7digital Development Team Performance

Period April 2012 –April 2013

Statistical Analysis Report

Contents

Summary

The 7digital Development team has been recording data on the work it's been undertaking for almost four years. This is the third long term analysis of data collected and covers the period of a year between April 2012 and April 2013. As well as looking at trends over that period the data will be compared to that collected in previous report as well as longer term trends of all data collected so far. Also new to this year is data on the historical team size (from January 2010) which has allowed us to look at the ratio of items completed to the size of the team and how the team size affects productivity.

In general the statistics are very positive and show very significant improvements in all measurements against the last reported period:

- **a 31% improvement in Cycle Times for all work items**
- **a 43% improvement in Cycle Times for Feature work**
- **a 108% increase in Throughput for all work items**
- **a 54% increase in Throughput for Feature work**
- **a 103% improvement in the ratio of Features to Production Bugs**
- **a 56% increase in the amount of Items completed per person per month**
- **a 64% increase in the amount of Features completed per person per month**

This continues to provide strong validation for our approach to software development at 7digital. It's particularly pleasing to see such a significant improvement in the ratio of Features to Production Bugs, something that the first two productivity reports showed we'd not been making any significant progress with.

It's worth noting that whilst we only record data for the Development teams these improvement are down to the entire Technology team, which includes our Infrastructure and DBA teams, and also our Product Management team whose trust and support we've absolutely needed to allow us to increase the amount of Infrastructure work we've been doing, which we're positive has had a huge influence in increasing our productivity.

As always, there's still a lot of room for improvement and we should expect to see these metrics continue to improve, but maybe not at quite such significant factors.

What this report cannot not tell us

(Plus a general warning about statistics)

There are many important thing which these statistics do not tell us, like whether we're building the right thing in the right order, how satisfied our customers are (stakeholders, partners or end consumers), if we're making more money for the company than we're spending and so on. So not much use at all then!

They're also just statistics. On their own they don't tell us anything and are only ever as reliable as the data collected. Statistics should always be used as an indicator and alongside other forms of analysis (such as product KPIs, financial targets, objectives etc.). Improving them should never be the end goal as it's widely known that doing so results in getting what you measure.

Background

The data

As part of an introduction of better practices to the 7digital development team we started recording data on all work undertaken. The records begin in April 2009. The period measured in this report begins 1st April 2012 and runs until 31st March 2013. In total 1329 items were recorded for that period. Each item has a type (e.g. Bug, Feature, Infrastructure), a development start date and a completion date. The data is recorded into spread sheets by the teams from their Kanban or task boards after the morning stand up meeting every day.

The team

By April 2012 there were around 29 people in the development team. This increased to 36 by June 2012 but went down to 33 people by April 2013 – a 14% increase overall by that time. The team mainly consists of developers but also includes three Quality Analysts and a UI/UX developer. For most of the period we've been grouped into 6 or 7 smaller delivery teams. The teams have largely remained consistent for that period but team size has fluctuated.

During the period we also increased the size of our DBA and Infrastructure teams to about double their previous size (mostly in 2013). We do not currently track work produced by these teams in such a manner we can add them to this report.

The practices

Our approach focuses on self-organising development teams with the overriding objective being *maintainable, sustainable development*. At a more granular level this includes a focus on practices such as *Continuous Delivery, Continuous Improvement, Kanban and Theory of Constraints, Test Driven Development, Refactoring, Pair Programming, and Emergent Design.*

At an architectural level we're big advocates of *Service Oriented Architecture* which as well as making our platform more scalable and fault tolerant, allows teams to work independently of each other, only having to work on small parts of the system without impacting each other. Where we still have interdependencies we are continuing to strive to remove them.

Work Type Definitions

Feature

A Feature (aka MMF – Minimum Marketable Feature) is any piece of work which is adding direct value to our services by adding or enhancing functionality.

Production Bug

A production bug is a fault with an application or service which has been identified and fixed in production, rather than a bug identified in a feature before it has reached production (which we call a Feature Bug). Here we are only tracking bugs which have been considered important enough to fix. There are probably bugs which get identified which never get fixed as they are considered not sufficiently critical that fixing them should take precedence over other work.

Infrastructure

Often referred to as “non-functional requirements”, infrastructure work is generally highly technically focused work with no visible improvements to functionality or user experience. Examples might include maintenance to servers or configuration to cope with scaling, resilience, security or to improve the performance and maintainability of automated testing and deployment.

Measurement Definitions

Described below are measurements we use to measure productivity and predictability. Read more about our approach to measuring productivity here:
http://blogs.7digital.com/dev/2012/01/06/productivity-throughput-and-cycle-time/

Cycle Time

A measurement of the number of working days an item of work takes from when development is started through to completion.

Throughput

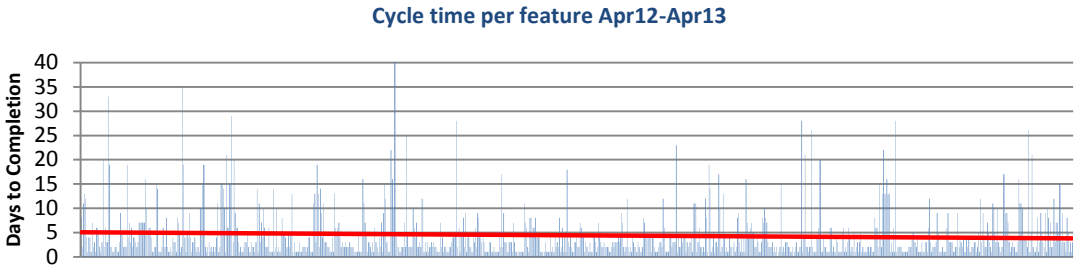
“Throughput can be best described as the rate at which a system generates its products / services per unit of time. “
http://en.wikipedia.org/wiki/Throughput_(business)

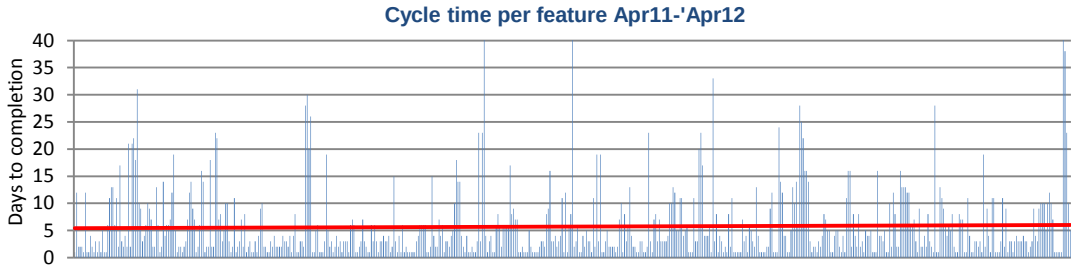
Productivity

Cycle time

Features

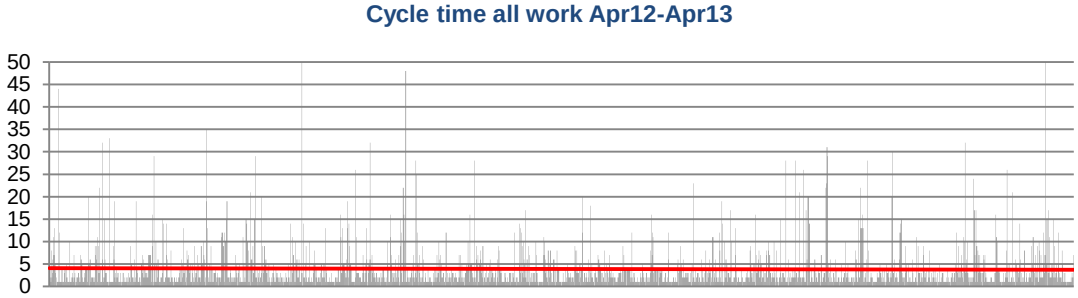
Our average cycle time for features in this period was 4.7 days, compared to 6.76 for the last period, **a 43% improvement**. The trend line shows a gradual improvement, compared to the previous period which was flat or even slightly increasing.

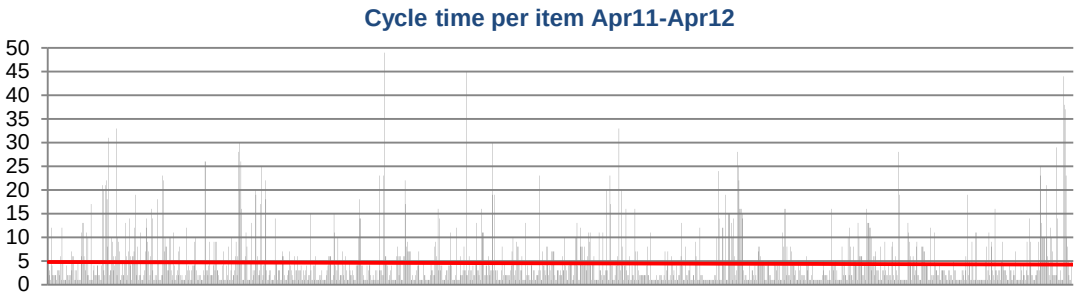




All work items

Our average cycle time for all items in this period was 4.7 days, compared to 5.2 days for the last period, **a 31% improvement**. The trend line shows a very slight improvement over the period.



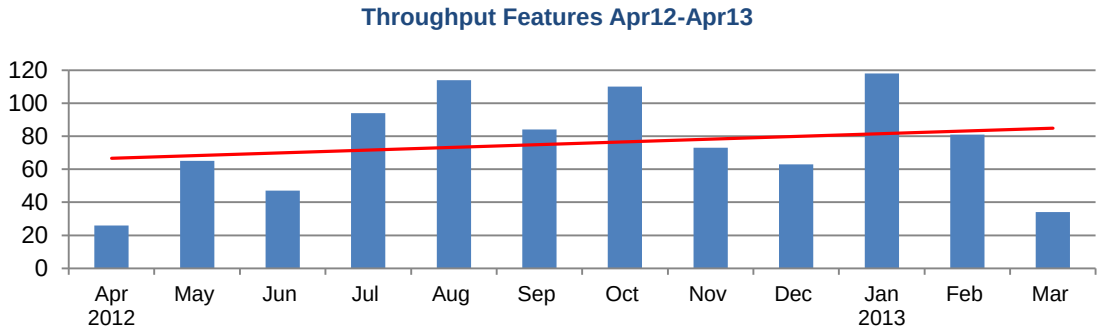


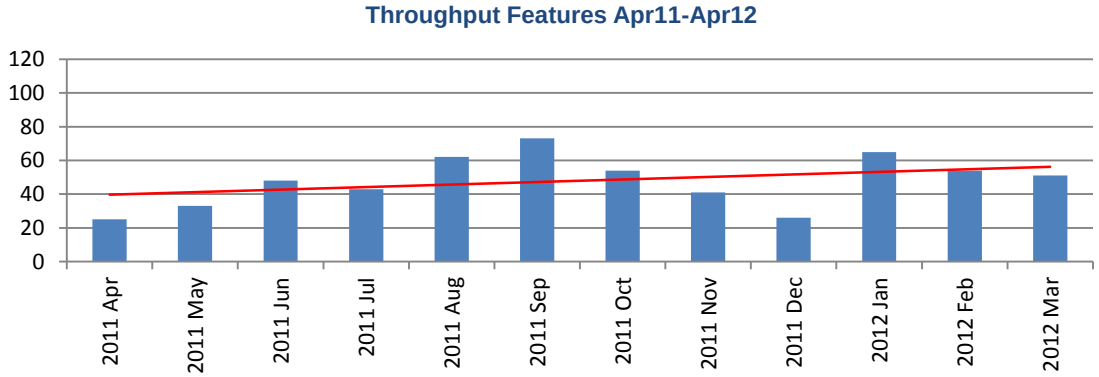
Throughput

Features

A really positive improvement in throughput of features (arguably the most valuable work we do). We averaged 76 features a month compared to 48 a month in the previous period and **a 55% increase in throughput** across the whole period.

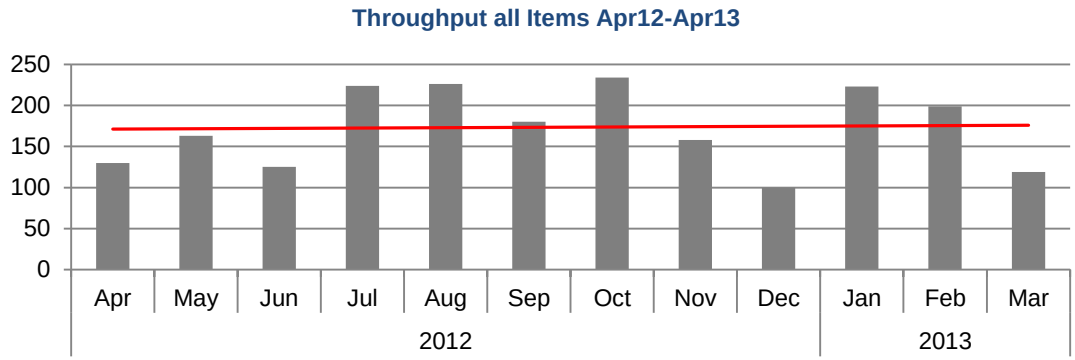
The charts below show a similar seasonal trend (but note the scale is much higher in this period compared to the previous shown below it). It's worth noting the low in March 2013 was most probably due to major infrastructure work occurring in that month (Database upgrade and data centre move).

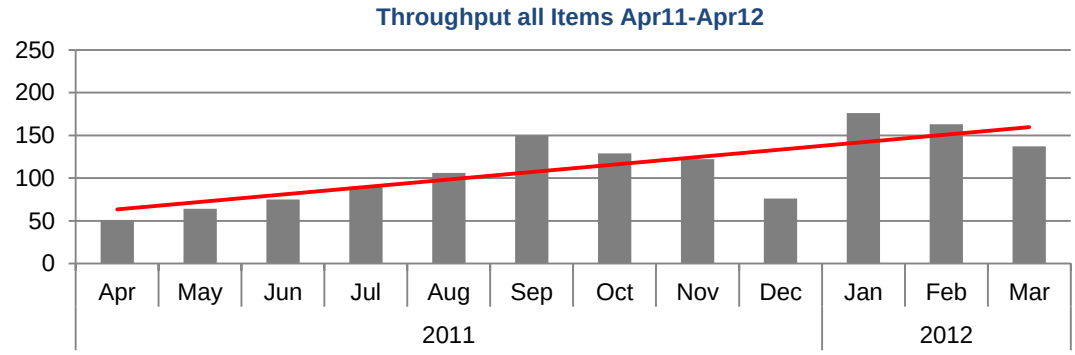




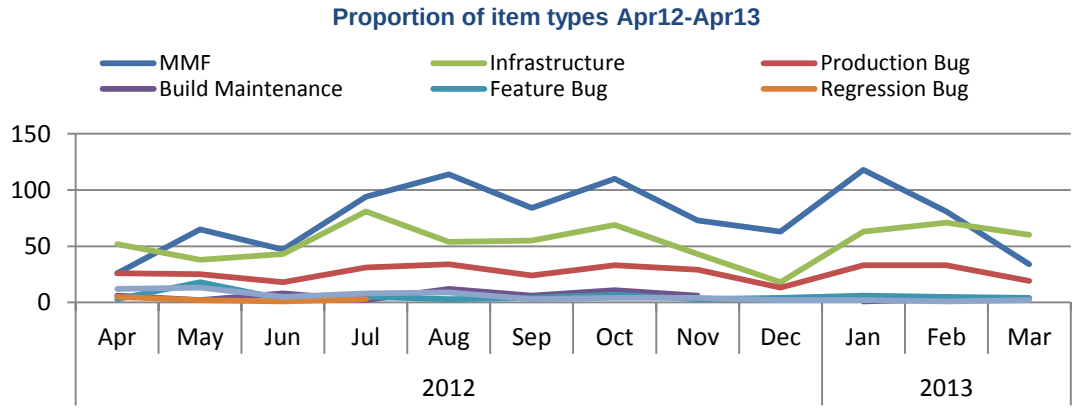
All Work Items

Similarly positive improvements in throughput all work items. We averaged 103 items a month compared to 60 a month in the previous period and **a 108% increase in throughput** across the whole period compared to the previous. The trend line is pretty flat, but comparing the vertical axis scale to the previous period shows we started improving significantly Jan-Mar 2012 and have maintained that momentum. Again Mar 2013 is low due to major infrastructure work during the month.





All work items increased at a greater rate than Features, the following chart shows the breakdown by all work items and suggests the difference is made up by Infrastructure work.



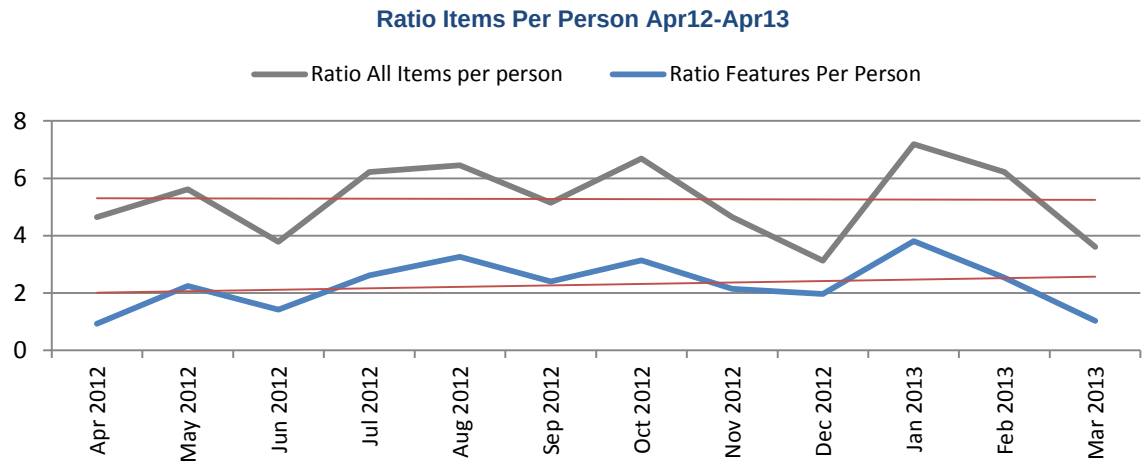
Items completed per person

For the first time we've also managed to collate our data on the overall team size on a monthly basis (back to Jan 2010). We can therefore compare the amount of items completed to team size on a monthly basis.

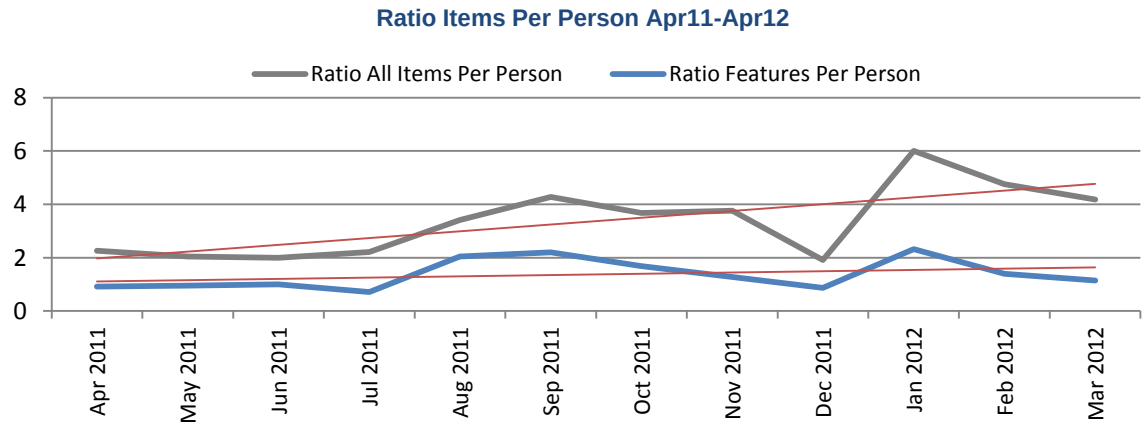
In this period we averaged 5.3 Items of work per person per month and within that 2.3 Features per person. In the previous period we averaged 3.4 Items and 1.4 Features:

A 56% increase in the amount of Items completed per person per month
A 64% increase in the amount of Features completed per person per month

Shown in a graph we can see this year we've not got a trend for the ratio improving much…

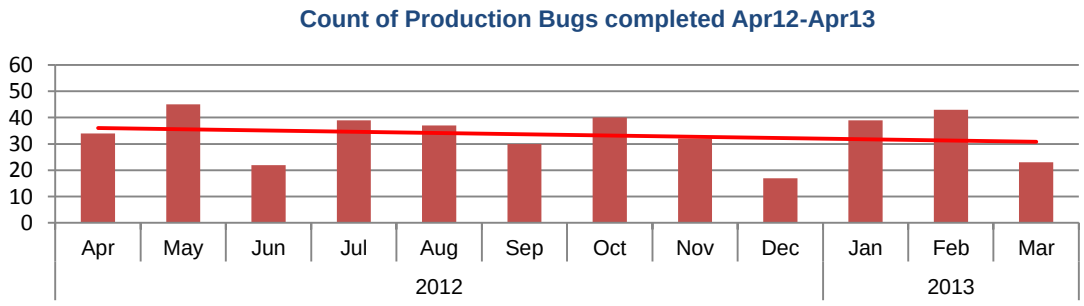


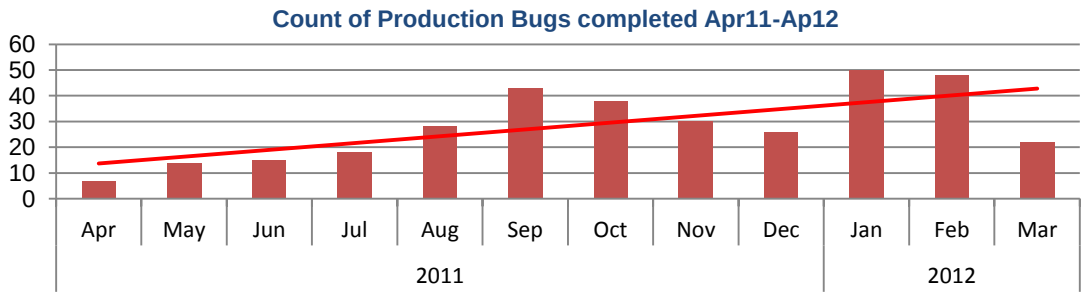
…however the previous year shows we had significant improvements, particularly at the beginning of 2012 onwards which we've managed to sustain.



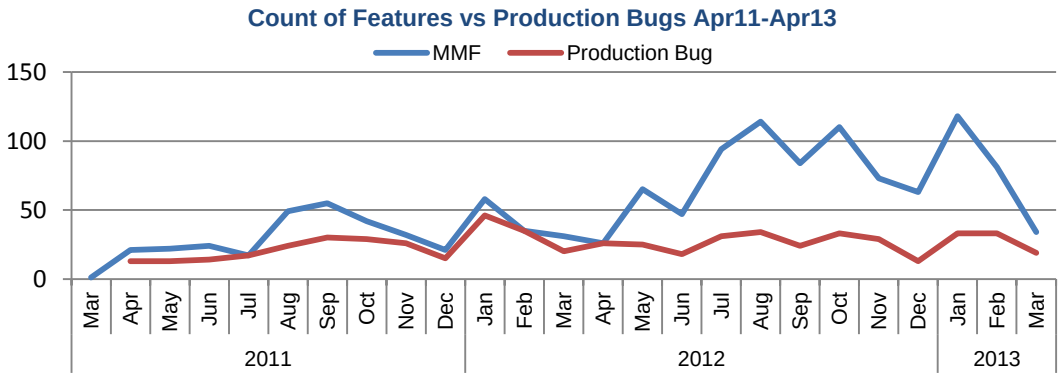
Production Bugs

In the previous report we did a lot of analysis here as we still didn't seem to be getting a handle on production bugs, with just a 6% reduction in the proportion of bugs to features. Much better news this time! From our peak in Jan 2012 there has since been a slightly positive trend downwards for the first time.





Most significantly though, as our throughput on Features has risen our ratio of bugs to features has dramatically improved. This graph for the last 2 period's shows we appear to have come out of the woods around May 2012:



In the previous period our average ratio of features to bugs was 1.37 (features for every production bug); in this period it was 2.77, **a 103% improvement**!

Long Term View

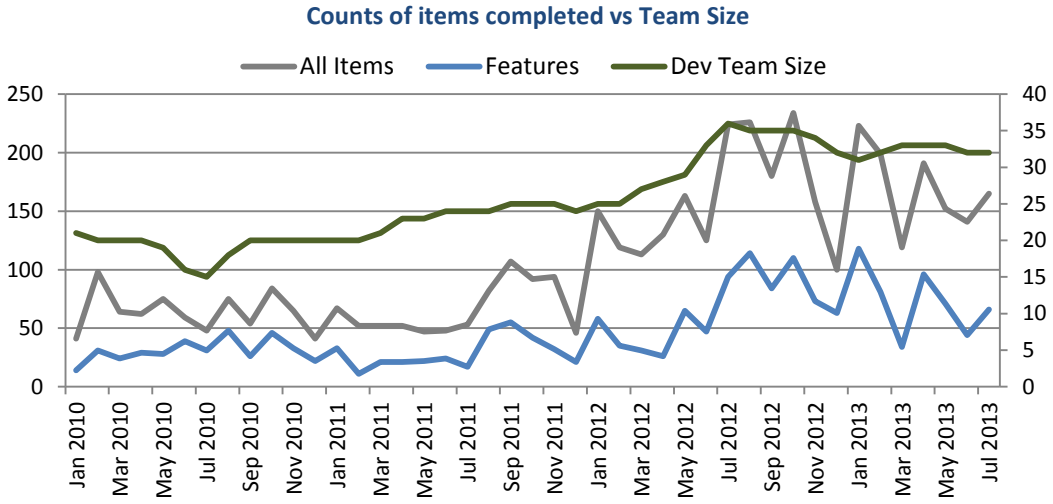
We've now been collecting metrics for around 4 years. Let's look back at some data across that entire period. It's worth noting that data from 2009 is not particularly reliable as not all teams were recording work (and when they were not recording it particularly reliably).

Generally everything suggests Jan 2012 was a key moment in our development as a team.

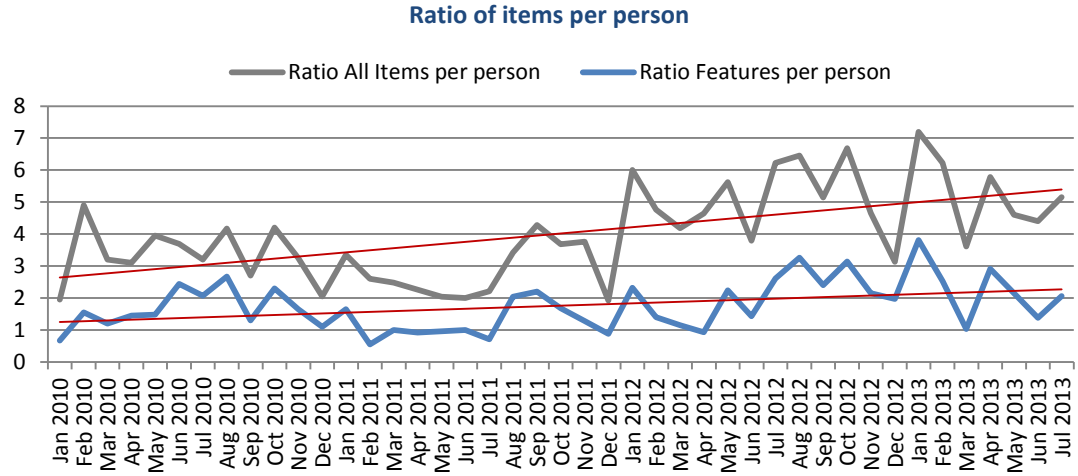
Effect of Team Size on Productivity

For the first time we've managed to collate our data on the overall team size on a monthly basis (back to Jan 2010), which allows us to compare the size of the team to the amount of work completed.

The graph below suggests we weren't getting an impact on productivity by increasing the team size until mid-late 2011.



Probably a better way to look at it is the ratio of items completed for each person in the team. We certainly seem to have gotten worse before we got better, with an all-time high of 7 items per person per month in Jan 2013.

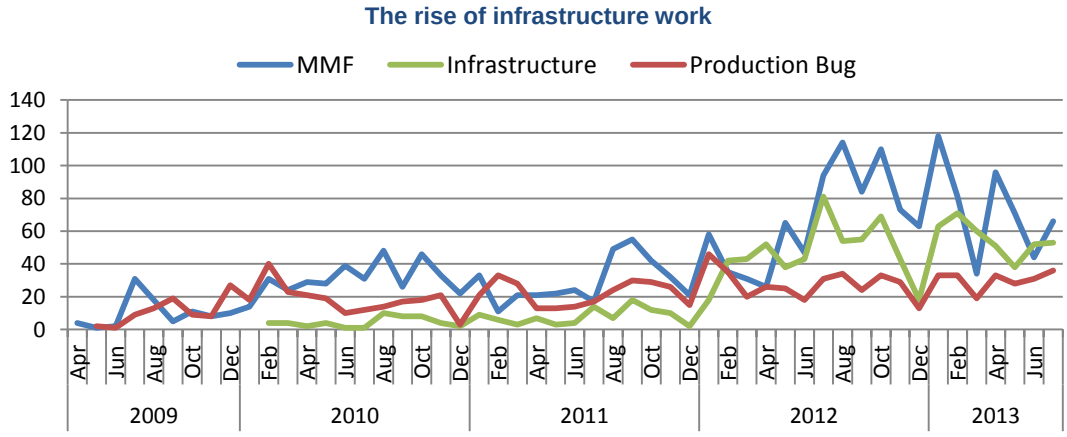


What is interesting with this view of our data is how you can see the impact of big events on our productivity.

- There's the typical seasonal drop off in December and a big spike up in Jan - we have a release freeze a couple of weeks before Xmas and most are away during the Xmas week.
- Feb-Jul 2011 is noticeably poor – this is largely applicable to a very painful period of database changes which affected all teams at that time.
- March 2013 shows a drop in productivity too – again a big DB infrastructure upgrade but also data centre moves which prevented us from making changes for significant periods.
- In June 2013 we moved offices (and also our office Data Centre which hosts our testing environments).

The rise of Infrastructure work

Noted in previous reports and very visible here is how Infrastructure work (see Work Type Definitions) has emerged:

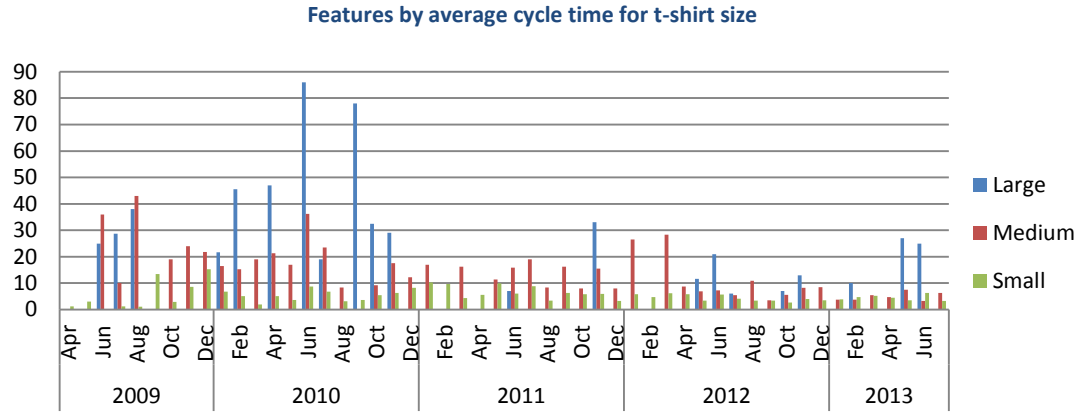


We started recording infrastructure work in Feb 2010. We see a significant increase in infrastructure work early 2012 which interestingly coincides with throughput of Features increasing and Production Bugs stabilising. A correlation here would not be a surprise as a significant proportion of infrastructure work is related to improving our automated testing and deployment pipeline.

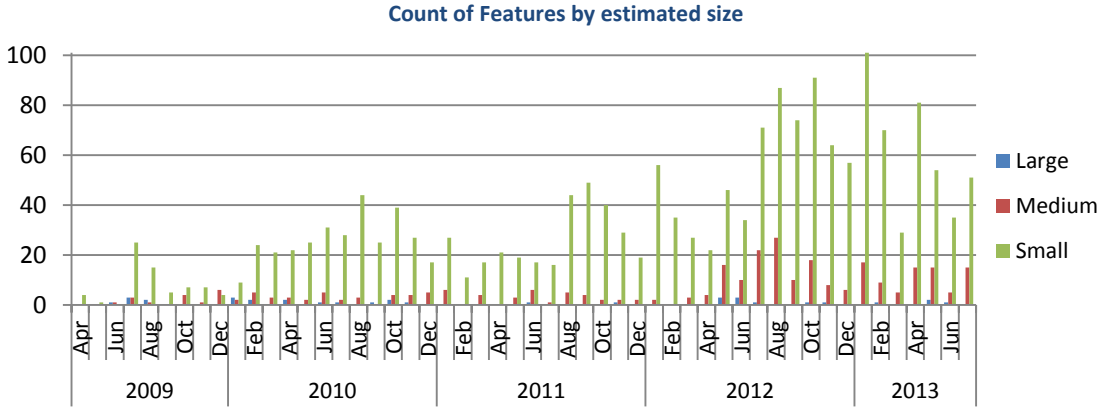
Size and Estimation

We do basic t-shirt size estimating on all Features – Small, Medium or Large. We will often use this data to help us try to project timescales on new work, but it also allows us to see how effective our estimating is and how well we're breaking up work (on the assumption that the smaller we can break things up into units of value (Minimum Marketable Features or MMFs) the more quickly we're adding value to the platform).

Below we can see the average cycle time for features by t-shirt size. In particular it shows that a large today is much smaller than a large in 2010 and about the same size as a Medium in early 2012. Interestingly it also looks like a modern Medium is not too dissimilar in actual size to a Small.



We're also doing fewer Large features. Lots and lots more Smalls, but also a recent growth in Mediums however, as above, a modern Medium estimate seems to be about the same amount of actual effort as a Small.



Conclusion

A brief summary of the last 4 years:

- Apr09-Apr11* Cycle Time improved (but not Throughput or Production Bugs)
- Apr11-Apr12 Throughput & Cycle Time improved (but not Production Bugs)
- Apr12-Apr13 All three measurements improved!

*The first productivity report collated 2 years' worth of data.

It's really pleasing to see we're finally starting to get a handle on Production Bugs and generally things continuing to improve. It's interesting to see this pattern for improvement. We haven't got any particularly good explanation for why things happened in that order and curious if other organisations have seen similar patterns or had different experiences. We'd expect it varies from organisation to organisation as the business context has a massive influence. 7digital is no different from any other organisation in that you have to be able to

balance short term needs against long term goals – there's no benefit in having a sustainable pace of work if it's at the expense of the organisation's ability to meet its financial obligations to its investors or owners. If anything else our experiences just further support the fact that real change takes time.

We must emphasis again that this data is pretty meaningless if you're not building the right thing, however we're strong believers in the concept that you've got to be able to "do it right" before you can "do the right thing", supported by the study by Shpilberg et al, Avoiding the Alignment Trap in IT.

It's also worth noting that whilst we only record data for the Development teams these improvement is down to the entire Technology team, which includes our Infrastructure and DBA teams, and also our Product Management team whose trust and support we've needed to allow us to increase the amount of Infrastructure work we've been doing, which we're positive has had a huge influence in increasing our productivity.

As always, there's still a lot of room for improvement and we should expect to see these metrics continue to improve, but maybe not at quite such significant factors.