding of the process and generally of low significance. These typically result in standard deviation that sits within the UCL and LCL. The team should follow standard project management practice and identify these as risks by analysis of the current project and review of the previous project review findings.

Special-cause events are emergent and unpredictable given our understanding of the process. They would be expected to result in non-standard deviation that sits outside of the UCL and LCL.

→ *Further Discussion: Project Risk and Issue Management*

## Response Options to Deviation

As previously mentioned there is the possibility that process events will not have the impact expected on the progress of delivery. It won't always be the case that Special-cause events result in non-standard deviation or that Common-cause events always stay within the bounds of the UCL and LCL lines.

There for all variation should be reviewed to ensure that the underlying cause is understood to some degree. Where it's agreed that a Special-cause event has occurred then an appropriate response should be agreed and executed. This review and agreement should be happening daily and for Scrum this would be in the Stand-Up Meetings. It's more likely that rapid identification of Root Causes will occur if analysis is performed proactively within a cross functional team such as those forming Scrum teams.

→ *Further Discussion: Root Cause Analysis*

## Effects of Responses

In the spirit of Scrum the goal is to deliver as much usable functionality as possible in the Sprint. There for being ahead of schedule may be as undesirable as being behind schedule. The team will need to assess if a task ahead of schedule means effort required has been grossly over estimated and whether additional tasks or functional complexity can be accommodated.

Appropriate response options may include extra resources applied to a task that is behind schedule or reduction of resource on tasks that are ahead of schedule. It may be that functional complexity needs to be reduced or there is some other response needed such as additional technical skills within the team, extra hardware or similar common project issues.

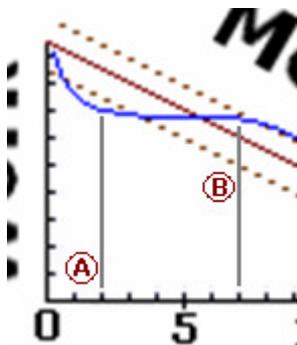


Diagram 5: Initial slippage (a) and the effect of the response to bring delivery back within control limits (b)

In diagram 5 we can see that the first few days the delivery is behind schedule (a), a response to this is executed and the delivery slowly gets back on schedule. However, a common risk is over responding and here we see that delivery may now go out of control in the opposite direction (b).

This situation highlights the need for effective management of the delivery to ensure the team can gauge what is an appropriate response more accurately and the need for frequent review.

## UCL and LCL Change Over Time

Overlaying the UCL and LCL lines fully parallel to the Mean for the duration of the Sprint assumes that we can tolerate and respond to the same level of variation equally throughout the Sprint. In practice this may not be the case as, referring back to diagram 4, we can see we would have to over run the final date of delivery by roughly two days to accommodate a delay near the end of the Sprint.

To keep the final delivery date on target we would need to reduce the overall variation that occurs as the Sprint progresses. In diagram 6 below we can see that tapering the UCL and LCL into the final delivery date will then indicate to us the reducing level of variation that can be tolerated as the Sprint progresses.

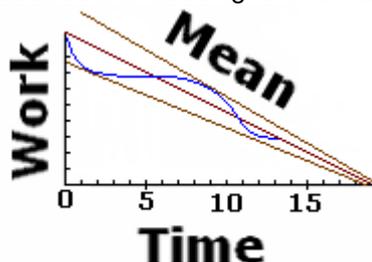


Diagram 6: Burndown Control Chart showing the reduction in allowable limits of variation over time

Strategies to enable a team to work with this tighter tolerance and factors that will affect it are similar to any traditional project, factors such as:

- Ensuring the more complex and higher risk items are worked early in the Sprint where possible.
- Management of the Packet size so the potential scale of issues causing variation is reduced.
- Understanding the availability of the team members and their level of competency.
- The accuracy of the analysis of development, testing needs, etc. and estimations of effort.

## Calculating the Mean

In order to calculate the Mean on the Burndown Control Chart the team simply need to calculate the mathematical average of the duration required for each work Packet in the Sprint. It's expected that packets will vary in size, however it's advisable they don't vary so much that the 'shifting' of the UCL and LCL from the Mean fails to provide the control needed.

If the organisation is practiced enough it should be possible to design the work Packets with roughly equal size and complexity and make it more likely the team will have, from project to project, Packets that take approximately the same effort to deliver.

On review it would seem to make sense the Mean would be more accurate and agreed on in a more informed way if the team had access to historic data related to previous projects. However, the question needs to be asked about how useful this really is when the Mean for a current project is so simple to calculate and will be more contextually correct.

## Calculating the Control Limits

The UCL and LCL should be set at a calculated distance from the longest or shortest duration Packets in the Sprint. For example, if we had a Sprint with 10 Packets and each Packet would take the following number of days: 4, 5, 5, 5, 6, 6, 6, 6, 7 and 7 we'd have 57 days of effort in the Sprint giving us a Mean of 5.2 days effort per Packet. We've now set the axes and the Mean line.

When estimating the duration needed to deliver each Packet the organisation must accept there will be variance against the planned duration. How much this variation is depends on how effective the team are at estimation and managing risks affecting delivery.

If when planning the estimations are given with a margin of variance already stated this can be used to set the UCL and LCL. For example, it may be stated "Duration = X days (+/-25%)" in which case the UCL for our example will be set at 8.75 ( $7 \times 1.25 = 8.75$ ) or just 9 days for ease of working which is as good as 4 days out from the Mean. The LCL will be set at 3.2 ( $4 / 1.25 = 3.2$ ) or 3 for ease of numbers which is 2 days out from the Mean. The +/- variation for each estimate will then sit within these limits if the +/- 25% holds true.

## A Note on the Delivery Line

The natural variation the organisation expects in the duration of delivery for individual Packets of work may present a risk where Packets are notably different in size. In our example we have a Packet of 4 days and one of 7 days which means the variation of +25% for the 4 day Packet will be within the UCL and LCL. This is an alarm condition in terms of delivery for the Packet but as the BCC is primarily for the Sprint all may appear OK for the untrained observer.

However, it's not only the location of the delivery line that is important but the shape. If the 4 day Packet ran on past 5 days the delivery line would flatten and this in itself would be an alarm for the Scrum team. It's also expected that the owner responsible for delivery the Packet of work will raise the alarm in the Stand-Up meeting.

## Conclusions

We can see there is a reasonably straight forward way to look at the delivery line on a Scrum Burndown Chart and not just observe the variation but understand what it means to us and how we might respond.

Is this an over elaboration of the Burndown Chart that perhaps adds complexity where it's meant to be avoided? The answer can be yes or no and depends on the context of your environment.

In an organisation where Scrum is being used effectively, estimation of effort is acceptably accurate and where continuous improvement is happening then control elements may not be needed. Delivery is within acceptable timeframes and it may be felt there is enough control to give the organisation the confidence they need. Conversely introduction of the Burndown Control Charts could be just the thing to further optimise the current process.

Where the organisation has perhaps just introduced Scrum and a closer eye needs keeping on the general process then control elements that Burndown Control Charts bring may prove useful. It will provide a way to recognise when variation from the Mean is becoming an issue and give the team an early warning to consider responding, then monitor in clearer terms the effect of the response. Conversely this could be step two after the recently introduced Scrum approach has had time to be adopted and established.