

Achieving Agile engineering excellence through Continuous Improvement

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 enterprises with large scale, distributed Agile teams.

1. Purpose of this paper

At Plandek we get the opportunity to work with a great variety of Agile engineering teams – from very large, distributed enterprises - to compact start-ups.

During our work, we have observed two hard facts:

- All these Agile teams are on a journey towards “Agile engineering excellence” (which means different things to different people)
- All struggle to balance the pressures of the “day job” (delivering features), with the need to manage and implement a structured and long term continuous improvement (CI) programme – so that they consistently get closer to the ultimate goal of “Agile engineering excellence”.

This short paper focuses on how you can implement a demonstrably effective CI programme even in the fastest moving and most resource constrained Agile environments – so that you deliver today and continue to improve at the same time.

2. Scope of the analysis and insight

This proprietary analysis is based on core Agile principles and experience of working with clients and analysing anonymised data, feedback and insight from Agile software development teams observed between September 2017 and December 2018.

The Plandek team continually talk with and interview Agile development teams and their stakeholders whilst piloting and implementing the Plandek BI and Analytics platform. The Plandek platform is a BI dashboard and analytics tool designed to help technology teams deliver Agile projects and programmes more productively and predictably.

3. The concept of Continuous Improvement – and why it’s so powerful

The concept of CI has been around for a long time and we are all no doubt pretty familiar with it. It was first applied most successfully in a business context in Japan and became popularised with Masaaki Imai’s 1986 book “Kaizen: the Key to Japan’s Competitive Success.”

The principle is very simple and very powerful, but surprisingly hard to deliver effectively. In essence, successful CI requires five important things:

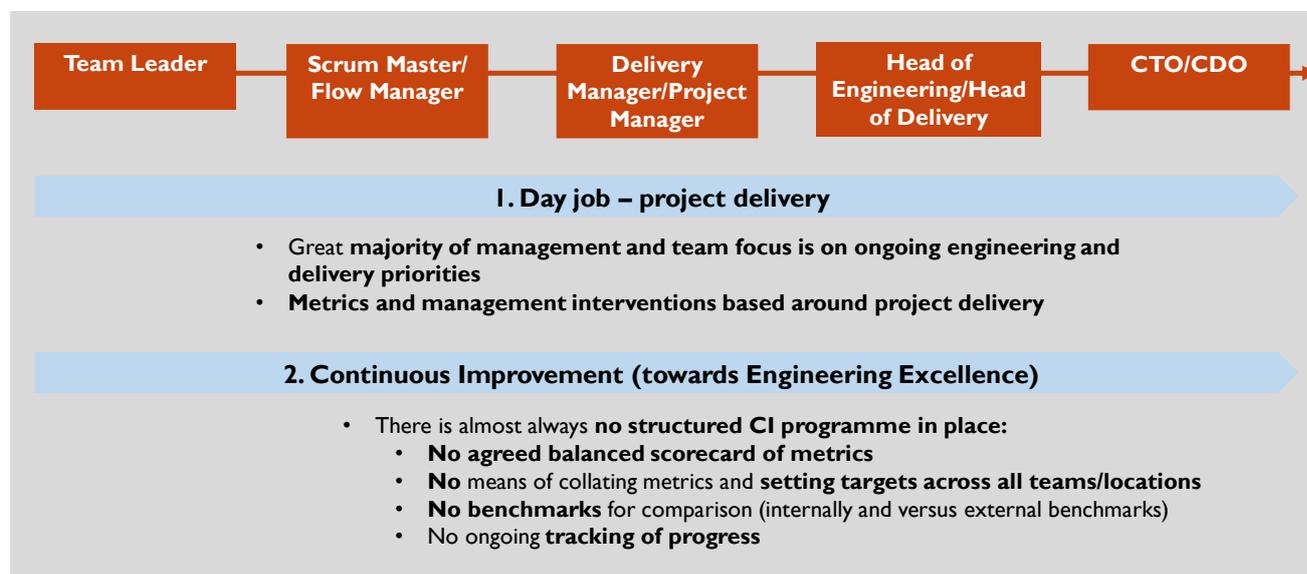
1. A serious long term commitment and sponsorship from the leadership team – as CI requires effort and resource over a prolonged period of time
2. An agreed, objective set of metrics to track progress – making sure that these metrics are actually the right ones – i.e. the metrics that are deterministic of desired outcome
3. An initial diagnostic to clarify the current situation and the scope of the functions, processes, teams etc that are to be improved
4. A means of tracking these metrics and setting targets (with targets calibrated against internal and external benchmarks)
5. An embedded process to manage the CI programme to set targets regularly, cascade targets out; track progress; make necessary interventions; celebrate success and move on.

4. The need for Continuous Improvement in Agile engineering teams

The CI philosophy is very complementary to the Agile philosophy – both share the aim of constant innovation and the recognition that the job is never done – you can always get better. Agile teams therefore tend to be very receptive to the CI philosophy.

That said, Agile teams are almost always busy and resource-constrained. As a result, the intention of always improving (in a structured and demonstrable way) often loses out to the pressures of the day job – delivering to the ceaseless demands of the business.

Typical Agile team management focus



We believe therefore that a structured, long term and well implemented CI programme is a vital addition to the great majority of Agile environments for two key reasons:

- without it, much of the inherent value of Agile is lost – as teams do not progress as hoped and large differences in competence and performance start to emerge between teams; and importantly
- it is not possible to track improvement over time – and celebrate the improvement internally and crucially demonstrate it externally to stakeholders.

Indeed, the evidence supports the logic of implementing CI – Agile engineering teams who embrace CI are demonstrably more productive and accurate in delivery timing, than those that do not.

5. Implementing an effective Continuous Improvement programme

5.1 Establishing commitment and sponsorship for CI

CI needs a serious commitment from the senior leadership team within engineering. This may not be possible if the organisation lacks the BI tools to provide the necessary metrics and reporting. Indeed, the reporting found within common tools like Jira is not designed to support a broad and effective CI programme.

The Plandek BI and analytics platform has been designed precisely with CI in mind. It provides all levels of the engineering team with a real-time and meaningful set of metrics and reporting around which a robust CI programme can be built.

A useful first step is to implement a tool like Plandek (though we don't know of any other tools that do the job as effectively!) – and set the key enabler in place to drive CI.

5.2 An agreed set of metrics around which to build the CI programme

By its very nature, Agile encourages a myriad of different methodologies and measures of success. These often vary from team to team. Whilst this is healthy, we believe that the senior leadership team needs a set of agreed Agility and effectiveness metrics, that are proven to be deterministic of better Agile outcomes – and which can be tracked and compared across teams.

At this point it is worth explaining what Plandek is/does.

Plandek is a specialist BI and analytics platform designed to optimise the software development process. It works by mining the data history from development teams' toolsets (e.g. Jira, Git) to reveal, track and manage levers (right down to the team and individual level), that are highly predictive of project productivity and timing - in order to significantly improve Agile project outcomes. These metrics are often not visible or accurate with current toolsets.

The deterministic metrics presented in real-time in the Plandek dashboard are designed to form the basis of a broad CI programme to steadily improve Agile engineering effectiveness.

The metrics are grouped into six key areas. These are:

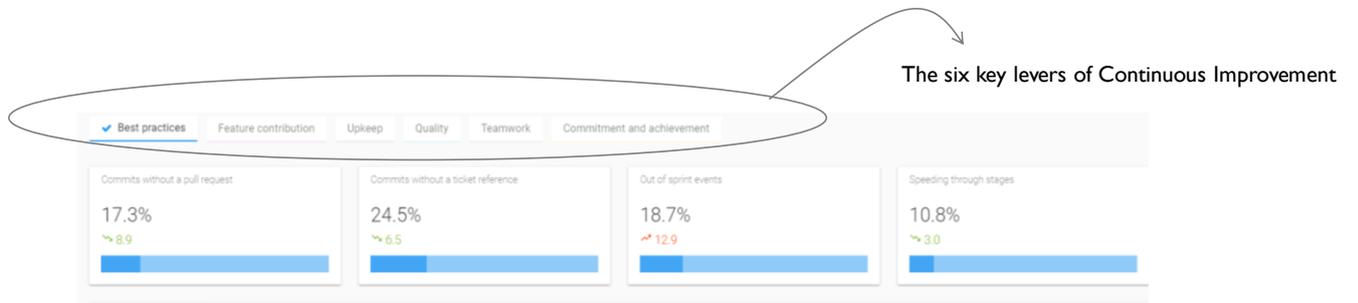
1. The critical enabler – best practice Agile process and tool use

..and the key practices...

2. Sprint disciplines and consistent delivery of sprint goals (Scrum Agile)
3. Proportion of time spent/efficiency of writing new features (productive coding)
4. QA failure rate and therefore...
5. Proportion of time spent/efficiency of bug fixing and re-working returns from QA
6. Teamwork and the ability to collaborate effectively.

Our view (supported by the empirical evidence) is that an engineering team CI programme should be built around these metrics – though may include many others (as Plandek does not yet for example cover release metrics within the broader DevOps cycle).

Example Plandek visual showing the six key levers around which an Agile engineering CI programme is built



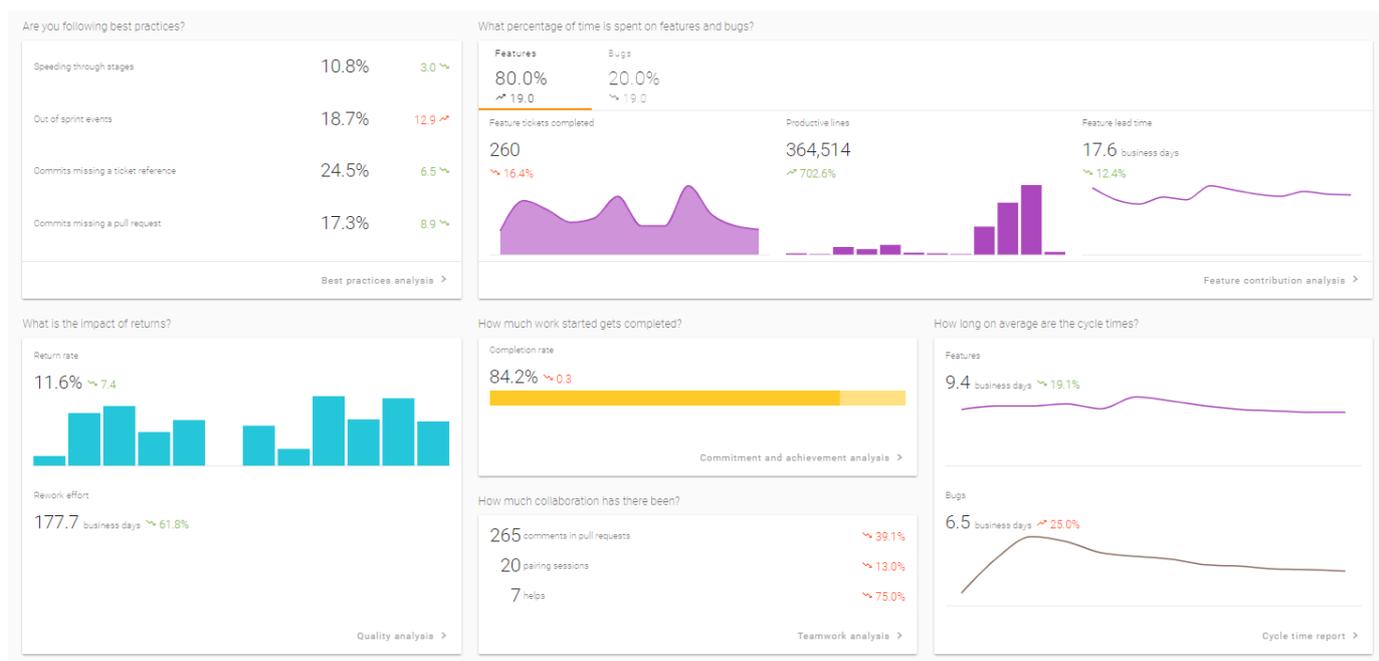
For a more detailed account of these metrics, please refer to the Plandek Whitepaper “The Six Key Drivers of Project Productivity and Predictability in Agile Environments”.

5.4 Diagnostic to clarify the current situation and scope for improvement

Having agreed a set of metrics that are meaningful (and genuinely deterministic of success), the CI programme can be started with an upfront analysis of the current situation across all teams, locations etc.

This shows the size of the prize/scope of the challenge; reveals deltas between programmes, projects, teams and locations and allows the senior leadership team to prioritise the initial focus of the CI process.

Example Plandek visual showing the typical elements of an initial CI diagnostic



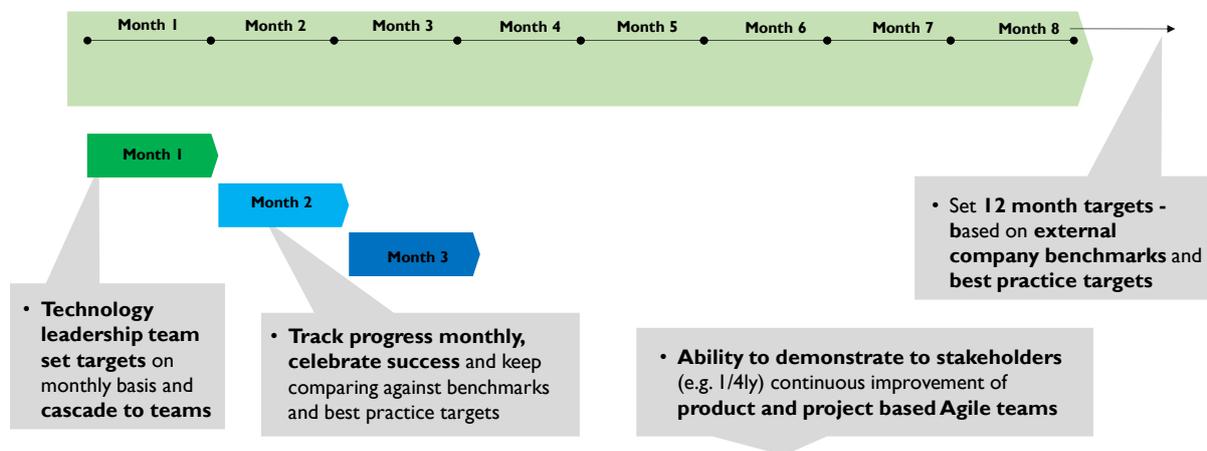
Very often the diagnostic reveals that the first priority for the CI programme should be a group of “best practice” metrics which when absent:

- prevent any meaningful view across teams (e.g. due to poor/inconsistent use of workflow management tools like Jira)
- create risk (e.g. infosec risk).

5.5 Running the CI programme – target setting, measuring progress, celebrating success

The CI programme is typically established on a monthly cycle of target setting; team implementation; review and celebration of success. The approach is represented graphically below.

Typical CI framework based around monthly CI cycles



The effort required becomes overwhelming with manual reporting tools (e.g. metrics collected across teams using spreadsheets). A real-time BI tool is the essential enabler. Priorities are set by the leadership team and targets cascaded down the organisation monthly. Targets can be specific to projects and teams, depending on observed strengths and weaknesses.

CI is designed to be a positive and motivating process – and it is vital that it is perceived as such. Celebrating success is therefore a key element of the process.

Use of team awards in Agile CI

Engineering CI - Team Awards



- **Monthly team prizes**
 - Location Winners, Global Winners
 - Location Most Improved, Global Most Improved
- **Recognition of Centres Of Excellence/Global Leaders** for certain competences to engage in championing the CI process
- **Annual awards ceremony/recognition**

5.6 Use of internal and external benchmarks in target setting

A key question arises around the setting of targets. Benchmarks are very useful in this context. We find that three key benchmarks are most useful:

1. Internal benchmarks (often set by centres of excellence within the organisation)
2. External competitor/comparator benchmarks – Plandek provides anonymised benchmarks across all metrics from similar organisations
3. Agile best-practice benchmarks - these are often very hard to attain but are obvious targets as the CI programme develops.

Plandek enables the CI programme director to view progress in real-time against these benchmarks. And then to look back over the duration of the programme to view the rate of improvement.

Example of a CI metric benchmarking approach

 Plandek	Metric	Actual	Comparative External Benchmark	Best Practice/Target	 Plandek	Metric	Actual	Comparative External Benchmark	Best Practice/Target	
Process Visibility & Infosec RISK	Missing Jira ticket reference in code repo	10%	15%	1%	Delivery TIMING ACCURACY	Out of sprint activity %	12%	15%	<5%	
	Speeding tickets (%)	30%	33%	1%		Sprint completion %	41%	45%	>90%	
	Commits missing pull request (%)	14%	18%	2%	Team & People effectiveness		Return rate %	41%	25%	10%
Process EFFICIENCY	Time spent on New Features (new code) (%)	75%	76%	80%		% returns from worst performing 20%	82%	80%	75%	
	Time spent bug fixing %	25%	24%	20%		Collaboration	Helping out (Ave per person per sprint)	0.2	0.8	2
	Flow efficiency (In progress % of total cycle time)	8%	7%	30%			Stakeholder collaboration rating (1-5) (1= poor)	2	3	4
	Rework Effort (Work Days) per month	5500	3500	2500						

6. The impact of an effective Continuous Improvement programme in Agile engineering teams

Agile engineering CI programmes harness the real power of the Agile philosophy and hence perform significantly better than Agile teams simply focused on their immediate delivery priorities.

There is clearly no overall measure of Agile engineering effectiveness, but well implemented CI programmes should deliver significant improvement in metrics that underpin productivity and timing accuracy – for example:

- 10-15% improvements in flow efficiency
- 15-20% reduction in return rates and time spent reworking tickets (returned from QA)
- 30-40% improvement in sprint completion accuracy (Scrum Agile)
- Greatly improved team collaboration and team happiness.

These improvements should be achievable over 12-18 months.

Introduction to Plandek

Plandek (www.plandek.com) is a BI and Analytics platform (and related continuous improvement methodology) to help large IT teams deliver Agile technology projects more productively and predictably. Plandek works closely with IT leadership faced with the challenge of managing complex Agile IT teams and delivering against the ever-increasing expectations of the business.

Plandek works by mining the data history from dev teams' toolsets (e.g. Jira, Git) to reveal and track levers (right down to the team and individual level), that are highly predictive of project velocity and timing - in order to significantly improve Agile project outcomes. These metrics are often not visible or accurate with current toolsets. Plandek is working with a wide range of organisations, to track and actively manage these metrics, to improve performance.

Plandek is a practical management tool that is used at all levels within the IT team (team leader to CTO) to:

- manage project delivery more efficiently (the day job); and
- drive and measure progress in clients' ongoing Agile Continuous Improvement programmes (the longer-term challenge).

If this Quick Reference White Paper has been helpful, feel free to contact the authors at cponsonby@plandek.com or nadia.tosheva@plandek.com, or the broader Plandek team via www.plandek.com.