

Novel Approach to Wireless Communications Software Development Performance and Productivity Measurement – SAFe Based

(Proceedings of the Int. Conference on “Microwave and THz Technologies and Wireless comm.”)

F. Muhammad¹, C. Cilliers¹

¹ VIAVI Solutions, Wireless, Longacres House, Six Hills Way, Stevenage, SG1 2AN, United Kingdom

Received 15 November 2018

Abstract: This paper gives an overview of a method that can be used to measure the productivity of software/hardware development teams. The approach is based on currently utilised Key Performance Indicator (KPI) in a product development programme as part of VIAVI's Wireless R&D organisation. The method with all associated processes and their applications are presented here. The outcome of the KPI is used to further improve the productivity of the team and ensure adequate Return on Investment (ROI) as well as to initiate Root Cause Analysis (RCA), where necessary. With this KPI, performance benchmarks can be set for Individual Cross Functional Teams (XFT) and the overall Agile Release Train(s) (ART) where productivity can be monitored over any time granularity. Other secondary benefits of this KPI is to facilitate management decisions which include; but is not restricted to; which site to use for R&D assignments, where to scale up/down, what work allocation to which team, true cost comparison between different geographical locations etc.

Keywords: Key performance indicator, wireless communication.

1. Introduction

Most R&D teams adopt some sort of KPIs as part of their development management whether Agile or Waterfall. These KPIs provide the necessary data which can be used to make informed decisions that positively impact engineering departments as well as the business as a whole.

The literature [2] suggests scores of different engineering KPIs but the actual value of their use depends on how a particular KPI is implemented and used in an R&D organisation. Collectively these KPIs complement each other to give management the full team performance picture across all teams in all locations in order to support overall business objectives.

Successful R&D teams adopt development processes, associated KPIs and analyses that are fit for purpose to support their business objectives. Analysing results help improve performance, identify and mitigate risks and issues, plan programmes and business evaluation.

Furthermore, these KPIs can improve resource allocation, planning, monitor progress, decision making and improves cost forecasting and gives a true productivity comparison between different teams in different locations operating on different cost bases.

This paper presents an overview of the Productivity Measurement process in software development on a particular product development. Associated KPIs and processes will also be discussed.

Introduction

Most R&D teams adopt some sort of KPIs as part of their development management whether Agile or Waterfall. These KPIs provide the necessary data which can be used to make informed decisions that positively impact engineering departments as well as the business as a whole.

The literature [2] suggests scores of different engineering KPIs but the actual value of their use depends on how a particular KPI is implemented and used in an R&D organisation. Collectively these KPIs complement each other to give management the full team performance picture across all teams in all locations in order to support overall business objectives.

Successful R&D teams adopt development processes, associated KPIs and analyses that are fit for purpose to support their business objectives. Analysing results help improve performance, identify and mitigate risks and issues, plan programmes and business evaluation.

Furthermore, these KPIs can improve resource allocation, planning, monitor progress, decision making and improves cost forecasting and gives a true productivity comparison

between different teams in different locations operating on different cost bases.

This paper presents an overview of the Productivity Measurement process in software development on a particular product development. Associated KPIs and processes will also be discussed.

2. Overall development flow

Most R&D organisations implement conventional homogeneous workflows based on a unified organisation with a single Way of Working (WoW). This is then with its supportive processes are applied to all areas of the organisation at all stages of product development. In this paper we are presenting a progressive method specifically applied to organisations where Time to Market (TTM) is critical, good quality on legacy and high stability is maintained throughout. These market requirements seem to be conflicting if approached by a single product development process, hence TTM-influenced development process initially and Time-In-Market TIM-influenced development process later.

With the above in mind we have devised a three-stage development process; Proof-of-Concept (PoC) stage, Rapid-Development (RD) stage and Product-Development (PD). These three stages will demand their own WoW, processes, team structures and supporting KPIs. In the following sections we will briefly describe the last two stages with emphasis on the product development stage. The productivity KPI is most applicable to the third stage. As illustrated in Figure (1) after the initial business and technical feasibility is carried out, the development method is decided according to its TTM criticality.

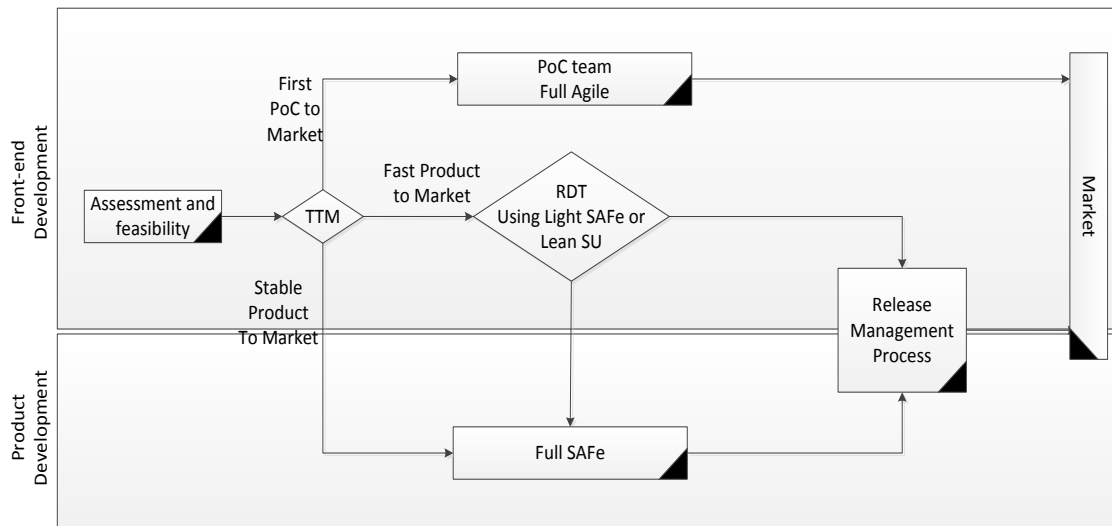


Figure (1)

a) PoC Stage:

At this stage, the team delivers any useful shippable software that helps the client progress their development. This team performs best in full Agile/Kanban, smallest planning and delivery increments, lightest processes, and most-relaxed quality gates approach. Nevertheless, release traceability and governance need to be maintained. Forward reusability is preferable but not mandatory to ensure TTM is met.

b) *RD Stage:*

At this stage, the foundations of the product are established. Features are developed in accordance with the Minimum Viable as TTM is still a major factor. Continuous Integration (CI) processes are used and short-lived temporary release branches are allowed when absolutely necessary. These teams work in an Agile Framework, e.g. customised SAFe, where faster cadence is required, in addition to shorter planning intervals, lighter processes, and relaxed quality gated for new features as new feature stability is still not a factor. Legacy feature quality must always be maintained. Co-located teams are necessary and closer collaboration with the support teams is vital.

c) *PD:*

This is the final development stage where prolonging TIM is key, which builds on the RD foundation and any initial prototypes developed. This stage is responsible for developing a resilient, scalable, stable and long-lasting product hence must follow a strict New Product Development (NPD) and quality control process. For this SAFe, CI and full quality gating are adhered to.

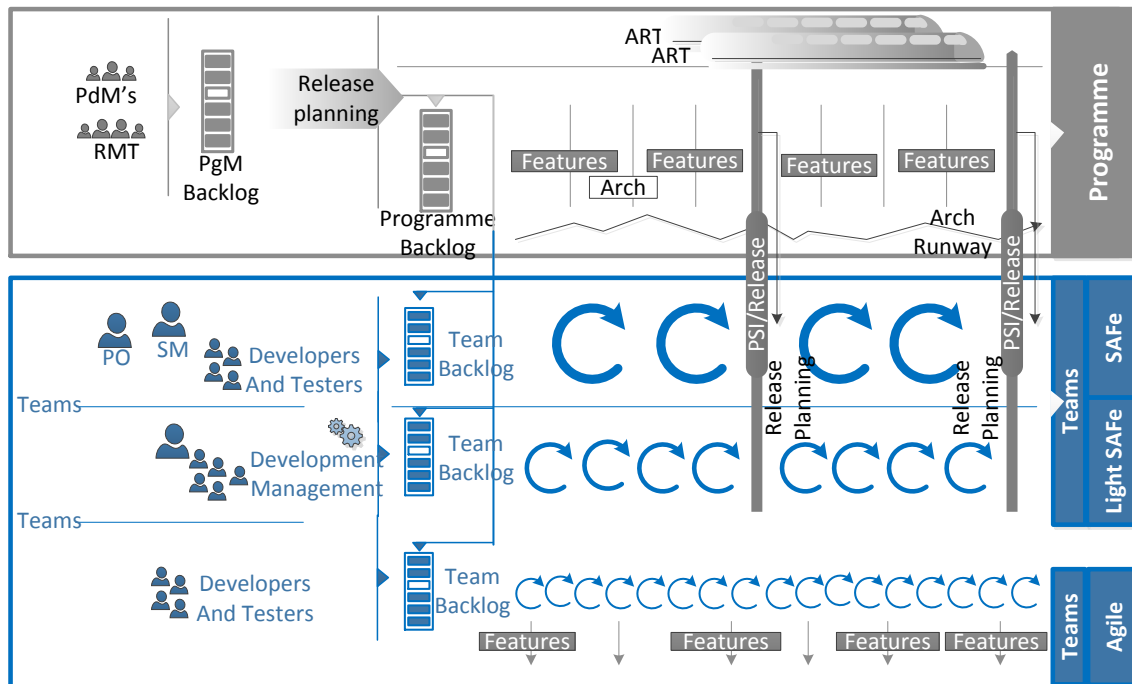


Figure (2)

3. Hybrid approach

Due to the aforementioned TTM and TIM reasons, different approaches to development are required at different stages. Hence it is recommended to adopt a hybrid Waterfall/Agile approach. At the initial stage of feature assessment, the development approach is decided, i.e. full Agile, Lean Start-up, or full SAFe, see Figure (2). Support and dependency teams, e.g. VHDL, Hardware, Platform, Quality Control, Algorithms teams, etc are common and shared across all development stages, i.e. PoC, RD and PD.

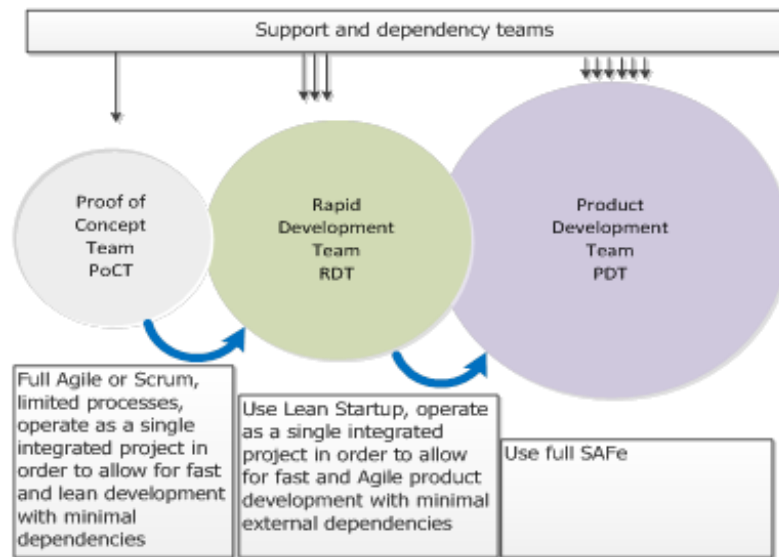


Figure (3)

4. Safe and capacity allocations

Team capacity allocation to plannable, non-plannable and interruptions depends on the type of product development. Typical allocation is as illustrated in the below Figure (4).

It is relatively easy to allocate capacity to plannable items, i.e. most of the 70% which is consumed with software development, design and code reviews and some plannable low priority defects. The 10% IP Sprint is taken up by technical refinement, upcoming PI planning, tasks that end up taking longer than first estimated, tasks that are looked over or missed during the PI planning event, and finally any time spent on new ideas and innovation. Teams often, experience a challenge in allocating capacity to the interruptive work and high priority defects. As such, 20% is typically reserved for these types of unpredictable tasks. These may include other tasks as illustrated in Figure (4).

5. How to deal with interruptions

There are many ways to deal with interruptions;

- a) *Stick to the Scrum Rules, the rules are clear, if its not part of the Sprint plan, then it shouldn't be done. This is not recommended as it is too strict and not interruption-friendly.*
- b) *Short Iterations which is shorter than interruption frequency. In this method, choose your iteration length to be so short that you can always start work on urgent interruptions. This is not recommended as well as it can be exhausting, but it is one ways to get the team and the organization to understand the large toll that these interruptions take.*
- c) *Handle interruptions on case-by-case bases, which requires constant change and crisis management and is highly unpredictable, [1].*

It is recommended to allocate a portion of time to interruptions based on Commitment Velocity where possible. After each sprint, consider how well the unplanned time allocation against that needed ended up for the sprint and adjust. This is something the XFT will improve

over time. Instead, it's a game of averages. The team needs to save the right amount of time for unplanned tasks on average. Some sprints will have more unplanned tasks occurring and some sprints will have fewer. When fewer occur, the team should get ahead on their stretched work, so that they're better prepared for when unplanned tasks occur further down the line.

The preceding advice works well for most SAFe teams where the level of interrupts is moderate. Some teams, however, are highly interrupted and therefore should have a different WoW, e.g. KANBAN approach. Hence another good approach is to create a dedicated team to deal with interruptions.

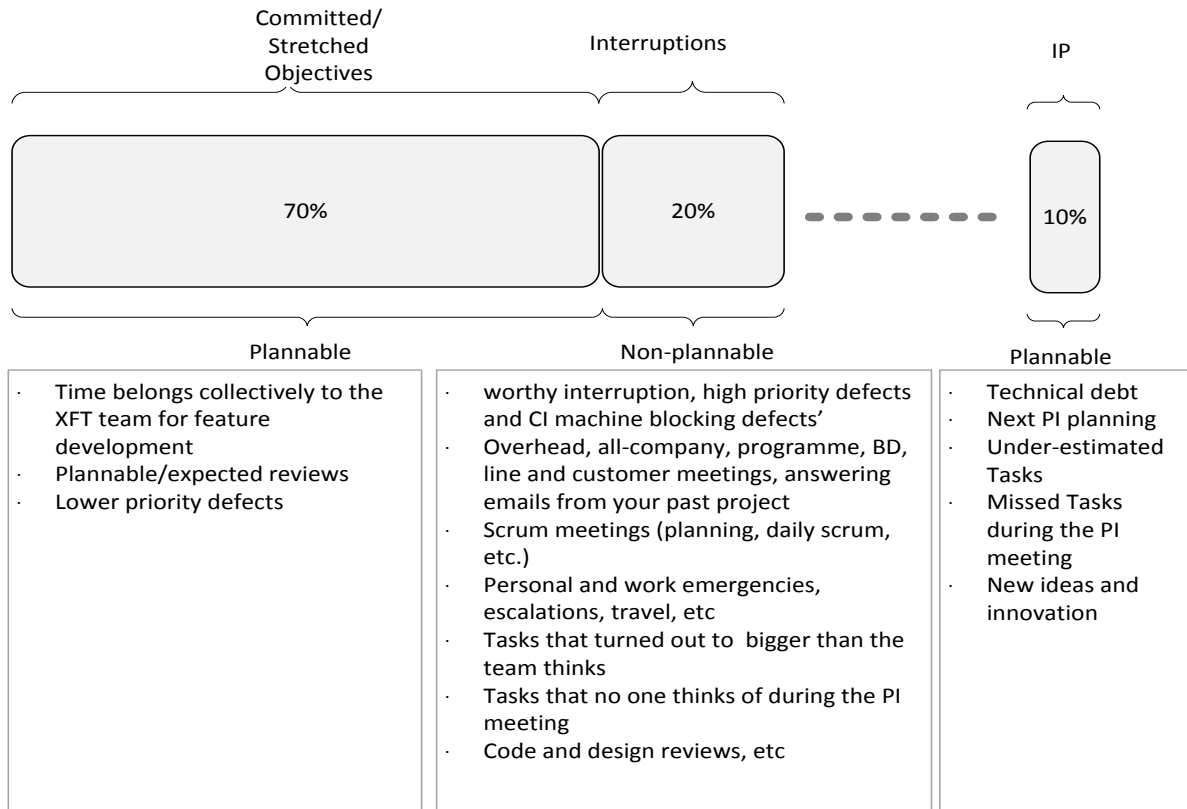


Figure (4)

In conclusion, no magic percentage exists but must be larger for the area of 'unplanned time' in Figure (4). These highly-interrupted SAFe teams still need to include space in their sprints for unplanned time.

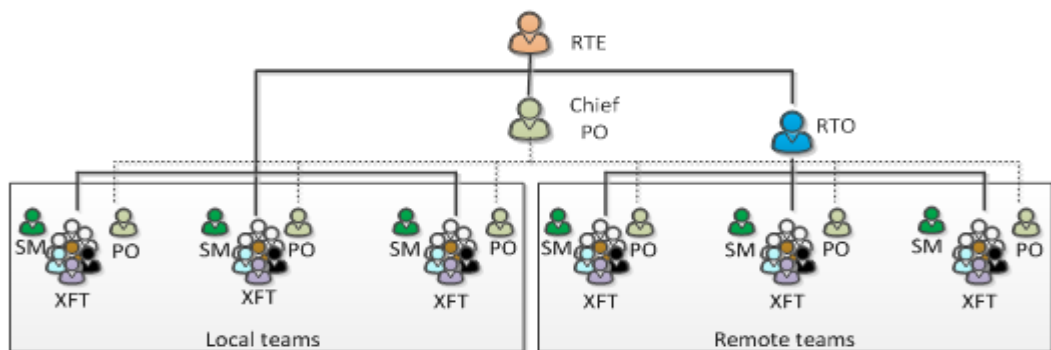


Figure (5)

6. Team structure

The below simplified Figure (5) illustrates a team structure for a single train consisting of local teams as well as remote teams. Each team consists of Agile team members, Scrum Master (SM) and Product Owner (PO). These teams interface directly to the Release Train Engineer (RTE). In case of the remote teams they interface to the RTE via Remote Team Owner (RTO) who is local to the RTE. Teams' POs are co-located with their teams; a single PO can be associated with more than one XFT and all POs are technically co-ordinated by the Chief PO although they are accountable to the RTE.

Each ART (or multiple ARTs) are supported by a number of Dependency Teams, which includes but is not limited to; Defect Triage Team, Architecture Team, Quality validation team, System Validation Team, VHDL team, Algorithms Team, HW Team etc. as illustrated in Figure (6).

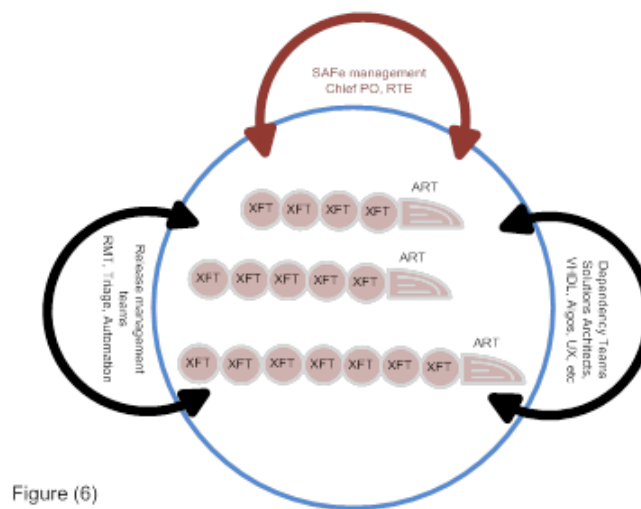


Figure (6)

7. Team and Art productivity evaluation

The performance of the team is measured using productivity, a novel approach which has been devised and implemented. This is a relative measurement (unit-less) and is based on a common reference and comparable currency, i.e. Complexity Value (CV), as the CV is related to the estimated feature development effort. It became apparent that just counting the number of features/work items delivered by a team is not a good measure, as features vary in size.

This KPI is used to measure productivity rather than efficiency. Efficiency and Productivity are often confused, yet the two have vastly different meanings when it comes to getting your work done. Being efficient, means you are working in a well-organised environment (work flow) while being productive means you are successful in producing the desired results. More on the definition of efficiency, productivity and effectiveness in a later section.

The starting point is to estimate the size of a feature by using a reference, i.e. the already known size of an implemented feature. The Modified Fibonacci sequence is used to determine the relative size of a feature as recommended by SAFe. Feature size is not connected to any specific unit of measurement. The size (effort) of each story is estimated relative to the smallest story, which is arbitrarily assigned a size of 1. SAFe applies the modified Fibonacci sequence (1, 2, 3, 5, 8, 13, 20, 40, 100) to reflect the inherent uncertainty in estimations, especially large numbers (e.g. 20, 40, 100, etc.).



Figure (7)

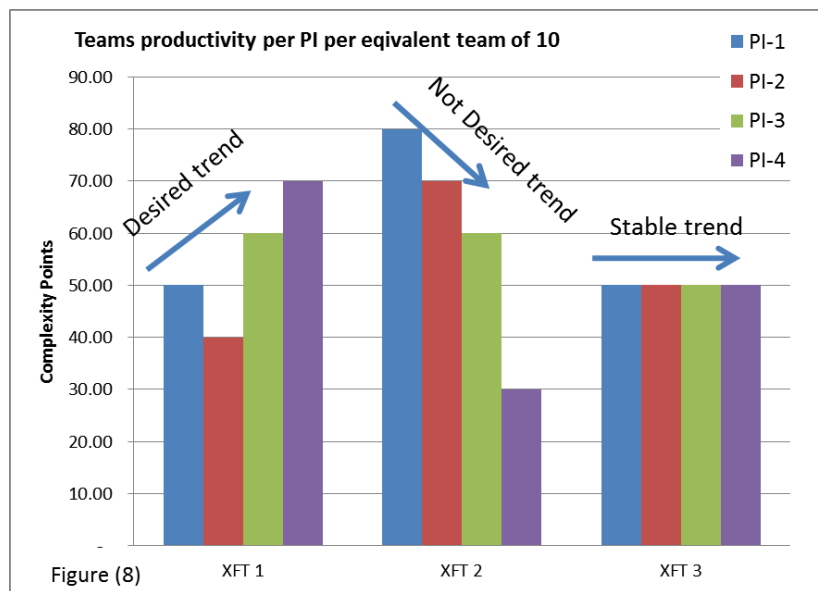


Figure (8)

Feature size to Effort Unit (can be days, weeks, etc) is based on past experience. A practical example can be found below.

For example, according to the below mapping, if a known feature size 1 is a two-month effort, then a size 10 feature is 20-week effort.

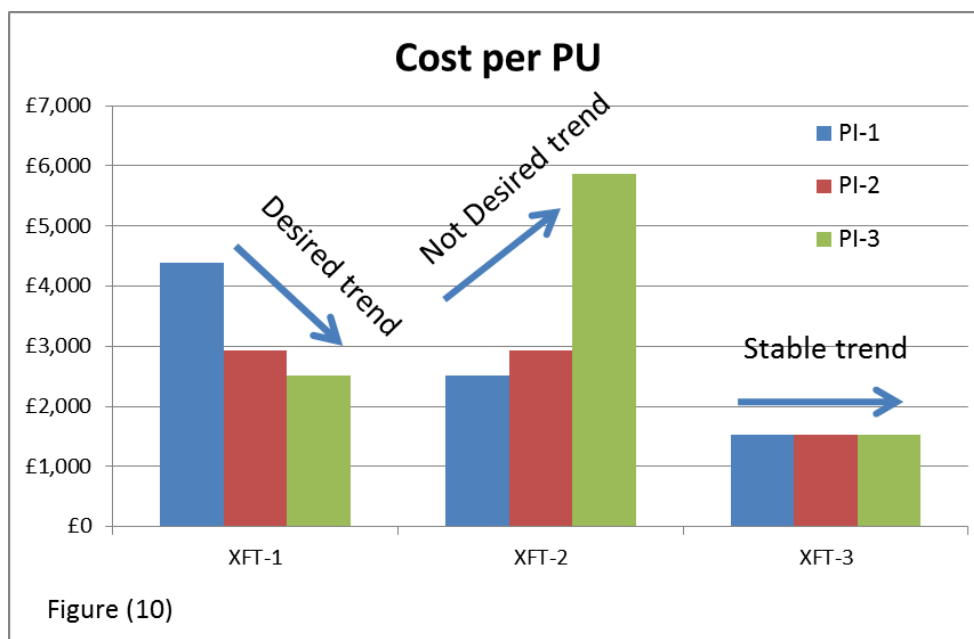
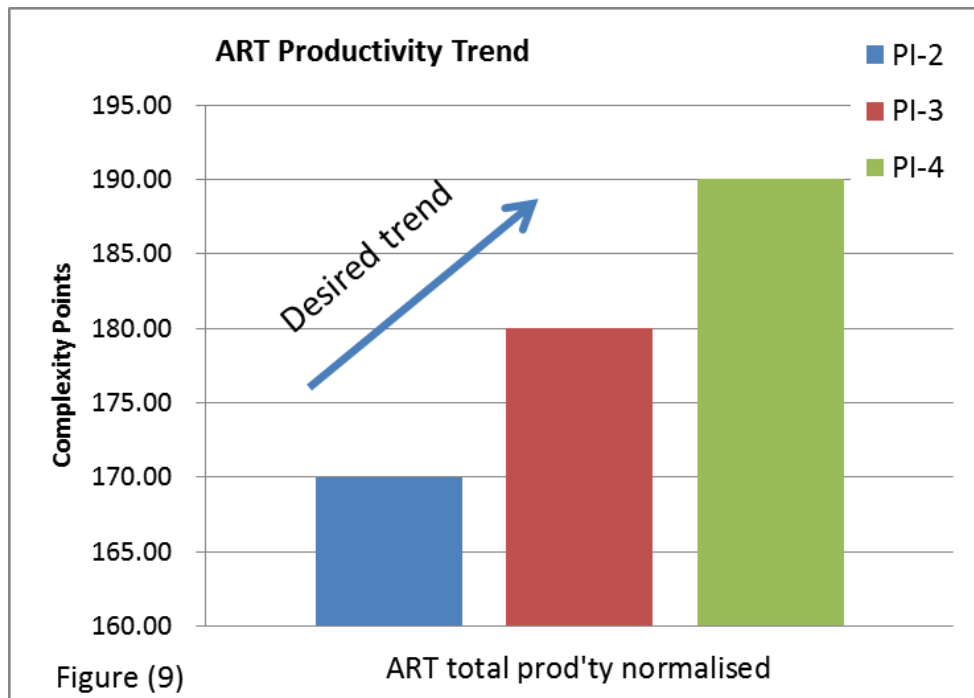
- Size 0 = negligible
- Size 1 = 2 effort units
- Size 2 = 4 effort units
- Size 3 = 6 effort units
- Size 5 = 10 effort units
- Size 8 = 16 effort units
- Size 13 = 26 effort units
- Size 20 = 40 effort units
- Size 40 = 80 effort units
- Size 100 = 200 effort units

For the Productivity KPI to be meaningful and in order to use it to compare different teams, estimates (feature sizing) have to be centralised and independent of the XFT, i.e. done by same person or same forum. Also the delivered Complexity Points by a team have to be normalised to a constant team size and fixed time-period, e.g. an XFT size of 10 members and PI of 10 weeks. Normalised to a team of 10, the total delivered Complexity Units for a team is;

$$CP = \frac{TCP}{TS} \times 10$$

where CP is the total normalised delivered Complexity Points by a team, TCP is the true Team Complexity Points delivered, and TS is the true Team Size. This can then be used to plot the total output per team over a given period (PI in this case), see Figure (8). This is agnostic of location, cost base or supplier.

Summed and normalised (to the total number of teams), the delivered Complexity Points gives you the overall ART productivity trend as illustrated in Figure (9). This can be used to measure the productivity increase.



8. Team and Art ROI evaluation

It is important is to determine the returned value on your investment (ROI). It is calculated below and example trends are illustrated in Figure (10).

$$CpCp = \frac{CpPpEU \times PIP \times TS}{CP}$$

Where $CpCP$ is Cost per Complexity Points, $CpPpEU$ Cost per Person per Effort Unit and PIP are the Programme Interval Period.

This KPI can be common to different teams' cost, e.g. Local or Offshore, permanent or contractor. It can also be measured over any period, i.e. week, month, PI, etc. This allows us to measure ROI taking into account different cost bases in different geographical locations.

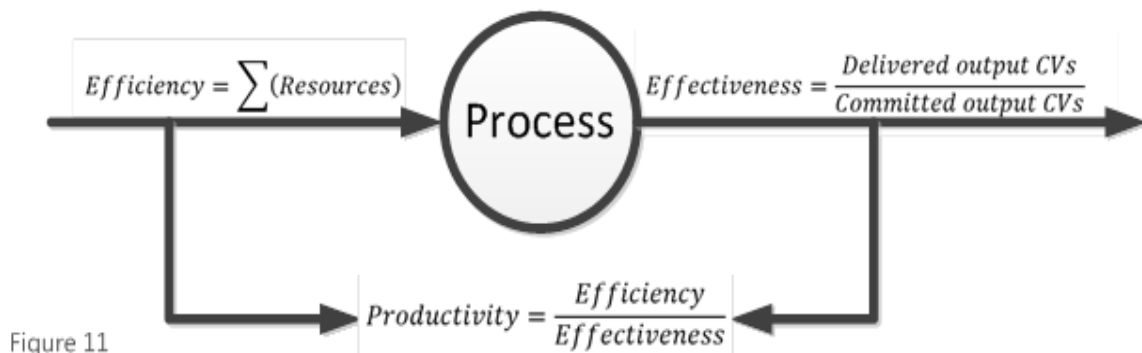
9. The relationship between efficiency, effectiveness and production

Often, we demand improvement by focusing exclusively on efficiency. There a limit to how much improvement you can make at the input (efficiency) side. Does it still make sense to try to reduce the team or the supplier cost further? By managing solely by keeping the cost price down, you may run a large risk of saving pennies on the expense of quality. We look less often at the output side – the effectiveness – of the team [3].

Efficiency is determined by the number of resources (time, money, and effort) that are necessary to produce the committed CVs. To meet the committed objectives, we commit a specific resource. For example, if we can meet the committed CVs with less resources we have operated more efficiently.

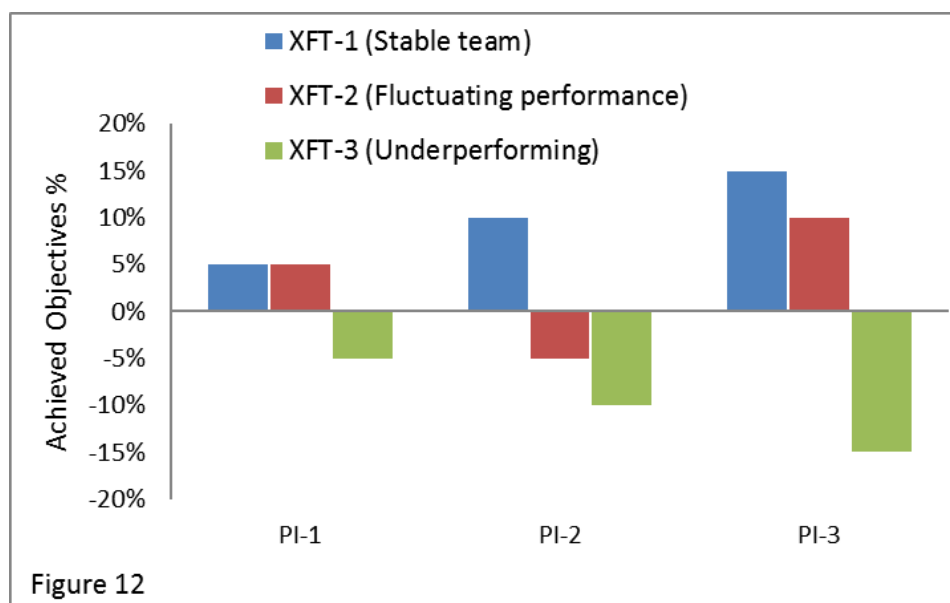
Productivity is determined by looking at the number of output CVs (effectiveness) versus the invested resources in order to achieve the output CVs (efficiency); in other words, if we can achieve more CVs with the same or less resources, productivity increases.

Effectiveness is determined by comparing what a team can produce with what they actually produce; therefore, effectiveness does not tell anything about the efficiency – the amount of resources that have to be committed to obtain that output CVs. If we are successful in producing more CVs in the same time period with the same resources, effectiveness will increase. See Figure 11 for understanding the relationship between the three parameters.



10. Team effectiveness

Team ‘effectiveness’ in other word is the actual achievement vs the theoretical commitment. During the PI event each team commits to “Committed” and “Stretch” Objectives, this KPI measures the “Effectiveness” of each team. Some teams deliver more than what is committed and some underachieve, as we can see in the example in Figure (12). This example presents the percentage of missed objectives (from committed objectives) that could not be delivered at the end of that PI, to the total number of objectives (in CP units). This gives an indicator of productivity, estimation accuracy, quality of deliverables etc. As a result, this may trigger an RCA.



11. Conclusion

There are many methods to govern and manage software development, we found SAFe as the most appropriate for VIAVI’s needs. R&D teams need to select and adopt development processes, associated KPIs and analyses that are fit for purpose to support their business objectives. Results help improve performance, identify and mitigate risks and issues, plan programmes and business evaluation. In this paper we presented a progressive method specifically applied to organisations where Time to Market (TTM) is critical, good quality on legacy and high stability is maintained throughout. We have devised a three-stage development process; Proof-of-Concept (PoC) stage, Rapid-Development (RD) stage and Product-Development (PD). Due to the TTM and TIM reasons, we recommend to adopt a hybrid Waterfall/Agile approach.

The above development approach is supported by a novel productivity KPI and other KPIs mentioned in this paper that can be used to measure the productivity of software and/or hardware development team.

Also, we concluded that there is no magic allocation percentage to cover for interruptions but has to be based on previous experience. This works for most SAFe teams where the level of interruptions is moderate, and if include sufficient space in their sprints for unplanned time. Some teams however, are highly interrupted and therefore should have a different WoW.

References

- [1] Agileadvicecom. 2012. Agile Advice. [Online]. [2 September 2018]. Available from: <http://www.agileadvice.com/2012/01/08/scrumxplean/seven-options-for-handling-interruptions-in-scrum-and-other-agile-methods/>
- [2] Pandian, C.R. 2015. Software Quality Metrics Overview. In: Pandian, C.R & Kumar, M eds. Simple Statistical Methods for Software Engineering: Data and Patterns. : CRC Press, pp. 85-126
- [3] Oeecoachcom. 2012. OEE Coach. [Online]. [4 September 2018]. Available from: <http://oeecoach.com/efficiency-effectiveness-productivity/>