Using end-to-end metrics to understand and mitigate software delivery (capability) risk

Whitepaper

This is part of a series of abridged whitepapers intended as quick reference sources for busy managers interested in the subject matter and faced with limited time to absorb lengthy research documentation.

It is based on research undertaken by Plandek drawn from anonymised data observed across a range of clients – from small start ups to large corporates with large scale, distributed Agile teams.

About Plandek

Plandek is the leading Agile and delivery metrics BI platform, providing an end-to-end view of your software delivery cycle.

Our SaaS solution allows mining the data history from the toolsets that engineers use for actionable insights.

We provide new insight derived from its unique end-to-end view of the delivery process.

Introduction

Purpose of this Paper

Governance and risk management is an increasingly active research area in Agile software delivery – particularly in large scale organisations. Moving to an effective Agile methodology is a major strategic decision. It takes a huge amount of time and effort and inevitably questions are asked (from the C-suite down) about its effectiveness and reliability for critical software delivery initiatives.

Moreover, Agile by its very nature, involves decentralising responsibility to small self-determining teams working in a more organic (agile) way than would be the case in a more traditional waterfall environment. This decentralised model (which is quite rightly at the heart of the Agile philosophy) can make understanding software delivery risk difficult, without effective metrics in place.

As a result, we often hear of the exasperation with existing RAG (Red, Amber, Green) progress reports – with workstreams classified as “Green” for weeks in a row, before flipping to “Red” with apparently no warning!

This short whitepaper discusses the analytics and metrics that can be applied to try and ensure that such surprises do not happen, as delivery managers have a much better understanding of the underlying risks within their software delivery teams (capability).

Delivery Capability Risk

For the purposes of this discussion, we are defining "delivery risk", as the risk of delivering software increments:

- later than expected; and/or
- of worse quality than expected; and/or
- requiring more effort/resource than anticipated.

Understanding software delivery risk in totality is a complex task, with a range of internal and external factors that drive delivery risk. This paper is interested in a key internal risk that is directly controllable by the delivery team, that we term Delivery Capability Risk (DCR).

The concept of DCR is summarised in the graphic below. There is a great range of Enterprise Agile Planning solutions that help you manage your delivery journeys (programmes). They track scope, effort and apparent progress. What they are unable to do is really understand how effectively the teams writing and releasing the software are working together.

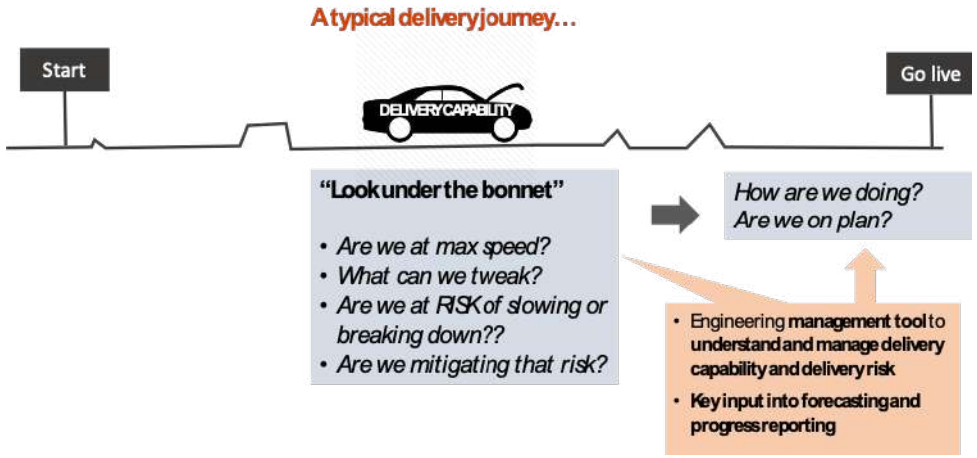

Figure 1 - Graphic showing the concept of Delivery Capability risk assessment

Understanding Delivery Capability Risk in complex IT programme management

There are a set of metrics that can quite accurately track delivery capability risk (DCR), but they are tricky to surface without specialist BI solutions like Plandek.

Plandek for example works by mining data from toolsets used by delivery teams (such as Jira, Git, CI/CD tools and Slack), to surface the metrics critical to identify and manage DCR.

It creates a balanced set of metrics that determine delivery capability risk, using both quant data from the underlying tools sets such as Jira, Git etc – and also from the engineers themselves via constant polling through Slack or other collaboration hubs.

The metrics fall into five logical categories which when synthesised together, give an accurate measure of DCR when tracked over time. These categories are:

- Backlog health analysis – metrics and analytics to understand as far as is possible the state of team's backlog, especially as it relates to the current and next programme cycle;
- Talent – quant metrics to understand your delivery teams' morale and views on process effectiveness (collected via polling on collaboration hubs);
- Process efficiency and transparency – metrics that reveal the effectiveness of the end-to-end delivery process (e.g. Flow Efficiency and Lead Time analysis) which reveal bottlenecks and friction in the process;
- Throughput and time to value – metrics showing volume of work being produced and time taken to deliver across the end-to-end SDLC;
- Delivery (sprint) accuracy – metrics showing teams' ability to meet their own sprint goals (for Scrum Agile) which is a key determinant of likelihood of delivering over longer time periods (e.g. Programme Increments).

Understanding Delivery Capability Risk in complex IT programme management

Examples of these metrics are shown in Figure 2 below .

Backlog analysis	Backlog size and health
Talent	Engineer morale score Engineer Sprint effectivenessscore Engineer teamwork rating
Process efficiency and transparency	Flow efficiency WIP per Developer Return rate %time Bugs/Features Deployment Frequency Deployment Failures Time to Restore Speedingtickets
Throughput and Time to Value	Cycle Time Lead Time Featuretickets completed
Delivery Accuracy	Sprint overall completion Sprint target completion

This balanced scorecard of capability risk metrics add a new dimension to overall programme risk management.

As Figure 2 shows, these metrics are principally designed for use in an Agile delivery context (with concepts of Cycle Times, Sprint Completion etc), but many can also be applied in a hybrid "Scrumfall" context (often adopted by larger organisations to deliver major projects).

For example:

- Metrics relating to backlog health are clearly key in any context (and reveal hidden risk);
- real-time understanding of engineer morale and engineer feedback as regards the delivery process are also critical leading indicators of (hidden) delivery risk; and so too are
- changes in time spent (and the efficiency of) fixing bugs and technical debt.

These are all "under the bonnet" metrics that when viewed together, give the experienced Delivery Manager a view on the health of the delivery "engine" – is it firing on all cylinders, or running on empty…?

Applying delivery capability risk to overall project risk management frameworks

Programme management techniques typically map the various workstreams and understand interdependencies and the critical path.

These techniques create well organised Gantt charts showing the theoretical progress of the project relative to planned milestones. However, **what these techniques cannot do, is effectively track the health of the underlying technology delivery capability.**

- **i.e. the Gantt chart may show that we just hit a key milestone, but an understanding of the health/stress of the underlying delivery team may paint a very different picture.** It may show that this was achieved in an unsustainable way (low morale, declining process efficiency, increasing technical debt etc) – hence the team is unlikely to hit the next milestone.

This is why an understanding of ***delivery capability risk*** (i.e. understanding the health of the underlying delivery "engine") can be a vital extra dimension in complex IT programme management.

This is indeed why Plandek is used as a delivery risk management tool to be applied in conjunction with existing Enterprise Agile Planning tools (such as Jira, Jira Align, Rally etc).

What we do

The Plandek SaaS solution mines and analyses data from key systems used by development teams and synthesises key metrics from these disparate data sources to give unique insight across your end-to-end delivery cycle.

For more content, drop us a message at

hello@plandek.com or visit our website plandek.com

Follow Plandek

facebook.com/plandekuk

@_Plandek_

www.linkedin.com/company/plandek

Plandek.com