

# Plandek's Top 5 Early Warning Signs of Problem Technology Projects

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

The analysis presented below is focused on the software development process and is particularly relevant for Scrum Agile environments.

It is designed to give delivery managers a practical guide to the KPIs & metrics that directly drive project productivity and hence are strong predictors of project velocity and delivery timing.

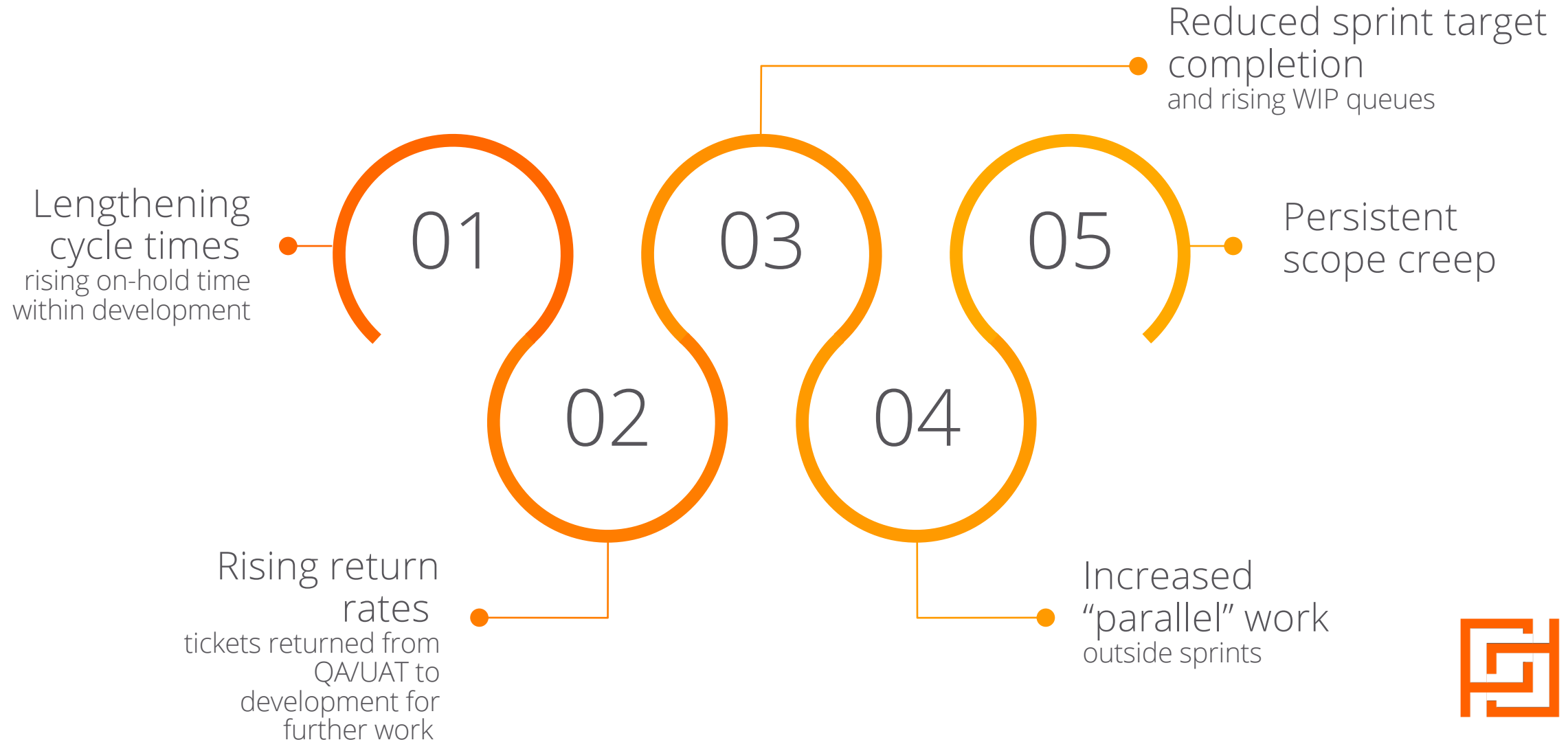
Reviewing these early warning signals regularly gives team leaders and delivery managers the ability to intervene early, in order to improve project outcomes.

## Nature of the Analysis

The following analysis is based on Plandek research undertaken in-house and in consultation with a number of clients, partners and interviewees principally based in the UK, consulted between September 2017 and April 2018.

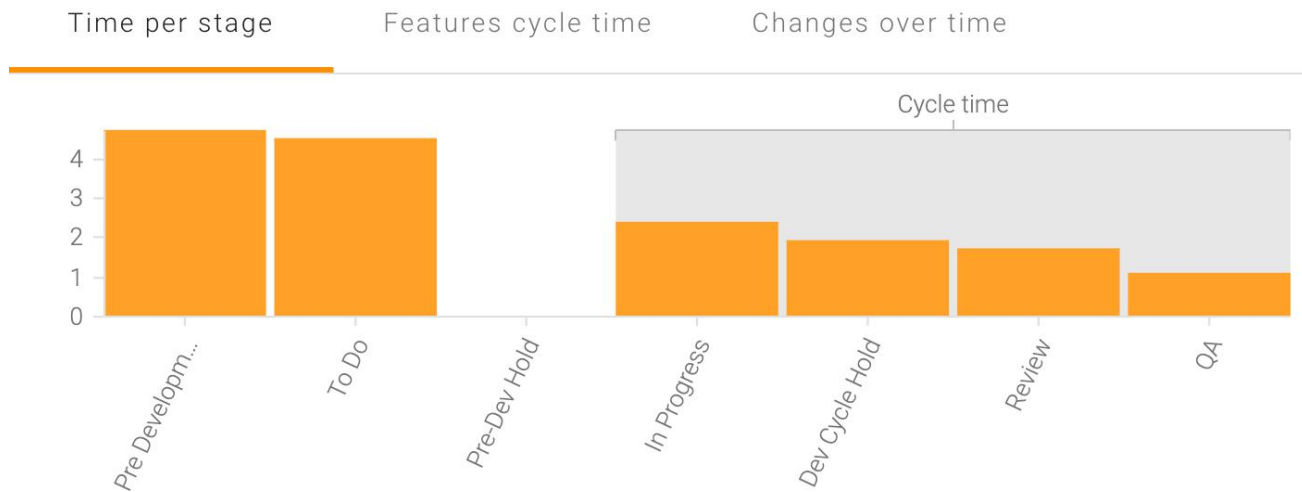


The top 5 KPIs found as indicative of future problem projects are as follows (in no particular order):



# KPI 1: Lengthening Cycle Times

Cycle times in both delivery of new features and upkeep vary for many different reasons. For example, they may lengthen as teams tackle complex project areas - but teams which show a persistent trend of lengthening cycle times require investigation and potential early intervention by PMs.



Regular analysis of the key stages within the cycle (from Jira statuses for example), shows the key bottlenecks and if these are increasing the overall cycle time.

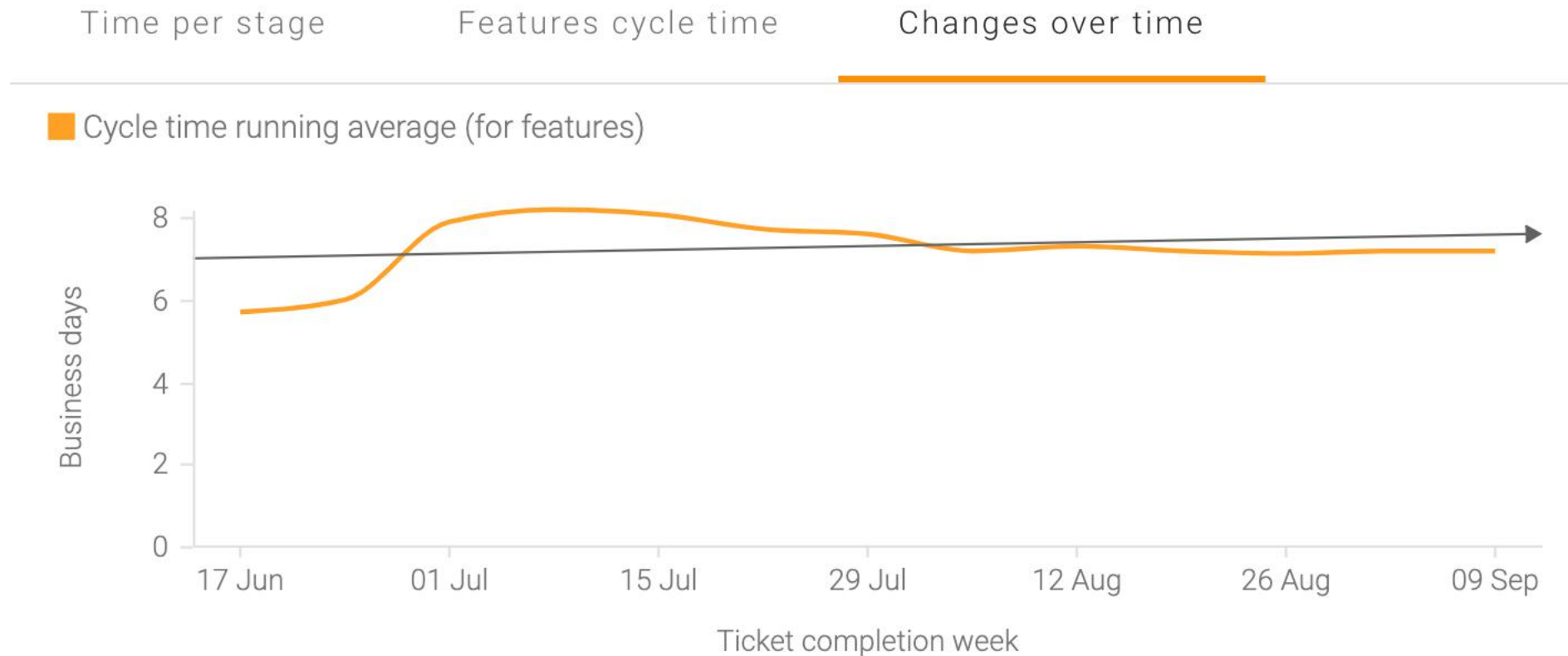
Typical bottlenecks visible include:

- A** Pre-development “On-hold” Time – where tickets are held waiting to be actioned as a result of input required (and not forthcoming) from a stakeholder or engineer
- B** Development Cycle “On-hold” Time – where tickets are frozen pending a dependency within the team
- C** QA delays -potentially as a result of problems, environment setup or resource issues within QA.



# KPI 1: Lengthening Cycle Times

Bottlenecks are near ubiquitous and need ongoing attention. But the key metric to review constantly also is the trend in cycle times as shown in the graphic below. These data are reviewed by individual team within a project and whilst the data needs to be considered very carefully, there are clearly observed cases where teams developing difficulties exhibit lengthening cycle times (often with quite rapid change over time).



## KPI 2: Rising Return Rates

Too few development teams take time to really quantify and understand the impact of rising return rates on overall velocity. As with any production process, returns cause friction and inefficiency. A deterioration in return rate can be a sign of the brightest and best engineers wrestling with the most complex of coding issues, but it can also be indicative of problems within a development team.

Plandek's research has shown a clearly observable 80:20 rule – with on average 80% of returns accounted for by 20% of engineers. Again, these data need to be reviewed judiciously and in context of the teams and projects in question. However, on deeper analysis it is often true that the data reveals engineers who may be new to the team or code base, who would benefit greatly from mentoring and collaboration with peers.

Return rate

13.2 % ↗ 0.6

This measures the number of times tickets are returned from QA. [Learn more](#)

Previous period

Return rate

Change

12.6%

0.6

Tickets

Number of returns

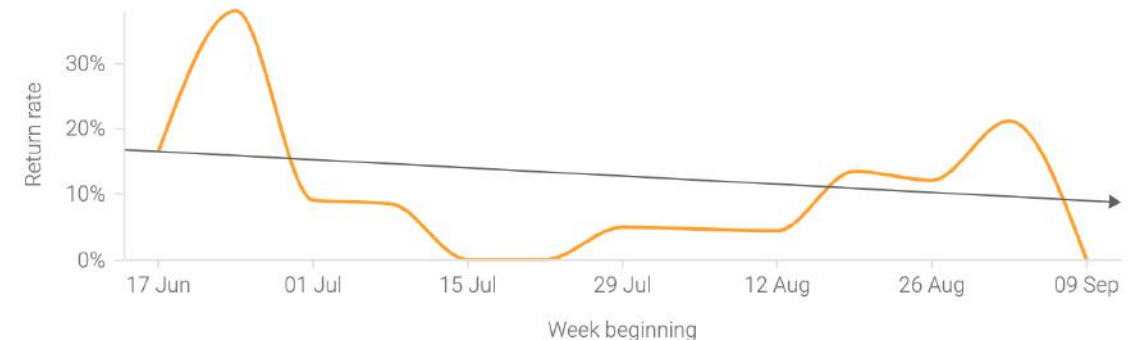
Total tickets

61

461

Changes over time

Return rate percentage



since 17th Jun 2019

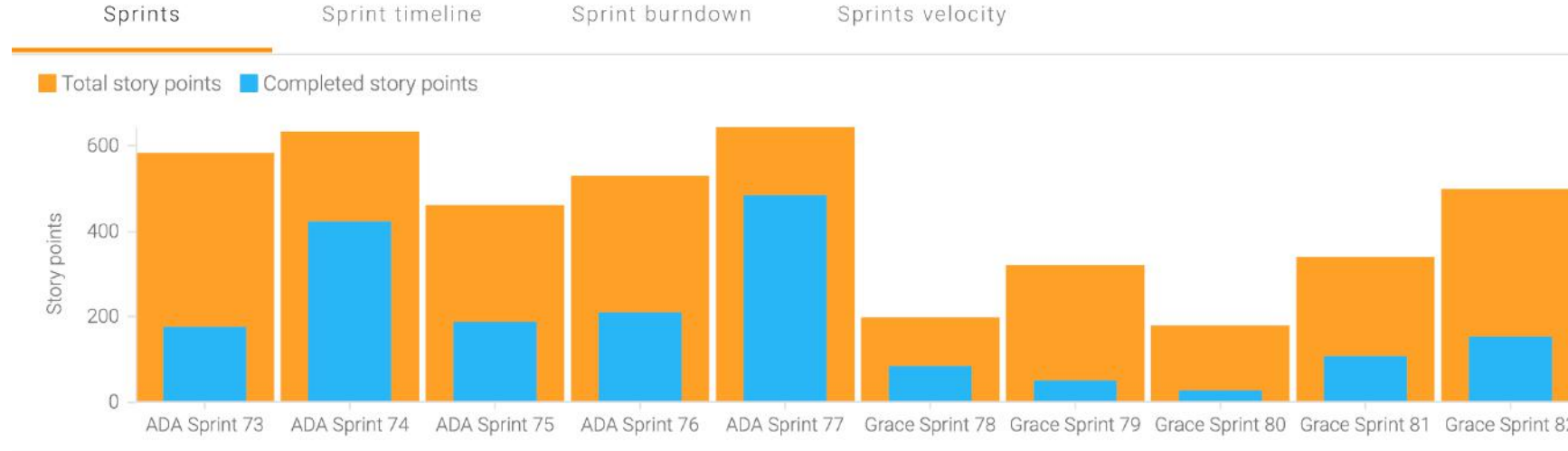




## KPI 3: Reduced Sprint Target Completion

It is a common finding that projects run on with a Scrum Agile methodology, struggle to achieve their sprint goals. It goes without saying, that failure of individual teams to estimate, plan and commit accurately becomes highly problematic in larger Agile projects. As a result, Sprint Target Completion is a measure that is vital to review regularly (and track over time) across all teams and sprints.

Analytics can show clearly individual team's sprint completion performance over time and why targets were not hit (i.e. where tickets were not completed, removed from the sprint or descoped). Big deltas will be observed between teams and whilst actively managing against sprint completion targets incentivises teams to reduce their targets (in order to ensure that they hit them), this may be a desirable outcome as it increases the predictability of output (even if theoretically slightly reduced).

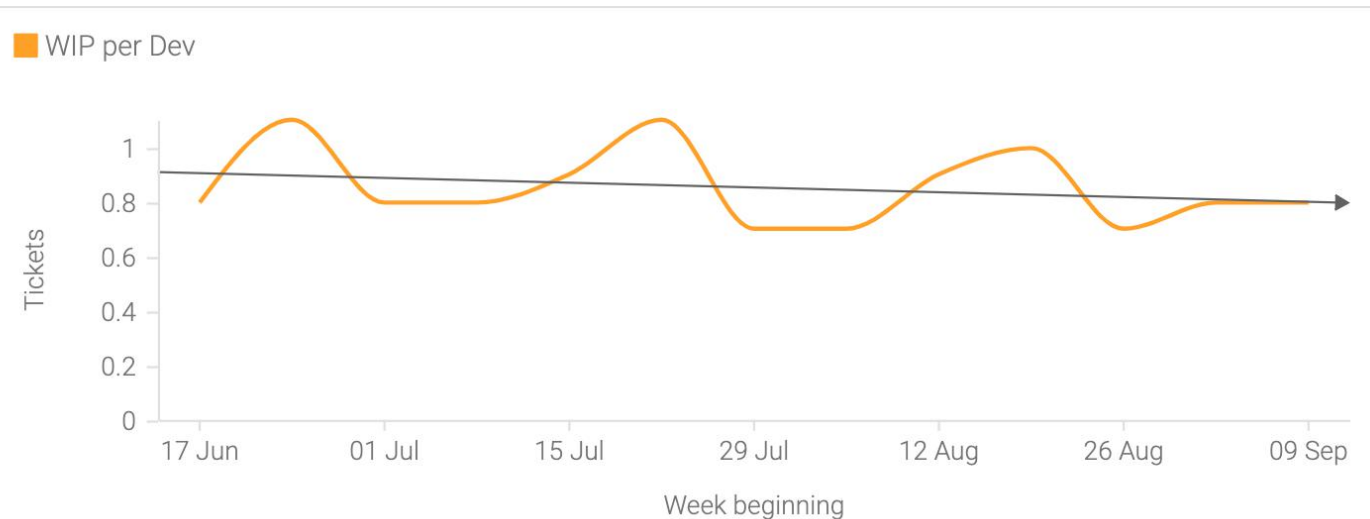


## KPI 3: Reduced Sprint Target Completion

Key metrics relating to teams' inability to deliver sprint goals include analysis of Overdues and WIP Per Developer. The latter can be an important early warning signal that should be constantly reviewed by PMs.

Ticket Overdues can be an interesting indicator of growing project problems. Teams can drown under the pressures of overdues and may need to insert additional sprints to clear the backlog. As such, upward trends in Overdues are best avoided

WIP per Dev















Similarly, keeping a careful eye on WIP Queue sizes across teams is very important. The WIP Per Developer need viewing in the context of the team size to which they relate and if growing, are a clear early indicator of a team that may be under stress.



## KPI 4: Increased “Parallel” Work Outside the Sprint

Teams tend to introduce a Scrum Agile methodology with tight disciplines at the outset. But projects are complex and the pressures on teams varied (often juggling multiple stakeholders, with conflicting priorities). As a result, a disciplined Scrum Agile methodology with clearly defined sprint goals, can often deteriorate.

Indeed, it is common for tickets to be worked on that are not included in the scheduled sprint activity – these maybe “stray” tickets from previous sprints – or in more serious cases – parallel streams of work being undertaken for a different stakeholder “pulling in a favour”. What is clear however, is that this parallel work needs to be identified early as it will inevitably affect overall project velocity.

Project name	Process events		Commits	
	↓ Ticket development outs...	Speeding through stages	Commits without a pull req...	Commits without a ticket r...
Grace	100.0% 	13.6% 	20.6% 	15.3% 
Katherine	100.0% 	14.6% 	41.7% 	75.0% 
Mary	100.0% 	16.6% 	19.0% 	39.0% 



Ticket development  
outside an allocated  
sprint



## KPI 5: Persistent Scope Creep

- The most common and often most significant early warning sign of potentially problematic projects is unsurprisingly scope creep. Agile brings stakeholders closer to the development process which is clearly a very good thing, but the downside is that it is easy (and tempting) for stakeholders to move the goal posts.
- All good PMs know that scope creep is happening, but the challenge often comes in quantifying it and communicating it back to stakeholders early and simply. This is complicated by the fact that Agile projects involve *scope change* almost by definition. *Scope creep* on the other hand, is where scope change becomes excessive and not within the agreed parameters at the outset of the project or milestone.
- Many projects persist for months with accusations of scope creep that never get properly quantified and discussed. As a result, stakeholders continue to push their luck and projects suffer significantly. Burndown charts (for a Version) in Jira can isolate scope change, but are not ideal in isolating *scope creep*.
- Simpler burndowns as shown below are designed to show scope creep and be understandable by stakeholders. Added story points are clearly visible and are linked back into Jira in order to confirm those story points are considered *scope creep* rather than valid scope change.



## KPI 5: Persistent Scope Creep

The story points defined as scope creep can then be subject to further cost/timing analysis, where the story points added are costed (at a weighted average cost of delivery) and expressed as a scope creep financial cost and scope creep time cost to the stakeholder.



Complete  
95.1%



Total Story  
Points 163.0



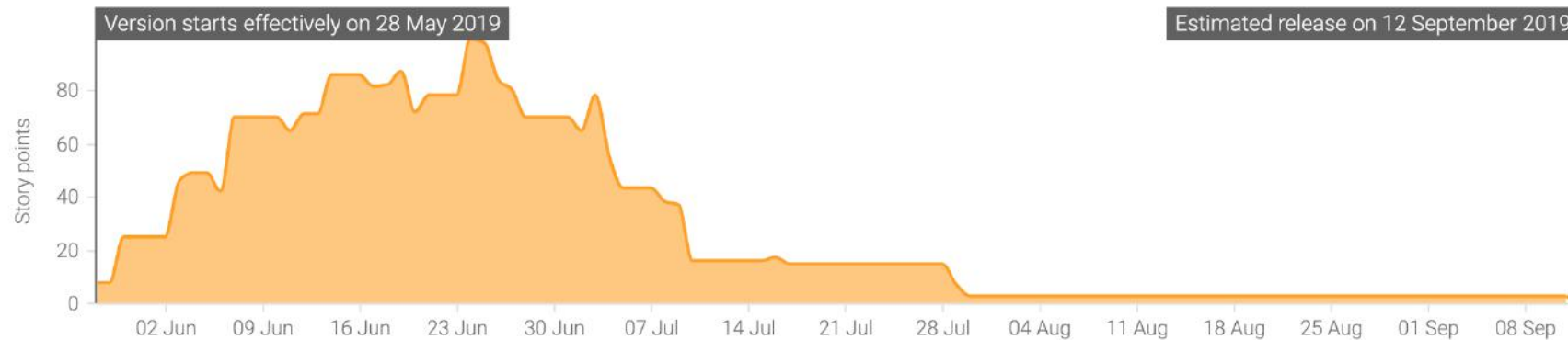
Remaining  
Story Points  
3.0



Average Daily  
Velocity 3.6

Version burndown

■ Remaining story points    ●●● Estimated release date



# Summary



- The early warning signs discussed above may appear obvious, but they are often not regularly scrutinised by PMs and delivery managers. Indeed, the scrutiny has to be directed at team and sprint level to be effective.



## 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](mailto:hello@plandek.com) or visit our website [plandek.com](https://plandek.com)



## Follow Plandek



[facebook.com/plandekuk](https://facebook.com/plandekuk)



[@\\_Plandek\\_](https://twitter.com/_Plandek_)



[www.linkedin.com/company/plandek](https://www.linkedin.com/company/plandek)



[Plandek.com](https://Plandek.com)

