

More grip, less guesswork in agile IT projects

Quantifying agile software development leads to better results

In a nutshell

Many managers lack a grip on agile software development projects in their companies. Effectiveness and efficiency in these projects leave much to be desired, because they are managed on the basis of uncertainties and unfounded assumptions.

Quantification offers a solution, even in agile projects in which not all the specifications have been established. Four key performance indicators (KPIs) and about 35 to 40 underlying metrics¹ will help you to estimate and steer IT projects better.

Most metrics can be determined objectively based on ISO standards. By establishing these before commencing, you prevent miscalculations and you determine the right supplier. You also use the metrics for evaluating the progress of the project. Moreover, you can deploy it to give the supplier more responsibility in exchange for sharing in the profits. This increases productivity and saves you a lot of money each year. Onshoring development will save 10% to 20% of your development costs; transferring your onshore development to offshore development will save IT costs even by a factor two.

¹ A software metric is an objective quantity for measuring software characteristics such as the number of Function Points, the number of Lines of Code (LOC), Mean Time To Repair/Recover (MTTR) or Mean Time Between Failures (MTBF)

Table of Contents

In a nutshell	2
1. Help, my scrum team consists of handymen!	4
2. Guesswork	4
3. Quantification.....	5
Not only for waterfall projects... ..	5
... but also suitable for agile projects	6
4. The correct values	7
Four core KPIs.....	7
At least 35 metrics.....	7
<i>Highlighted: ISO standard</i>	8
<i>Highlighted: A fixed price per function point, is that possible?</i>	9
Highlighted: Story points versus function points	10
5. Real-time insight.....	11
<i>Checking whether you are deviating from agreements during each sprint</i>	11
Dashboard	11
6. Implementation.....	12
Continuous fact-based delivery.....	12
Continuous output-based delivery.....	12
7. Involve the supplier in setting up the metrics!	13
<i>Highlighted: Save lots of money each year</i>	13
8. Conclusion	14
Highlighted: More information?	14

1. Help, my scrum team consists of handymen!

In spite of scrum's popularity, many managers lack a grip on agile IT projects.

Agile software development is more popular than ever. In agile development, developers supply small software components regularly (increments), making interim adjustments very easy to implement. This prevents a situation in which you complete a project and then discover that the system you built does not meet the wishes and requirements. This method is especially useful if the specifications are not entirely clear in advance.

In general, agile methodology saves you development time and costs. It allows you to respond swifter to changing market circumstances, which is an important condition for staying ahead of the competition. This is also the reason that many organisations have embraced agile software development. This also emerges from a study done by KPMG in 2017 among 62 organisations in the Netherlands and Belgium.

- 58% consider the agile methodology an important asset for implementing fast changes
- 76% expect more agile development projects rather than traditional ones within three years
- 85% anticipate that most organisations will continue to deploy both agile and waterfall methodologies

Yet, by far not everyone is enthusiastic about agile development. Although many IT projects use an agile approach, management often feels that it is not 'in control'. Nearly half (48%) of the interviewed managers say that they have too little grip on agile projects².

Thus, the effectiveness and efficiency of agile software development projects often leave much to be desired. Projects are managed based on uncertainties and unfounded assumptions. How big is the job actually; how experienced are the developers; how complex are the settings; are we market compliant? Much of this remains guesswork.

2. Guesswork

Who bases projects on lack of clarity, uncertainty and erroneous estimates?

The frameworks of many IT projects are based on guesswork, also projects that use the agile approach. However, the variables³ behind this highly affect the quality, the total costs and the turnaround times of an IT project. And that's while most projects are limited in time, budget and resources.

Managers usually do not have the right figures at their disposal for determining crucial issues. With the lack of figure-based substantiation of the most important variables, it is no

² Source: Van der Meijs, S. & Van Brummelen, J. (2017), *Agile project delivery*, KPMG Advisory

³ Referring to the 14 factors on page 9

wonder that they make the wrong estimates. That way they often underestimate the impact of the functional system scope on the project duration and the effort demanded by the project.

Do they realise, however, that the number of man-years that a development project takes increases exponentially with respect to its functional scope⁴? That it is significant for the complexity of the software whether the software is intended for a hospital, a bank or a nuclear power station⁵? And that the effort put into the project greatly depends on the effectiveness of the individual team members and the programme languages used for the project⁶?

Moreover, do the managers understand how the technical debt⁷ affects their IT plans? That they need to reserve time to make sure that things remain correct 'under the bonnet'?

3. Quantification

KPIs and metrics provide better estimation and steering of IT projects, even agile projects.

Without a correct insight, you can never make the right decisions necessary for making an IT project successful. Decisions that can be revised gradually based on facts instead of assumptions or opinions.

You obtain a correct insight by quantifying. No guesswork, but by using figures that become transparent, substantiating the project approach and monitoring progress. All this ensures that you compare apples with apples and – based on the correct data – you come to the right agreements with the right supplier. Figures with which you objectively compare the quality and prices of suppliers and find out which items conform to the market. You will be stunned by the differences. In terms of productivity and/or quality differences may differ by a factor 2 to 10 and this is no exception.

Not only for waterfall projects...

Quantification is done by using key performance indicators (KPIs) and underlying metrics. This gives you better steering before and during a project. KPIs and metrics are quite common in IT projects and especially in clear-cut waterfall projects. In such projects, the project result is largely established already at the start.

Many companies consider agile IT projects, in which the specifications are usually not established or in which there is no internal agreement yet, less suitable for quantification based on KPIs. That's mainly due to the uncertainties that are inherently associated with agile software development. However, SAFe⁸ (Scaled Agile Framework), a method for

⁴ Source: Prof. dr. Verhoef, C. (October 20002), *Quantitative IT portfolio management*, <https://bit.ly/2FgTeAA>

⁵ Source: McConell, S. (2015). *Software estimation*, ISBN: 978-0-7356-0535-0

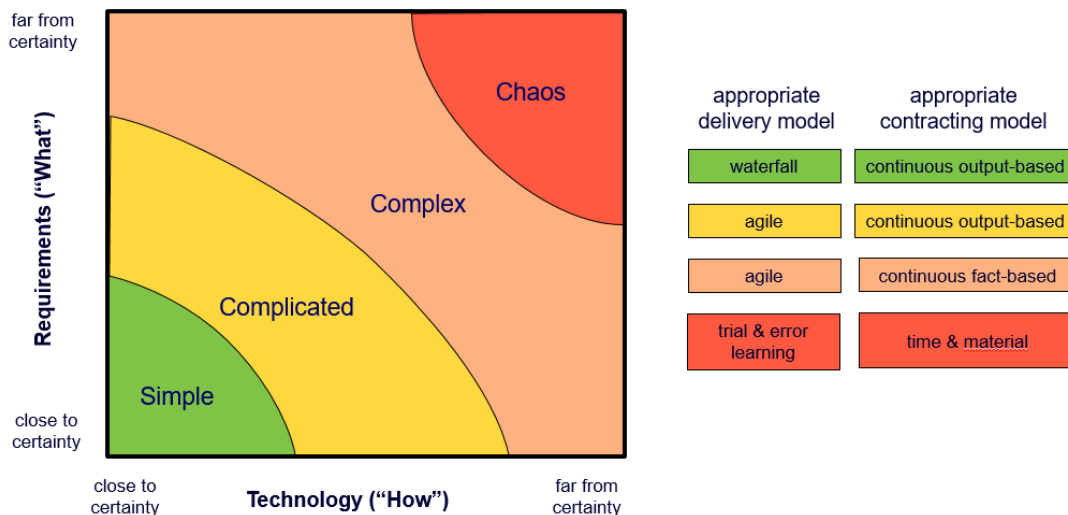
⁶ Source: Jones, C. (2000), *Software Assessments, Benchmarks, and Best Practices*, Addison-Wesley

⁷ Source: Agile Alliance, *Introduction to the Technical Debt Concept*, <https://bit.ly/2UxqM43>

⁸ Source: SAFe, <https://www.scaledagileframework.com>

scaling up the number of agile teams and for enhancing agile work with lean principles, does take metrics into account. However, these are mainly relative and are focused on the business value and software development processes. This means that the SAFe framework is characterised less by product KPIs and metrics that are related to the building system.

Ralph Stacey complexity matrix



The complexity matrix of Ralph Stacey. The traditional waterfall model for software development is suitable for projects with a large degree of certainties and agreements (figure 1). If there is any uncertainty and/or discord, then opt for agile. Both models can use the same set of metrics. The more the project characteristics move to right hand corner (figure 1) the shorter the ‘trail and error’ iterations must become. Time & Material is the most appropriate contract model.

... but also suitable for agile projects

So, are KPIs and metrics suitable only for waterfall projects? Absolutely not. In practice, it appears less problematic to use them in agile IT projects than you would expect. Indeed: we have been using project quantification for years for both small and big clients and for agile and waterfall projects. Our work method has been approved by one of the world’s biggest IT consultancy firms and has been applied in one of the Netherlands’ biggest output projects, and in many other projects in the Netherlands and abroad.

To determine the KPIs and metrics, we do not rely solely on our years of experience. We also use a healthy dose of common sense and the necessary ISO standards⁹.

⁹ Source: ISO/IEC 25010:2011, ISO/IEC 25020:2007, ISO/IEC 24570:2018 NESMA en ISO/IEC 20269:2009 IFPUG

4. The correct values

Discuss with the supplier(s) which value you consider acceptable per metric.

You can quantify even agile IT projects based on KPIs and metrics. Proper attuning assures profits for your own company as well as for the external or internal software supplier(s). Quantification reduces the risks of miscalculating and keeps everyone on their toes during the turnaround time of the project.

Four core KPIs

KPIs and metrics offer figure-based substantiation of the size and status of a project and allow you to steer in the direction of success. For quantifying IT projects, we define four main KPIs (core metrics¹⁰) and a set of about 35 to 40 underlying metrics, which are applicable in every software development project.

1. The quality of the software supplied ('How do we make it better?')
2. Efficiency of the software development ('How do we make it cheaper?')
3. Productivity of the software development ('How can we deliver faster and more frequently?')
4. Satisfaction of the stakeholders ('How do we please everyone?')

Of these four KPIs, the quality of the software that will be built is by far the most important factor. After all, the aftertaste of poor quality remains longer than the sweet taste of a low price.

At least 35 metrics

You measure every core KPI based on metrics. There are usually about 35 to 40 metrics. Prior to the IT project, you discuss with your IT department or supplier(s) which value you consider acceptable per metric. The stakeholders decide what is acceptable. A questionnaire is a solution for this aspect. Have the questionnaire filled in by Business, IT staff and Management and you will have a good picture of the wishes and expectations.

If subsequently, you and the supplier and/or internal IT department cannot concur about the metrics and their values, then you are perhaps on the verge of doing business with the wrong party.

By far most of the metrics are related to quality. The most important determining factor of quality is the scope, or the **functional scope** of the IT system that will be built. The functional scope gives an indication of the physical scope and complexity of the system. This includes the number of functionalities that the user wants to use, such as consulting customer data and preparing invoices.

¹⁰ Source: Putman, L (2003), *Five core metrics*, ISBN 0-932633-55-2

You use the functional scope to get a realistic estimate of the project costs. One of the ways to determine the scope is to use function point analysis. You add up the number of function points of the IT system that will be built. A **function point** is a unit that you measure based on objective ISO standards, such as ISO/IEC 24570:2018 NESMA¹¹.

Highlighted: ISO standard

In the Netherlands, NESMA is the most commonly used ISO standard for determining functional scope. But there are also other standards for that purpose, such as ISO/IEC 20269:2009 IFPUG 4.3.1 and ISO/IEC 19761:2017 COSMIC FFP. In some cases, these are more suitable for measuring scope. Sometimes there is no suitable ISO standard. In that case, you would have to discuss your own units with the agile team. If you solve this without an ISO standard, it might lead to additional discussions.

As you can see in the table below, the choice of the right functional scope metric is very important. A large number of metrics are completely (100%) related to this functional scope.

KPIs Core level	KPIs Second level	Underlying metrics	Related to functional scope [function points]	Degree to which you can automate the measurement
1. Better	1. Code quality	+ 8 metrics	No	100%
	2. Functional testing coverage	+ 2 metrics	100%	100%
	3. Errors	+ 9 metrics	100%	Largely
	4. Solution times of errors	+ 8 metrics	No	Largely
2. Happier	5. Satisfaction	+ 5 metrics	No	0%
3. Faster	6. Time to market	+ 4 metrics	100%	Largely
4. Cheaper	7. Productivity	+ 2 metrics	100%	Largely

Figure 2: Relationship between functional scope and metrics

Altogether, we usually define about 35 to 40 metrics, and based on these, you can quantify the four core KPIs. Below is a selection of these:

- Quality of the source code, the IT architecture and the related documentation
- Presence of technical and functional tests and their coverage (code and decision coverage)
- The number of errors per urgency class per environment, related to the functional scope
- Maximum time span for solving the defects per urgency class per environment
- The efficiency with which we remove the errors from the software during the development process (defect removal efficiency)

¹¹ Source: Nesma (2018), *Definitions and counting guidelines for the application of function point analysis, version: 2.3*, ISBN: 978-90-76258-45-4

- The number of function points, hours per function point and the price per function point that a single scrum team can deliver, and when do we scale up to several teams
- Satisfaction among the stakeholders (product owners, process owners, development team)
- Usage of the features in production
- The extent in which a sprint target is achieved

Highlighted: A fixed price per function point, is that possible?

Indirectly we determine the costs of an IT project based on the price per function point. Not everyone agrees with this. There is a myth in the market that a fixed price per function point is impossible. We see it differently.

If together with the supplier or IT department, you establish the crucial factors, you can certainly agree on a fixed price per function point. After all, other professional groups, such as tilers, define the conditions and wishes and based on these, determine their price accordingly. Why could not that work for software development?

Beside the functional scope, other factors too affect the price per function point. At least 14 factors listed below will appear in every contract.

1. *Functional scope*
2. *Characteristics of the team: knowledge, experience, attitude*
3. *The technology that will be used*
4. *The method that will be used for determining the functional scope*
5. *Detail level for determining the functional scope*
6. *Does this concern newly-built software or software maintenance and which guidelines apply?*
7. *Does this concern the functional scope of one project or of one product?*
8. *Production capacity per time unit (turnaround time, team size)*
9. *Number of software errors per functional scope in acceptance phase and in production phase*
10. *Number of products to be delivered*
11. *Delivery criteria for products to be delivered*
12. *Complexity of the business and the processes*
13. *Non-functional requirements (such as user-friendliness, performance, security)*
14. *IT architecture*

Highlighted: Story points versus function points

Agile IT development projects are already using another type of point system: story points. These are used by agile teams to plan sprints and to obtain commitments^{12 13}. The number of story points is directly linked to the expected efforts to yield a certain **product backlog item**, often a user story.

Why do not we use story points to quantify and compare efforts, quality, costs and suppliers? In comparison with function points, story points are not really suited for getting an objective picture of a project. After all, determining function points are based on **international ISO standards** that are unequivocal and transparent. This enables you to compare in a universally objective manner.

The agile team determines the number of story points per user story during a poker session. There are no uniform counting guidelines. Therefore, you can not validate or audit anything. Determination often occurs in a subjective and relative manner, therefore in relation to other efforts. Commercial interests usually also play a role in which it might be useful to fictitiously increase the number of story points for the supplier or the IT department.

In theory, you can best deploy story points in order to compare productivity between agile teams. However, there are important conditions included in this. For example, you need a mutual reference of the story points with respect to the various teams and the story points in each team must be determined in a consistent way during the poker sessions. Each team must be sufficiently mature to make the correct estimates *and* not to allow any commercial interests to interfere.

In practice, it is nearly impossible to comply with every one of these conditions. Therefore, it will always remain difficult, if not impossible, to quantify and compare efforts¹⁴, quality, costs and suppliers on the basis of story points. **Function points are objective and thus better**. Experience and research teach us that function points are more reliable¹⁵ and can be counted faster than Story Points, as long as they are carried out by experienced counters (preferably certified).

¹² Source: Schofield, J. & Armentrout, A & Trujillo, R., (2013), *Function Points, Use Case Points, Story Points*, <https://bit.ly/2Og0cKi>

¹³ Source: Estimancy, software cost automation, (2018), <https://bit.ly/2OP2WOH>

¹⁴ Source: Little, T. & Verhoef, C., *To Estimate or #NoEstimates, that is the Question*, <https://bit.ly/2bdnFsl>

¹⁵ Source: Kampstra. P. & Verhoef, C., *Reliability of function point counts*, <https://www.cs.vu.nl/~x/rofpc/rofpc.pdf>

5. Real-time insight

Checking whether you are deviating from agreements during each sprint

Ascertaining with your supplier(s) or IT department which value you consider acceptable per metric is not something you only do before an IT project. Also during project implementation, you continuously check whether the actual values do not deviate too much from the agreed-upon values. In each case, you evaluate the metrics at every sprint in the retrospective ceremony of the agile delivery model.

Measuring the values occurs as much as possible in an *automated* fashion. There are various tools available for this purpose, such as Microsoft Azure Devops Server, Azure Devops Services, Jira, TIOBE, SonarQube, CAST, Docker, Atlassian Crucible, JetBrains Upsource, Jenkins, OpenCover, Istanbul, JaCoCo and/or Apache Maven. Automated measuring is difficult in only four of the minimum 35 metrics. These concern the number of hours spent, satisfaction rates, the observed number of software errors and the definition of the functional scope. With respect to these values, the measuring is manual (or partly manual). Experience shows that this has a nearly negligible impact on the project budget. Even if you fill in all 35 metrics manually in an Excel sheet, it will take you a maximum of 30 minutes per sprint.



Figure 3: Balanced Scorecard - KPI dashboard

Transparency is an important condition for deploying KPIs and metrics. All the parties have to know how the metrics are calculated and which guidelines apply. In addition, they must be able to easily figure out where the project stands at any given moment. The progress and the current metric values must be transparent for everyone, so that everyone is looking at 'the same movie'. Unfortunately, it happens far too often that a steering group has a different picture of the project status than the scrum teams.

Dashboard

A good way to make the various metrics transparent at the highest level – and to maintain that level – is to use balanced scorecards in a real-time dashboard (see figure 3). This type of dashboard shows immediately where the project might derail. Subsequently, at a lower level you can retrieve the details of the obstacles.

6. Implementation

The more responsibilities the supplier takes upon himself, the better the result

How do you tackle an agile IT project in which you steer based on KPIs and metrics? You incorporate the KPIs, metrics and values that you agreed upon with the supplier(s) into a **Statement of Work**. The SoW states the metric values, such as the functional scope and the price per function point. Furthermore, the documents expound the main lines of the project. What are the work activities, the deliverables, the schedule and the price? Which steps are necessary, which people and which check points? The SoW also states what are known as the Definition of Ready and the Definition of Done. These, respectively, are the conditions that the projects/sprints must meet at the start and the conditions that must be met at every sprint.

Continuous fact-based delivery

The agile process and the metric supervision of it happens in two ways¹⁶, depending on the engagement and maturity of the supplier (or the internal IT department) and the responsibility that he is willing to take for the final product.

Does the supplier/IT department only agree with continuous adjustments by the customer based on metrics (**continuous fact-based delivery**) in which the customer is still the one taking the greatest risk? After all, the customer will pay per hour. The IT club delivers, the customer pays.

Continuous output-based delivery

Or does the supplier or IT department dare to bear more risks in exchange for a higher profit? In that case, you charge him on the final result (price per function points) and the efforts that the IT club exerted. We call this **continuous output-based delivery**. The IT department delivers, shares in the risk, but also in the final profit.

The IT department or supplier takes care of putting together a team and the software for delivery, which the customer evaluates based on the agreed-upon metrics. Only after all the metrics have been met each sprint and the Definition of Done is achieved, the supplier or the IT department hands over the product, followed by the invoice. This method of work benefits productivity. The supplier receives a bonus in case of fast delivery and you get the software faster in-house. Definitely a win-win-situation.

¹⁶ The main difference between fact-based and output-based is that fact-based means that the supplier will be paid at an hourly rate and output-based means that the supplier will be paid per delivered functionality (e.g. price per Function Point). In both cases metrics and values of the metrics are exactly the same.

7. Involve the supplier in setting up the metrics!

... and change his scepticism to enthusiasm.

Experience has taught us that suppliers and internal IT departments are usually not enthusiastic about continuous output-based delivery. To begin with, they are scared of the use of KPIs and metrics. They often have no idea of the quality of their services. Therefore, they are distances away from knowing the quality, efficiency and productivity of their employees and they often underestimate projects. If the supplier or the internal IT department does not dare to enter an output-based delivery, then be warned: in this project you will not really know what the quality and costs will be.

Highlighted: Save lots of money each year

By deploying KPIs and metrics, you can save lots of money each year. How?

- *Suppose you are looking for a supplier for building a new IT system. A function point analysis shows that the scope of the IT project entails 5,000 function points.*
- *Two suppliers have offered their services. The first one quotes 1,000 euro per function point and the second will price it at 1,100 euro per function point.*
- *The estimate is that the team will realise 2,000 function points per year.*
- *If you select supplier 1, you will save € 500,000. This means that you spend 5 million euro (5,000 x 1,000) and with the second one 5.5 million (5,000 x 1,100).*
- *You can only consider this thoroughly if your quality metrics are in order. Make them a part of the Definition of Done and pay the supplier only once all the DoD conditions are met.*

Yet, continuous output-based delivery also has great advantages for suppliers. The most important one is that they share in the profits. To get them over the threshold, it is advised to first come to agreements about continuous fact-based delivery. Although the customer still bears the most risk, you have more grip on the project than in a standard contract with invoicing per hour.

Moreover, with a continuous fact-based delivery, you build a history of metrics with respect to the internal or external supplier. With this valuable information, in six months from now for example, you can convince him to take more responsibility in exchange for shared profits. This will benefit the quality of the software and productivity of the project.

8. Conclusion

Using KPIs and metrics has many advantages for organisations. If management needs to have *more grip on agile IT projects*, then quantification is necessary. If prior to commencing a project, you make clear agreements with the supplier(s) or IT department, agreements that you can substantiate and monitor with figures, you prevent false expectations, erroneous estimates and interminable projects.

Experience has shown us that a set of *four main KPIs and minimally 35 metrics* is an excellent base for smart agile software development. Quality, efficiency, productivity and satisfaction are the core KPIs. Of these, quality is by far the most important one. Based on concretely formulated metrics, it is possible to steer towards success.

The most important metrics for IT projects are based on the *functional scope* of the system that will be built. You determine the scope based on function points. As long as you do not know the functional scope of a project, you must not start to work. This also applies to agile projects.

A condition for a successful deployment of metrics in agile IT projects is transparency. All stakeholders must be able to easily see the project's current status. A *dashboard with balanced scorecards* is appropriate for this. One of the world's biggest IT consultancy firms has approved this approach and it is applied in small and very big IT projects in the Netherlands and abroad.

Not every IT organisation is eager to quantify its service provision. This step exposes its shortcomings, such as the low productivity of its agile team or lack of knowledge. Quantification prevents you from going into business with the wrong party. You can also deploy quantification in order to increase the *involvement and motivation of the IT supplier (internal or external)*. By making agreements about continuous output-based delivery, you make the supplier more responsible for the software project and you let him share in the profits. This absolutely benefits the quality of the software and productivity.

Highlighted: More information?

Do you want to know more about deploying KPIs and metrics in agile software development? Curious about the countless projects that we raise to higher levels thanks to quantification and output-based delivery? Contact Richard Sweer, project manager at Finidy BV. Richard is available at 06-22130006 and by email: richard@finidy.nl.